UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

TESIS DOCTORAL

Proactive Power and Thermal Aware Optimizations for Energy-Efficient Cloud Computing

Autor:

Patricia Arroba García

Ingeniero de Telecomunicación

Directores:

José Manuel Moya Fernández

Doctor Ingeniero de Telecomunicación

José Luis Ayala Rodrigo

Doctor Ingeniero de Telecomunicación

External Mentor:

Rajkumar Buyya

Doctor of Philosophy in Computer Science and Software Engineering

2017

Patricia Arroba García
E-mail: parroba@die.upm.es

Ph.D. Thesis

Título:	Proactive Power and Thermal Aware Optimizations For Energy-Efficient Cloud Computing
Autor:	Patricia Arroba García
Tutor:	José Manuel Moya Fernández José Luis Ayala Rodrigo
External mentor:	Rajkumar Buyya
Departamento:	Departamento de Ingeniería Electrónica

Miembros del tribunal:

Presidente:
Secretario:
Vocal:
Vocal:
Vocal:

Suplente:
Suplente:

Los miembros del tribunal arriba nombrados acuerdan otorgar la calificación de:

Madrid, de de 2017

"Do. Or do not. There is no try."

— Yoda, *The Empire Strikes Back. Star wars*

Acknowledgment

"You'll stay with me? – 'Until the very end'."
— J.K. Rowling, *Harry Potter and the Deathly Hallows*

Hace unos años, cuando decidí dedicar mi trabajo a la investigación, aposté por una línea emergente, que se salía de lo establecido en ese momento. Desde entonces he dedicado mi esfuerzo a una investigación en la que creo, porque pienso que tiene un objetivo importante a nivel social y que tiene un gran impacto a nivel global. Y es una decisión que me ha hecho muy feliz.

Mis directores de tesis son los principales responsables de sacar adelante esta línea de investigación. Gracias por ser tan valientes, por apostar por lo que creéis y por apoyarme siempre a nivel profesional y a nivel personal. A Josem, por ser un visionario, pensar siempre a lo grande, y ser el más optimista. A Jose, porque siempre me impresionan tus conocimientos tan multidisciplinares, por ser tan inteligente y aportarme tanta serenidad. Sois excepcionales.

Durante mi investigación he tenido la suerte de trabajar con dos grupos con gente maravillosa. A mis compañeros de GreenLSI, me encanta trabajar con vosotros, juntos somos un gran equipo. A los investigadores de ArTeCS, porque rodearse de gente que es muchísimo mejor que tú hace que puedas crecer. A Marina, por los deadlines, por los cafés y por las llamadas Boston-Melbourne y Laussane-Tokyo. Gracias a todos por enseñarme, me queda mucho que aprender de vosotros.

I would like to express my sincere gratitude to my mentor Prof. Rajkumar Buyya, from the University of Melbourne, for his continuous support and immense knowledge. I'd like to thank my sensei Prof. Satoshi Matsuoka for his advice and support during my research stay at his group at the Tokyo Institute of Technology. Thanks also to the people of the CLOUDS Lab. and the Matsuoka Lab. I really appreciate your guidance and generosity.

A mis amigos, mis Jamelgas que siempre están ahí, a Serrano y Alicia por acordarse de mí aunque esté bien lejos, a Ale porque a veces crees tú más en mí que yo, y porque teneros cerca y compartir cosas con vosotros me hace muy feliz.

Gracias a toda mi superfamilia, porque celebrar los buenos momentos con vosotros es lo que merece la pena. Papá, mamá, Edu, siempre nos hemos cuidado y apoyado en todo y esto no iba a ser menos. Siempre nos habéis enseñado que el esfuerzo y la constancia tienen su recompensa, y ésta es una más que disfrutar juntos. Os quiero!

Y sobre todo a David, por estar ahí todos los días, en Australia, en Japón, en Madrid, y en todas partes. Te admiro y te quiero. Juntos mejor. Siempre.

"Tengo un millón de cosas que agradecerte, pero también toda una vida para hacerlo."
— Dr. D. Fraga, *Trust-ware: A Methodology to Analyze, Design, and Secure Trust and Reputation Systems*

Abstract

"Hey, mamma, Look at me,
I'm on the way to the promised land"

— AC/DC, *Highway to hell*

Cloud computing addresses the problem of costly computing infrastructures by providing elasticity to the dynamic resource provisioning on a pay-as-you-go basis, and nowadays it is considered as a valid alternative to owned high performance computing clusters. There are two main appealing incentives for this emerging paradigm: first, utility-based usage models provided by Clouds allow clients to pay per use, increasing the user satisfaction; then, there is only a relatively low investment required for the remote devices that access the Cloud resources [1].

Computational demand on data centers is increasing due to growing popularity of Cloud applications. However, these facilities are becoming unsustainable in terms of power consumption and growing energy costs. Nowadays, the data center industry consumes about 2% of the worldwide energy production [2]. Also, the proliferation of urban data centers is responsible for the increasing power demand of up to 70% in metropolitan areas where the power density is becoming too high for the power grid [3]. In two or three years, this situation will cause outages in the 95% of urban data centers incurring in annual costs of about US$2 million per infrastructure [4]. Besides the economical impact, the heat and the carbon footprint generated by cooling systems in data centers are dramatically increasing and they are expected to overtake airline industry emissions by 2020 [5].

The Cloud model is helping to mitigate this issue, reducing carbon footprint per executed task and diminishing CO_2 emissions [6], by increasing data centers overall utilization. According to the Schneider Electric's report on virtualization and Cloud computing efficiency [7], Cloud computing offers around 17% reduction in energy consumption by sharing computing resources among all users. However, Cloud providers need to implement an energy-efficient management of physical resources to meet the growing demand of their services while ensuring sustainability.

The main sources of energy consumption in data centers are due to computational Information Technology (IT) and cooling infrastructures. IT represents around 60% of the total consumption, where the static power dissipation of idle servers is the dominant contribution. On the other hand, the cooling infrastructure originates around 30% of the overall consumption to ensure the reliability of the computational infrastructure [8]. The key factor that affects cooling requirements is the maximum temperature reached on the servers due to their activity, depending on both room temperature and workload allocation.

Static consumption of servers represents about 70% of the IT power [9]. This issue is intensified by the exponential influence of temperature on the leakage currents. Leakage power is a component of the total power consumption in data centers that is not traditionally considered in the set point temperature of the room. However, the effect of this power contribution, increased with temperature, can determine the savings associated with the proactive management of the cooling system. One of the major challenges to understand the thermal influence on static energy at the data center scope consists in the description of the trade-offs between leakage and cooling consumption.

The Cloud model is helping to reduce the static consumption from two perspectives based on VM allocation and consolidation. First, power-aware policies reduce the static consumption by increasing overall utilization, so the operating server set can be reduced. Dynamic Voltage and Frequency Scaling (DVFS) is applied for power capping, lowering servers' energy consumption. Then, thermal-aware strategies help to reduce hot spots in the IT infrastructure by spreading the workload, so the set point room temperature can be increased resulting in cooling savings. Both thermal and power approaches have the potential to improve energy efficiency in Cloud facilities. Unfortunately, these policies are not jointly applied due to the lack of models that include parameters from both power and thermal approaches. Deriving fast and accurate power models that incorporate these characteristics, targeting high-end servers, would allow us to combine power and temperature together in an energy efficient management.

Furthermore, as Cloud applications expect services to be delivered as per Service Level Agreement (SLA), power consumption in data centers has to be minimized while meeting this requirement whenever it is feasible. Also, as opposed to HPC, Cloud workloads vary significantly over time, making optimal allocation and DVFS configuration not a trivial task. A major challenge to guarantee QoS for Cloud applications consists in analyzing the trade-offs between consolidation and performance that help to combine DVFS with power and thermal strategies.

The main objective of this Ph.D. thesis is to address the energy challenge in Cloud data centers from a thermal and power-aware perspective using proactive strategies. Our research proposes the design and implementation of models and global optimizations that jointly consider energy consumption of both computing and cooling resources while maintaining QoS from a new holistic perspective.

Thesis Contributions: To support the thesis that our research can deliver significant value in the area of Cloud energy-efficiency, compared to traditional approaches, we have:

- Defined a taxonomy on energy efficiency that compiles the different levels of abstraction that can be found in data centers area.
- Classified state-of-the-art approaches according to the proposed taxonomy, identifying new open challenges from a holistic perspective.
- Identified the trade-offs between leakage and cooling consumption based on empirical research.
- Proposed novel modeling techniques for the automatic identification of fast and accurate models, providing testing in a real environment.
- Analyzed DVFS, performance and power trade-offs in the Cloud environment.
- Designed and implemented a novel proactive optimization policy for dynamic consolidation of virtual machines that combine DVFS and power-aware strategies while ensuring QoS.
- Derived thermal models for CPU and memory devices validated in real environment.
- Designed and implemented new proactive approaches that include DVFS, thermal and power considerations in both cooling and IT consumption from a novel holistic perspective.
- Validated our optimization strategies in simulation environment.

Resumen

"Is there anyone here who speaks English?
Or maybe even ancient Greek?"

— Marcus Brody, *Indiana Jones and the Last Crusade*

La computación en la nube, o *Cloud computing*, aborda el problema del alto coste de las infraestructuras de computación, proporcionando elasticidad al aprovisionamiento dinámico de recursos. Este paradigma de computación está basado en el *pago por uso* y actualmente se considera como una alternativa válida a la adquisición de un clúster de computación de altas prestaciones (HPC).

Existen dos principales incentivos que hacen atractivo a este paradigma emergente: en primer lugar, los modelos basados en el pago por uso proporcionados por la nube permiten que los clientes paguen sólo por el uso que realizan de la infraestructura, aumentando la satisfacción de los usuarios; por otra parte, el acceso a los recursos de la nube requiere una inversión relativamente baja.

La demanda computacional en los centros de datos está aumentando debido a la creciente popularidad de las aplicaciones *Cloud*. Sin embargo, estas instalaciones se están volviendo insostenibles en términos de consumo de potencia y debido al precio al alza de la energía. Hoy en día, la industria de los centros de datos consume un 2% de la producción mundial de energía [2]. Además, la proliferación de centros de datos urbanos está generando una demanda de energía cada vez mayor, representando el 70% del consumo en áreas metropolitanas, superando la densidad de potencia soportada por la red eléctrica [3]. En dos o tres años, esta situación supondrá cortes en el suministro en el 95% de los centros de datos urbanos incurriendo en unos costes anuales de alrededor de US$2 millones por infraestructura [4]. Además del impacto económico, el calor y la huella de carbono generados por los sistemas de refrigeración de los centros de datos están aumentando drásticamente y se espera que en 2020 hayan superado a las emisiones de la industria aérea [5].

El modelo de computación en la nube está ayudando a mitigar este problema, reduciendo la huella de carbono por tarea ejecutada y disminuyendo las emisiones de CO_2 [6], mediante el aumento de la utilización global de los centros de datos. Según el informe de Schneider Electric sobre virtualización y eficiencia energética de la computación en la nube [7], este modelo de computación ofrece una reducción del 17% en el consumo de energía, compartiendo recursos informáticos entre todos los usuarios. Sin embargo, los proveedores de la nube necesitan implementar una gestión eficiente de la energía y de los recursos computacionales para satisfacer la creciente demanda de sus servicios garantizando la sostenibilidad.

Las principales fuentes de consumo de energía en centros de datos se deben a las infraestructuras de computación y refrigeración. Los recursos de computación representan alrededor del 60% del consumo total, donde la disipación de potencia estática de los servidores es la contribución dominante. Por otro lado, la infraestructura de refrigeración origina alrededor del 30% del consumo total para garantizar la fiabilidad de la infraestructura de computación [8]. El factor clave que afecta a los requisitos de refrigeración es la temperatura máxima alcanzada en los servidores debido a su actividad, en función de la temperatura ambiente así como de la asignación de carga de trabajo.

El consumo estático de los servidores representa alrededor del 70% de la potencia de los recursos de computación [9]. Este problema se intensifica con la influencia exponencial de la temperatura en las corrientes de fugas. Estas corrientes de fugas suponen una contribución importante del consumo total de energía en los centros de datos, la cual no se ha considerado tradicionalmente en la definición de la temperatura de de la sala. Sin embargo, el efecto de esta contribución de energía, que aumenta con la temperatura, puede determinar el ahorro asociado a la gestión proactiva del sistema de refrigeración. Uno de los principales desafíos para entender la influencia térmica en la componente de energía estática en el ámbito del centro de datos consiste en la descripción de las ventajas y desventajas entre las corrientes de fugas y el consumo de refrigeración.

El modelo de computación en la nube está ayudando a mitigar el problema de consumo estático desde dos perspectivas basadas en la asignación de máquinas virtuales (MVs) y en su consolidación. En primer lugar, las políticas conscientes de la potencia reducen el consumo estático mediante el aumento de la utilización global, por lo que el conjunto de servidores operativos puede reducirse. El escalado dinámico de frecuencia y tensión (DVFS) se aplica para reducir el consumo de energía de los servidores. Por otra parte, las estrategias conscientes de la temperatura ayudan a la reducción de los puntos calientes en la infraestructura de computación mediante la difusión de la carga de trabajo, por lo que la temperatura ambiente de la sala se pueden aumentar con el consiguiente ahorro en el consumo de la refrigeración. Ambos enfoques tienen el potencial de mejorar la eficiencia energética en instalaciones de la nube. Desafortunadamente, estas políticas no se aplican de manera conjunta debido a la falta de modelos que incluyan parámetros relativos a la potencia y a la temperatura simultáneamente. Derivar modelos de energía rápidos y precisos que incorporen estas características permitiría combinar ambas estrategias, conscientes de la potencia y la temperatura, en una gestión global eficiente de la energía.

Por otra parte, las aplicaciones características de la computación en la nube tienen que cumplir unos requisitos específicos en términos de tiempo de ejecución que están previamente contratados mediante el acuerdo de nivel de servicio (SLA). Es por esto que la optimización del consumo de energía en estos centros de datos tiene que considerar el cumplimiento de este contrato siempre que sea posible. Además, a diferencia de HPC, las cargas de trabajo de la nube varían significativamente con el tiempo, por lo que la asignación óptima y la configuración del DVFS no es una tarea trivial. Uno de los retos más importantes para garantizar la calidad de servicio de estas aplicaciones consiste en analizar la relación entre la consolidación y el rendimiento de la carga de trabajo, ya que facilitaría la combinación del DVFS con las estrategias térmicas y energéticas.

El principal objetivo de esta tesis doctoral se centra en abordar el desafío de la energía en centros de datos dedicados a la computación en la nube desde una perspectiva térmica y con conciencia de la potencia utilizando estrategias proactivas. Nuestro trabajo propone el diseño e implementación de modelos y optimizaciones globales que consideren conjuntamente el consumo de energía tanto de los recursos informáticos y de refrigeración, manteniendo la calidad de servicio, desde una nueva perspectiva holística.

Contribuciones clave: Para apoyar la tesis de que nuestra investigación puede proporcionar un valor significativo en el ámbito de la eficiencia energética en la computación en la nube, en comparación con enfoques tradicionales, nosotros hemos:

- Definido una taxonomía en el área de la eficiencia energética que se compone de diferentes niveles de abstracción que aparecen en el ámbito de los centros de datos.

- Clasificado propuestas del estado del arte de acuerdo a nuestra taxonomía, identificando posibles contribuciones, desde una perspectiva holística.

- Identificado el compromiso entre las fugas de potencia y el consumo de refrigeración basado en un estudio empírico.

- Propuesto nuevas técnicas de modelado para la identificación automática de modelos precisos y rápidos, proporcionando una validación en entorno real.

- Analizado el compromiso entre DVFS, rendimiento y consumo en el entorno de computación en la nube.

- Diseñado e implementado una nueva política de optimización proactiva para la consolidación dinámica de máquinas virtuales que combina DVFS y estrategias conscientes de la potencia, manteniendo la calidad de servicio.

- Derivado modelos térmicos para procesador y memoria validados en un entorno real.

- Diseñado e implementado nuevas políticas proactivas que incorporan consideraciones de DVFS, térmicas y de potencia en para el consumo de las infraestructuras de computación y refrigeración desde una nueva perspectiva holística.

- Validado nuestras estrategias de optimización en un entorno de simulación.

Contents

List of Tables

List of Figures

1. Introduction

"This is your last chance. After this, there is no turning back. You take the blue pill – the story ends, you wake up in your bed and believe whatever you want to believe. You take the red pill – you stay in Wonderland, and I show you how deep the rabbit hole goes."

— Morpheus, *The Matrix*

The amount of energy consumed by data centers is growing disproportionately, thus becoming a critical element in maintaining both economic and environmental sustainability. This work presents a study of the literature that comprises recent research on energy-aware policies highlighting the scope of Cloud computing data centers. The present research is intended to give an overview of the energy-efficiency issue throughout the different abstraction levels, from hardware technology to data center infrastructures.

1.1 Motivation

The trend towards Cloud computing has lead to the proliferation of data centers since they are the infrastructure that provides this new paradigm of computing and information storage. Reference companies such as Amazon [10], Google [11], Microsoft [12], and Apple [13] have chosen this computational model where information is stored in the Internet Cloud offering services more quickly and efficiently to the user.

Nowadays, data centers consume about 2% of the worldwide energy production [2], originating more than 43 million tons of CO_2 per year [14]. Also, the proliferation of urban data centers is responsible for the increasing power demand of up to 70% in metropolitan areas, where the power density is becoming too high for the power grid [3]. In two years, the 95% of urban data centers will experience partial or total outages, incurring in annual costs of about US$2 million per infrastructure. The 28% of these service outages are expected to be due to exceeding the maximum capacity of the grid [4].

The advantages of Cloud computing lie in the usage of a technological infrastructure that allows high degrees of automation, consolidation and virtualization, which results in a more efficient management of the resources of a data center. The Cloud model allows a large number of users as well as the use of concurrent applications that otherwise would require a dedicated computing platform.

Cloud computing, in the context of data centers, has been proposed as a mechanism for minimizing environmental impact. Virtualization and consolidation increase hardware utilization (of up to 80% [15]) thus improving resource efficiency. Moreover, a Cloud usually consists of distributed resources dynamically provisioned as services to the users, so it is flexible enough to find matches between different parameters to reach performance optimizations.

Cloud market opportunities in 2016 achieved up to $209.2 billion [16], but the rising price of energy had an impact on the costs of Cloud infrastructures, increasing the Total Cost of Ownership (TCO) and reducing the Return on Investment (ROI). Gartner expectations predict that by 2020, Cloud adoption strategies would impact on more than the 50% of the IT outsourcing deals in an effort to cost optimize the infrastructures that use from 10 to 100 times more power than typical office buildings [17] even consuming as much electricity as a city [18].

The main contributors to the energy consumption in a data center are: (i) the Information Technology (IT) resources, which consist of servers and other IT equipment, and (ii) the cooling infrastructure needed to ensure that IT operates within a safe range of temperatures, ensuring reliability. The remaining 10% comes from (iii) the power consumption that comes from lightning, generators, Uninterrupted Power Supply (UPS) systems and Power Distribution Units (PDUs) [19].

The IT power in the data center is dominated by the power consumption of the enterprise servers, representing up to 60% of the overall data center consumption. The power usage of an enterprise server can be divided into dynamic and static contributions. Dynamic power depends on the switching transistors in electronic devices during workload execution. Static consumption associated to the power dissipation of servers, represents around 70% and is strongly correlated with temperature due to the leakage currents that increase as technology scales down. However, traditional approaches have never incorporated the impact of leakage consumption, which grows at higher temperatures. Subsection 1.2.1 analyzes this issue from the low-level scope.

On the other hand, data center cooling is one of the major contributors to the overall data center power budget, representing around 40% of the total power consumed by the entire facility [20]. This is the main reason why recent research aim to achieve new thermal-aware techniques to optimize the temperature distribution in the facility, thus minimizing the cooling costs. Moreover, temperature in data centers is increasing substantially due to the activity of servers that result from growing resource demand. The heat is evacuated outwards in the form of thermal pollution avoiding server failures. In Subsection 1.2.2, we explain why cooling is necessary to avoid these failures and also irreversible damage in the IT infrastructure, from a technological perspective.

Controlling the set point temperature of cooling systems in data centers is still to be clearly defined and it represents a key challenge from the energy perspective. This value is often chosen based on conservative suggestions provided by the manufacturers of the equipment and it is calculated for the worst case scenario resulting on overcooled facilities. Increasing the temperature by 1°C results in savings of 7.08% in cooling consumption, so a careful management can be devised ensuring a safe temperature range for IT resources. Subsection 1.3.1 presents how this challenge can be tackled from a higher-level perspective, considering the efficiency at the data center scope.

From the application-framework viewpoint, Cloud workloads present additional restrictions as 24/7 availability, and Service Level Agreement (SLA) constraints among others. In this computation paradigm, workloads hardly use 100% of Central Processing Unit (CPU) resources, and their execution time is strongly constrained by contracts between Cloud providers and clients. These restrictions have to be taken into account when minimizing energy consumption as they impose additional boundaries to efficiency optimization strategies. In Subsection 1.3.2 we provide some considerations about Cloud applications.

Besides the economical impact, the heat and the carbon footprint generated by these facilities are dramatically harming the environment and they are expected to overtake the emissions of the airline industry by 2020 [5]. Data centers are responsible for emitting tens of millions of metric tons of greenhouse gases into the atmosphere, resulting in more than 2% of the total global emissions [21]. It is expected that, by implementing the Cloud computing paradigm, energy consumption will be decreased by 31% by 2020, reducing CO_2 emissions by 28% [22]. Just for an average $100kW$ data center, a 7% in annual savings represent around US $5 million per year [23].

These power and thermal situations have encouraged the challenges in the data center scope to be extended from performance, which used to be the main target, to energy-efficiency. This context draws the priority of stimulate researchers to develop sustainability policies at the data center level, in order to achieve a social sense of responsibility while minimizing environmental impact.

The following sections profusely explain the main issues from the perspective of technology, data center and Cloud areas:

1.2 Technological Considerations

1.2.1 Impact on IT Leakage Power

As previously mentioned, power dissipation introduced by leakage has a strong impact on the overall consumption of the Complementary Metal-Oxide-Semiconductor (CMOS) devices.

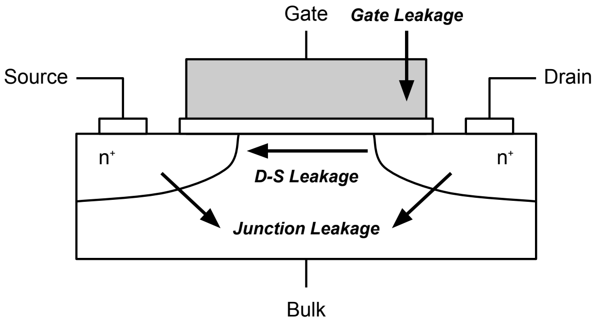


Figure 1.1: Leakage currents in a MOS transistor

Theoretically, any current should not circulate through the substrate of a Metal-Oxide-Semiconductor (MOS) transistor between drain and source when off due to an infinite gate resistance. However, in practice this is not true, and leakage currents flow through the reverse-biased source and drain-bulk pn junctions in dynamic logic as represented in Figure 1.1. Also due to the continuous technology scaling, the influence of leakage effects is rising, increasing the current by 5 orders of magnitude according to Rabaey [24].

Therefore, it is important to consider the strong impact of static power consumed by devices as well as its temperature dependence and the additional effects influencing their performance. Thus, theoretical and practical models for the calculation of power consumption in Cloud servers should also consider these issues in their formulations. This research work is intended to include the static consumption, its effects and temperature dependence in power models to globally optimize energy consumption in Cloud computing data centers.

1.2.2 Impact on Reliability

The high-density computing causes the appearance of hot spots in data center facilities, which are intensified due to the heterogeneous workload distribution. This issue has a strong impact on the reliability of systems, reducing their mean time to failure and, in some cases, generating irreversible damage to the infrastructure. Some adverse effects arising from hot spots in a circuit are the following:

- Single event upset (SEU): Effect that results in the change of state caused by radiation, and experiments have indicated that it increases further with temperature [25].
- Electromigration (EM): Phenomenon that consists in the transference of a material caused by the gradual movement of ions in a conductor due to the momentum transfer between conduction electrons and metal atoms. It depends on the operating temperature and causes short circuits and important failures.

- Time-dependent dielectric-breakdown (TDDB): It appears due to the breakdown of the oxide gate resulting from the electron tunneling current to the substrate when a Metal-Oxide-Semiconductor Field-Effect Transistor (MOSFET) perform close to their operational voltages.

- Stress migration (SM): Failure phenomenon due to the open circuit or the vast resistance associated to large void formations, resulting in vacancy migration driven by a hydrostatic stress gradient that can be increased by temperature.

- Thermal cycling (TC): It produces accumulative damage each time the device goes cycling through extreme temperatures usually changing at high rates and setting boundaries to the system lifetime.

- Negative bias temperature instability (NBTI): It results in degradation due to an increase in threshold voltage. High temperatures slow down integrated circuits due to the degradation of carrier mobility, thus minimizing device lifetime.

- Hot carrier injection (HCI): Phenomenon that comes from the emergence of a potential barrier resulting in the kinetic energy gained by an electron or a hole, breaking an interface state. Although the term "hot" refers to the model carrier density effective temperature, not to the temperature of the device, tests [26] show dependence due to temperature impact.

1.3 Data Center Considerations

1.3.1 Impact on Cooling Efficiency

The cooling power is one of the major contributors to the overall data center power budget, consuming over 30% of the overall electricity bill in typical data centers [27]. In a typical air-cooled data center room, servers are mounted in racks, arranged in alternating cold/hot aisles, with the server inlets facing cold air and the outlets creating hot aisles. The computer room air conditioning (CRAC) units pump cold air into the data room and extract the generated heat (see Figure 1.2). The efficiency of this cycle is generally measured by the Coefficient of Performance (COP).

One of the techniques to reduce the cooling power is to increase the COP by increasing the data room temperature. We will follow this approach to decrease the power used by the cooling system to a minimum, while still satisfying the safety requirements of the data center operation, also considering temperature impact on IT power.

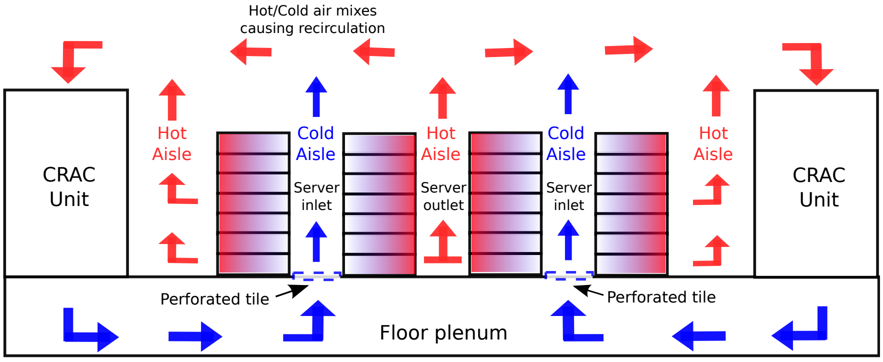


Figure 1.2: Data center air cooling scheme

1.3.2 Impact on Cloud applications

While High Performance Computing (HPC) workloads typically consist of large scientific and experimental research applications with heavy computational requirements, Cloud workloads are more commercial applications that process incoming requests on a service-based manner. As opposed to HPC, Cloud workloads vary significantly over time as they are interactive and real-time workloads, making optimal allocation configuration not a trivial task. As CPU and memory usage are variable due to interaction with users, a major challenge to guarantee QoS for Cloud applications consists of analyzing the trade-offs between consolidation and performance that help on enhancing optimization strategies.

The resource usage of Cloud applications is lower and much more variable than in the HPC context. In this framework, virtualization allows workload consolidation by the migration of Virtual Machines (VM) thus helping to improve host utilization during runtime. Moreover, depending on the type of applications executed in the data center, the use of the computing resources would be different. Cloud data centers typically run workloads as web search engines, data mining and media streaming, among others.

- *Search engine* applications index terabytes of data gathered from online sources. They need to support a large number of concurrent queries against the index, resulting in an intensive use of CPU, memory and network connections.

- *Data mining* applications analyze, classify and filter human-generated information. They handle large-scale analysis making intensive use of CPU.

- *Media streaming* applications offer an ubiquitous service to access media contents. They make an appreciable use of CPU to packetize the information of the media file (with a size range from megabytes to terabytes) for being sent through the Internet. They also make an intensive use of network connections.

Consequently, the workload requirements in terms of resource demand and performance will determine the optimizations supported by the computational model. Therefore, power strategies must be able of accurately predicting the consumption during workloads with high variability in resource utilization.

1.4 Optimization paradigm

This section explains our hypothesis to provide a feasible proactive and holistic solution to approach the issues that have been motivated. As stated in the previous section, the temperature has a relevant impact on the data center consumption at different levels. Regarding technical considerations, there exist a power leakage that increases due to temperature. Currently, to the best of our knowledge, there are no models that incorporate this dependence to power consumption within the data center scope. For this reason, we provide server power models that include thermal effects together with other contributions as dynamic power consumption among others. As servers are complex systems, we propose different modeling methodologies that enhance the modeling process including automaticity in both feature selection and model generation

The temperature also impacts on the reliability of data center subsystems, so there is a necessity of considering the thermal behavior during runtime to ensure safe-operation ranges. For this purpose, we provide thermal models that estimate the temperature of different devices inside the server, depending on the cooling setpoint temperature and the variable resource demand of the workload.

Regarding data center considerations, these power and thermal models help us to provide a holistic strategy to reduce both IT and cooling consumption globally. From the perspective of the application framework, Cloud applications present a highly variable demand of resources during their execution. So, we provide dynamic VM consolidation techniques to optimize the system during runtime from our holistic perspective. Also, as Cloud workloads hardly consume the 100% of the server resources, we apply DVFS-awareness to leverage the overdimensioning of the processing capacity, while maintaining SLA.

The new holistic paradigm presented in this research considers the energy at the Cloud data center from a global and proactive perspective for the first time in literature. So, our proposed optimization algorithms are aware of the evolution of the global energy demand, the thermal behavior of the room and the workload considerations at all the data center subsystems during runtime.

1.5 Problem formulation

The work developed in this Ph.D. thesis proposes a global solution based on the energy analysis and optimization for Cloud applications from a holistic perspective. The envisioned modeling and optimization paradigm is summarized in Figure 1.3. This framework takes as input all the information gathered from the data center during the workload execution, at different abstraction layers (i.e. server and data room), via sensor measurements of both physical and computational metrics. Data is stored to generate models, also at different abstraction levels. The models obtained enable the design of proactive optimization strategies. The results of these optimizations are evaluated in order to integrate the decisions taken.

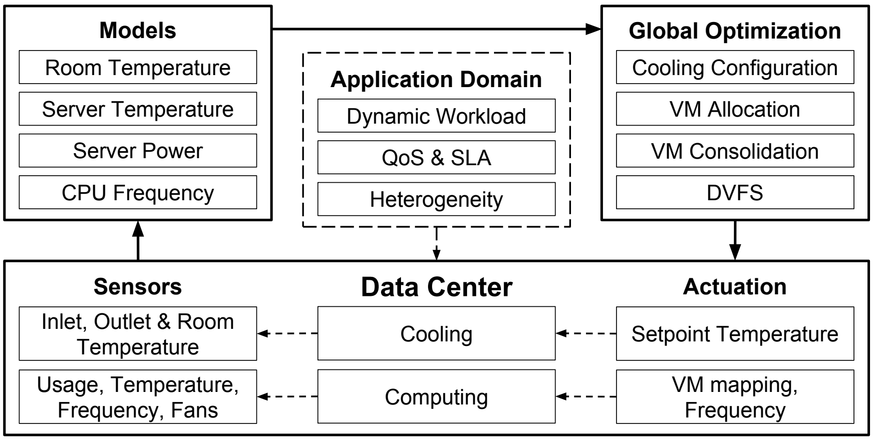


Figure 1.3: Overview of the proactive analysis and optimization system.

The scenario chosen for the development of this Ph.D. thesis, is a Cloud application framework that can be seen in Figure 1.4. These applications require constantly monitoring of their computational demand in order to capture their variability during runtime and to perform VM migrations when needed in order to avoid Quality of Service (QoS) degradation.

As we do not have access to an operative Cloud data center, we leverage real traces publicly released by Cloud providers to simulate the operation of the infrastructure. These traces consist of periodic resource usage reports that provide specific information as CPU demand percentages and memory and disk usage and provisioning values for all VMs. The utilization traces are our only input for our optimization based on proactive strategies. Finally, for each optimization slot, we obtain the allocation for the VMs in the system, as well as the servers' Dynamic Voltage and Frequency Scaling (DVFS) configuration and the cooling set point temperature.

In Figure 1.5, we observe our proposed optimization based on proactive strategies more in detail. For each optimization slot, in which we have input traces, we detect overloaded hosts for the current placement of VMs that are already deployed on the system, where oversubscription is allowed. Overloaded hosts are more likely to suffer from performance degradation, so some VMs have to be migrated from them to other hosts.

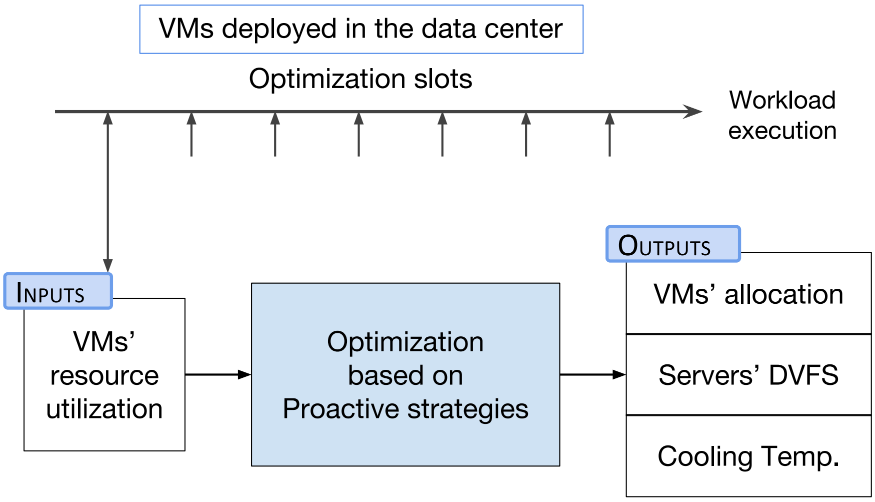

VMs deployed in the data center
Optimization slots
Workload execution
Inputs
VMs' resource utilization
Optimization based on Proactive strategies
Outputs
VMs' allocation
Servers' DVFS
Cooling Temp.

Figure 1.4: Overview of the inputs and outputs of the proposed optimization framework.

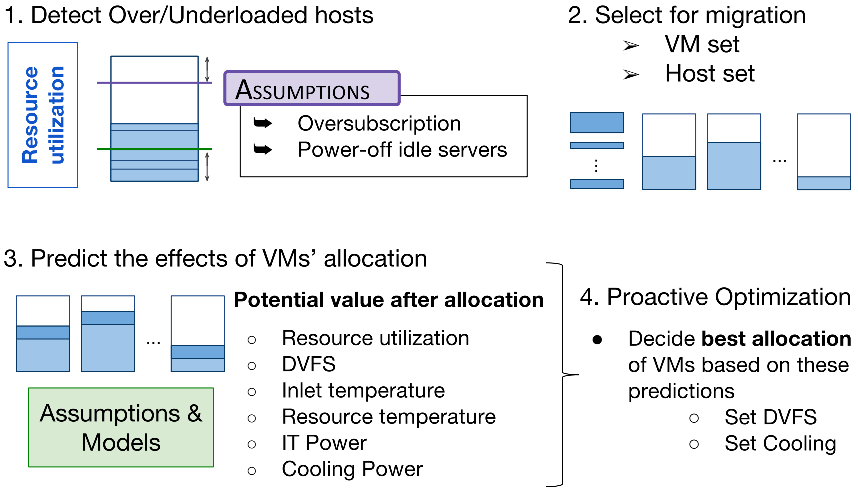


Figure 1.5: Overview of the optimization diagram for the proposed framework.

Based on this information, the consolidation algorithm selects: 1) the set of VMs that have to be migrated from the overloaded physical machines and 2) the set of servers that are candidates to host these VMs. Then, the models proposed in this Ph.D. thesis help to predict the effects of potential allocations for the set of VMs. These models are needed to provide values of both the parameters that are observed in the infrastructure (temperature and power of the different resources) and the control variables (VM placement, DVFS and cooling set point temperature). Finally, the proactive optimization algorithm decides the best allocation of VMs based on these predictions. After this first iteration, if underloaded hosts are found, this optimization process is repeated in order to power off idle servers if possible.

To this end, the scenario chosen for the development of energy-aware techniques at the data center level is a virtualized Cloud data center. We assume this data center may be homogeneous in terms of IT equipment. Live migration, oversubscription of CPU, DVFS and automatic scaling of the active servers' set are enabled. We assume a traditional hot-cold aisle data center layout with CRAC-based cooling. In particular, at the data center level we consider a raised-floor air-cooled data center where cold air is supplied via the floor plenum and extracted in the ceiling.

Real server architectures are tested by means of the monitoring and modeling of various presently-shipping enterprise servers. Also, at the data center scope other reduced scenarios are used to test the proposed optimization policies.

1.6 Contributions of this Ph.D. Thesis

The contributions of this Ph.D. thesis can be broadly described into 4 categories: (i) analysis of the state-of-the-art in energy efficiency, (ii) study of the thermal trade-offs at the data center scope, (iii) modeling of data center power, temperature and performance, and (iv) implementation of global optimizations.

- State-of-the-Art on Energy Efficiency
  - Define a taxonomy on energy efficiency that compiles the different levels of abstraction that can be found in data centers' area.
  - Classify the different energy efficiency approaches according to the proposed taxonomy, evaluating the impact of current research on energy efficiency.
  - Identify new open challenges that have the potential of improving sustainability on data centers significantly by applying our proposed holistic optimization approach, including information from different abstraction levels.
- Thermal Trade-offs
  - Detect the need of addressing leakage power of servers jointly with cooling efficiency to achieve substantial global savings, evaluating the relationships between temperature and power consumption.
  - Identify the trade-off between leakage and cooling consumption based on empirical research.
  - Our empirical results show that increasing data room setpoint temperature in 6° C, increases application power consumption in about 4.5% and reduces cooling power by 11.7%.
- Data Center Modeling
  - Detect parameters mainly affected by leading power-aware strategies for improving Cloud efficiency as well as thermal considerations.
  - Analyze and implement novel modeling techniques for the automatic identification of fast and accurate models that help to target heterogeneous server architectures.
  - Derive models that incorporate these contributors that help to find the relationships required to devise global optimizations combining power and thermal-aware strategies.
  - Provide training and testing in a real environment. Our models, which are aware of DVFS and temperature, present an average testing error in the range of 4.87% to 3.98%, outperforming current approaches whose accuracies are in the range of 7.66% to 5.37%.
- Global Energy Optimizations
  - Analyze and model DVFS, performance and power trade-offs.
  - Design and implement a novel proactive optimization policy for dynamic consolidation of Cloud services that combine DVFS and power-aware strategies while ensuring QoS.

  - Provide validation in a simulation environment. Our DVFS-aware holistic approach provides energy savings of up to 45.76% for the IT infrastructure, also increasing global server utilization to 83% in average, when compared to a power-aware baseline.
  - Derive thermal models for CPU and memory devices trained and tested in real environment with average testing errors of 0.84% and 0.5049% respectively.
  - Design and implement new proactive approaches that include DVFS, thermal and power considerations in both cooling and IT consumption from a holistic perspective.
  - Provide validation in simulation environment. Our DVFS and thermal aware holistic strategy presents maximum savings of up to 14.09% and 21.74% with respect to our state-of-the-art baselines.

1.7 Structure of this Ph.D. Thesis

The rest of the document of this Ph.D. thesis is organized as follows:

- Part I explains our research on state-of-the-art on energy-efficient data centers.
  - Chapter 2 presents a taxonomy and survey, highlighting the main optimization techniques within the state-of-the art.
  - Chapter 3 describes the problem statement and our positioning.
- Part II proposes our research on modeling power consumption including DVFS and thermal awareness.
  - Chapter 4 provides further information about thermal considerations on IT infrastructures.
  - Chapter 5 introduces our modeling approach.
  - Chapters 6, 7 and 8 further describe the modeling techniques proposed in these Ph.D. thesis.
- Part III explains our research on energy optimization at the data center level, taking advantage of the power models that we have derived from a holistic viewpoint.
  - Chapter 9 presents our power optimization based on DVFS-aware dynamic consolidation of virtual machines.
  - Chapter 10 explains our research on proactive power and thermal aware allocation strategies for optimizing Cloud data centers.
- Chapter 11 summarizes the conclusions derived from the research that is presented in this Ph.D. thesis, as well as the contributions to the state-of-the-art on energy efficiency in data centers. The chapter also includes a summary on future research directions.

Figure 1.6 provides the reader with an overview of the structure of this Ph.D. thesis and how the different chapters are organized. Chapters are arranged from lower to higher abstraction level, and describe the different modeling and optimization techniques developed in this work.

1.8 Publications

The results of this Ph.D. thesis, together with other related research have been published in international conferences and journals. In this section, we briefly present these publications and highlight the chapter in which the specific contributions can be found.

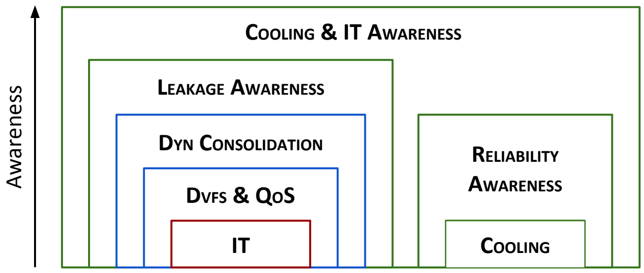


Figure 1.6: Overview of the Ph.D. thesis proactive proposal structure.

1.8.1 Journal papers

In terms of scientific publications, this Ph.D. thesis has generated the following articles in international journals:

- P. Arroba, J. M. Moya, J. L. Ayala, and R. Buyya, "Dynamic voltage and frequency scaling-aware dynamic consolidation of virtual machines for energy efficient cloud data centers", *Concurrency and Computation: Practice and Experience*, vol. 29, no. 10, 2017, ISSN: 1532-0634. DOI: `10.1002/cpe.4067` [JCR Q2 IF=0.942] *(Chapter 9 of this Ph.D. thesis)*

- P. Arroba, J. L. Risco-Martín, M. Zapater, J. M. Moya, and J. L. Ayala, "Enhancing regression models for complex systems using evolutionary techniques for feature engineering", *J. Grid Comput.*, vol. 13, no. 3, pp. 409–423, Sep. 2015, ISSN: 1570-7873. DOI: `10.1007/s10723-014-9313-8` [JCR Q2 IF=1.561] *(Chapter 8 of this Ph.D. thesis)*

- P. Arroba, J. L. Risco-Martín, M. Zapater, J. M. Moya, J. L. Ayala, and K. Olcoz, "Server power modeling for run-time energy optimization of cloud computing facilities", *Energy Procedia*, vol. 62, pp. 401 –410, 2014, ISSN: 1876-6102. DOI: `10.1016/j.egypro.2014.12.402` *(Chapter 6 of this Ph.D. thesis)*

1.8.2 Conference papers

Also, this Ph.D. thesis has generated the following articles in international peer-reviewed conferences:

- P. Arroba, J. M. Moya, J. L. Ayala, and R. Buyya, "Proactive power and thermal aware optimizations for energy-efficient cloud computing", in *Design Automation and Test in Europe. DATE 2016, Dresden, Germany*, Ph.D Forum, Mar. 2016 *(Chapter 1 of this Ph.D. thesis)*

- P. Arroba, J. M. Moya, J. L. Ayala, and R. Buyya, "Dvfs-aware consolidation for energy-efficient clouds", in *2015 International Conference on Parallel Architecture and Compilation, PACT 2015, San Francisco, CA, USA, 2015*, 2015, pp. 494–495. DOI: `10.1109/PACT.2015.59` [Core A conference] *(Chapter 9 of this Ph.D. thesis)*

- P. Arroba, J. L. Risco-Martín, M. Zapater, J. M. Moya, J. L. Ayala, and K. Olcoz, "Server power modeling for run-time energy optimization of cloud computing facilities", in *2014 International Conference on Sustainability in Energy and Buildings, Cardiff, Wales, UK*, Jun. 2014. DOI: `10.1016/j.egypro.2014.12.402` *(Chapter 6 of this Ph.D. thesis)*

- P. Arroba, M. Zapater, J. L. Ayala, J. M. Moya, K. Olcoz, and R. Hermida, "On the Leakage-Power modeling for optimal server operation", in *Innovative architecture for future generation high-performance processors and systems (IWIA 2014), Hawaii, USA*, 2014 *(Chapter 4 of this Ph.D. thesis)*

- P. Arroba, J. L. Risco-Martín, M. Zapater, J. M. Moya, J. L. Ayala, and K. Olcoz, "Evolutionary power modeling for high-end servers in cloud data centers", in *Mathematical Modelling in Engineering & Human Behaviour, Valencia, Spain*, 2014 *(Chapter 8 of this Ph.D. thesis)*

1.8.3 Other publications

Finally, the author has also contributed in the following articles in international peer-reviewed conferences, journals and book chapters not specifically related to the contents of this Ph.D. thesis:

- M. T. Higuera-Toledano, J. L. Risco-Martin, P. Arroba, and J. L. Ayala, "Green adaptation of real-time web services for industrial cps within a cloud environment", *IEEE Transactions on Industrial Informatics*, vol. PP, no. 99, pp. 1–1, 2017, ISSN: 1551-3203. DOI: 10.1109/TII.2017.2693365 [JCR Q1 IF=4.708]

- M. Zapater, J. L. Risco-Martín, P. Arroba, J. L. Ayala, J. M. Moya, and R. Hermida, "Runtime data center temperature prediction using grammatical evolution techniques", *Applied Soft Computing*, Aug. 2016, ISSN: 15684946. DOI: 10.1016/j.asoc.2016.07.042 [JCR Q1 IF=2.857]

- I. Aransay, M. Zapater, P. Arroba, and J. M. Moya, "A trust and reputation system for energy optimization in cloud data centers", in *2015 IEEE 8th International Conference on Cloud Computing*, 2015, pp. 138–145. DOI: 10.1109/CLOUD.2015.28

- M. Zapater, P. Arroba, J. L. Ayala, K. Olcoz, and J. M. Moya, "Energy-Aware policies in ubiquitous computing facilities", in *Cloud Computing with e-Science Applications*, CRC Press, Jan. 2015, pp. 267–286, ISBN: 978-1-4665-9115-8. DOI: 10.1201/b18021-13

- M. Zapater, P. Arroba, J. L. Ayala, J. M. Moya, and K. Olcoz, "A novel energy-driven computing paradigm for e-health scenarios", *Future Generation Computer Systems*, vol. 34, pp. 138 –154, 2014, Special Section: Distributed Solutions for Ubiquitous Computing and Ambient Intelligence, ISSN: 0167-739X. DOI: 10.1016/j.future.2013.12.012 [JCR Q1 IF=2.786]

- J. Pagán, M. Zapater, O. Cubo, P. Arroba, V. Martín, and J. M. Moya, "A Cyber-Physical approach to combined HW-SW monitoring for improving energy efficiency in data centers", in *Conference on Design of Circuits and Integrated Systems*, Nov. 2013

- M. Zapater, P. Arroba, J. M. Moya, and Z. Banković, "A State-of-the-Art on energy efficiency in today's datacentres: researcher's contributions and practical approaches", *UPGRADE*, vol. 12, no. 4, pp. 67–74, 2011, Awarded best paper of the year 2011, ISSN: 1684-5285

1.9 Grants and Research Projects

The author of this dissertation has been awarded the following research grants:

- Mobility Grant from the European Commission under the Erasmus Mundus Euro-Asian Sustainable Energy Development programme, for a 6-month research stay at the Matsuoka Lab. at the Tokyo Institute of Technology (Tokyo, Japan) *[9,000€ awarded on a competitive basis, January 2016]*.

- Mobility Grant from the European Network of European Network of Excellence on High Performance and Embedded Architecture and Compilation (HiPEAC), for a 3-month research stay at the Cloud Computing and Distributed Systems (CLOUDS) Lab. at the University of Melbourne (Melbourne, Australia) *[5,000€ awarded on a competitive basis, May 2014]*

Moreover, the research work developed during this Ph.D. thesis was partially funded by the following R&D projects and industrial contracts:

- GreenStack project: This project focuses on the development of energy optimization policies in OpenStack, providing it with awareness of the behavior of the data center to accurately anticipate actual needs. Funded by the National R&D&i Programme for Societal Challenges, RETOS-COLABORACION of the Spanish Ministry of Economy and Competitiveness (MINECO).

- GreenDISC project: development of HW/SW Technologies for Energy Efficiency in Distributed Computing Systems. The project proposes several research lines that target the power optimization in computing systems. Funded by the National Programme for Fundamental Research Projects of Spanish Ministry of Economy and Competitiveness (MINECO).

- LPCloud project: This project focuses on the optimum management of low-power modes for Cloud computing. Funded by the National Programme for Public-Private Cooperation, INNPACTO of MINECO.

- CALEO project: Thermal-aware workload distribution to optimize the energy consumption of data centres. Funded by Centro para el Desarrollo Tecnológico e Industrial (CDTI) of Spain.

The following part of this Ph.D. thesis presents a taxonomy and survey, providing current approaches on energy-efficient data centers that are analyzed from the perspective of understanding the specific issues of each abstraction level. We also present our problem statement and positioning, giving further information on how our research contributes to the existing state-of-the-art.

Part I

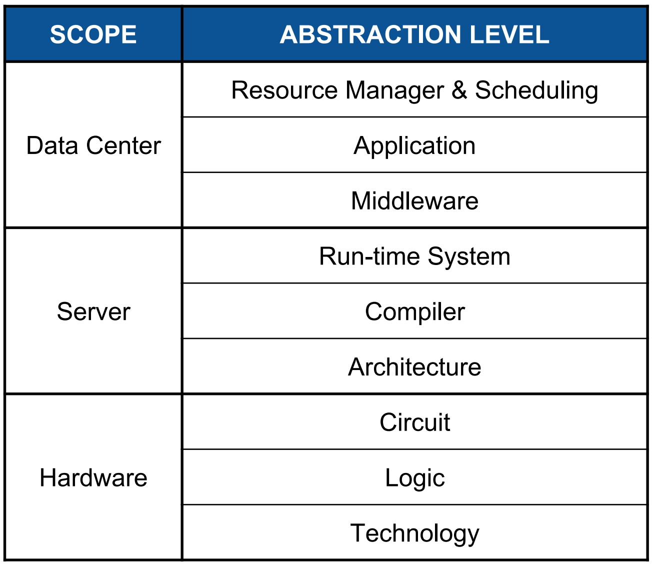
State-of-the-art on Energy-Efficient Data Centers

2. Taxonomy and Survey on Energy-Efficient Data Centers

> *"Man has always learned from the past. After all, you can't learn history in reverse!"*
>
> — Archimedes, *The Sword in the Stone. Disney*

Due to the impact of energy-efficient optimizations in an environment that handles so impressive high figures as data centers, many researchers have been motivated to focus their academic work on obtaining solutions for this issue. Therefore, the survey in this chapter aims for the analysis of the problem from different abstraction levels to draw conclusions about the evolution of energy efficiency. Practical approaches are analyzed from the perspective of understanding the specific issues of each abstraction level. The study examines the technology, logic and circuitry that comprise the hardware, followed by the analysis of the server scope that takes into account the architecture, type of compilation and runtime-system, and concluding with the analysis of the entire data center as shown in Table 2.1. Data center scope is discussed further as represents the main challenge of our work, being splited into middleware-level, explaining the benefits of virtualized systems, application-level, and resource management and scheduling-level.

Table 2.1: Main abstraction levels highlighted by hardware, server and data center scope



SCOPE	ABSTRACTION LEVEL
Data Center	Resource Manager & Scheduling
	Application
	Middleware
Server	Run-time System
	Compiler
	Architecture
Hardware	Circuit
	Logic
	Technology

The following survey analyzes today's energy-efficient strategies from both the IT power-aware and the thermal-aware perspectives. The proposed taxonomy classifies current research on efficient data centers, helping to identify new open challenges and promoting further progress towards sustainability.

2.1 Power-Aware Practical Approaches

2.1.1 Hardware Efficiency

The main challenge within energy-efficient hardware design consists in reaching a compromise between the QoS and energy consumption so that the performance is not degraded. To achieve this goal, the different components of the system should be analyzed, as well as the interaction between them when they operate as a whole.

Technology Level

From vacuum tubes to modern transistors, miniaturization of electronic devices by reducing the energy requirements and manufacturing costs have been the major instrument for the progress and sustainability of technologies. Transistor density of new systems is increasing due to the miniaturization per Moore's law so, power delivery and temperature management have become critical issues for computing. However, the main source of inefficiency is due to the leakages caused by submicron technologies. The main achievements in energy-efficiency at the technology level focus on technology scaling, voltage and frequency reduction, chip layout optimization and capacitances minimization. These techniques have been applied to technologies based on CMOS transistors achieving energy savings of about 18% [43].

Duarte et al. [44] show that migration from scaled down technologies reduces energy consumption considerably. For $0.07\mu m$, $0.05\mu m$ and $0.035\mu m$, savings obtained were 8%, 16% and 23% respectively. Timing speculation, which consists of increasing frequency at constant voltage and correcting resultant faults, is studied by Kruijf et al. [45] to achieve energy savings. Their outcomes show energy savings by 13% for high-performance low-power CMOS and by 32% using ultra-low power CMOS technology.

The design of more efficient chip layouts is becoming an important target in terms of energy savings as technology scales down. Muttreja et al. [46] combine fin-type field-effect transistors (FinFETs) with threshold voltage control through multiple supply voltages (TCMS) to explore the synthesis of low power interconnections for 32nm technology and beyond. Their work achieves power savings of about 50.41% by reducing the layout area by $9,17\%$.

Logic Level

Logic-level design for energy efficiency mainly focuses on optimizing the switching activity. Minimizing the switching capacitance directly optimizes the dynamic power consumption by reducing the energy per transition on each logic device [47]. Clock management, asynchronous design and optimized logic synthesis also provide power savings by using accurate delay modeling and minimization of charging loads, taking into account slew rates and considering the dynamic power dissipation occurred due to short-circuit currents [48].

Power gating, also known as MTCMOS or Multi-Threshold CMOS, allows to put to sleep transistors by cutting off the power supply from a circuit when it is not switching, as well as disconnecting the ground lines from the cells eliminating leakage. Madan et al. [49] have devised a robust management policy of guarded power gating, suggesting efficient guard mechanisms that guarantee power savings. Per-core power gating (PCPG) is proposed in [50] as a power management solution for multi-core processors. PCPG allows cutting voltage supply to selected cores, resulting on savings in energy consumption up to 30% in comparison to DVFS, also reducing the leakage power for the gated cores to almost zero.

Clock gating provides substantial energy savings by disabling parts of a circuit avoiding switch states. PowerNap [51] reduces power of idle components by using clock gating dropping power requirements about 20%. Voltage scaling significantly reduces the circuit consumption due to the quadratic relationship between supply voltage and dynamic power consumption. Henrty et al. [52] present Sense Amplier Pass Transistor Logic, a new logic style designed for ultra low voltage that results in 44% drop in energy consumption.

Circuit Level

The major challenges in circuit-level design are based on efficient pipelining and interconnections between stages and components. Pipelining technique is commonly used to boost throughput in high performance designs at the expense of reducing energy efficiency due to the increasing area and execution time. Seok et al. [53] present a strategy for optimizing energy efficiency based on an aggressive pipelining achieving energy savings around 30%. On the other hand, the pipelining technique proposed by Jeon et al. [54], reduces the logic depth between registers increasing pipeline stages when compared with conventional strategies for ultra low voltages. They achieve a 30% of energy savings also incorporating two-phase latches to provide better variation tolerance.

The interconnection-based energy consumption is another issue regarding nanometer CMOS regime due to the constant scaling of technology. Logic delays are drastically reduced while increasing interconnection delays results in larger repeater sizes and shorter flip-flop distances. The high-density and the complexity increment require large wiring densities leading to significant capacity couplings inducing the risk of crosstalk in adjacent wires. The problem is due to the capacitive coupling, which in deep sub-micron technology could be comparable to the ground line capacitance of the wire itself, increasing both crosstalk energy and bus dissipation. Brahmbhatt et al. [55] propose an adaptive bus encoding algorithm considering both self and coupling capacitance of the bus, thus improving energy savings by 24%. The research by Seo et al. [56] defines an edge encoding technique that achieves energy savings over 31% by an optimized bus design without overloaded latency, reducing the capacitive coupling. This work shows high-robustness to process variations concerning energy savings.

Another important technique used to achieve energy savings in the context of circuit level is based on charge recycling. The main objective of charge recycling is to reduce power consumption during transitions active-to-sleep and sleep-to-active by charge sharing between active circuit and sleep circuit capacitors. By using this technique, Pakbaznia et al. [57] achieve energy savings of 46%.

2.1.2 Server Efficiency

Currently, IT energy consumption in data centers is mainly determined by computing resources as CPU, memory, disk storage, and network interfaces. Compared with other resources, CPU is the main contributor to power consumption so many research focus on increasing its energy efficiency. However, the impact of memory, disk and network is not negligible in modern servers, also establishing these resources as important factors for energy optimization. This section will provide an overview of the architecture, compilation, and runtime-system approaches by analyzing relevant research in terms of reducing server consumption.

Architectural Level

Power savings are typically achieved at the architectural-level by optimizing the balance of the system components to avoid wasting power. Some techniques focus on reducing the complexity and increasing the efficiency by using strategies that use specific hardware resources to obtain higher performance in idle and active modes. In addition, DVFS is widely used to minimize energy consumption.

DVFS is by far the most used technique at the architectural-level as well as one of the currently most efficient methods to achieve energy savings, especially when combined with other strategies. DVFS is a technique that scales power according to the workload in a system by reducing both operating voltage and frequency. Reducing the operating frequency and the voltage slows the switching activity achieving energy savings, but it also slows down the system performance. The DVFS implementation on a CPU results in an almost linear relationship between its power and its frequency, taking into account that the set of states of frequency and voltage of the CPU is limited. Only by applying this technique on a server CPU, up to 34% energy savings can be reached as presented in the research proposed by Le Sueur et al. [58].

Since memory and disk consumption is becoming even more important and comparable with CPU power, DVFS has been used in current research to also improve energy efficiency in these IT resources. Deng et al. [59] apply DVFS to memory management achieving 14% of energy savings. Lee et al. [60] propose an efficiency solution that applies DVFS on an energy-efficient storage device design, thus achieving reductions in energy consumption around 20-30%.

However, DVFS is even more effective when combined with other energy-aware strategies. As reported in the research presented by Heo et al. [61], by combining DVFS with Feedback On/Off techniques, they achieve savings around 55% for highly loaded systems. Moreover, Per-Core Power Gating, used together with DVFS can reach around 60% savings [50]. The Energy-Delay-Product-aware DVFS technique proposed by Swaminathan et al. [62], dynamically adjusts the frequency of processor cores achieving improvements of up to 44% in energy efficiency for heterogeneous CMOS-Tunnel field-effect transistor (TFET) multicores.

Apart from DVFS, there are other strategies to optimize the energy consumption of servers in the architectural-level. The work presented by Seng et al. [63] studies the issue of wasteful CPU consumption from three different perspectives: the execution of unnecessary instructions, the speculation waste due to the instructions that do not commit their results, and the architectural waste, given by suboptimal sizing of processor structures. Their paper discusses that, by eliminating the sources of waste, reductions about 55% can be obtained for the processor energy.

The rise in the number of server cores together with virtualization technologies have widely increased storage demand, establishing this resource as one of the most important factors of energy optimization [64]. In the research presented by Zhu et al. [65], the authors present an off-line power-aware greedy algorithm that saves 16% more disk energy in comparison to the Least Recently Used algorithm. On the other hand, Pinheiro et al. [66] suggest the use of multi-speed disks, so each device could be slowed down to reach lower energy consumption during low-loaded periods showing energy savings of 23%.

Compiler Level

The challenge of compiler-level optimizations is to generate code that reduces the system energy consumption with or without a penalty in performance. Optimized compilers improve application performance by optimizing software code for better exploitation of underlying processor architecture, thus avoiding serious issues due to system failure. Compiler optimization mechanisms have been proposed to reduce power consumption by code optimization, profiling and annotation approaches.

Jones et al. [67] propose a compiler to an efficient placement of instructions performing energy savings in the instruction cache. The compiler comprises most frequently used instructions at the beginning of the binary, to be subsequently placed explicitly in the cache. Compared with the state-of-the-art, their work highlights a 59% in energy savings, in contrast to the 32% achieved by the hardware implementation. Fei et al. [68] propose the usage of source code transformations for operating system embedded programs to reduce their energy consumption, achieving up to 37.9% (23.8%, on average) energy reduction compared with highly compiler-optimized implementations.

Furthermore, the current compilation techniques are not able to exploit the potential of new system parallelism, such as the technologies based on multiple memory banks. Consequently, the compiler-generated code is far from being efficient. Shiue et al. [69] present an energy-aware high-performance optimized compiler that reaches 48.3% improvement in system's performance and average energy savings around 66.6%.

Run-time System Level

The run-time systems development is a very interesting matter within the scope of server energy efficiency, as it allows the monitoring of systems through both server logs or physical sensors that help to control the most relevant features. These systems also provide predictive implementations making anticipation possible for future workloads. As a consequence, statistical models can be introduced to allow run-time system-wide prediction of servers power consumption [70], [71].

Network dynamic management is also used in order to increase savings. Nedevschi et al. [72] combine the reduction of energy consumption in the absence of packets, via sleeping networking components, during idle times with an algorithm that adapts the rate of network operations according to the workload increasing latency in a controlled manner, without a perceptible loss increase.

Son et al. [73] present the implementation of a run-time system that improves disk power efficiency. In this approach the compiler provides key information to the run-time system to perform pattern recognition of disk access, so the system can act accordingly. Their outcomes reach among 19.4% and 39.9% savings when compared with the energy consumed by hardware or software based solutions. A prediction-based scheme for run-time adaptation is presented by Curtis-Maury et al. [74] improving both performance and energy savings by 14% and 40% respectively. In the field of multiprocessor system on chip, significant improvements have been reached at run-time system level. Yang et al. [75] combine the low complexity of the design-time scheduling with the flexibility of a run-time scheduling to achieve an energy efficiency of 72%.

2.1.3 Data center Efficiency

The advantages of Cloud computing lie in the usage of a technological infrastructure that allows high degrees of automation, consolidation and virtualization, which result in a more efficient management of the resources of a data center. The Cloud model allows working with a large number of users as well as concurrent applications that otherwise would require a dedicated computing platform. Cloud computing in the context of data centers, has been proposed as a mechanism for minimizing environmental impact.

Virtualization and consolidation increase hardware utilization (of up to 80% [15]) thus improving resource efficiency. Also, a Cloud usually consists of heterogeneous and distributed resources dynamically provisioned as services to the users, so it is flexible enough to find matches between these different parameters to reach performance optimizations. The Cloud market keeps growing and is expected to grow 10% in 2017 to reach to total $246.8 billion, up from $209.2 billion [16]. Also, according to Gartner, Cloud system infrastructure services and Cloud application services are projected to grow about a 36.8% and a 20.1% respectively in 2017.

It is expected that, by implementing this type of computing, energy consumption will be decreased by 31% by 2020, reducing CO_2 emissions by 28%. Even so, consumption and environmental impact are far from being acceptable [22]. Just for an average $100kW$ data center, a 7% in annual savings represents around US $5 million per year [23]. This context draws the priority of encourage research to develop sustainability policies at the data center scope, in order to achieve a social sense of responsibility while minimizing environmental impact.

Middleware Level

Within the scope of data centers, the middleware abstraction level includes the concepts of virtualization, consolidation and modification of the dynamic operating server set. Virtualization allows the management of the data center as a pool of resources, providing live migration and dynamic load balancing, as well as the fast incorporation of new resources and power consumption savings. Due to virtualization, a single node can accommodate simultaneously various virtual machines (also based on different operating systems) that can be dynamically started and stopped according to the system load that shares physical resources. By virtualizing a data center, savings in the electricity bill can be achieved of around 27%.

On the other hand, consolidation uses virtualization to share resources. Recent studies, as seen in the research presented by Lee et al. [76], highlight that power consumption in a server scales linearly with resource utilization. The task consolidation reduces energy consumption by increasing resource utilization, allowing multiple instances of operating systems to run concurrently on a single physical node. Energy savings reported by virtualization and consolidation are ranging from 20% to 75% by accommodating several VMs on the same host as can be seen in [77] and [78]. To reduce energy usage in data center infrastructures, some approaches have studied the consumption related to the basic operations of VMs such as boot or migrations. Lefèvre et al. [79] propose an energy-aware migration policy that achieves around 25% savings.

Moreover, the available computational resources of data centers are over-dimensioned for its typical incoming workload, and in very few load peaks they are used entirely. This is the main reason why a significant fraction of the infrastructure would be underutilized the most part of the time and thus an important amount of energy could be saved by turning off idle servers. Typically, an idle server consumes up to 66% of the total power consumption, so many efforts have been made in this area.

Chen et al. [80] focus on 4 algorithms that estimate the CPU usage in order to reduce the active server set by turning off machines. The researchers have achieved savings around 30%. The Limited Lookahead Control algorithm provided by Kusic et al. [81] also consists in a predictive control to select the active server set in virtualized data centers. This approach takes into account the control and the switching associated costs, including time and power consumed while a machine is powered up or down, achieving average savings of 22% while maintaining QoS goals.

The management of the active server set is especially useful considering service requirements fluctuation. Dynamic consolidation techniques presented by Beloglazov et al. [82] obtain a increase of 45% in energy savings. This approach takes into account migration costs as well as turning off idle machines of the server set. Similar policies consisting on server shutdown are used in the research proposed by Niles [83] and Corradi et al. [78] obtaining energy savings of 69% and 75% respectively.

Application Level

As the connectivity in personal and working environments is gaining importance, an increasing number of services with diverse application-level requirements are offered over the Internet [84]. The integration of application-level strategies together with server consolidation techniques is a major challenge to maximize energy savings [85]. The amount of resources that are required by the VMs are not always known a priori, so it is recommendable to have an application-level performance controller to adapt resources optimally to the studied variations in the application requirements. This issue is particularly accentuated when servers are overloaded and applications cannot access enough resources to operate efficiently. Therefore, it is a great recommendation to use consolidation algorithms that dynamically reallocate VMs on different physical servers throughout the data center in order to optimize resource utilization.

MapReduce, popularized by Google [86] is widely used in application-level energy-aware strategies due to simplified data processing for massive data sets in order to increase data center productivity [23]. When a MapReduce application is submitted, it is separated into multiple Map and Reduce operations so its allocation may influence the task performance [87]. Research by Wirtz et al. [88] achieved energy savings from 9% to 50% by combining this kind of applications with Hadoop frameworks for large clusters [89]. Also including dynamic operating server set techniques, research by Leverich et al. [90] and Maheshwari et al. [91] provide an energy efficiency improvement of 23% and 54% respectively.

On the other hand, PowerPack [92] is a proposal that involves circuit-level application profiling in order to determine how and where the power is consumed. Researchers also have developed a scheduling policy achieving energy savings of 12.1% by observing consumption profiles and their correlation with the application execution patterns. In future work they propose to adopt PowerPack for thermal profiling.

Resource Management and Scheduling Level

Resource management refers to the efficient and effective deployment of computational resources of the facility where they are required. The resource management techniques are used to allocate, in a spatio-temporal way, the workload to be executed in the data center thus optimizing a particular goal.

One of the key issues to consider in Cloud data centers, is the optimization of the current allocation of VMs. It is important to study which of them would get better energy-performance running on different hosts and therefore how the system should perform globally after migrations. Live migration of virtual machines involves the transfer of a virtual machine to another physical node at runtime without interrupting the service. Beloglazov et al. [93] present different migration procedures to manage the migration of VMs from overloaded hosts to avoid performance degradation. Their results show that energy consumption can be significantly reduced relatively to non-power aware and non-migration policies by 77% and 53% respectively maintaining up to 5.4% of SLA violations.

EnaCloud resource manager presented by Li et al. [94] helps to maintain the data center pool utilization at 90%, providing energy savings between 10% and 13%. Liu et al. [95] have designed GreenCloud, which enables online-monitoring of the system, live migration and searching the optimal placement of VMs by using a heuristic algorithm. This approach evaluates the costs of these migrations and then it turns on or off servers accordingly. Results show savings up to 27% obtaining a near-optimal solution in less than $300ms$ in a test environment.

2.2 Thermal-Aware Practical Approaches

Currently, data centers save lots of energy by cooling in an efficient manner. Hot-spots throughout the facilities are the main drawback according to system failures, and due to this factor, some data centers maintain very low room temperatures of up to 13$°C$ [15]. Most recently, data centers are turning towards new efficient cooling systems that make higher temperatures possible, around 24$°C$ or even 27$°C$.

2.2.1 Server Efficiency

The problem due to increasing power density of new technology has resulted in the incapacity of processors to operate at maximum design frequency, while transistors have become extremely susceptible to errors causing system failures [96]. Moreover, adding to this issue that system errors increase exponentially with temperature, new techniques that minimize both effects are required. As a result, optimized compilers are being developed taking into account both the thermal issues and power dissipation.

The register-file is one of the most affected components inside the processor due to its high temperature during activity. The development of compiler-managed register-file protection schemes is more efficient than hardware designs in terms of the power consumption. However, Lee et al. [97] propose a compile-time analysis as a solution to optimize further issues to significantly improve energy efficiency by an additional 24$\%$.

2.2.2 Data Center Efficiency

Within the data center scope, many of the reliability issues and system failures are given by the adverse effects due to hot spots. However one of the major problems is given by the heterogeneous distribution of the workload across the IT infrastructure, generating heat sources in localized areas. This fact causes the room cooling to be conditioned by hot spots temperature so CRAC units have to distribute air colder than necessary in many other areas thus avoiding failures in the whole data center server set. Therefore, in order to prevent the room overcooling, several techniques have been developed to optimize scheduling and resource management in data centers. The motivation in this area focuses on the efficient usage of resources and the thermal-aware policies to contribute to the energy efficiency and the sustainability of the facilities.

Wang et al. [98] present a Thermal Aware Scheduling Algorithm (TASA) that achieves simulation results of up to 12% in savings for the cooling system, representing about 6% of the power consumption of the entire infrastructure, which corresponds to a reduction of around 5000 kWh. Mukherjee et al. [99] use genetic algorithms to balance the workload to minimize thermal cross-interference, saving 40% of the energy when compared to first-fit placement techniques.

In addition, in the research presented by Tang et al. [100] authors have addressed the problem of minimizing the peak inlet temperature in data centers through task assignment. They use a recirculation model that minimizes cooling requirements, in comparison with other approaches. According to their results, the inlet temperature of the servers can be reduced from $2°C$ to $5°C$, saving about 20-30$\%$ of the cooling energy. The resource manager presented by Beloglazov et al. [101], while combined with techniques for minimizing the number of migrations, achieves energy savings about 83% due to the optimization of resource utilization, host temperature and network topology, also ensuring QoS.

2.3 Thermal and Power-Aware Practical Approaches

2.3.1 Server Efficiency

Joint thermal and power-aware strategies can be found within the server scope, considering fan control together with scheduling in a multi-objective optimization approach [102]. The work by Chan et al. [103] proposes a technique that combines both energy and thermal management technique to reduce the server cooling and memory energy costs. They propose a model to estimate temperature that uses electrical analogies to represent the thermal and cooling behavior of components. However, their work does not split the contributions of leakage and cooling power, so their minimization strategy is unaware of the leakage-cooling trade-offs.

2.3.2 Data Center Efficiency

By virtualizing a data center, savings in the electricity bill can be achieved of around 27%. However, by combining improvements in power of both computation and cooling devices, savings have the potential to reach about 54% [83]. This is the main challenge to reduce data center energy from a global perspective.

On its own, virtualization has the potential of minimizing the hot-spot issue by migrating VMs. Migration policies allow to distribute the workload also considering temperature variations during run-time without stopping task execution. Some Cloud computing solutions, such as introduced by Li et al. [104], have taken into account the dependence of power consumption on temperature, due to fan speed and the induced leakage current.

Abbasi et al. [105] propose heuristic algorithms to address this problem. Their work presents the data center as a distributed Cyber Physical System (CPS) in which both computational and physical parameters can be measured with the goal of minimizing energy consumption. However, the validation of these works is kept in the simulation space, and solutions are not applied in a real data center scenario.

The current research in the area of joint workload and cooling control strategies is not addressing the issue of proactive resource management with the goal of total energy reduction. Instead, techniques so far either rely on the data room thermal modeling provided by Computational Fluid Dynamics (CFD) software, or just focus on measuring inlet temperature of servers. However, as opposed to the holistic approach proposed in this dissertation, the models at the data room level do not monitor the CPU temperature of servers nor adjusting the cooling proactively or performing a joint workload and cooling management during run-time for arbitrary workloads.

2.4 State of the Art Discussion

Research on problem solving focuses on understanding the processes to reach a solution within the context of available knowledge. Therefore, understanding the energy efficiency contributions becomes essential to reach optimal results in terms of savings in consumption, together with reducing the economic and environmental impact involved. In this section, the challenges related to the different abstraction levels are discussed. Table 2.2 summarizes the bibliography and the higher savings value organized by scope for each abstraction level of the presented taxonomy.

Table 2.2: Bibliography and savings for each abstraction level

SCOPE		References	Savings
DATA CENTER	RM & Scheduling	[93]- [95], [98]- [106], [104], [105]	83%
	Application	[84]- [92]	54%
	Middleware	[76]- [78]	75%
SERVER	Run-time System	[70]- [75], [102]- [103]	72%
	Compiler	[67]- [69], [97]	67%
	Architecture	[58]- [66]	60%
HARDWARE	Circuit	[53]- [57]	46%
	Logic	[47]- [52]	44%
	Technology	[43]- [46]	32%

Temperature and power dissipation are the major issues faced by technology due to Moore's law scaling evolution, resulting in static and dynamic energy consumption, and noise. Thereby the energy efficiency achieved at this level, reaching 32%, is restricted to the integration density allowed by current technology. At logic-level the main strategies for reducing energy are based on the minimization of switching activity and capacity loads achieving savings about 44%. Interconnection-based consumption improvements, correcting logic delays, and prevention against capacitive coupling due to technology scaling are the main goals at the circuit level, obtaining reductions about 46% in energy consumption.

Regarding the server scope, architectural-level energy savings of 60% are achieved by avoiding unnecessary power waste and optimizing the balance of the system activity. Preventing hot spots within the server is one of the main challenges of optimized compilation. These approaches allow the processor to run at the best possible design conditions taking advantages of parallel architectures. These techniques represent energy savings up to 66.6%. Through server logs or hardware sensing, usage patterns can be recognized in order to monitor run-time systems and therefore to develop both reactive and proactive policies. Thus, run-time system optimizations, which are based on the optimal management of resources and the control of temperature variations, improve energy efficiency up to 72%.

By addressing the energy challenge of the data center as a whole, savings in consumption have assumed such proportions, which result in strong improvements both in economic and environmental impact. Due to the oversized data centers and the high consumption of idle and underutilized servers, the main target at middleware-level is to optimize the resource utilization of servers, even turning them on/off , thus balancing the workload. Virtualization impacts on these optimization approaches that, for Cloud computing, achieves energy savings up to 75%. However, migration of VMs introduces overheads that are not always considered. Application-level strategies mainly focus on the adaptation between the resources offered by the data center infrastructure and the variations of the applications requirements. This issue is even worse for overloaded servers and for workloads that dynamically fluctuate over time. At this abstraction level, practical approaches achieve energy savings of up to 54%.

The resource management and scheduling level focus on optimizing the allocation of VMs, by increasing the efficiency of reallocations and by modifying the operating server set. Maintaining QoS is also a major challenge for Cloud data centers as services are expected to be delivered by SLA. At this abstraction level, thermal-aware approaches are more relevant in current research, developing strategies that help to avoid the overcooling of the data room. Resource management and scheduling techniques achieve energy savings of up to 83%, thus improving the sustainability of these infrastructures.

This Ph.D. thesis proposes a novel holistic paradigm to consider the energy globally within the data center scope, from the IT to the cooling infrastructures that has not been applied before in the literature. Taking into account the energy contributions and the thermal impact at the different levels of abstraction would lead to more efficient global optimizations that are aware of the information available from the different subsystems. The following chapter presents our problem statement and positioning, giving further information on how the present research offers a new optimization paradigm, contributing to the existing state-of-the-art.

3. Problem Statement and Positioning

"Venture outside your comfort zone.
The rewards are worth it."
— Rapunzel, *Tangled, Disney*

The survey presented in the previous section studies the main energy-efficient strategies from both the power-aware and the thermal-aware perspectives. This section analyzes the trade-offs between energy efficiency and abstraction levels in order to help to identify new open challenges at the data center scope. The box plot in Figure 3.1 graphically depicts the range of energy savings by abstraction level for the practical approaches in Table 2.2.

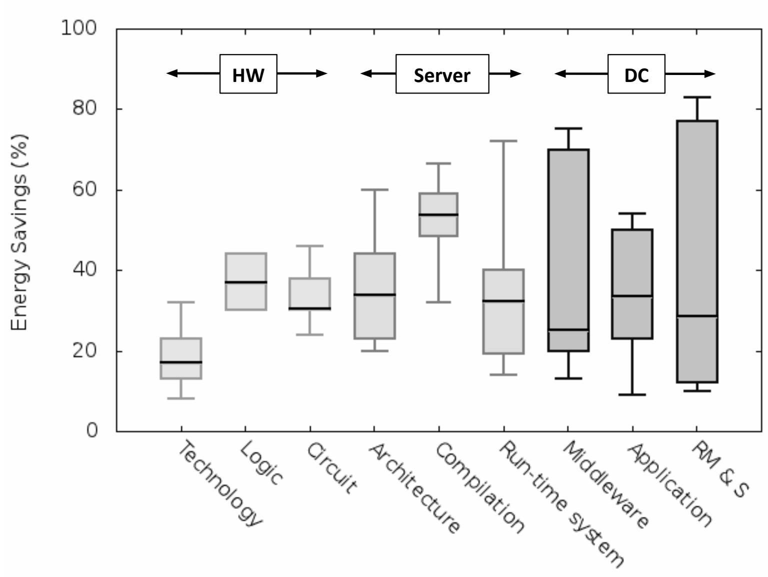


Figure 3.1: Box-and-whisker diagram representing the range of energy savings for the proposed taxonomy

For each abstraction level, the input data is a numeric vector including all the energy savings' values that can be found in the previous section. On each box, the central mark provides the median value, and the bottom and top edges of the box specify the 25^{th} and 75^{th} percentiles, respectively. The whiskers' edges represent the most extreme data points not considered outliers. For the values considered in our research, no outliers have been found.

3.0.1 The holistic perspective

Regarding Figure 3.1, some assessments can be made. First, we can observe that for increasing abstraction levels, the maximum savings value also improves. Data center energy consumption depends on the contribution of diverse subsystems and how they interact. Thus, the energy performance of these facilities is based on the complex relationships between many different parameters. As the energy challenge is tackled from a more global perspective, the optimization strategies can be aware of the status of a larger set of elements. This trend shows that, higher abstraction levels help to take better decisions that result in higher savings, as they have deeper knowledge about subsystems interactions and how they affect to the global consumption. So, this Ph.D. thesis will face the energy efficiency issue from a high-level perspective.

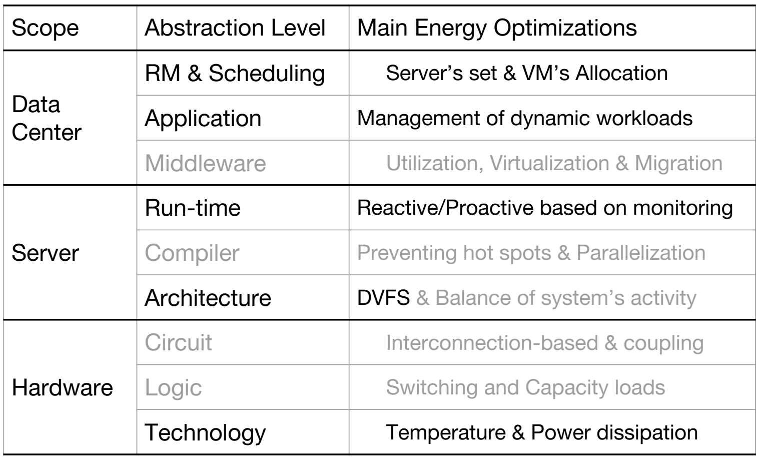
However, research found in the area of application-level optimizations shows energy savings that are lower than expected, according to their level of abstraction. These results indicate that further research on VM consolidation for dynamic workloads is required to better understand the problem, proposing new optimizations to increase the current energy savings at this scope. Thus, our work will emphasize on analyzing the features of dynamic workloads that best describe real environments.

Also, we find that Architecture and Run-time system levels have the potential to increase energy efficiency, as there exists a significant gap between the maximum and the average efficiency obtained. Due to this, our research will also focus on DVFS and proactive techniques based on runtime monitoring.

Our contributions to the state-of-the-art

Table 3.1 presents current state-of-the-art approaches and highlights those techniques used in our research, which we consider to have the highest potential for improve energy efficiency in Cloud data centers when applying our holistic paradigm.

Table 3.1: Current state-of-the-art approaches for energy efficiency



Scope	Abstraction Level	Main Energy Optimizations
Data Center	RM & Scheduling	Server's set & VM's Allocation
	Application	Management of dynamic workloads
	Middleware	Utilization, Virtualization & Migration
Server	Run-time	Reactive/Proactive based on monitoring
	Compiler	Preventing hot spots & Parallelization
	Architecture	DVFS & Balance of system's activity
Hardware	Circuit	Interconnection-based & coupling
	Logic	Switching and Capacity loads
	Technology	Temperature & Power dissipation

The thermal-aware strategies are becoming relevant when applied to reduce the energy consumption of cooling infrastructures. These policies help to avoid hot-spots within the IT resources so the cooling temperature can be increased, resulting in lower energy requirements while maintaining safe operation ranges. However, current approaches do not incorporate the impact of leakage consumption found at the technology level when controlling the cooling set point temperature at the data center level. This power contribution, which grows for increasing temperatures, is neither considered when optimizing allocation strategies in terms of energy.

Thus, deriving power models that incorporate this information at different abstraction levels, which impact on power contributors, helps to find the relationships required to devise global optimizations that combine power and thermal-aware strategies in a holistic way. In our work, we will focus on the combination of strategies that are aware of both IT and cooling consumption, also taking into account the thermal impact.

Modeling the power consumption is crucial to anticipate the effects of novel optimization policies to improve data center efficiency from a holistic perspective. The fast and accurate modeling of complex systems is a relevant target nowadays. Modeling techniques allow designers to estimate the effects of variations in the performance of a system. Data centers, as complex systems, present non-linear characteristics as well as a high number of potential variables. Also, the optimal set of features that impacts on the system energy consumption is not well known as many mathematical relationships can exist among them.

Analytical models, as closed form solution representations, require specific knowledge about the different contributions and their relationships, becoming hard and time-consuming techniques for describing complex systems. Complex systems comprise a high number of interacting variables, so the association between their components is hard to extract and understand as they have non-linearity characteristics [107]. Also, input parameter limitations are barriers associated to classical modeling for these kind of problems. Therefore, new modeling techniques are required to find consumption models that take into account the influence of temperature on both IT and cooling. This work is intended to offer new power models that also take into account the contributions of non-traditional parameters such as temperature. So, deriving fast and accurate models will allow us to combine both power and thermal-aware strategies.

In this Ph.D. thesis we develop a novel methodology based on a Grammatical Evolution metaheuristic for the automatic inference of accurate models. Hence, we propose a methodology that considers all these factors thus providing a generic and effective modeling approach that could be applied to numerous problems regarding complex systems, where the number of relevant variables or their interdependence are not known. Our methodology allows to derive complex models without designer's effort, automatically providing an optimized set of features combined in a model that best describe the power consumption of servers.

Modeling the main contributors to the consumption of the whole data center, including information about the impact on power at different abstraction levels, offers estimations that can be used during runtime in a proactive manner. Based on the information provided by the models on how optimization techniques impact on both IT and cooling infrastructures, new local and global optimization strategies may be devised to reduce the whole consumption of the data center. In this research we propose proactive optimization approaches based on best fit decreasing algorithms and simulated annealing metaheuristics to reduce the global energy consumption, both IT and cooling contributions, while maintaining the QoS.

The new holistic paradigm proposed in this work focuses on considering the energy globally and proactively for the first time in literature. So, all the data center elements are aware of the evolution of the global energy demand and the thermal behavior of the room. Decisions are based on information from all available subsystems to perform energy optimizations.

The work presented in the following part of this Ph.D. thesis profusely describes our strategies to model power consumption that include both DVFS and thermal awareness. In this research we detect the need of addressing leakage power and focus on the different modeling methodologies to provide power models that outperform the state-of-the-art.

Part II

Modeling Power Consumption: DVFS and Thermal Awareness

4. Thermal considerations in IT infrastructures

"Nothing shocks me, I'm a scientist."
— Indiana Jones, *Indiana Jones and the Temple of Doom*

4.1 Introduction

Leakage power consumption is a component of the total power consumption in data centers that is not traditionally considered in the set point temperature of the room. However, the effect of this power component, increased with temperature, can determine the savings associated with the careful management of the cooling system, as well as the reliability of the system. The work presented in this chapter detects the need of addressing leakage power in order to achieve substantial savings in the energy consumption of servers.

Interestingly, the key issue on how to control the set point temperature, at which to run the cooling system of a data center, is still to be clearly defined [108]. Data centers typically operate in a temperature range between 18° C and 24° C, but we can find some of them as cold as 13° C degrees [109], [110]. Due to the lack of scientific data in the literature, these values are often chosen based on conservative suggestions provided by the manufacturers of the equipment.

Some authors estimate that increasing the set point temperature by just one degree can reduce energy consumption by 2 to 5 percent [109], [111]. Microsoft reports that raising the temperature from two to four degrees in one of its Silicon Valley data centers saved $250,000 in annual energy costs [110]. Google and Facebook have also been considering increasing the temperature in their data centers [110].

Power consumption in servers can be estimated by the summation of the dynamic power consumption of every active module, dependent on the computing activity, and the static power consumption, present even if the server is idle. However, a power leakage is also present, and it is strongly correlated with the integration technology. In particular, leakage power consumption is a component of the total power consumption in data centers that is not traditionally considered in the set point temperature of the room. The effect of this power component, increased with temperature, can determine the savings associated with the careful management of the cooling system, as well as the reliability of the system itself.

The work presented in this chapter detects the need of addressing leakage power in order to achieve savings in the energy consumption of servers and makes the following contributions:

- we establish the need of considering leakage power consumption and its dependence with temperature for modern data centers;

- we detect the impact of leakage power in the total power consumption of high-performance servers;

4.2 Background on data center power modeling

The main contributors to the energy consumption in a data center are the IT power, i.e. the power drawn by servers in order to run a certain workload, and the cooling power needed to keep the servers within a certain temperature range that ensures safe operation. Traditional approaches have tried to reduce the cooling power of data center infrastructures by increasing the supply temperature of CRAC units. However, because of the direct dependence of leakage current with temperature, the leakage-temperature trade-offs at the server level must be taken into account when optimizing energy consumption.

In this section we show the impact of these trade-offs on the total energy consumption of the data center, as well as how the ambient room temperature affects the cooling power. This fact can be exploited to optimize the power consumption of the infrastructure as a whole.

4.2.1 Computing power

Current state-of-the-art resource management and selection techniques were considering only the dynamic power consumption of servers when allocating tasks or selecting machines. Moreover, the devised power models have not traditionally included the impact of leakage power consumption and its thermal dependence, driving to non-optimal solutions in their energy optimization plans.

Dynamic consumption has historically dominated the power budget. But when scaling technology below the $100nm$ boundary, static consumption has become much more significant, being around 30- 50% [112] of the total power under nominal conditions. Moreover, this issue is intensified by the influence of temperature on the leakage current behavior. With increasing temperature, the on-current of a transistor is reduced slightly. However the reduction of the threshold voltage is not sufficient to compensate the decreased carrier mobility and has a strong exponential impact on leakage current. Hence, the increasing temperature reduces I_{on}/I_{off} ratio, so that static consumption becomes more relevant in the total power budget as shown in Figure 4.1.

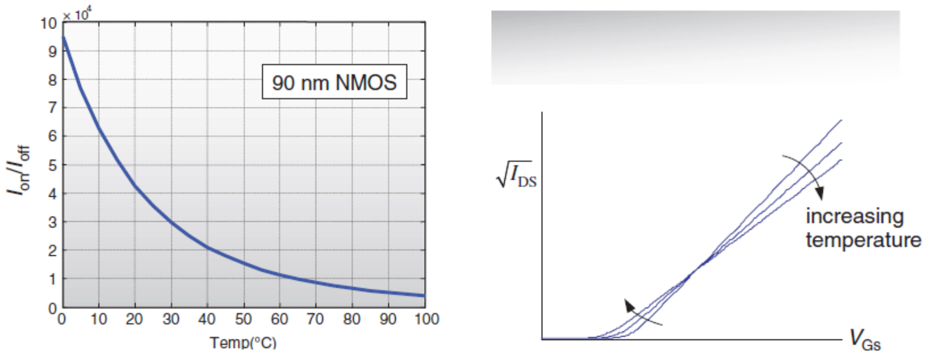


Figure 4.1: Performance variations with temperature [24]

There are various leakage sources in devices such as gate leakage or junction leakage but at present, sub-threshold leakage is the most important contribution in modern designs. This phenomenon occurs because, when the gate voltage is lowered under the threshold voltage, the transistor does not turn off instantly, entering a sub-threshold regime also known as “weak inversion”. During this process drain-source current increases exponentially in terms of V_{GS}

voltage as seen in Figure 4.2.

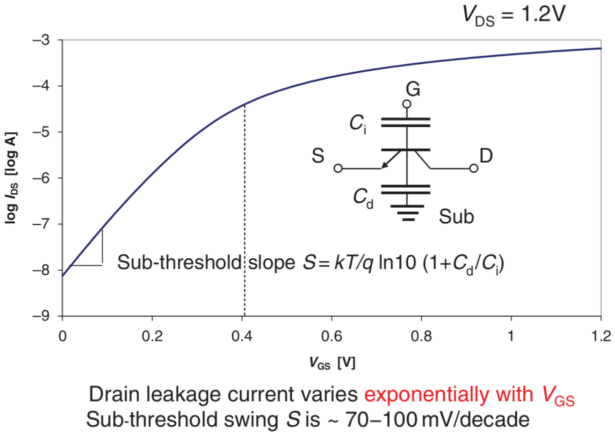


Figure 4.2: The transistor in "weak inversion" [24]

Also leakage current increases exponentially with a linear reduction in threshold voltage as can be seen in the example 4.3, where leakage current at $V_{GS} = 0$ for a lower threshold transistor ($V_{TH} = 0.1V$) is approximately four orders of magnitude higher than that for a high threshold ($V_{TH} = 0.4V$) device.

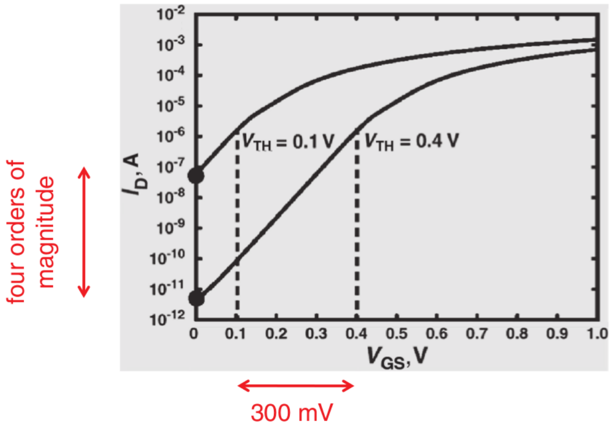


Figure 4.3: Comparison between leakage current at different thresholds [24]

The effect of Drain-Induced barrier lowering (DIBL) aggravates the problem because the threshold is reduced approximately linearly with V_{DS} voltage as seen in equation 4.1.

$$V_{TH} = V_{TH0} - \lambda_d \cdot V_{DS} \tag{4.1}$$

This issue occurs in short-channel devices, where the source-drain distance is comparable to the widths of depletion regions. This means that the DIBL effect makes the threshold voltage to become a variable that varies with the signal, so the drain voltage can modulate the threshold. Figure 4.4 shows this effect for different drain-source channel lengths.

The current that is generated in a MOS device due to leakage is the one shown in equation 4.2.

$$I_{leak} = I_s \cdot e^{\frac{V_{GS}-V_{TH}}{nkT/q}} \cdot (1 - e^{\frac{V_{ds}}{kT/q}}) \tag{4.2}$$

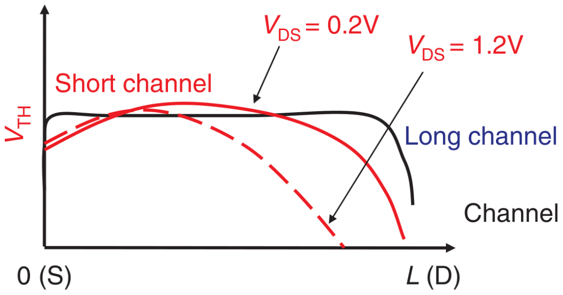


Figure 4.4: Threshold variations due to different drain voltages in short-channel devices [24]

Research by Rabaey [24] shows that if $V_{DS} > 100mV$ the contribution of the second exponential is negligible, so the previous formula can be rewritten as in Equation 4.3:

$$I_{leak} = I_s \cdot e^{\frac{V_{GS}-V_{TH}}{nkT/q}} \tag{4.3}$$

where technology-dependent parameters can be grouped together to obtain the formula in Equation 4.4:

$$I_{leak} = B \cdot T^2 \cdot e^{\frac{V_{GS}-V_{TH}}{nkT/q}} \tag{4.4}$$

where B defines a constant that depends on the manufacturing parameters of the server.

According to this formulation, the effect of leakage currents due to temperature may be devised, at the server scope, in order to accurately model the power contribution of IT together with cooling variations.

4.2.2 Cooling power

The cooling power is a major contributor to the overall electricity bill in data centers, consuming over 30% of the power budget in a typical infrastructure [27]. In an air-cooled data center room, servers mounted in racks are arranged in alternating cold/hot aisles, with the server inlets facing cold air and the outlets creating hot aisles. The CRAC units pump cold air into the data room's cold aisles and extract the generated heat. The efficiency of this cycle is generally measured by the COP or coefficient of performance. The COP is a dimensionless value defined as the ratio between the cooling energy produced by the air-conditioning units (i.e. the amount of heat removed) and the energy consumed by the cooling units (i.e. the amount of work to remove that heat), as shown in Equation 4.5.

$$COP_{MAX} = \frac{\text{output cooling energy}}{\text{input electrical energy}} \tag{4.5}$$

Higher values of the COP indicate a higher efficiency. The maximum theoretical COP for an air conditioning system is described by Carnot's theorem as in Equation 4.6:

$$COP_{MAX} = \frac{T_C}{T_H - T_C} \tag{4.6}$$

where T_C is the cold temperature, i.e. the temperature of the indoor space to be cooled, and T_H is the hot temperature, i.e. the outdoor temperature (both temperatures in Celsius). As the difference between hot and cold air increases, the COP decreases, meaning that the air-conditioning is more efficient (consumes less power) when the temperature difference between the room and the outside is smaller.

According to this, one of the techniques to reduce the cooling power is to increase the COP by increasing the data room temperature. We will follow this approach to decrease the power wasted on the cooling system to a minimum, while still satisfying the safety requirements of the data center operation.

4.3 Experimental methodology

The experimental methodology in this section pursues the goal of finding leakage-temperature trade-offs at the server level, by means of measuring the power consumption of an enterprise server at different temperatures in a real data room environment where the air-conditioning can be controlled. After this, we will be able to evaluate the energy savings that could be obtained in a data center when our modeling strategy is applied.

4.3.1 Data room setup

To find if a dependence exists between temperature and power consumption in an infrastructure that resembles a real data center scenario, we install eight Sunfire V20z servers in a rack inside an air-cooled data room, with the rack inlet facing the cold air supply and the outlet facing the heat exhaust. We selected this type of servers because they present an enterprise server architecture used in current infrastructures that exhibits leakage-dependent properties with temperature. The air conditioning unit mounted in the data room is a Daikin FTXS30 unit, with a nominal cooling capacity of 8.8kW and a nominal power consumption of 2.8KW. We assume an outdoor temperature of 35°C and use the manufacturers technical data to obtain the COP curve depending on the room temperature [113]. This temperature is only used to estimate the energy savings based on the curve provided by the manufacturer and does not affect the experimental results.

As can be seen in Figure 4.5, as the room temperature and the heat exhaust temperature rise, approaching the outdoor temperature, the COP increases thus improving the cooling efficiency.

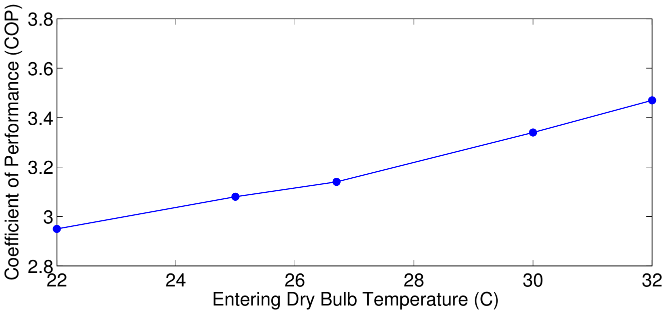
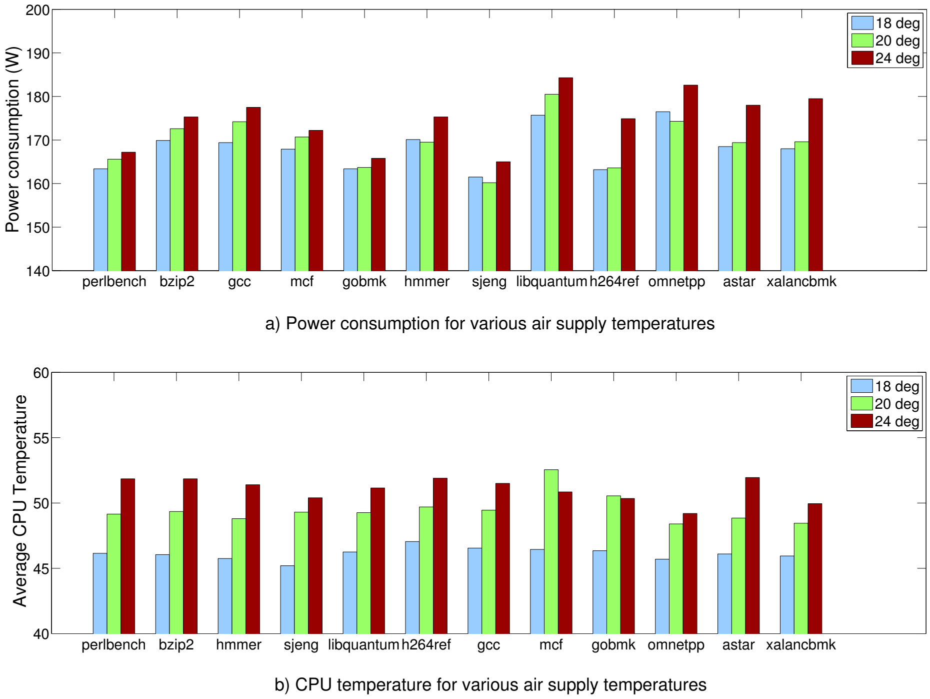


Figure 4.5: Evolution of the air-conditioning COP with room temperature

We monitor all the servers by means of the Intelligent Platform Management Interface (IPMI) tool to gather the server internal sensors and we use current clamps to obtain power consumption. We set the air supply temperature at various values ranging from 18°C to 24°C, and run from 1 to 4 simultaneous instances of the different tasks of the Standard Performance Evaluation Corporation (SPEC) CPU 2006 benchmark suite [114] in the servers of the data room. Our goal is to verify the leakage-temperature dependence, finding the maximum air-supply temperature that makes the servers work in the temperature region where leakage is negligible.

4.4 Results

We run the tasks of the SPEC CPU 2006 benchmark suite in the AMD servers under different data room conditions. In our experiments, we run from 1 to 4 instances of SPEC CPU in the AMD servers at different room temperatures of 18° C, 20° C, 22° C and 24° C. Figure 4.6a shows the power consumption values for two simultaneous instances of the SPEC CPU 2006 benchmark at an air supply set point temperature of 18° C, 20° C and 24° C, respectively. Figure 4.6b shows the CPU temperature for each of these tests under the same conditions.



a) Power consumption for various air supply temperatures

b) CPU temperature for various air supply temperatures

Figure 4.6: Power consumption and CPU temperature of SPEC CPU 2006 at different air supply temperatures

Because all other variables are constant (we removed server fans so there is no variable fan power), and as the measurement error with the current clamp is already controlled, the changes in the power consumption for each test can be due to the differences in ambient temperature. As can be seen in the plots, even though there are differences in the average CPU temperature between the 18° C and the 20° C case, for most of the benchmarks CPU temperature does not go above the 50° C, staying in the negligible leakage area. In fact, the power consumption differences between the 18° C and the 20° C case are in the range of ±5W, so we cannot consider them to be due to leakage, but to the inaccuracy of our current clamp. However, for the 24° C case, CPU temperatures raise above 50° C and power consumption for most of the benchmarks is considerably higher than in the 18° C scenario, achieving differences higher than 8W for gcc, libquantum, astar and xalancbmk benchmarks that represent an increase of about 4.5% in IT power. Thus, in this region we begin to observe temperature-dependent leakage.

The experimental results for our data room scenario show that if we allow temperature to rise above this 24° C barrier, the contribution of the leakage increases, thus increasing the computing power drawn by our infrastructure. However, for our data room configuration and under our workload, leakage is negligible in the 18° C - 24° C range and, thus, we can rise the ambient temperature within this range in order to reduce cooling power.

If we increase the air supply temperature from 18° C to 24° C, the room temperature increases and the COP varies (see Figure 4.5) from 2.95 to 3.47, increasing the energy efficiency of the cooling equipment and reducing the cooling power. This increase has a proportional impact on the energy savings of the infrastructure, leading to savings of 11.7% in cooling power as predicted by the curve.

4.5 Summary

Power consumption in servers can be estimated by the summation of the dynamic power consumption of every active module (which depends on the activity) and the static power consumption, but the leakage contribution to power that is strongly correlated with the integration technology may be considered. However, traditional approaches have never incorporated the impact of leakage power consumption in these models, and the noticeable values of leakage power consumption that appear at higher temperatures.

The work presented in this chapter detects the need of addressing leakage power in order to achieve substantial savings in the energy consumption of servers. In particular, our work shows that, by a careful detection and management of the impact of thermal-dependent leakage, energy consumption of the data-center can be optimized by a reduction of the cooling budget. Finally, we validate these facts with an experimental work that resembles the infrastructure of current enterprises. Our empirical results show that increasing the cooling setpoint temperature in 6° C reduces cooling power by 11.7%, but also increases IT power consumption in about 4.5%. The state-of-the-art only take into account cooling power reduction due to set point temperature increments. So, our research outperform current approaches by also considering the power increase in IT under these conditions.

The next chapter provides an analytical power model that provides the dependence of power consumption on temperature for the first time at server level. We also present a metaheuristic-based method, to optimize this analytical model, to enhance the accuracy between model estimations and power measurements.

5. Power Modeling Considerations

"I always like to look on the optimistic side of life, but I am realistic enough to know that life is a complex matter."

— Walt Disney

The management of energy-efficient techniques and proactive optimization policies requires a reliable estimation of the effects provoked by the different approaches throughout the data center. However, data center designers have collided with the lack of accurate power models for the energy-efficient provisioning and the real-time management of the computing facilities. Thus, new power models will facilitate the analysis of several parameters from the perspective of the power consumption, and they will allow us to devise efficient techniques for energy optimization.

In the last years, there has been a rising interest in developing simple techniques that provide basic power management for servers operating in a Cloud, i.e. reducing the active servers set by turning on and off servers, putting them to sleep or using DVFS to adjust servers' power states by reducing clock frequency. Many of these recent research works have focused on reducing power consumption in cluster systems [115]–[118]. In general, these techniques take advantage of the fact that application performance can be adjusted to utilize idle time on the processor to save energy [119]. However, their application in Cloud servers is difficult to achieve in practice as the service provider usually over-provisions its power capacity to address worst case scenarios. This often results in either waste of power or severe under-utilization of resources.

On the other hand, increasing server's utilization may increase the temperature of the infrastructure, so the effects on the overall power consumption of servers should be devised, considering the leakage currents that are correlated with temperature. Thus, it is critical to quantitatively understand the relationship between power consumption, temperature and load at the system level by the development of a power model that helps on optimizing the use of the deployed Cloud services.

5.1 Related Work

Currently the state of the art offers various power models. However the majority of these models are analytical, architecture-dependent and do not include the contribution of static power consumption, or the capability of switching the frequency modes. Also, no power model can be found that describes the dependence of server's power with temperature. Many authors develop linear regression models that present the power consumption of a server as a linear function of the CPU usage of that server [120]–[122].

Some other models can be found where server power is formulated as a quadratic function of the CPU usage [123]–[125]. Still, as opposed to ours, these models do not include the estimation of the static power consumption (which has turned to have a great impact due to the current server technology). Besides, these models have not been exploited in a multi-objective optimization methodology to minimize the power consumption of servers for Cloud services.

The approach presented by Bohra et al. [126] is based on a robust fitting technique to determine the power model, taking also into account the correlation between the total system power consumption and the utilization of the different resources. Our work follows a similar approach but it also incorporates the contribution of the static power consumption, its dependence on temperature, and the effect of applying DVFS techniques. We will show later that this is a critical upgrade of the model as it allows to improve the accuracy in over-loaded and top-notch servers.

Interestingly, one key aspect in the management of a data center is still not very well understood: controlling the ambient temperature at which the data center operates. Due to the lack of accurate power models, the effect of the ambient temperature set point on the power consumption of high-end servers has not been clearly analyzed. The experimental evaluation presented in this work has been performed in ambient temperatures ranging from 17°C to 27°C. This range follows nowadays' practice of operating at higher temperatures [127] being close to the limits recommended by the American Society of Heating Refrigerating and Air-Conditioning Engineers (ASHRAE). Increasing ambient temperature of data centers obtains energy savings in the cooling expense [128]. But the lack of detailed server power models, which consider the effect of a temperature increment on server consumption, prevents the application of thermal-aware optimization policies to reduce the power consumption of the facility as a whole.

A complex system can be described as an interconnected agents system exhibiting a global behavior that results from agents interactions [129]. Nowadays, the number of agents in a system grows in complexity, from data traffic scenarios to multi-sensor systems, as well as the possible interactions between them. Therefore, inferring the global behavior, not imposed by a central controller, is a complex and time-consuming challenge that requires a deep knowledge of the system performance. Due of these facts, new automatic techniques are required to facilitate the fast generation of models that are suitable for complex systems presenting a large number of variables. The case study presented in this work exhibits high complexity in terms of number of variables and possible traditional and non-traditional sources of power consumption.

A Grammatical Evolution based modeling technique has been also proposed by J.C. Salinas-Hilburg [130] to model specific contributions to power for CPU and memory devices for HPC workloads. However, to the best of our knowledge, this approach has not been yet used to model the power consumption of the whole server and also for workloads that vary significantly during runtime.

The work presented in this section outperforms previous approaches in the area of power modeling for enterprise servers in Cloud facilities in several aspects. Our different approaches provide the identification of accurate power models that are consistent with current architectures. We propose models that consider main power consumption sources that involve traditional and non-traditional parameters and that have an impact on the servers' power consumption. Thus, our power models consider the effects of nowadays' Cloud optimizations, being able to be used during runtime and under variable workload constrains.

5.2 Modeling considerations

Our modeling framework is presented in Figure 5.1. For modeling purposes, we run specific workloads in a real server in order to monitor their performance during runtime. We profile the applications according to those parameters that impact on power consumption, considering current state-of-the-art Cloud optimizations. The model features are provided or, in the case of automatic modeling approaches, different rules are considered for their generation. Then, the fitting objective helps to improve models accuracy.

By applying the different model techniques presented in this part of the research, we obtain accurate power models that could be used during runtime. These models, when incorporated to a simulation environment, help to analyze the impact of Cloud optimizations in a wider range of servers for traces from real Cloud infrastructures that are publicly available.

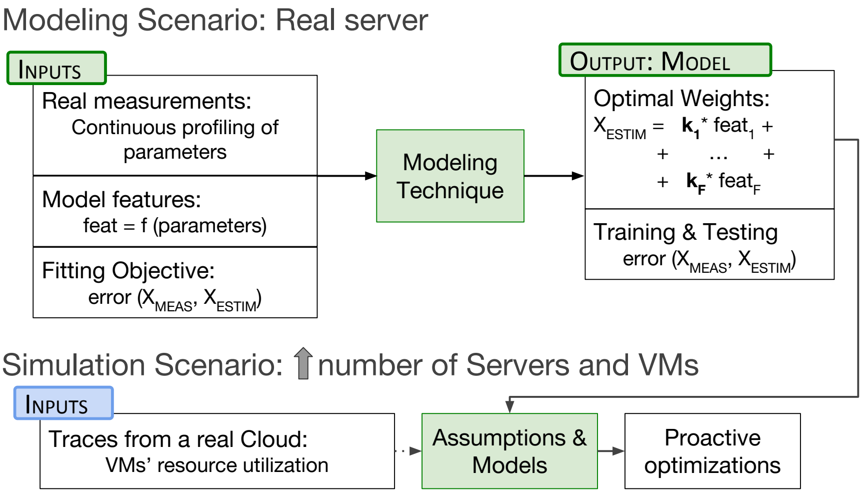

Modeling Scenario: Real server
Inputs
Real measurements:
Continuous profiling of parameters
Model features:
feat = f (parameters)
Fitting Objective:
error (X_{MEAS}, X_{ESTIM})
Modeling Technique
Output: Model
Optimal Weights:
$X_{ESTIM} = \mathbf{k_1} * feat_1 + \dots + \mathbf{k_F} * feat_F$
Training & Testing
error (X_{MEAS}, X_{ESTIM})
Simulation Scenario: ⬆ number of Servers and VMs
Inputs
Traces from a real Cloud:
VMs' resource utilization
Assumptions & Models
Proactive optimizations

Figure 5.1: Modeling vs. Model usage.

5.2.1 Data compilation

Data have been collected gathering real measures from a Fujitsu RX300 S6 server based on an Intel Xeon E5620 processor. This high-end server has a RAM memory of 16GB and is running a 64bit CentOS 6.4 Operating System (OS) virtualized by the Quick Emulator (QEMU) hypervisor. Physical resources are assigned to four Kernel-based Virtual Machine (KVM) VMs, each one with a CPU core and a 4GB RAM block. We selected this type of server because it presents an enterprise server architecture used in current infrastructures that exhibits leakage-dependent properties with temperature.

The power consumption of a high-end server usually depends on several factors that affect both dynamic and static behavior [33]. Our proposed case study takes into account the following 7 variables:

- *Ucpu*: CPU utilization (%)
- *Tcpu*: CPU temperature (Kelvin)
- *Fcpu*: CPU frequency (GHz)
- *Vcpu*: CPU voltage (V)
- *Umem*: Main memory utilization (Memory accesses per cycle)
- *Tmem*: Main memory temperature (Kelvin)
- *Fan*: Fan speed (revolutions per minute (RPM))

Power consumption is measured with a current clamp with the aim of validating our approach. CPU and main memory utilization are monitored using the hardware counters collected with the *perf* monitoring tool. On board sensors are checked via IPMI to get both CPU and memory temperatures and fan speed. CPU frequency and voltage are monitored via the *cpufreq-utils* Linux package. To build a model that includes power dependence with these variables, we use this software tool to modify CPU DVFS modes during workload execution. Also room temperature has been modified during run-time with the goal of finding non-traditional consumption sources that are influenced by this variable.

5.2.2 Experimental workload

We define three workload profiles (i) synthetic, (ii) Cloud and (iii) HPC over Cloud as they emulate different utilization patterns that could be found in typical Cloud infrastructures.

Synthetic benchmarks

The use of synthetic load allows to specifically stress different server resources. The importance of using synthetic load is to include situations that do not meet the actual real workloads that we have selected. Thus, the range of possible values of the different variables is extended in order to adapt the model to fit future workload characteristics and profiles. *Lookbusy*[1] stresses different CPU hardware threads to a certain utilization avoiding memory or disk usage. The memory subsystem is also stressed separately using a modified version of *RandMem*[2]. We have developed a program based on this benchmark to access random memory regions individually, with the aim of exploring memory performance. *Lookbusy* and *RandMem* have been executed, in a separated and combined fashion, onto 4 parallel Virtual Machines that consume entirely the available computing resources of the server.

On the other hand, real workload of a Cloud data center is represented by the execution of *Web Search*, from *CloudSuite*[3], as well as by *SPEC CPU 2006 mcf* and *SPEC CPU 2006 perlbench* [131].

Cloud workload

Web Search characterizes web search engines, which are typical Cloud applications. This benchmark processes client requests by indexing data collected from online sources. Our *Web Search* benchmark is composed of three VMs performing as index serving nodes (ISNs) of Nutch 1.2. Data are collected in the distributed file system with a data segment of 6 MB, and an index of 2 MB that is crawled from the public Internet. One of this ISNs also executes a Tomcat 7.0.23 front end in charge of sending index search requests to all the ISNs. The front end also collects ISNs responses and sends them back to the requesting client. Client behavior is generated by Faban 0.7 performing in a fourth VM. Resource utilization depends proportionally on the number of clients accessing *Web Search*. Our number of clients configuration ranges from 100 to 300 to expose more information about the application performance. The four VMs use all the memory and CPU resources available in each server.

HPC over Cloud

In order to represent HPC over a Cloud computing infrastructure, we choose *SPEC CPU 2006 mcf* and *perlbench* as they are memory and CPU-intensive, and CPU-intensive applications, respectively. *SPEC CPU 2006 mcf* consists in a network simplex algorithm accelerated with a column generation that solves large-scale minimum-cost flow problems. On the other hand, a mail-based benchmark is performed by *SPEC CPU 2006 perlbench*. This program applies a spam checking software to randomly generated email messages. Both SPEC applications run in parallel in 4 VMs using entirely the available resources of the server.

5.2.3 Data set variability

Our data set presents high variability for the different parameters compiled from the server as can be seen in the following compilation of ranges obtained after workload execution.

- CPU operation frequency (*Fcpu*) is fixed to $f_1 = 1.73$ GHz, $f_2 = 1.86$ GHz, $f_3 = 2.13$ GHz, $f_4 = 2.26$ GHz, $f_5 = 2.39$ GHz and $f_6 = 2.40$ GHz; thus modifying CPU voltage (*Vcpu*) from 1.73 V to 2.4 V.

[1] http://www.devin.com/lookbusy/

[2] http://www.roylongbottom.org.uk

[3] http://parsa.epfl.ch/cloudsuite

- Room temperature has been modified in run-time, from 17°C to 27°C. Therefore, temperatures of CPU and memory (*Tcpu* and *Tmem*) range from 306 K to 337 K, and from 298 K to 318 K respectively.
- CPU and memory utilizations (*Ucpu* and *Umem*) take values from 0% to 100% and from 0 to 0.508 memory accesses (cache-misses) per CPU cycle respectively.
- Finally, due to both room temperature, and CPU and memory utilization variations, fan speed values (*Fan*) range from 3540 RPM to 7200 RPM.

5.3 Modeling Techniques

In this work we pursue four different modeling strategies for the identification of power models of enterprise servers in Cloud data centers. In Chapter 6 we propose (i) an analytical model that does not only consider the workload consolidation for deriving the power model, but also incorporates other non traditional factors like the static power consumption and its dependence with temperature. We also provide (ii) an automatic method, based on Multi-Objective Particle Swarm Optimization, to simplify the number of parameters used during power estimation in our proposed analytical model.

However, this modeling technique performs only as a parameter identification mechanism so, it may not provide the features that best represent the system's power consumption and other features could be incorporated to enhance the power estimation. Chapter 7 presents (iii) an automatic method based on Grammatical Evolution to obtain a power model that provides both Feature Engineering and Symbolic Regression. This technique helps to incorporate model features that only depend on the most suitable variables, with little designer's expertise requirements and effort.

Otherwise, classical regressions provide models with linearity, convexity and differentiability attributes, which are highly appreciated for describing systems performance. In Chapter 8, we proposes (iv) an automatic methodology for modeling complex systems based on the combination of Grammatical Evolution and a classical regression to obtain an optimal set of features that take part of a linear and convex model.

5.4 Comparison with state-of-the-art models

In order to evaluate the performance of the models that we obtain in this part of the research, we use as baseline different alternatives found in the state-of-the-art and that are available in current Cloud simulation environments [132].

- Linear model: Power is linear with CPU utilization (u_{cpu}).

$$P_{linear} = k_{linear1} \cdot u_{cpu} + k_{linear2} \tag{5.1}$$

- Quadratic model: Power is quadratic with CPU utilization.

$$P_{quadratic} = k_{quadratic1} \cdot u_{cpu}^2 + k_{quadratic2} \tag{5.2}$$

- Cubic model: Power is cubic with CPU utilization.

$$P_{cubic} = k_{cubic1} \cdot u_{cpu}^3 + k_{cubic2} \tag{5.3}$$

- Square root model: Power presents a square root dependence with CPU utilization.

$$P_{sqrt} = k_{sqrt1} \cdot \sqrt{u_{cpu}} + k_{sqrt2} \tag{5.4}$$

Also, as can be seen in Equation 6.8, dynamic power consumption of the CPU can be modeled considering the CPU supply voltage $V(k)$ and the working frequency of the machine $f(k)$ in a specific k DVFS mode. On the other hand, as seen in [133] fan power is a cubic function of fan speed represented as FS. We apply these statements to provide two more baseline power models that would help us to show how the thermal awareness, proposed in our research of the following chapters, would outperform non-thermal-aware approaches.

- DVFS model: Power presents a dependence with DVFS considering CPU voltage and frequency.
$$P_{DVFS} = k_{DVFS1} \cdot V^2(k) \cdot f(k) \cdot u_{cpu} + k_{DVFS2} \tag{5.5}$$
- DVFS and Fan model: Power presents a dependence with DVFS and with fan speed.
$$P_{DVFS\&fan} = k_{DVFS\&fan1} \cdot V^2(k) \cdot f(k) \cdot u_{cpu} + k_{DVFS\&fan2} \cdot FS^3 + k_{DVFS\&fan3} \tag{5.6}$$

k_{MX} are constants depending on the model M and the number of the constant X. These constants, which can be seen in Table 5.1 are obtained using a classic regression ($lsqcurvefit$ provided by Matlab) for training the models using the training data set collected from our Fujitsu RX300 S6 server explained in Subsection 5.2.2.

Table 5.1: Model constants for the baseline models

Model	k_{M1}	k_{M2}	k_{M3}
Linear	34.52	154.53	-
Quadratic	28.84	163.33	-
Cubic	26.77	166.85	-
Sqrt	40.44	144.35	-
DVFS	5.16	157.30	-
DVFS&fan	4.98	$8.086 \cdot 10^{-11}$	152.64

6. Analytical Power Modeling and PSO-based Model Optimization

Modeling the power consumption of data center infrastructures is crucial to anticipate the effects of aggressive optimization policies, but accurate and fast power modeling is a complex challenge for high-end servers not yet satisfied by current analytical approaches.

This chapter proposes an analytical model and an automatic method, based on Multi-Objective Particle Swarm Optimization (OMOPSO), for the identification of power models of enterprise servers in Cloud data centers. Our approach, as opposed to previous procedures, does not only consider the workload consolidation for deriving the power model, but also incorporates other non traditional factors like the static power consumption and its dependence with temperature. Our thermal and DVFS considerations are based on physical phenomena observed at the transistor level. Our experimental results show that we reach slightly better models than classical approaches, but simultaneously simplifying the power model structure and thus the numbers of sensors needed, which is very promising for a short-term energy prediction. This work, validated with real Cloud applications, broadens the possibilities to derive efficient energy saving techniques for Cloud facilities.

6.1 Introduction

Analytical models, as closed form solution representations, require the classification of the parameters that regulate the performance and power consumption of a computing system. Also, it is mandatory to find the complex relationships between these parameters to build the analytical functions [134].

However, incorporating a large amount of considerations to an analytical model may impact on its complexity not only in terms of non-linear relationships but also in the number of features. Also, analytical models enforce the usage of features that may have a low impact on the modeling target, thus degrading the performance of the curve fitting. For this reason we provide a model optimization using higher-level procedures that may help to simplify the complexity of analytical models. In this research we propose the use of metaheuristics because they are particularly useful in solving optimization problems that are noisy, irregular and change over time. These optimization algorithms make few assumptions about the optimization problem, providing adequately good solutions that could be based on fragmentary information [135], [136]. In this way, metaheuristics appear as a suitable approach to meet our optimization problem requirements in order to provide simplified accurate models that could be used in a Cloud during runtime. Finally our work makes the following contributions:

- We propose an accurate analytical power model for high-end servers in Cloud facilities. This model, as opposed to previous approaches, does not only consider the workload assigned to the processing element, but also incorporates the need of considering the static power consumption and, even more interestingly, its dependence with temperature.

- Moreover, this power model, applied to both the processing core and the memories of the system, includes voltage and frequency as parameters to be tuned during run-time by the DVFS policies.

- The model has been built and tested for an enterprise server architecture and with several real applications that can be commonly found in nowadays' Cloud server machines, achieving low error when compared with real measurements.
- We have optimized the power model for our target architecture using OMOPSO, a novel technique to perform the curve fitting. This algorithm allows the simplification of our analytical model attending to each server architecture.

The power model is presented in Section 6.2. Section 6.3 provides the background algorithm used for the model optimization. In Section 6.4 we describe the algorithm setup to adapt its parameters to our optimization problem. Section 6.5 describes profusely the experimental results. Finally, in Section 6.6 the main conclusions are drawn.

6.2 Power modeling

Leakage current increases strongly with temperature [24], also in deep sub-micron technologies [137], consequently increasing power consumption. Therefore, it is important to consider the strong impact of static power consumed by devices, as well as its dependence with temperature, and the additional effects influencing their performance. In this section, we derive a leakage model for the static consumption of servers attending to these concepts. The model is tested with real measurements taken in the enterprise server of our case study.

The current that is generated in a MOS device due to leakage is given by

$$I_{\text{leak}} = I_{\text{s}} \cdot e^{\frac{V_{\text{GS}}-V_{\text{TH}}}{nkT/q}} \cdot (1 - e^{\frac{-V_{\text{DS}}}{kT/q}}) \tag{6.1}$$

Research by Rabaey [24] shows that if $V_{\text{DS}} > 100mV$ the contribution of the second exponential in (6.1) is negligible, so the previous formula can be rewritten as

$$I_{\text{leak}} = I_{\text{s}} \cdot e^{\frac{V_{\text{GS}}-V_{\text{TH}}}{nkT/q}} \tag{6.2}$$

where leakage current depends on the slope factor n, the surface mobility of the carriers μ, the capacitance of the insulating material for the oxide gate C_{ox} and the ratio between the width and length of the transistors $\frac{W}{L}$ as can be seen in the following equation. Technology-dependent parameters can be grouped together to obtain an α constant.

$$I_{\text{s}} = 2 \cdot n \cdot \mu \cdot C_{\text{ox}} \cdot \frac{W}{L} \cdot \frac{kT}{q}^2 \tag{6.3}$$

$$I_{\text{leak}} = \alpha \cdot T^2 \cdot e^{\frac{V_{\text{GS}}-V_{\text{TH}}}{nkT/q}} \tag{6.4}$$

Using (6.4) in the leakage power equation $P_{\text{leak}} = I_{\text{leak}} \cdot V_{\text{DD}}$, the leakage power for a particular machine m can be derived:

$$P_{\text{leak}}(m) = \alpha(m) \cdot T^2(m) \cdot e^{\frac{V_{\text{GS}}-V_{\text{TH}}}{nkT/q}} \cdot V_{\text{DD}}(m) \tag{6.5}$$

Since our goal is to fit a model for the leakage power, we expand the polynomial function (6.5) into its Taylor third order series in order to easily regress the function, which leads to

$$P_{\text{leak}}(m) = \alpha_1(m) \cdot T^2(m) \cdot V_{\text{DD}}(m) + \alpha_2(m) \cdot T(m) \cdot V_{\text{DD}}^2(m) + \alpha_3(m) \cdot V_{\text{DD}}^3(m) \tag{6.6}$$

where $\alpha_1(m)$, $\alpha_2(m)$ and $\alpha_3(m)$ define the specific constants due to the manufacturing parameters of a server.

Incorporating frequency and voltage dependence in models is interesting due to the current trend of using DVFS modes to control the power consumption of servers.

Two of the main contributors to power consumption in servers are the CPU and the memory subsystem. We can easily find DVFS modes in CPUs, but there are currently very few memory devices with these capabilities. Power consumption of both disk and network have not been taken into account because of their lower impact in our scenario, high variability and heterogeneity of their technology in data centers.

Below is the formulation of the static consumption in a scenario with a CPU providing $k \in \{1 \ldots K\}$ different DVFS modes and a memory performing at a constant voltage. The model considers the different contributions due to temperature dependence. Also $\gamma(m)$ has been taken into account as it represents the fan power contribution constant. As seen in [133] fan power is a cubic function of fan speed represented as $FS(m)$. $\lambda(m)$ represents the total consumption of the rest of the server resources and devices that operate at a constant voltage and frequency.

$$\begin{aligned} P_{\mathrm{leak}}(m,k) &= \alpha_1(m) \cdot T_{\mathrm{cpu}}^2(m) \cdot V_{\mathrm{DD}}(m,k) + \alpha_2(m) \cdot T_{\mathrm{cpu}}(m) \cdot V_{\mathrm{DD}}^2(m,k) + \\ &+ \alpha_3(m) \cdot V_{\mathrm{DD}}^3(m,k) + \beta_1(m) \cdot T_{\mathrm{mem}}(m) + \beta_2(m) \cdot T_{\mathrm{mem}}^2(m) + \\ &+ \gamma(m) \cdot FS^3(m) + \lambda(m) \end{aligned} \tag{6.7}$$

As temperature-dependent leakage cannot be measured separately from the dynamic power in a server, we execute the *lookbusy*[1] synthetic workload to stress the system during monitored periods of time. *Lookbusy* can stress all the hardware threads to a fixed CPU utilization percentage without memory or disk usage. The use of a synthetic workload to derive the leakage model has many advantages, the most important of which is that dynamic power can be described as linearly dependent with CPU utilization and Instructions Per Cycle (IPC). Equation 6.8 provides the formula for dynamic power consumption.

$$P_{\mathrm{cpu}}^{\mathrm{dyn}}(m,k,w) = \alpha_4(m) \cdot V_{\mathrm{DD}}^2(m,k) \cdot f(m,k) \cdot u_{\mathrm{cpu}}(m,k,w) \tag{6.8}$$

In the previous formula $\alpha_4(m)$ is a constant that defines the technological parameters of the machine m, $V_{\mathrm{DD}}(m,k)$ is the CPU supply voltage and $f(m,k)$ is the working frequency of the machine in a specific k DVFS mode. $u_{\mathrm{cpu}}(m,k,w)$ is the averaged CPU percentage utilization of the specific physical machine m that operates in the k DVFS mode, running a workload w. $u_{\mathrm{cpu}}(m,k,w)$ is proportional to the number of cycles available in the CPU and accurately describes power consumption.

In order to stress the memory system we have developed a specific benchmark based on *RandMem*[2]. The program accesses random memory regions of an explicit size to explore the memory power consumption. Dynamic power consumption depends on the high level data cache misses characterized during profiling. As memory performs at a constant frequency and voltage, Equation 6.9 describes its dynamic power consumption.

$$P_{\mathrm{mem}}^{\mathrm{dyn}}(m,k) = \beta_3(m) \cdot u_{\mathrm{mem}}(m,k,w) \tag{6.9}$$

The constant $\beta_3(m)$ is defined by the technological features of the device, including both the constant frequency and voltage, and $u_{\mathrm{mem}}(m,k,w)$ represents the memory utilization expressed in memory accesses per cycle in a k DVFS mode ($k = 1$ represents a powered down server).

Finally, total power can be described as in Equation 6.10.

$$\begin{aligned} P_{\mathrm{tot}}(m,k,w) &= P_{\mathrm{cpu}}(m,k) + P_{\mathrm{mem}}(m,k) + P_{\mathrm{others}}(m,k) && (6.10) \\ P_{\mathrm{cpu}}(m,k,w) &= \alpha_1(m) \cdot T_{\mathrm{cpu}}^2(m) \cdot V_{\mathrm{DD}}(m,k) + \alpha_2(m) \cdot T_{\mathrm{cpu}}(m) \cdot V_{\mathrm{DD}}^2(m,k) + \\ &+ \alpha_3(m) \cdot V_{\mathrm{DD}}^3(m,k) + \\ &+ \alpha_4(m) \cdot V_{\mathrm{DD}}^2(m,k) \cdot f(m,k) \cdot u_{\mathrm{cpu}}(m,k,w) && (6.11) \\ P_{\mathrm{mem}}(m,k,w) &= \beta_1(m) \cdot T_{\mathrm{mem}}(m) + \beta_2(m) \cdot T_{\mathrm{mem}}^2(m) + \beta_3(m) \cdot u_{\mathrm{mem}}(m,k,w) && (6.12) \\ P_{\mathrm{others}}(m,k) &= \gamma(m) \cdot FS^3(m) + \lambda(m) && (6.13) \end{aligned}$$

[1] http://www.devin.com/lookbusy/

[2] http://www.roylongbottom.org.uk

6.3 Model identification

As stated above, our proposed power model consists of 9 parameters. Depending on the target architecture, some parameters might have more impact than others, as shown in our results. In our case, identification is performed as a multi-objective optimization and compared with a classical regression method. With a multi-objective optimization, we simultaneously optimize the average and the maximum errors to avoid peaks in the error function. To this end, we have selected a multi-objective Particle Swarm Optimization (PSO) algorithm to identify our power model. The reason for selecting multi-objective PSO is that this stochastic evolutionary computation technique, based on the movement and intelligence of swarms, has obtained excellent results specially in instances with real variables [138]. Next we provide a brief background about multi-objective optimization and the algorithm selected.

6.3.1 Multi-objective optimization

Multi-objective optimization tries to simultaneously optimize several contradictory objectives. For this kind of problems, single optimal solution does not exist, and some trade-offs need to be considered. Without any loss of generality, we can assume the following multi-objective minimization problem:

$$\begin{aligned} \text{Minimize} \quad & \vec{z} = (f_1(\vec{x}), f_2(\vec{x}), \ldots, f_m(\vec{x})) \\ \text{Subject to} \quad & \vec{x} \in X \end{aligned} \tag{6.14}$$

where $\vec{z}$ is the objective vector with m objectives to be minimized, $\vec{x}$ is the decision vector, and X is the feasible region in the decision space. A solution $\vec{x} \in X$ is said to dominate another solution $\vec{y} \in X$ (denoted as $x \prec y$) if and only if the following two conditions are satisfied:

$$\forall i \in \{1, 2, \ldots, m\}, f_i(\vec{x}) \leq f_i(\vec{y}) \tag{6.15}$$

$$\exists i \in \{1, 2, \ldots, m\}, f_i(\vec{x}) < f_i(\vec{y}) \tag{6.16}$$

A decision vector $\vec{x} \in X$ is non-dominated with respect to $S \subseteq X$ if another $\vec{x}' \in S$ such that $\vec{x}' \prec \vec{x}$ does not exist. A solution $\vec{x}^* \in X$ is called Pareto-optimal if it is non-dominated with respect to X. An objective vector is called Pareto-optimal if the corresponding decision vector is Pareto-optimal.

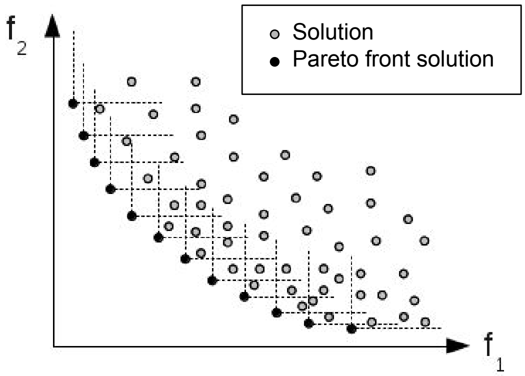


Figure 6.1: Non-dominated solutions of a set of solutions in a two objective space.

The non-dominated set of the entire feasible search space X is the Pareto-Optimal Set (POS). The image of the POS in the objective space is the Pareto-Optimal Front (POF) of the multi-objective problem at hand. Figure 6.1 shows a particular case of the POF in the presence of two objective functions. A multi-objective optimization problem is solved, when its complete POS is found.

6.3.2 PSO and OMOPSO

PSO is a metaheuristic search technique that simulates the movements of a flock of birds that aim to find food. The relative simplicity of PSO and the fact that is a population-based technique have made it a natural candidate to be extended for multi-objective optimization [139].

In PSO, particles are "flown" throughout a hyper-dimensional search space. Changes to the position of particles within the search space are based on social-psychological tendencies of individuals to emulate the success of other individuals. Hence, the position of each particle is changed according to its own experience and the experience of its neighbors. Let $x_i(t)$ denote the position of particle p_i, at time step t. The current position of p_i is then changed by adding a velocity vector $v_i(t)$ to the previous position, i.e.:

$$\vec{x}_i(t) = \vec{x}_i(t-1) + \vec{v}_i(t) \tag{6.17}$$

The velocity vector reflects the socially exchanged information and is defined in the following way:

$$\vec{v}_i(t) \quad = \quad W\vec{v}_i(t-1) + C_1\vec{r}_{i1}(\vec{x}_{ipbest} - \vec{x}_i(t-1)) + C_2\vec{r}_{i2}(\vec{x}_{ileader} - \vec{x}_i(t-1)) \tag{6.18}$$

where:

- W is the inertia weight and controls the impact of the previous history of velocities.
- C_1 and C_2 are the learning factors. C_1 is the cognitive learning factor and represents the attraction that a particle has towards its own success. C_2 is the social learning factor and represents the attraction that a particle has towards the success of its neighbors.
- $\vec{r}_{i1}$, $\vec{r}_{i2}$ are random vectors, each component in the range $[0, 1]$.
- $\vec{x}_{ipbest}$ is the personal best position of p_i , namely, the position of the particle that has provided the greatest success.
- $\vec{x}_{ileader}$ is the position of the particle that is used to guide p_i towards better regions of the search space.

Particles tend to be influenced by the success of any other element they are connected to. These neighbors are not necessarily particles close to each other in the decision variable space, but instead are particles that are close to each other based on a neighborhood topology, which defines the social structure of the swarm [139].

M. Reyes and C. Coello proposed a multi-objective PSO approach based on Pareto dominance, named OMOPSO [138]. This algorithm uses a crowding factor for the selection of leaders. This selection is made by binary tournament. This proposal uses two external archives: one for storing the leaders currently being used for performing the flight and another one for storing the final solutions. Only the leaders with the best crowding values are retained. Additionally, the authors propose a scheme in which the population is subdivided in three different subsets. A different mutation operator is applied to each subset. We use OMOPSO in the identification of our proposed power model, identifying the set of parameters that are representative for each target architecture.

6.4 Algorithm setup

PSO, as a metaheuristic, makes few assumptions about the optimization problem. As a consequence, the algorithm requires a preliminary configuration to provide adequate solutions. In this section we explain both the constraints and the parameter setup to adapt the metaheuristic to our optimization problem.

6.4.1 Multi-objective function

The problem to be solved is the estimation of the power consumption in virtualized enterprise servers performing Cloud applications. Our power model considers the heterogeneity of servers, as the specific technological features of each processor architecture result in a different power consumption. The resultant power model is non-linear (as shown in the previous section) and presents a large set of constraints. As stated above, the model identification is tackled as a multi-objective optimization simultaneously minimizing both the average and maximum errors :

$$\begin{array}{rrcl} \text{Minimize} & \vec{z} & = & (e_{\text{avg}}(\vec{x}), e_{\max}(\vec{x})) \\ \text{Subject to} & \vec{x}_{\min} & \leq & \vec{x} < \vec{x}_{\max} \\ \text{where} & \vec{x} & = & (\alpha_1, \ldots, \alpha_4, \beta_1, \ldots, \beta_3, \gamma, \lambda) \in X \end{array} \tag{6.19}$$

$\vec{x}$ is the vector of n decision variables and $\vec{z}$ is the vector of 2 objectives. $e_{\text{avg}}(\vec{x})$ is the average relative error percentage, $e_{\max}(\vec{x})$ is the maximum of the relative error percentage (Equation 6.21) and X is the feasible region in the decision space. Although we are interested in the minimization of the average relative error, we also use the maximum error percentage to avoid singular high peaks in the estimated model.

$$e_{\text{avg}}(\vec{x}) = \frac{1}{N} \cdot \sum_{n} \left| \frac{(P - P_{\text{tot}}) \cdot 100}{P} \right| \tag{6.20}$$

$$e_{\max}(\vec{x}) = \max \left| \frac{(P - P_{\text{tot}}) \cdot 100}{P} \right| \tag{6.21}$$

P is the power consumption measure given by the current clamp, P_{tot} is the power consumption estimated by our model (Equation 6.10) and n is each sample of the entire set of N samples used for the algorithm training. We use OMOPSO [140] to obtain a set of candidate solutions in order to solve our problem. Using this formulation, we are able to obtain a power consumption that is realistic with the current technology.

6.4.2 Algorithm parameters

Our power modeling problem requires a set of solutions with low error when compared with the real power consumption measures. In order to obtain suitable solutions we tune the OMOPSO algorithm using the following parameters:

- Swarm size: 100 particles.
- Number of generations: 2000. We avoid the PSO algorithm to be trapped in a local minimum by exhaustively analyzing this parameter. We have performed 20 optimizations for each number of generations ranging from 200 to 2400 as can be seen in Figure 6.2. For each number of generations, the input data is a numeric vector including all the values for the average error and maximum error respectively that can be found at the end of the PSO algorithm for all the particles. On each box, the central mark provides the median value, and the bottom and top edges of the box specify the 25^{th} and 75^{th} percentiles, respectively. The whiskers' edges represent the most extreme data points not considered outliers. Finally, the outliers are presented using the symbol "+". As can be seen in the figure, no improvements can be found when running more than 2000 generations.
- Perturbation: Uniform and non-uniform mutation. Both with a perturbation index of 0.5 and with mutation probability inversely proportional to the number of decision variables, 1/9 in our case.
- W, C1 and C2 are generated for each particle in every iteration as a random value in [0.1, 0.5], [1.5, 2] and [1.5, 2], respectively.

6.5 Experimental results

6.5.1 Training

We use our data set from the execution of the synthetic workload to perform a regression to our model by applying both nonlinear curve-fitting algorithms. First, we fit the power curve using OMOPSO optimizations and then we compare the results with MATLAB *lsqcurvefit* fitting function to analyze its benefits. The function *lsqcurvefit* is defined to solve nonlinear curve-fitting problems in least-squares sense.

The data collected during the execution of the training set are used to perform 30 iterations of the OMOPSO optimization. We obtain 30 sets of solutions, each of them defining a Pareto front for the two objectives defined in our problem, as seen in Figure 6.3. The hypervolume of these Pareto fronts shows an average value of -1.0109 and a standard deviation of 0.0229; hence, it can be concluded that the algorithm is not trapped into a local minimum. Once we combine these Pareto fronts into a final one, we achieve the final set of solutions for our power modeling problem, also shown in Figure 6.3.

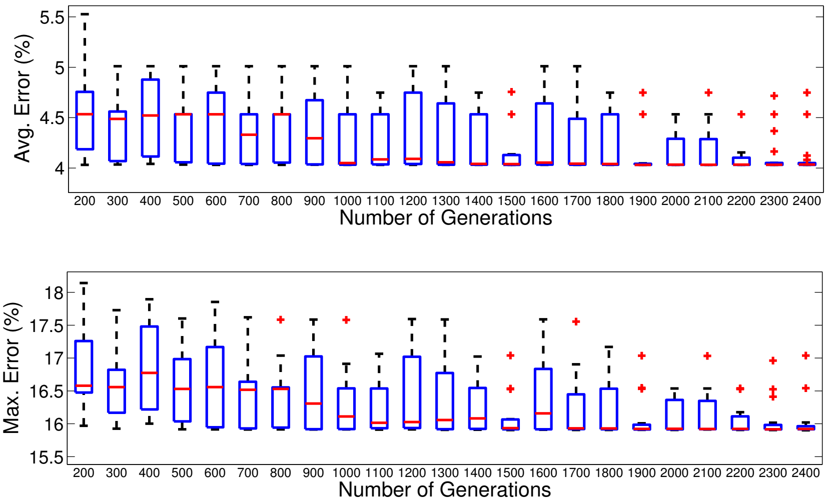


Figure 6.2: 20 optimizations for each number of generations.

6.5.2 Results

In order to present some results that support the benefits reached by OMOPSO applied to our optimization problem, we choose a solution from the final Pareto front. We also obtain the only solution of the *lsqcurvefit* optimization applied to the same training data set so that we can compare both approaches. Table 6.1 shows the values of both the average relative error and maximum relative error percentages obtained applying OMOPSO and *lsqcurvefit*, whereas Table 6.2 shows the corresponding solution for these two objectives, i.e., the best values reached for the 9 constants included in our power model.

Table 6.1: Objectives for Power curve fitting

Algorithm	Avg.Error	Max.Error
OMOPSO	4.0328%	17.0693%
lsqcurvefit	4.8501%	16.9401%

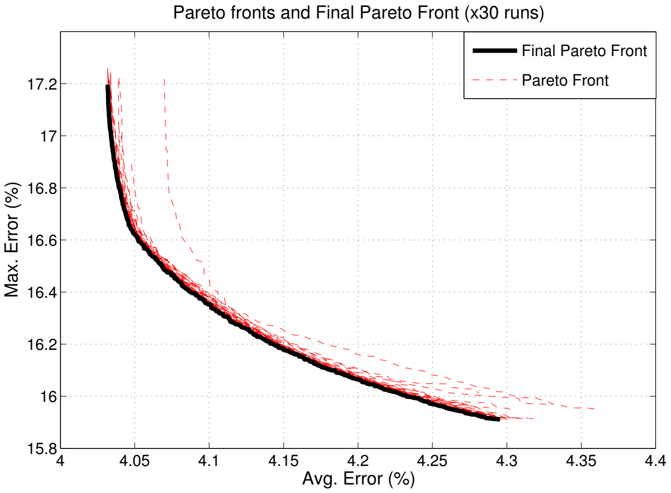


Figure 6.3: Pareto Fronts for 30 optimizations.

Table 6.2: Constants obtained for Power curve fitting

Algorithm	α_1	α_2	α_3	α_4	β_1	β_2	β_3	γ	λ
OMOPSO	0	0	0	3.32	0	$1.63 \cdot 10^{-3}$	0	$4.88 \cdot 10^{-11}$	0
lsqcurvefit	$2.71 \cdot 10^{-12}$	$3.70 \cdot 10^{-10}$	$1.48 \cdot 10^{-8}$	3.50	$2.55 \cdot 10^{-10}$	$1.60 \cdot 10^{-3}$	$7.63 \cdot 10^{-9}$	$5.12 \cdot 10^{-11}$	$3.76 \cdot 10^{-10}$

These results show that, while *lsqcurvefit* uses all the constants of the model, OMOPSO provides nonzero values to three constants simplifying the power model. The optimized model provided by our OMOPSO-based methodology is presented in Equation 6.22. This also means that for *lsqcurvefit* we need to collect information from seven sensors and, for OMOPSO, only from five sensors, resulting in computational savings in the monitoring system. We validate the solutions obtained for the power model with both algorithms, OMOPSO and *lsqcurvefit*, using real Cloud computing workload. The testing of the model is conducted for the data set gathered during the execution of the three different tests that represent real workload of a Cloud computing data center: *Web Search* application, *SPEC CPU 2006 mcf* and *SPEC CPU 2006 perlbench* [131].

$$\begin{aligned} P_{\text{Fujitsu}}(k,w) &= 3.32 \cdot V_{\text{DD}}^2(k) \cdot f_{op}(k) \cdot u_{\text{CPU}}(k,w) + 1.63 \cdot 10^{-3} \cdot T_{\text{mem}}^2 + \\ &+ 4.88 \cdot 10^{-11} \cdot FS^3 \end{aligned} \tag{6.22}$$

We calculate the values of $e_{\text{avg}}(\vec{x})$ and $e_{\text{max}}(\vec{x})$ for the test data sets using the solutions of both OMOPSO and *lsqcurvefit* algorithms. The average percentage error results, $e_{\text{avg}}(\vec{x})$, can be seen in Table 6.3. These results show that OMOPSO not only simplifies the optimization problem for our power model but also provides better error results than *lsqcurvefit* for three of the four tests conducted. *Web Search* presents higher peaks of memory accesses per cycle in comparison with the rest of the tests. The *lsqcurvefit* algorithm takes into account additional power contributions that are not present in the OMOPSO formulation, which are more sensitive to the high variability in the memory utilization. However, the difference in the power estimated by both algorithms in this test is only 0.3W. Finally, for our

Table 6.3: Average error percentage comparison for the tests performed

Workload	Training	Web Search	SPEC mcf	SPEC perlbench
OMOPSO	4.0328%	4.6028%	4.1242%	5.1148%
lsqcurvefit	4.8501%	4.4253%	6.1736%	5.2453%

OMOPSO-optimized model we obtain a mean error between the estimated power and the real measurement of -2.291 watts and a standard deviation of 5.199 watts. Figure 6.4 shows the power error distribution for this model, where it can be seen that a Gaussian-shaped distribution has been obtained. According to this, we can conclude that the error in terms of power of the 68% of the samples ranges from -7.4 to 2.9 watts. In Figure 6.5, the fitting of our optimized power model is provided.

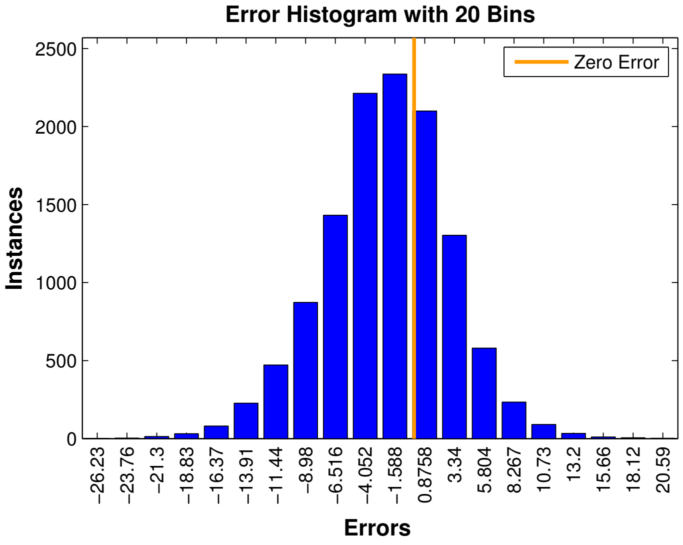


Figure 6.4: Power error distribution for our OMOPSO-optimized model.

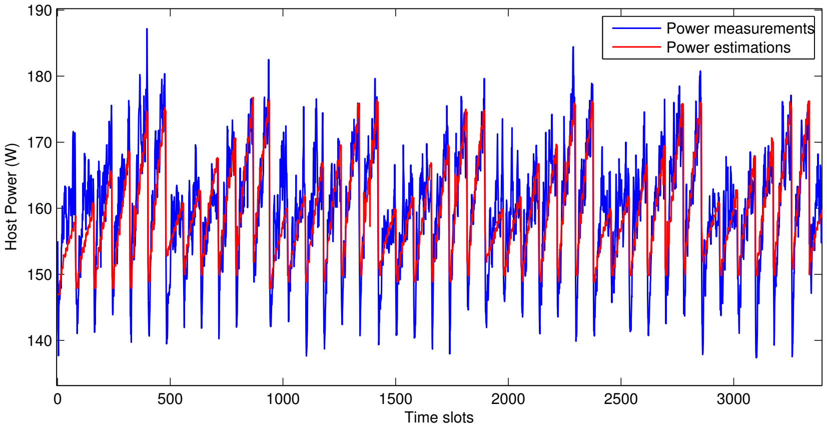


Figure 6.5: Model fitting for our OMOPSO-optimized model.

Our results consist of 30 different runs of the metaheuristic, with a randomly generated initial collection of particles as described in Section 6.4.2. All the solutions offered by our PSO-based algorithm, shown in Figure 6.3, present an average error between 4.03% and 4.36%, better than the 4.85% provided by the *lsqcurvefit*-based approach.

In order to compare the performance of our models, Table 6.4 presents the average error percentage for all the baseline models and for our proposed models $lsqcurvefit$ and OMOPSO. Linear, quadratic, cubic and sqrt models provide training errors that are higher, from 7.66% to 6.20%. It is important to note that the DVFS, DVFS&fan and our $lsqcurvefit$ and OMOPSO models aggregate incrementally DVFS-awareness, fan speed-awareness and thermal-awareness respectively. Each of them have been trained independently so their constants are different and can be seen in Tables 5.1 and 6.2. Thus, including DVFS-awareness to the power model, as in the DVFS model, improves the error when compared to linear, quadratic, cubic and Sqrt models from 6.20% to 5.77%. Moreover, incorporating also fan speed awareness, as done in the DVFS&fan model, also outperforms these models from 5.77% to 5.37%. By including the thermal-awareness as in our analytical model trained using a classic regression, $lsqcurvefit$, the average error is reduced from 5.37% to 5.24%. Finally, by optimizing the number of features of the analytical model using OMOPSO, the average error of the model is further reduced from 5.24% to 4.87%.

Table 6.4: Average error percentage comparison with the baseline models

Model	Linear	Quadratic	Cubic	Sqrt	DVFS	DVFS&fan	lsqcurvefit	OMOPSO
Training	5.83%	6.25%	6.46%	5.61%	5.58%	5.32%	4.85%	4.03%
Testing	6.58%	7.21%	7.66%	6.20%	5.77%	5.37%	5.24%	4.87%

Given the obtained results, we can conclude that 1) our contributions of including thermal-awareness to our models is relevant in order to estimate the power consumption of a server with higher accuracy; and 2) that the proposed methodology based on OMOPSO algorithms is an efficient technique for the envisioning of complex, multi-parametric power models for state-of-the-art Cloud computing servers. Moreover, the proposed technique allows to target several optimization problems that work on setting an energy-efficient working point by deciding the optimal clock frequency, voltage supply level and thermal-aware workload assignment.

6.6 Summary

This work has made successful advances in the provisioning of accurate power models of enterprise servers in Cloud services. The work presented in this research outperforms previous approaches in the area of power modeling for enterprise servers in Cloud facilities in several aspects. First, our work presents the relevance of including DVFS and thermal-awareness for power modeling in servers, enhancing their accuracy. Then, our approach consists on an automatic method for the identification of an accurate power model particularized for each target architecture. We propose an extensive power model consistent with current architectures. It is based on a generic analytical model where the main power consumption sources are considered. The model is multiparametric to allow the development of power optimization approaches. Our generic power model is optimized using metaheuristics, resulting in a specific model instance for every target architecture. Also the execution of the resulting power model is fast, making it suitable for run-time optimization techniques. Current models (linear, quadratic, cubic and square root among others), which do not consider both DVFS and thermal-awareness, present power accuracies that range from 7.66% to 5.37%. Our power model provides an error when compared with real measurements of 4.03% for training and 4.87% for testing in average, thus outperforming the state-of-the-art.

The use of multi-objective metaheuristic optimization algorithms allows us to include the traditional and non-traditional sources of power consumption, as well as the effect of several system-level parameters that affect the energy footprint. The experimental work, conducted with realistic workload, has shown the accuracy of the proposed methodology as compared with traditional regression algorithms. In addition, the multi-objective optimization approach followed in this chapter opens the door to proactive energy minimization techniques where the parameters are considered as decision variables.

The following chapter presents a novel methodology where the power models can be obtained automatically with no design effort. This modeling strategy helps to provide the optimal set of features and to infer the model, using metaheuristics based on evolutionary computation, without the necessity of deriving an analytical equation.

7. Automatic GE-based Power Modeling Methodology

Data centers, as complex systems, involve a vast number of different nature interacting variables that include non-linear relationships. So, extracting and understanding the connections between the most representative parameters and the power consumption require an enormous effort and knowledge about the problem. Analytical approaches have not yet fulfilled efficiently the complex challenge of easily generate accurate models for data center's servers.

In the previous chapter we have used a PSO to identify analytical models, providing accurate power estimations [33]. PSO helps to simplify the resultant power model by reducing the number of predefined parameters, variables and constants used in our analytical formulation. However, this technique is a parameter identification mechanism so, it does not provide the features that best represent the system's power consumption. Some other features could be incorporated to enhance the power estimation.

In this chapter, we propose an automatic method based on Grammatical Evolution to obtain a set of models that minimize power estimation error. This technique provides both Feature Engineering and Symbolic Regression to infer accurate models, which only depend on the most suitable variables, with little designer's expertise requirements and effort. This work has been validated using a real infrastructure running real Cloud applications resulting in a testing average power estimation error of 4.22%.

The work presented in this chapter outperforms previous approaches in the area of power modeling for enterprise servers in Cloud facilities in several aspects. Our approach is an automatic modeling method that offers an extensive power model consistent with current architectures. Also, our procedure takes into account the main sources of power consumption resulting in a multiparametric model, allowing the development of novel power optimization approaches. Different parameters are combined by FE, thus enhancing the generation of an optimized set of features. The resulting models, describe the power performance of high-end servers during runtime, for workloads that are subject to vary significantly. Our work improves the possibilities of deriving proactive energy-efficient policies in data centers that are simultaneously aware of complex considerations of different nature.

7.1 Introduction

One of the biggest barriers in data centers, as complex systems scenarios, is the huge number of variables that are potentially correlated. This problem complicates the inference of general power models from a macroscopic analytical perspective. The dependence of power on some traditionally ignored factors, which are increasingly influencing the consumption patterns of these infrastructures, must now be considered in order to achieve accurate power models. Factors like static power consumption, along with its dependence on temperature, or the power due to internal server cooling, are just some examples of parameters that typically have not been considered in the published models.

Also, Cloud data centers run workloads that show significant variations over time. So, power models need to be aware of the fluctuation of the different parameters during runtime.

Consequently, a fast and accurate method is required to model server performance, achieving a more accurate overall power prediction under varying workloads and working situations.

Analytical models require specific knowledge about the different power contributions and their relationships, becoming hard and time-consuming techniques for describing these complex systems. Moreover, models are architecture-dependent, so the modeling process has to be replicated for each different server structure. Conversely, metaheuristics, as high level procedures, help to find good enough solutions for modeling heterogeneous, scalable and distributed systems based on fragmentary information and making few assumptions about the problem [135].

Our research in the previous chapter [33] shows that accurate results for power estimation can be obtained by applying a PSO metaheuristic. PSO helps to constrain the predefined set of parameters of the analytical formulation, thus simplifying the resulting power model. However, this technique does not provide the optimal collection of features that best describe the system power performance, because it only works as an identification mechanism.

Feature Engineering (FE) methods are used to select adequate features, avoiding the inclusion of irrelevant parameters that reduce generality [141]. FE properties help, not only to find relevant variables, but their combinations and correlations, offering a straightforward problem formulation thus generating better solutions. Grammatical Evolution (GE) is an evolutionary computation technique used to perform Symbolic Regression (SR) [142].

GE is particularly useful to solve optimization problems and build accurate models of complex systems. This technique provides solutions that include non-linear terms while still offering FE capabilities, thus bypassing the barrier of analytical modeling. Also, as GE is an automatic technique, little designer's expertise is required to process high volumes of data. In this work we propose a GE-based approach to find optimized power models that accurately describe and estimate the consumption of high-end Cloud servers, providing a general methodology that can be applied to a broad set of server architectures and working conditions.

Our work makes the following **key contributions**:

- We propose a method for the automatic generation of fast and accurate power models to describe the performance of high-end Cloud servers.

- Resulting models are able to include combinations and correlations of variables due to FE and SR performed by GE. Therefore, the power models incorporate the selection of representative features that best describe power performance.

- Different models have been built and tested for a high-end server architecture using real applications that can be commonly found in nowadays' Cloud data centers, achieving low error in terms of power estimation when compared with real measurements.

- The validation of the resulting models for different applications (including web search engines, and intensive applications) shows an average testing error of 4.22% in power estimation.

The remainder of this chapter is organized as follows: Section 7.2 provides some background on GE. The power modeling evaluation is presented in Section 7.3. Section 7.4 describes profusely the experimental results. Finally, in Section 7.5 the main conclusions are drawn.

7.2 Feature selection and Modeling Process

Data centers, as complex systems can be defined as an interconnected agents system that exhibits a global behavior resulting from the interaction of these agents [129]. So, inferring the global performance, which is not induced by a central controller, requires a deep knowledge of the operation and the physical phenomena, being a complex and time-consuming challenge. Therefore, fast and automatic modeling techniques are required that are more suitable for systems that have a huge amount of parameters.

Our research focuses on obtaining a mathematical expression that represents server power consumption. In this work, the power formulation is derived from experimental data collected in a real infrastructure. Our data set compiles values of the different variables that have been considered such as processor and memory temperatures, fan speeds, processor and memory utilization percentages, processor frequencies and voltages. In this context, we consider the selection of the relevant features that will take part of our model as a SR problem. SR helps to simultaneously obtain a mathematical expression and include the relevant features to reproduce a set of discrete data.

Genetic Programming (GP) has proven to be effective in solving a number of SR problems [143], but it presents some limitations like the excessive growth of memory computer structures, often produced by representing the phenotype of the individuals. In the last years, GE has appeared as a simpler optimization variant of GP [144]. GE allows the generation of mathematical models applying SR, where the model generation is achieved thanks to the use of grammars that define the rules for obtaining mathematical expressions.

In this work we use GE together with grammars expressed in Backus Naur Form (BNF) [144] as this representation has been satisfactorily used by the authors to solve similar problems when combined with regressive techniques [29]. A BNF specification is a set of derivation rules, expressed in the form:

$$\texttt{<symbol>::=<expression>} \tag{7.1}$$

BNF rules are represented as sequences of non-terminal and terminal symbols. The former symbols use to appear on the left side of the equation, but they may appear also on the right, while the later are shown on the right side. In Equation 7.1, we can affirm that `<symbol>` and `<expression>` are non-terminals, although these do not represent a complete BNF specification, since those are always enclosed between the pair `< >`. This equation represents that the non-terminal `<symbol>` will be replaced (indicated `::=`) by an expression. The rest of the grammar may define the set of different alternatives for the expression.

A grammar is defined by the 4-tuple N, T, P, S, being N the set of non-terminals, T the set of terminals, P the production rules for the replacement of elements between N and T, and S is the start symbol that should appear in N. The symbol "|" separates the different options within a production rule.

```
N = {EXPR, OP, PREOP, VAR, NUM, DIG}
T = {+, -, *, /, sin, cos, log, x, y, z, 0, 1, 2, 3, 4, 5, (, ), .}
S = {EXPR}
P = {I, II ,III ,IV ,V ,VI}
I   <EXPR> ::= <EXPR><OP><EXPR> | <PREOP>(<EXPR>) | <VAR>
II  <OP>   ::= + | - | * | /
III <PREOP>::= sin| cos | log
IV  <VAR>  ::= x | y | z | <NUM>
V   <NUM>  ::= <DIG>.<DIG> | <DIG>
VI  <DIG>  ::= 0 | 1 | 2 | 3 | 4 | 5
```

Figure 7.1: Example BNF grammar designed for symbolic regression

Figure 7.1 shows an example of a BNF grammar, designed for symbolic regression.

The final obtained expression will consist of elements of the set of terminals T. These have been combined with the rules of the grammar, as explained previously. Also, grammars can be adapted to bias the search of the relevant features because there is a finite number of options in each production rule.

The final expression resulting from the GE execution will only consist of terminals of the T set. Non-terminals will be translated to terminal options by using the production rules in set

P. Grammars can be adapted to bias the search of the relevant features because of the finite number of options in each production rule.

In this work, our variables set `<VAR>` consists of different parameters, as usage, frequency, voltage and temperature of computing resources, which contribute to the server power consumption. Our set of terminals consist of these variables, the operators $+, -, *$ and $/$, the preoperators exp, sin, cos and ln, and base-10 exponent format constants. Finally, they will be combined in a final expression describing power consumption.

GE works like a classic Genetic Algorithm (GA) [145] in terms of structure and internal operators. Each individual is defined by a chromosome and a fitness value. Each chromosome consists of a specific number of genes, also called codons. Then, the population formed by a set of individuals is evolved by the algorithm. While in SR, the fitness function is commonly a regression metric, as a Mean Squared Error (MSE), a Root Mean Square Deviation (RMSD) or a Coefficient of Variation (CV), in GE, the chromosome consists of a string of integers. The GA operators are applied iteratively in order to improve the fitness function during the algorithm execution. These operators are the selection of the population, the crossover process, which combines the chromosomes, and the mutation of the resulting individuals, which occurs with a certain probability defined as mutation probability. Then, in the decoding stage, the GE algorithm computes the fitness function for each iteration, extracting the expression defined in each individual. To obtain further information about GA operators, we refer the reader to the work presented by D. E. Goldberg [146]. Through the following example, we explain the GE decoding process to clearly explain how this algorithm selects the features. In this example, we decode the following 7-gene chromosome using the BNF grammar shown in Figure 7.1.

    21-64-17-62-38-254-2

First, we begin to decode the expression using the Start symbol `S={EXPR}` defined by the grammar in Figure 7.1.

    Solution = <EXPR>

Then, we decode the first gene of the chromosome, 21, in rule `I` of the grammar. This rule has 3 different choices: (i) `<EXPR><OP><EXPR>`, (ii) `<PREOP><EXPR>` and (iii) `<VAR>`. Hence, the modulus operator is applied as a mapping function:

    21 MOD 3 = 0

As result of the mapping function, the first option `<EXPR><OP><EXPR>` is selected, so this expression is used to substitute the non-terminal. The current expression after this decoding step is the following:

    Solution = <EXPR><OP><EXPR>

The process continues with the substitution of the first non-terminal of the current expression `<EXPR>`, using the next codon, 64. The modulus is applied again to rule `I`.

    64 MOD 3 = 1

In this case, the algorithm selects the second option offered by the grammar for this rule, `<PREOP>(<EXPR>)`. The current expression is the following:

    Solution = <PREOPR>(<EXPR>)<OP><EXPR>

The GE takes the next gene, 17, for decoding. At this point, `<PREOP>` is the first non-terminal in the current expression. Therefore, we apply the modulus operator to rule `III` to choose one of the three different choices.

    17 MOD 3 = 2

So, the third option `log` is selected. The output of the decoding process at this point results in the expression:

    Solution = log(<EXPR>)<OP><EXPR>

The following codon, 62, decodes `<EXPR>` using rule `I`.

    62 MOD 3 = 2

Value 2 means to select `<VAR>`, the third option, resulting in the expression:

    Solution = log(<VAR>)<OP><EXPR>

Next codon, 38, uses rule `IV` to decode `<VAR>`.

    38 MOD 4 = 2

Non-terminal `z` is selected as the mapping function output means the third option.

    Solution = log(z)<OP><EXPR>

Codon 254 decodes the Non-terminal `<OP>` with rule `II`:

```
   254 MOD 4 = 2
```

This value means the third option, terminal $\star$:

```
   Solution = log(z)*<EXPR>
```

The last codon of the chromosome in this example, decodes `<EXPR>` with rule `I`:

```
   2 MOD 3 = 2
```

The third option is selected so, the expression is substituted by the non-terminal `<VAR>`. In this step, the current expression is the following:

```
   Solution = log(z)*<VAR>
```

At this point, GE algorithm has run out of codons. However, the decoding process has not obtained an expression with Terminals in each of its components. GE, solves this problem by reusing codons, starting from the first one in the chromosome, and during the decoding process, it is possible to reuse the codons more than once. This technique is known as wrapping, and it is inspired in the gene-overlapping phenomenon present in many organisms [147]. By applying wrapping to our example, the GE reuses the first gene, 21, which decodes `<VAR>` with rule `IV`.

```
   21 MOD 4 = 1
```

The result of the mapping function selects the second option, non-terminal `y`, giving the final expression of the phenotype:

```
   Solution = log(z)*y
```

As can be seen in this example, the process performs parameter identification like in classic regression methods. Moreover, when used together with an appropriate fitness function, GE is also able to infer the optimal set of features that best describes the target system. So, the evolutionary algorithm computes the mathematical expression, performing both model identification and feature selection, being able to result into the most accurate power model.

7.3 Modeling Evaluation

This work focuses on the accurate estimation of the power consumption in virtualized enterprise servers running Cloud applications. Our power model considers heterogeneity, specific technological features and non-traditional parameters of the target architecture that impact on power. Hence, we propose a modeling process that considers all these factors, applying GE to infer an expression that characterizes the power consumption. As a result, we derive a highly accurate power model, targeting a specific server architecture, that is automatically generated by our evolutionary methodology with little effort for the designer.

7.3.1 GE Configuration

Modeling systems can target two different purposes. On the one hand, there exist modeling procedures that intend to interpret systems' behavior. They have the purpose of acquiring additional knowledge from the final models once these have been derived. However, this kind of models is not optimized in terms of accuracy, but incur a loss of precision in favor of being more explanatory. On the other hand, when building an accurate predictor, the presence of irrelevant features and restrictions on operations hinders generalization and conciseness of models. The proposed grammar in Figure 7.2 allows the occurrence of a wide variety of operations and operands to favor building optimal models. The variables $x[0] - x[6]$ are the parameters measured during data compilation as explained in Section 5.2.

Fitness Evaluation

During the evolutionary process, GE will evolve towards optimizing the fitness function. Our fitness function includes the power estimation error in order to build accurate models by constraining the difference between the real measurement $P(n)$ and the estimated value $\widehat{P}(n)$. The fitness function presented in Equation 7.2 leads the evolutionary process to obtain optimized solutions thus minimizing RMSD.

```
N =   {EXPR, OP, PREOP, CTE, BASE, EXPON, SIGN, VAR}
T =   {+, -, *, /, exp, sin, cos, ln, 0,..., 99, x[0],...,x[6],(,)}
S =   {EXPR}
P =   {I, II ,III ,IV, V, VI, VII, VIII}
I     <EXPR>  ::= <EXPR><OP><EXPR> | <PREOP>(<EXPR>) | <VAR> | <CTE>
II    <OP>    ::= + | - | * | /
III   <PREOP> ::= exp | sin | cos | ln
IV    <CTE>   ::= <BASE>E<sign><EXPON>
V     <BASE>  ::= 1 | 2 | 3 |...| 98 | 99
VI    <EXPON> ::= 1 | 2 | 3 |...| 8 | 9
VII   <SIGN>  ::= + | -
VIII  <VAR>   ::= x[0] | x[1] |...| x[6]
```

Figure 7.2: Proposed grammar in BNF. x represent our variables, with $i = 0 \dots 6$.

$$\begin{aligned} F &= \sqrt{\frac{1}{N} \cdot \sum_{n} {e_{\text{n}}}^2} && (7.2) \\ e_{\text{n}} &= |P(n) - \widehat{P}(n)|, \qquad 1 \leq n \leq N && (7.3) \end{aligned}$$

The estimation error e_{n} represents the deviation between the power measure given by the current clamp P, and the power consumption that is estimated by the model phenotype $\widehat{P}$. n represents each sample of the entire data set of N samples obtained during the execution of the proposed workloads.

7.4 Experimental results

Our data set (explained in Section 5.2) has been split into a training and a testing set. We have used the three kind of workloads for both training and testing the evolutionary algorithms instead of restricting the use of synthetic workloads only for training and Cloud benchmarks exclusively for testing. This procedure provides automation for the progressive incorporation of additional benchmarks to the model. The training stage performs feature selection and builds the power model according to our grammar and fitness function. Then, the testing stage tests the power model accuracy.

Our GE-based algorithm is executed 30 times using the same grammar and fitness function configuration. For each run, we randomly select 50% of each data set for training and 50% for testing, thus obtaining 30 power models. This random-split technique validates the variability and versatility of the system, by analyzing the occurrence of local minima in optimization scenarios.

7.4.1 Models training

We use GE to obtain a set of candidate solutions with low error, when compared with the real power consumption measurements in order to solve our modeling problem. To obtain adequate solutions we adjust the algorithm parameters to the values shown in Table 7.1, where the mutation probability is inversely proportional to the number of rules.

It is worthwhile to mention that we have performed a variable standardization for every parameter (in the range $[1, 2]$) in order to ensure the same probability of appearance for all the variables, thus enhancing SR. Table 7.2 shows the phenotype of the best model and the fitness function (RMSD errors) at the training stage obtained for each of the 30 iterations. As can be seen in this table, the minimum training RMSD is 0.11 and those phenotypes presenting this error value depend on the three same parameters: *Tcpu*, *Tmem* and *Ucpu*.

Table 7.1: Algorithm configuration parameters

Parameter	Configuration value
Population size	250 individuals
Number of generations	30000
Chromosome length	100 codons
Mutation probability	1/8
Crossover probability	0.9
Maximum wraps	3
Codon size	256
Tournament size	2 (binary)

Table 7.2: Phenotype, RMSD and Average testing error percentages for 30 iterations

Run	Expression	Train	Testing	Synthetic	mcf	perlb	WebSearch	Total
	Power estimation $\widehat{P}$	(RMSD)	(RMSD)	(%)	(%)	(%)	(%)	(%)
1	((Tcpu-58E-3)+(ln(Ucpu)/exp(Fcpu)))	0.13	0.12	4.81	4.60	5.00	5.47	4.88
2	ln(ln(exp((Tcpu+((Tmem+Ucpu)-sin(35E-3))))))	0.11	0.11	4.16	4.95	4.44	4.62	4.24
3	ln(((Ucpu+Tcpu)+Tmem))	0.11	0.12	4.10	5.15	4.54	4.81	4.22
4	exp((Tcpu/ln(5E+1)))	0.13	0.12	5.22	4.42	4.17	5.52	5.21
5	(59E-2*(Tcpu+(exp(37E-6)+3E-9)))	0.13	0.13	4.88	5.52	4.13	5.17	4.91
6	(67E-1/(ln(10E+1)-cos((2E+5-(Tcpu-60E+1)))))	0.13	0.13	5.39	4.23	4.30	5.75	5.38
7	exp(((exp(Tcpu)+Ucpu)*sin((19E+4*sin(33E-8)))))	0.12	0.12	5.16	4.03	4.23	4.84	5.07
8	ln((Ucpu+(Tcpu+Tcpu)))	0.12	0.13	5.19	4.87	6.19	5.16	5.20
9	exp((exp(Tcpu)*89E-3))	0.13	0.12	5.39	4.49	4.04	5.81	5.38
10	ln((Tmem+(Ucpu+(Tcpu-46E-9))))	0.12	0.11	4.10	5.15	4.54	4.81	4.22
11	exp((exp(Tcpu)*88E-3))	0.12	0.13	5.26	4.62	4.02	5.63	5.25
12	(cos(sin(Umem))-(Tcpu*sin(12E+4)))	0.12	0.12	5.27	4.62	4.02	4.66	5.15
13	exp((Tcpu/40E+1))	0.13	0.12	5.15	4.67	4.26	5.24	5.12
14	ln(((Tcpu+Ucpu)+Tmem))	0.11	0.12	4.10	5.15	4.54	4.81	4.22
15	ln(((Tcpu+Ucpu)+Tmem))	0.11	0.12	4.10	5.15	4.54	4.81	4.22
16	exp((84E-3*exp(Tcpu)))	0.12	0.13	4.91	5.25	4.18	5.00	4.91
17	ln((Tmem+(Tcpu+(Ucpu-2E-8))))	0.11	0.11	4.10	5.15	4.54	4.81	4.22
18	(50E-6+ln((Tmem+31E-2+Fcpu+(Ucpu/Fcpu))*exp(64E-9)))	0.12	0.12	4.56	5.12	4.80	5.25	4.66
19	ln(((cos(cos(33E+4))+Tcpu)/52E+2))	0.12	0.13	5.12	4.62	4.04	5.53	5.13
20	ln((exp(Tcpu)+ln((Fan/(Tcpu/Ucpu)))))	0.12	0.12	4.97	4.23	4.75	4.97	4.94
21	(sin((48E-2*Tcpu))-ln(cos(91E+6)))	0.13	0.13	5.61	4.00	4.27	6.32	5.61
22	exp((exp(Tcpu)*84E-3))	0.13	0.12	4.91	5.25	4.18	5.00	4.91
23	ln(((Tcpu+Ucpu)+Tmem))	0.11	0.11	4.10	5.15	4.54	4.81	4.22
24	exp((Tcpu*25E-2))	0.12	0.13	5.15	4.67	4.26	5.24	5.12
25	ln((Tcpu+(Tmem+Ucpu)))	0.11	0.11	4.10	5.15	4.54	4.81	4.22
26	exp((25E-2*Tcpu))	0.13	0.12	5.15	4.67	4.26	5.24	5.12
27	exp((Tcpu/38E+1))	0.13	0.12	5.41	4.16	4.17	5.97	5.41
28	ln(((Ucpu+Tcpu)+Tmem))	0.11	0.12	4.10	5.15	4.54	4.81	4.22
29	exp(sin((90E-3*exp(Tcpu))))	0.13	0.12	5.21	4.59	4.02	5.60	5.21
30	exp((Tcpu/39E+1))	0.12	0.13	5.23	4.39	4.17	5.57	5.22

7.4.2 Models testing and analysis of results

After training, we evaluate the quality of the power models obtained for each of the 30 iterations. Testing results are analyzed particularly for each benchmark in order to verify the reliability of the power estimation for different workloads. Table 7.2 shows average error percentages for each data set. These values have been obtained using the following equation:

$$e_{\text{AVG}} = \sqrt{\frac{1}{N} \cdot \sum_{n} \Big(\frac{|P(n) - \widehat{P}(n)| \cdot 100}{P(n)} \Big)^2}, 1 \leq n \leq N \tag{7.4}$$

As can be seen in Table 7.2, those solutions that present a lower training error also show the best values obtained for testing. Our results demonstrate a minimum value of 4.22% for the total average error (*Total*) when evaluating the entire testing data set. The average value can be broken down for the different tests, achieving testing errors of 4.1%, 5.15%, 4.54% and 4.81% for *Synthetic*, *mcf*, *perlbench* and *WebSearch* workloads respectively.

Also, those phenotypes presenting the lowest error are logarithmically dependent on the summation of *Tcpu*, *Tmem* and *Ucpu*. Solution in Equation 7.5 appears in 8 of the 30 iterations (assuming negligible additional constants) and is the best power model obtained during the training stage. *Ucpu* and *Tcpu* provide physical information about the dynamic consumption of the CPU and its variability with the workload. *Tcpu* is also correlated to DVFS modes, fan speed and the static contribution of the CPU that depends on the inlet temperature of the server. On the other hand, *Tmem* provides the information regarding the dynamic consumption of the memory and is correlated to fan speed and to the static consumption, which present a dependence with the inlet temperature of the server. Furthermore, this power model simultaneously simplifies the power model (from 7 to 3 parameters), reducing the number of required sensors, which is very promising for run-time prediction. These experimental results show that simplified models that use relevant features provide an accurate prediction that can be calculated during run-time for real workloads.

$$\widehat{P} = ln(Ucpu + Tcpu + Tmem) \tag{7.5}$$

Figure 7.3 shows the power fitting of the model shown in Equation 7.5 normalized in the range [1,2]. In this figure, each sample represents the averaged values of an execution of 5 minutes of workload. It can be seen that this model offers a good fitting to the normalized monitoring curve on scenarios with workloads that vary significantly during runtime. Finally,

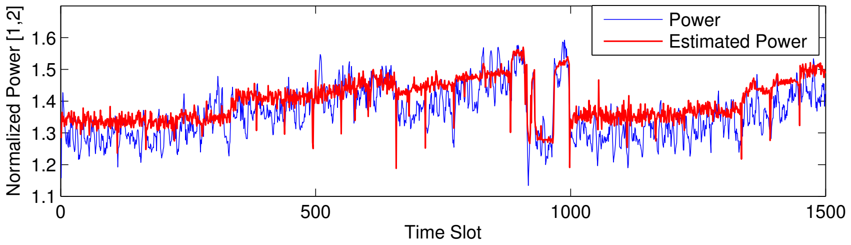


Figure 7.3: Model fitting for our GE-based power model

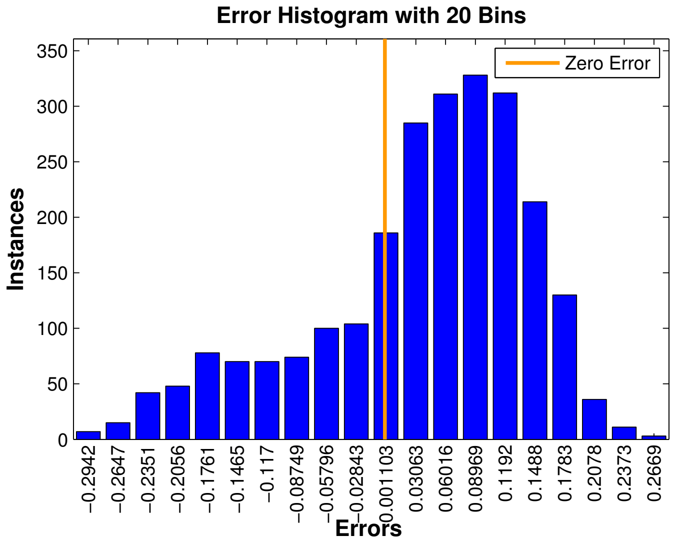


Figure 7.4: Power error distribution for our GE-based power model

for our GE-optimized model we obtain a mean error between the estimated power and the real measurement of -0.0357 and a standard deviation of 0.1082. Figure 7.4 shows the power error distribution for this model, where it can be seen a Gaussian-shaped distribution. According to this, we can conclude that the error in terms of power of the 68% of the samples ranges from -0.0725 to 0.1439.

Following the criteria of reducing the number of parameters, and with the aim of simplifying the computational complexity of the models, we also analyze the effect of the number of parameters in the error of the power estimation. Figure 7.5 shows the trade-off between the testing error percentage and the number of parameters that appear in the final power expression per run. When considering those models whose expressions have 3 variables, we obtain a testing error ranging from 4.22% and 4.94%. A global minimum can be found at 4.22% for phenotypes matching Equation 7.5. We can see that the rest of the parameters that do not appear in Equation 7.5 are less relevant for describing power performance or they are correlated with these three variables. So, their inclusion would decrease the model accuracy as it occurs in the case of including *Fan* and *Fcpu* in runs 20, 1 and 18.

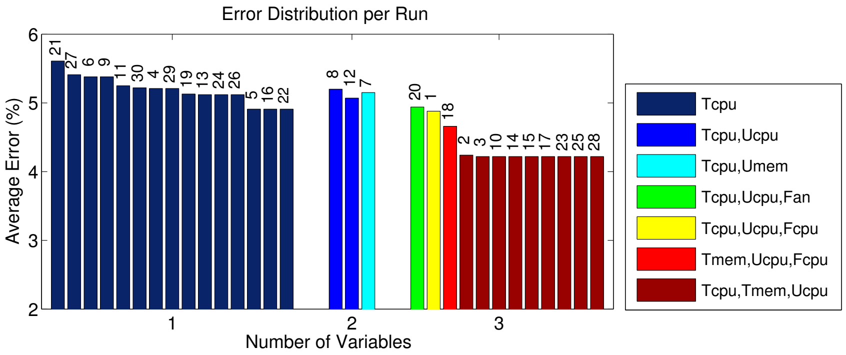


Figure 7.5: Average error per run and per variables set

For the obtained models presenting expressions with 2 variables, results show that as *Tmem* parameter is missing, testing results worsen ranging from 5.07% to 5.20%. Testing results for 1 variable-expressions range from 4.91% to 5.61% showing that the most relevant feature when estimating server power is *Tcpu*. A local minimum can be found at 4.91% as, in some cases better solutions can be easily found for expressions depending only on *Tcpu* than on 2 variables.

In order to compare the performance of our models, Table 7.3 presents the average error percentage for all the baseline models, for our models proposed in Chapter 6, $lsqcurvefit$ and OMOPSO and for the model obtained in this work, GE. Linear, quadratic, cubic and sqrt baselines provide errors that are higher, from 7.66% to 6.20%. Including DVFS-awareness and also fan speed-awareness to the power model, as in the DVFS and DVFS&fan models, improve the error when compared to linear, quadratic, cubic and sqrt models from 6.20% to 5.77% and to 5.37% respectively. By including the thermal-awareness as in our analytical model trained using a classic regression, $lsqcurvefit$, and using OMOPSO, the average error is reduced from 5.37% to 5.24% and to 4.87% respectively. Finally, by optimizing the feature selection, using an automatic modeling based on GE, for the testing data set, the average error is further reduced from 4.87% for the OMOPSO model to 4.22% for our GE model.

Table 7.3: Average error percentage comparison with the baseline models

Model	Linear	Quad.	Cubic	Sqrt	DVFS	DVFS&fan	lsqcurv.	OMOPSO	GE
Training	5.83%	6.25%	6.46%	5.61%	5.58%	5.32%	4.85%	4.03%	4.10%
Testing	6.58%	7.21%	7.66%	6.20%	5.77%	5.37%	5.24%	4.87%	4.22%

These results show that, by using the proposed methodology, 1) the model can be obtained automatically with no design effort applying FE to provide the optimal set of features; 2) adequate solutions can be found if the number of variables is a major constraint when finding a suitable power model in order to meet system requirements; and 3) optimizing the set of features using GE leads to more accurate models. As training and testing data sets are randomly selected in each of the 30 runs, this analysis confirms that FE works well for our scenario.

7.5 Summary

The work presented in this chapter makes relevant contributions on the accurate power modeling of high-end Cloud servers. Our GE-based automatic approach does not require designer's expertise to describe the complex relationships between parameters and power consumption sources. FE and SR help to infer accurate models by incorporating only those features that best describe the power consumption.

The proposed modeling method has shown relevant benefits with respect to state-of-the-art modeling techniques. Our approach automatically derives power models, with almost no designer's effort, helping to consider a broader number of input parameters. The experimental evaluation, using real infrastructure, shows high accuracy for the estimation of power. Moreover, the models provided for different executions of our modeling algorithm, where data sets are split randomly, converge to the same solution that also present a lowest average testing error of 4.22% when compared with real measurements. So, our automatically generated model outperforms current models (linear, quadratic, cubic and square root among others), which do not consider both DVFS and thermal-awareness, whose power accuracies range from 7.66% to 5.37%.

According to these results, we can infer that our methodology based on GE algorithms is effective for performing feature selection and building accurate multi-parametric power models for high-end Cloud servers. Finally, our modeling methodology is a starting point for proactive energy-efficient policies as resulting models can be exploited by further power optimization techniques that consider the joint effect of different features. Our approach helps to implement proactive energy optimization techniques based on these power models, thus considering the combined effect of consolidation, DVFS and temperature. This research help to optimize not only the computing resources of the data center, but also the cooling contribution.

Classical regressions may provide models with linearity, convexity and differentiability attributes, which are highly appreciated for describing systems power consumption. However, these properties are difficult to be provided by GE approaches that use BNF grammars. In the following chapter we propose a novel methodology for the automatic inference of accurate models that combines the benefits offered by both classic and evolutionary strategies.

8. Automatic Power Modeling Methodology based on GE and LASSO

This chapter proposes an automatic methodology for modeling complex systems. Our methodology is based on the combination of GE and classical regression to obtain an optimal set of features that take part of a linear and convex model. The management of energy-efficient techniques and aggressive optimization policies in a Cloud, requires a reliable prediction of the effects caused by the different procedures throughout the data center. The heterogeneity of servers and the diversity of data center configurations make it difficult to infer general models. Also, power dependence with non-traditional factors that affect consumption patterns of these facilities may be devised in order to achieve accurate power models.

For this case study, our methodology minimizes error in power prediction. This research has been tested using real Cloud applications resulting on an average error in power estimation of 3.98%. Our work improves the possibilities of deriving Cloud energy efficient policies in Cloud data centers being applicable to other computing environments with similar characteristics.

8.1 Introduction

As we present in the previous chapter, some metaheuristics as GP, perform FE for selecting an optimal set of features that best describe an optimization problem. Those features consist of measurable properties or explanatory variables of a phenomenon. Finding relevant features typically helps with prediction, but correlations and combinations of representative variables, also provided by FE, may offer a straightforward view of the problem thus generating better solutions. Also, designer's expertise is not required to process a high volume of data as GE is an automatic method. However, GE provides a vast space of solutions that may be bounded to achieve algorithm efficiency.

Otherwise, classical regressions as least absolute shrinkage and selection operator techniques provide models with linearity, convexity and differentiability attributes, which are highly appreciated for describing systems performance. Thus, combining the modeling properties of classical techniques with metaheuristics may be interesting in order to find automatic modeling approaches that also present these kind of desirable attributes. In our previous work presented in the previous chapter, we apply an evolutionary algorithm to infer power models for enterprise servers. This technique achieves very good accuracy results, but does not provide linearity, convexity and differentiability properties to models.

In this work we propose a novel methodology for the automatic inference of accurate models that combines the benefits offered by both classic and evolutionary strategies. First, SR performed by a GE algorithm finds optimal sets of features that best describe the system behavior. Then, a classic regression is used to solve our optimization problem using this set of features providing the model coefficients. Finally, our approach provides an accurate model that is linear, convex and derivative and also uses the optimal set of features. This methodology can be applied to a broad set of optimization problems regarding complex systems. This chapter presents a case study for its application in the area of Cloud power modeling as it is a relevant challenge nowadays.

The work proposed in this chapter makes substantial contributions in the area of power modeling of Cloud servers taking into account these factors. We envision a powerful method for the automatic identification of fast and accurate power models that target high-end Cloud server architectures. Our methodology considers the main sources of power consumption as well as the architecture-dependent parameters that drive today's most relevant optimization policies.

8.1.1 Contributions

Our work makes the following contributions:

- We propose a method for the automatic generation of fast and accurate models adapted to the behavior of complex systems.
- Resulting models include combination and correlation of variables due to the FE and SR performed by GE. Therefore, the models incorporate the optimal selection of representative features that best describe system performance.
- Through the combination of GE and classical regression provided by our approach, the inferred models present linearity, convexity and differentiability properties.
- As a case study, different power models have been built and tested for a high-end server architecture running several real applications that can be commonly found in nowadays' Cloud data centers, achieving low error when compared with real measurements.
- Testing for different applications (web search engines, and both memory and CPU-intensive applications) shows an average error of 3.98% in power estimation.

The remainder of this chapter is organized as follows: Section 8.2 provides the background algorithms used for the model optimization. The methodology description is presented in Section 8.3. In Section 8.4 we provide a case study where our optimization modeling methodology is applied. Section 8.5 describes profusely the experimental results. Finally, in Section 8.6 the main conclusions are drawn.

8.2 Algorithm description

8.2.1 Least absolute shrinkage and selection operator

Tibshirani proposes the least least absolute shrinkage and selection operator (lasso) algorithm [148] that minimizes residual summation of squares according to the summation of the absolute value of the coefficients that are lower than a defined constant.

The algorithm combines the favorable features of both subset selection and ridge regression (like stability) and offers a linear, convex and derivable solution. *Lasso* provides interpretable models shrinking some of the coefficients and setting others to exactly zero values for generalized regression problems.

For a given non-negative value of λ, the *lasso* algorithm solves the following problem:

$$\min_{\beta_0,\beta}\left(\frac{1}{2N}\sum_{i=1}^{N}(y_i-\beta_0-x_i^T\beta)^2+\lambda\sum_{j=1}^{p}|\beta_j|\right) \tag{8.1}$$

where:

- β is a vector of p components. *Lasso* algorithm involves the L^1 norm of β
- β_0 defines a scalar value.

- N is the number of observations.
- y_i provides the response at observation i.
- x_i presents the vector of p values at observation i.
- λ is a non-negative regularization parameter corresponding to one value of Lambda. The number of nonzero components of β decreases as λ increases.

At the end, we combine the use of GE, which generates the set of relevant features, with *lasso*, which computes the coefficients and the independent term in the final linear model.

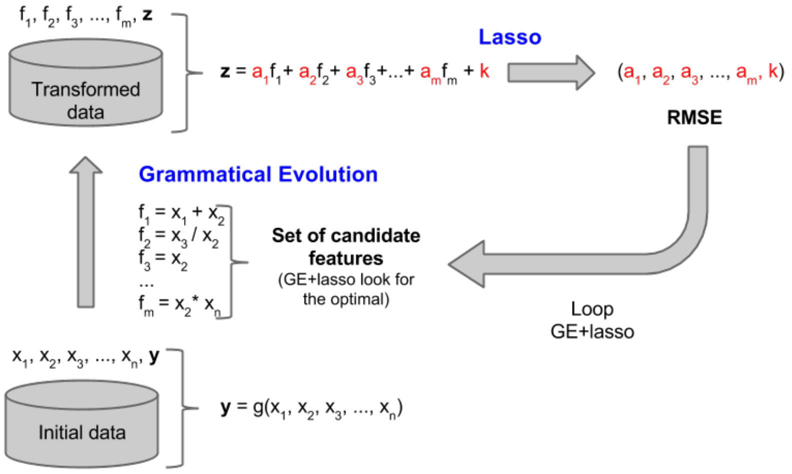
As a result, our GE+*lasso* framework solves our optimization problem that targets the generation of accurate power models for high-end servers.

8.3 Devised Methodology

The fast and accurate modeling of complex systems is a relevant target nowadays. Modeling techniques allow designers to estimate the effects of variations in the performance of a system. Complex systems present non-linear characteristics as well as a high number of potential variables. Also, the optimal set of features that impacts the system performance is not well known as many mathematical relationships can exist among them.

Hence, we propose a methodology that considers all these factors by combining the benefits of both GE algorithms and classical *lasso* regressions. This technique provides a generic and effective modeling approach that could be applied to numerous problems regarding complex systems, where the number of relevant variables or their interdependence are not known.

Figure 8.1 shows the proposed methodology approach for the optimization of the modeling problem. Detailed explanations of the different phases are summarized in the following subsections.



$f_1, f_2, f_3, ..., f_m$, **z**
Transformed data
z = $a_1f_1 + a_2f_2 + a_3f_3 + ... + a_mf_m + k$
Lasso
$(a_1, a_2, a_3, ..., a_m, k)$
RMSE
Grammatical Evolution
$f_1 = x_1 + x_2$
$f_2 = x_3 / x_2$
$f_3 = x_2$
...
$f_m = x_2 * x_n$
Set of candidate features
(GE+lasso look for the optimal)
Loop GE+lasso
$x_1, x_2, x_3, ..., x_n$, **y**
Initial data
y = $g(x_1, x_2, x_3, ..., x_n)$

Figure 8.1: Optimized modeling using our GE+*lasso* methodology.

8.3.1 GE feature selection

Given an extensive set of parameters that may cause an effect on system performance, FE selects the optimal set that best describes the system behavior. Also, this technique, which is provided by GE, avoids the inclusion of irrelevant features while incorporating correlations and combinations of representative variables.

The input to our approach consists of a vector of initial data that includes the entire set of variables x_n extracted from the system.

$$\vec{y} = g_1(x_1, x_2, x_3, \ldots, x_n) \tag{8.2}$$

All these parameters are entered in the GE algorithm to start the optimization process.

Each individual of the GE encodes its own set of candidate features $f_1, f_2, f_3, \ldots, f_m$. The candidate features follow the rules imposed by a BNF grammar allowing the occurrence of a wide variety of operations and operands to favor building optimal sets of features. Figure 8.2 shows an example of a BNF grammar for this approach.

```
<list_features> ::= <feature> | <feature>;<list_features>
<feature>       ::= (<feature><op><feature>)
| <preop>(<feature>) | <var>
<op>            ::= + | - | * | /
<preop>         ::= exp | sin | cos | ln
<var>           ::= x[0] | x[1] | x[2] | x[3] | ... | x[n]
```

Figure 8.2: Grammar in BNF format. x variables, with $i = 0 \ldots n$, represent each parameter obtained from the system.

This grammar provides the operations $+$, $-$, $*$, / and preoperators exp, sin, cos, ln. The space of solutions is easily modified by incorporating a broader set of relationships between operands to the BNF grammar.

The output of the GE stage consists of a matrix that includes all the candidate features provided by individuals. The output vector of each individual has its own set of m candidate features that intends to minimize the fitness function provided for the system optimization.

$$\vec{z} = g_2(f_1, f_2, f_3, \ldots, f_m) \tag{8.3}$$

8.3.2 *Lasso* generic model generation

Modeling procedures usually intend to interpret systems' behavior. They have the purpose of acquiring additional knowledge from the final models once these have been derived. Linearity, convexity and differentiability offered by the *lasso* classic regression help modeling to be a more explanatory and repeatable process. In addition, whereas GE is able to find complex symbolic expressions, it does not perform well in parameter identification, mainly because the exploration of real numbers is not easily representable in BNFs. Due to these facts, we have included the *lasso* algorithm in our methodology in order to manage the coefficient generation of the model.

As can be seen in Figure 8.1, each individual of the GE provides a set of candidate features to *lasso*. This classical regression is in charge of deriving the optimized model for each individual by solving the following equation.

$$\vec{z} = a_1 f_1 + a_2 f_2 + a_3 f_3 + \cdots + a_m f_m + k \tag{8.4}$$

Lasso offers the set of optimized coefficients (a_1, a_2, a_3, . . . , a_m, k) for each individual that minimizes the fitness function. This process provides the goodness of each individual. All this information feeds back the GE algorithm to generate the next population of individuals through selection, crossover and mutation, creating a loop. The loop continues executing until it completes the number of generations defined by the GE. This process results in the set of models that best fits power performance.

8.3.3 Fitness evaluation

As our main target is to build accurate models, our fitness function includes the error resulting in the estimation process. The fitness function presented in 8.5 leads the evolution to obtain optimized solutions thus minimizing the RMSD.

$$F = \sqrt{\frac{1}{N} \cdot \sum_{n} {e_{\mathrm{n}}}^{2}} \tag{8.5}$$

$$e_{\mathrm{n}} = |P(n) - \widehat{P}(n)|, \qquad 1 \leq n \leq N \tag{8.6}$$

The estimation error, defined as e_{n}, represents the deviation between the measure given by system monitoring P, and the estimation obtained by the model $\widehat{P}$. n represents each sample of the entire set of N samples used to train the algorithms.

8.4 Case Study

In this section we describe a particular case study for the application of the devised methodology presented in Section 8.3. The problem to be solved is the fast and accurate estimation of the power consumption in virtualized enterprise servers performing Cloud applications. Our power model considers heterogeneity of servers, as well as specific technological features and non-traditional parameters of the target architecture that affect power consumption. Hence, we propose our modeling technique that considers all these factors by combining the benefits of both GE algorithms and classical *lasso* regressions.

First, a GE algorithm is applied to extract those relevant features that best describe power consumption sources. The features may also include correlations and combinations of representative variables due to the FE performed by GE. Then, the *lasso* algorithm takes the optimal set of features in order to infer an expression that characterizes the power behavior of the target architecture of a Cloud server. As a result, we derive a highly accurate, linear and convex power model, targeting a specific server architecture, which is automatically generated by our evolutionary methodology.

We apply our methodology described in Section 8.3 to real measures gathered from a high-end Cloud server in order to infer an accurate power model. Also, we provide an experimental scenario for various workloads with the purpose of building and testing our methodology.

Instead of restricting the use of synthetic workloads only for training the algorithms, and limiting the use of real Cloud benchmarks exclusively for testing, we have used both workloads for the two purposes. This procedure provides automation for the progressive incorporation of additional benchmarks to the model.

8.5 Experimental results

Our data collection (explained in Section 5.2) has been split into training and testing sets. Training stage performs feature selection and builds the power model according to our grammar and fitness function. Next, the testing stage examines the power model accuracy. The algorithm proposed by our methodology runs completely 20 times using the same grammar and fitness function configuration. For each run, we randomly select 50% of each data set for training and 50% for testing stage, thus obtaining 20 final power models. This procedure validates the variability and versatility of the system, by analyzing the occurrence of local minima in optimization scenarios.

8.5.1 Algorithm setup

GE setup parameters

We use GE to obtain a set of candidate features that best describe our optimization problem. To obtain adequate solutions we tune the algorithm using the following parameters:

- Population size: 250 individuals
- Number of generations: 3000
- Chromosome length: 100 codons
- Mutation probability: inversely proportional to the number of rules, $1/4$ in our case.
- Crossover probability: 0.9
- Maximum wraps: 3
- Codon size: 256
- Tournament size: 2 (binary)

As we strictly seek for simple combination of features, our proposed BNF grammar only provides the operations $+| - | * |/$. The space of solutions is easily increased by incorporating more complex relationships between operands to the BNF grammar. Figure 8.3 shows the BNF grammar proposed for this case study.

```
<list_features> ::= <feature> | <feature>;<list_features>
<feature>       ::= (<feature><op><feature>) | <var>
<op>            ::= + | - | * | /
<var>           ::= x[0] | x[1] | x[2] | x[3] | x[4] | x[5] | x[6]
```

Figure 8.3: Grammar in BNF format. x variables, with $i = 0 \ldots 6$, represent processor and memory temperatures, fan speed, processor and memory utilization percentages, processor frequency and voltage, respectively.

Lasso setup parameters

We use the *lasso* algorithm to obtain a set of candidate solutions with low error, when compared with the real power consumption measurements in order to solve our optimization modeling problem. *Lasso* setup parameters are the following:

- Number of observations: 100
- λ regularization parameter: Geometric sequence of 100 values, the largest just sufficient to produce zero coefficients.
- λ regularization parameter: $1 \cdot 10^{-4}$

8.5.2 Training stage

We have performed variable standardization for every feature (in the range $[1, 2]$) to ensure the same probability of appearance for all the variables and to enhance the GE symbolic regression. Experiments with more than 4 features do not provide better values for RMSD. Hence, we have bounded their occurrence to a maximum of 4 by penalizing higher number of features in our fitness evaluation function. This also facilitates the generation of simpler models, which are faster and easy to be applied, in order to be used for real-time power optimizations.

Table 8.1 shows the phenotypes obtained for each feature combined with the coefficients provided by *lasso*, which are obtained for 20 complete runs of our methodology algorithm. Fitness results, which correspond to the RMSD between measured and estimated power consumption (see Equation 8.5), are shown in Table 8.2 for the training stage. Both Table 8.1 and Table 8.2 present the results for the best model of each execution.

Table 8.1: Power models obtained by combining GE features and *lasso* coefficients for 20 executions

Run	$a_1 \cdot f_1 + a_2 \cdot f_2 + a_3 \cdot f_3 + a_4 \cdot f_4$ + K
1	0.288 · Tcpu + 0.127 · (((Tcpu*Ucpu)-Umem)*Fan) + 0.220 · (Fan*Tmem) + -0.450 · Fan + 1.043
2	0.173 · Ucpu + 0.438 · Tcpu + -0.209 · Fan + 0.070 · (Tmem/(Umem/(Fan*Tmem))) + 0.636
3	0.256 · (Fan/(Ucpu/Tmem)) + 0.346 · Ucpu + -0.694 · (Fan/Tcpu) + 1.151
4	-0.376 · Tmem + -0.033 · ((((Fan/Tcpu)/(Fcpu+Fcpu))/((Fan/(Vcpu+Umem))/Ucpu))*Fan) + 0.606 · ((Fan/((Umem+(Fan+(Fcpu/Fcpu)))*(Fan/Ucpu)))+(Fan+Tmem)) + 0.786 · ((Fcpu-(Fcpu+Fan))/Tcpu) + 0.810
5	0.181 · Ucpu + 0.254 · (Fan*Tmem) + 0.378 · Umem + -0.345 · (((Umem+Umem)*Fan)/Tcpu) + 0.939
6	0.483 · (Ucpu-Fan) + 0.030 · ((Tmem+Fan)*((Fan-(Tmem/((Ucpu+Vcpu)+(Fan+Fan))))*(Fan*Fan))) + 0.220 · Tmem + 0.430 · (Tcpu/Ucpu) + 0.402
7	0.506 · Tcpu + 0.195 · ((Ucpu/(Vcpu+(Tmem-Umem)))*Vcpu) + -0.319 · Fan + -0.199 · (((Fan+Umem)*((Umem-Tmem)/Tcpu))*Fan) + 0.704
8	0.084 · (Ucpu/Vcpu) + 0.473 · Tmem + 0.499 · (Ucpu/(Ucpu*(((Fcpu-Vcpu)+Tmem)/Tcpu))) + -0.019 · (Fan-(((Fan+Vcpu)*Tcpu)*((Ucpu*Tmem)-(Vcpu-Fcpu)))) + 0.046
9	0.927 · Ucpu + -0.380 · Fan + 0.232 · (((Tmem*((Fan+Umem)-Ucpu))+(Tcpu-(Ucpu*Umem)))-Ucpu) + 0.180 · Tcpu + 0.365
10	-0.073 · Tmem + 0.106 · (((Tmem+Fan)*Fan)-Umem) + 0.194 · (Ucpu+Tmem) + 0.437 · (Tcpu-Fan) + 0.665
11	-0.117 · (Tmem*(Ucpu-(Tmem*Fan))) + 0.317 · Ucpu + 0.377 · (Tcpu-Fan) + 0.810
12	-0.070 · Umem + 0.174 · Ucpu + 0.647 · (Tcpu/Tmem) + 0.647 · Tmem + -0.318
13	0.291 · (Tmem+Fan) + -0.409 · (Fan/Tcpu) + 0.234 · Tcpu + 0.423 · (Ucpu/(Tmem+Umem)) + 0.442
14	0.093 · (Ucpu+(Ucpu+(Tmem*Tmem))) + -0.019 · ((Tcpu-((Tmem*Fan)-Vcpu))-Vcpu) + -0.081 · (Tmem+Umem) + 0.462 · Tcpu + 0.526
15	-0.004 · Fcpu + 0.380 · (Ucpu/(Umem+Fan)) + 0.054 · (Tmem*(Tmem+Fan)) + 0.454 · Tcpu + 0.347
16	-0.010 · Fan + -0.155 · (((Fan/Tmem)-(Tmem/Ucpu))*Fan) + 0.282 · Ucpu + 0.417 · Tcpu + 0.393
17	0.242 · (Fan*(Tmem/Ucpu)) + 0.396 · (Tcpu-Fan) + 0.001 · Fcpu + 0.344 · Ucpu + 0.508
18	0.448 · Tmem + -0.178 · Umem + -0.221 · (((((Tcpu/(Vcpu/(Fcpu-Vcpu)))-Ucpu)+Fan)/Tmem)-(Tcpu-(Tmem-(Tcpu+Fan)))) + 0.100 · (Umem/Fan) + 0.271
19	0.134 · Ucpu + 0.241 · (Tmem*Fan) + 0.066 · Ucpu + -0.403 · ((Fan-Tcpu)/Umem) + 0.653
20	-0.433 · (((Fan-(Ucpu+Umem))/Fan)-(Tcpu+Fan)) + -0.295 · Umem + -0.102 · Fan + 0.235 · (((Tmem-Umem)-Ucpu)+Fan) + 0.184

Table 8.2: RMSD and Average testing error percentages for 20 executions

Run	Train (RMSD)	Testing (RMSD)	Synthetic (%)	mcf (%)	perlb (%)	WebSearch (%)	Total (%)
1	0.1069	0.1068	3.985	4.097	4.463	4.147	4.173
2	0.1068	0.1067	3.984	4.099	4.463	4.110	4.164
3	0.1070	0.1068	3.995	4.110	4.504	4.145	4.189
4	0.1070	0.1071	4.007	4.085	4.469	4.155	4.179
5	0.1069	0.1069	3.991	4.106	4.494	4.113	4.176
6	0.1071	0.1068	3.988	4.085	4.459	4.153	4.171
7	0.1070	0.1072	3.995	4.042	4.462	4.101	4.150
8	0.1071	0.1072	3.994	3.996	4.559	4.101	4.162
9	0.1072	0.1072	4.033	3.884	3.990	4.059	3.991
10	0.1067	0.1072	4.052	3.894	3.969	4.031	3.986
11	0.1073	0.1075	4.023	3.926	3.963	4.063	3.994
12	0.1071	0.1076	4.098	3.896	3.951	4.030	3.994
13	0.1070	0.1070	4.073	3.939	4.173	4.243	4.107
14	0.1072	0.1072	4.088	3.935	4.174	4.184	4.096
15	0.1071	0.1070	4.083	3.922	4.161	4.246	4.103
16	0.1071	0.1070	4.060	3.937	4.164	4.217	4.095
17	0.1079	0.1057	3.951	4.136	4.208	4.056	4.088
18	0.1081	0.1060	3.981	4.171	4.180	4.050	4.095
19	0.1082	0.1060	3.953	4.190	4.224	4.212	4.145
20	0.1082	0.1059	3.974	4.205	4.178	4.074	4.108

As can be seen in Table 8.1, power model solutions combine features that correspond to a single variable with others that merge a combination of several parameters. On the one hand, there are single-variable features that appear in up to 50% of the power model solutions. This shows that there are linear dependencies with certain parameters, as *Ucpu*, *Tcpu*, and *Tmem* that are consistent regardless of the workload that is used for training and testing. On the other hand, variables as *Vcpu*, *Fcpu* and *Umem* are seldom treated as a feature in the model solutions. However, they systematically appear when combined with other variables.

Ucpu and *Tcpu* provide physical information about the dynamic consumption of the CPU and its variability with the workload. *Tcpu* is also correlated to the fan speed and the static contribution of the CPU that depends on the inlet temperature of the server. *Vcpu*, *Fcpu* present the dependence with the DVFS mode of the CPU. On the other hand, *Umem* and *Tmem* provide the information regarding the dynamic consumption of the memory and *Tmem* is also correlated to fan speed and to the static consumption, which present a dependence with the inlet temperature of the server. These results show that there exist input parameters that are not relevant for the modeling or that they are correlated to other features, and how their inclusion could decrease the model accuracy. Model training for run 10 shows the lowest RMSD error of 0.1067.

8.5.3 Model testing

At this stage, we analyze the quality of the models that we have simultaneously tested for the 20 complete executions of our methodology algorithm. Results are also analyzed particularly for the testing data that corresponds to each benchmark data set in order to verify the estimation reliability of the models for different workloads. Table 8.2 shows testing average error percentages particularized for the different benchmark data sets. These values have been obtained according to the following formulation:

$$e_{\mathrm{AVG}} = \sqrt{\frac{1}{N} \cdot \sum_{n} \left(\frac{|P(n) - \widehat{P}(n)| \cdot 100}{P(n)} \right)^2} , 1 \leq n \leq N \tag{8.7}$$

where P is the power measurement given by the current clamp and $\widehat{P}$ is the power estimated by the model phenotype. n represents each sample of the entire set of N samples.

Total average error for the testing data set shows a lowest error of 3.98% (as shown in Table 8.2). Best testing error corresponds to the solution with lower training error. Solutions can be broken down for those samples that belong to different tests, achieving testing errors of 4.052%, 3.894%, 3.969% and 4.031% for synthetic, *SPEC CPU 2006 mcf*, *SPEC CPU 2006 perlbench* and *WebSearch* workloads respectively. This fact confirms that our methodology works well for our scenario, extracting optimized sets of features and coefficients that are consistent even for 20 runs with random selection of both training and testing data sets.

For our optimized model using GE+Lasso, we obtain a mean error between the estimated power and the real measurement of $-1.47 \cdot 10^{-5}$ and a standard deviation of 0.1069. Figure 8.4 shows the power error distribution for this model. According to this, we can conclude that the error in terms of power of a high percentage of the samples ranges from -0.1069 to 0.1069. In Figure 8.5, the fitting of our optimized power model is provided.

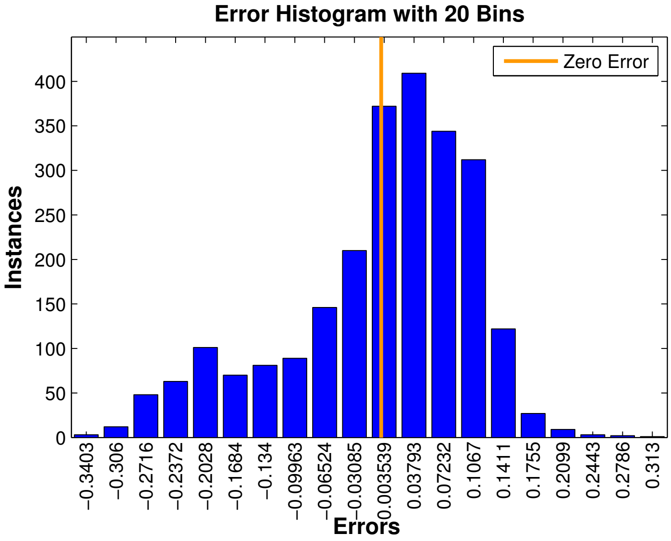


Figure 8.4: Power error distribution for our optimized model using GE+Lasso.

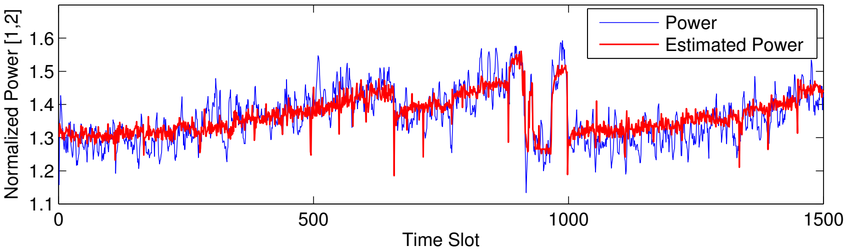


Figure 8.5: Model fitting for our optimized model using GE+Lasso.

In order to compare the performance of our models, Table 8.3 presents the average error percentage for all the baseline models, for our models proposed in Chapter 6 ($lsqcurvefit$ and OMOPSO) and Chapter 7 (GE) and for the model obtained in this work, GE+lasso. Linear, quadratic, cubic and sqrt baselines provide errors that are higher, from 7.66% to 6.20%. Including DVFS-awareness, fan speed-awareness and thermal-awareness to the power model (as in the DVFS, DVFS&fan and $lsqcurvefit$ models) improves the error when compared to linear, quadratic, cubic and sqrt models from 6.20% to 5.77%, 5.37% and 5.24% respectively. By optimizing the set of features of our analytical model using OMOPSO, the average error is reduced from 5.24% to 4.87% respectively. Then, optimizing the feature selection, using an automatic modeling based on GE, the average error is reduced from 4.87% for the OMOPSO model to 4.22% for our GE model. However, the appearance of linearity, derivability and convexity properties using GE is difficult using BNF grammars. Our GE+lasso methodology intends to enhance the modeling process, so the functions that make up the model are closer to the physical model according to our hypothesis. Finally, enforcing linearity, derivability and convexity properties using our GE+lasso methodology, the testing error outperforms the GE value from 4.22% to 3.98%.

Table 8.3: Average error percentage comparison with the baseline models

Model	Linear	Quad.	Cubic	Sqrt	DVFS	DVFS&fan	lsqcurv.	OMOPSO	GE	GE+lasso
Training	5.83%	6.25%	6.46%	5.61%	5.58%	5.32%	4.85%	4.03%	4.10%	4.05%
Testing	6.58%	7.21%	7.66%	6.20%	5.77%	5.37%	5.24%	4.87%	4.22%	3.98%

Our methodology application shows very accurate testing results for all of the complete runs ranging from 3.98% to 4.18%, thus outperforming all our baselines. The obtained results are robust, as they have been obtained for a heterogeneous mix of workloads so the power models are not workload-dependent. According to these results, we can infer that our methodology is effective for performing feature selection and building accurate multi-parametric, linear, convex and differentiable power models for high-end Cloud servers. This technique can be considered as a starting point for implementing energy optimization policies for Cloud computing facilities.

8.6 Summary

This chapter has presented a novel work in the field of FE and SR for the automatic inference of accurate models. Resulting models include combination and correlation of variables due to the FE and SR performed by GE. Therefore, the models incorporate the optimal selection of representative features that best describe the target problem while providing linearity, convexity and differentiability characteristics due to lasso incorporation.

As a proof of concept, the devised methodology has been applied to a current computing problem, the power modeling of high-end servers in a Cloud environment. In this context, the proposed methodology has shown relevant benefits with respect to state-of-the-art approaches, like better accuracy and the opportunity to consider a broader number of input parameters that can be exploited by further power optimization techniques.

Our approach presents an automatic method for the identification of an accurate power model particularized for each target architecture, which is consistent with current architectures. In this research, optimal features provided by GE are included in a classical regression resulting in a specific model instance for every target architecture that is linear, convex and derivable. Also the execution of the resulting power model is fast, making it suitable for run-time optimization techniques. Current models (linear, quadratic, cubic and square root among others), which do not consider both DVFS and thermal-awareness, present power accuracies that range from 7.66% to 5.37%. Our power model, inferred automaticaly, provides a testing error of 3.98%, outperforming the state-of-the-art approaches.

This part of the present dissertation introduces our power models that are aware of DVFS and thermal impact on power consumption. The information regarding their dependence with these non-traditional parameters provides the base for deriving new strategies to improve energy savings in the data center infrastructure. The following part presents DVFS and thermal aware proactive consolidation strategies that use these models to consider the energy globally. Thus, decisions are based on information from all available subsystems to perform different energy optimizations from a holistic perspective.

Part III

Data Center Proactive Energy Optimization

9. DVFS-Aware Dynamic Consolidation for Energy-Efficient Clouds

"They see the immediate situation. They think narrowly and they call it 'being focused'. They don't see the surround. They don't see the consequences."

— Michael Crichton, *Jurassic Park*

Data centers are becoming unsustainable in terms of power consumption and growing energy costs so Cloud providers have to face the major challenge of placing them on a more scalable curve. Also, Cloud services are provided under strict Service Level Agreement conditions, so trade-offs between energy and performance have to be taken into account. Techniques as DVFS and consolidation are commonly used to reduce the energy consumption in data centers, although they are applied independently and their effects on Quality of Service are not always considered. Thus, understanding the relationship between power, DVFS, consolidation and performance is crucial to enable energy-efficient management at the data center level.

In this chapter we propose a DVFS policy that reduces power consumption while preventing performance degradation, and a DVFS-aware consolidation policy that optimizes consumption, considering the DVFS configuration that would be necessary when mapping VMs to maintain QoS. We have performed an extensive evaluation on the CloudSim toolkit using real Cloud traces and one of our accurate power models based on data gathered from real servers. Our results demonstrate that including DVFS awareness in workload management provides substantial energy savings of up to 45.76%, for scenarios under dynamic workload conditions, when compared with a state-of-the-art baseline. These outcomes outperform previous approaches, which do not consider integrated use of DVFS and consolidation strategies.

9.1 Introduction

The main contributor to the energy consumption in a data center is the IT infrastructure, which consists of servers and other IT equipment. The IT power in the data center is dominated by the power consumption of the enterprise servers, representing up to 60% of the overall data center consumption [14]. The power usage of an enterprise server can be divided into dynamic and static contributions. Dynamic power depends on the switching transistors in electronic devices during workload execution. Static consumption associated to the power dissipation of powered-on servers represents around the 70% of the total server consumption and is strongly correlated with temperature due to the leakage currents that increase as technology scales down.

The Cloud computing paradigm helps improving energy efficiency, reducing the carbon footprint per executed task and diminishing CO_2 emissions [6] by increasing data centers overall utilization. The main reason is that, in the Cloud model, the computing resources are shared among users and applications so, less powered-on servers are needed, which means less static consumption. In this way, smaller facilities are able to consolidate higher incoming workloads, thus reducing the computing and cooling energy requirements.

To meet the growing demand for their services and ensure minimal costs, Cloud providers need to implement an energy-efficient management of physical resources. Therefore, optimization approaches that rely on accurate power models and optimize the configuration of server parameters (voltage and working frequency, workload assignment, etc.) can be devised. Furthermore, as many applications expect services to be delivered as per SLA, power consumption in data centers may be minimized without violating these requirements whenever it is feasible.

From the application-framework viewpoint, Cloud workloads present additional restrictions as 24/7 availability, and SLA constraints among others. In this computing paradigm, workloads hardly use 100% of CPU resources, and their execution time is strongly constrained by contracts between Cloud providers and clients. These restrictions have to be taken into account when minimizing energy consumption as they impose additional boundaries to efficiency optimization strategies. QoS would be determined by these constraints and it would be impacted by performance degradation.

Also, Cloud scenarios present workloads that vary significantly over time. This fluctuation hinders the optimal allocation of resources, which requires a trade-off between consolidation and performance. Workload variation impacts on the performance of two of the main strategies for energy-efficiency in Cloud data centers: DVFS and Consolidation.

DVFS strategies modify frequency according to the variations on the utilization performed by dynamic workload. These policies help to dynamically reduce the consumption of resources as dynamic power is frequency-dependent. DVFS has been traditionally applied to decrease the power consumption of underutilized resources as it may incur on SLA violations. On the other hand, consolidation policies decrease significantly the static consumption by reducing the number of active servers, increasing their utilization.

Dynamic workload scenarios would require policies to adapt the operating server set to the workload needs during runtime in order to minimize performance degradation due to oversubscription. However, both strategies are applied independently, regardless the effects that consolidation have on DVFS and vice versa. Therefore, the implementation of DVFS-aware consolidation policies has the potential to optimize the energy consumption of highly variable workloads in Cloud data centers.

The **key contributions** of our work are 1) a DVFS policy that takes into account the trade-offs between energy consumption and performance degradation; 2) a novel consolidation algorithm that is aware of the frequency that would be necessary when allocating a Cloud workload in order to maintain QoS. Our frequency-aware consolidation strategy reduces the energy consumption of the data center, making use of DVFS to reduce the dynamic power consumption of servers, also ensuring SLA. The algorithm is light and offers an elastic scale-out under varying demand of resources.

The rest of the chapter is organized as follows: Section 9.2 gives further information on the related work on this topic. The proposed DVFS policy that considers both energy consumption and performance degradation is presented in Section 9.3. This section also provides our Frequency-Aware Optimization strategy for the energy optimization of the IT infrastructure in Cloud data centers. The simulation configuration is detailed in Section 9.4. In Section 9.5, we describe profusely the performance evaluation and the experimental results. Finally, Section 9.6 concludes the work.

9.2 Related Work

Recently, there has been a growing interest in developing techniques to provide power management for servers operating in a Cloud. The complexity of the power management and workload allocation in servers has been described by Gandhi et al. [149] and Rafique et al. [150], where the authors show that the optimal power allocation is non-obvious, and, in fact, depends on many factors such as the power-to-frequency relationship in processors, or the arrival rate of jobs. Thus, it is critical to understand quantitatively the relationship between power consumption and DVFS at the system level to optimize the use of the deployed Cloud services.

DVFS is by far the most frequent technique at the architectural-level as well as one of the currently most efficient methods to achieve energy savings. This technique scales power according to the workload in a system by reducing both operating voltage and frequency. Reducing the operating frequency and voltage slows the switching activity achieving energy savings but also decreasing the system performance. DVFS implementation on CPU results in an almost linear relationship between power and frequency, taking into account that the set of states of frequency and voltage of the CPU is limited. Only by applying this technique on a server CPU, up to 34% energy savings in dynamic consumption can be reached as presented by Le Sueur et al. [58].

DVFS has been mainly applied to enhance energy efficient scheduling on idle servers, or those performing under light workload conditions [151], and during the execution of noncritical tasks [152]. However, a recent research shows that DVFS can be also used to meet deadlines in mixed-criticality systems [153]. Furthermore, DVFS-based scheduling research on multiprocessor systems shows promising results. Rizvandi et al. [154] achieved considerable energy savings by applying this technique on up to 32-processor systems for HPC workload. However, the effects of DVFS in loaded servers have not been analyzed yet for Cloud scenarios. The QoS offered depends on the SLA contracted to Cloud providers could be violated under certain frequency-voltage conditions. DVFS-aware approaches could help to reduce the energy consumption of Cloud facilities but new algorithms have to be devised for large scale data center infrastructures also taking into account the SLA considerations of Cloud workloads.

On the other hand, many of the recent research works have focused on reducing power consumption in cluster systems by power-aware VM consolidation techniques, as they help to increase resource utilization in virtualized data centers. Consolidation uses virtualization to share resources, allowing multiple instances of operating systems to run concurrently on a single physical node. Virtualization and consolidation increase hardware utilization (up to 80% [15]) thus improving resource efficiency.

The resource demand variability of Cloud workloads is a critical factor in the consolidation problem as the performance degradation boundary has to be considered for both migrating VMs and reducing the active server set [155]. Balancing the resource utilization of servers during consolidation was performed by Calheiros et al. [156] to minimize power consumption and resource wastage. In the research proposed by Hermenier et al. [157], their consolidation manager reduces the VM migration overhead. Also, there exist interesting works that focus on modeling the energy consumption of the migration process as the research proposed by Haikun et al. [158] and De Maio et al. [159].

However, DVFS-Aware consolidation techniques that maintain QoS in data centers have not been fulfilled yet. Although some combined application of DVFS and consolidation methods for Cloud environments can be found, no one of them are considering performance degradation due to VM migration or resource over-provisioning. In the research presented by Wang et al. [160], the consolidation is performed regardless the frequency impact, and the DVFS is applied separately. The approach presented by Petrucci et al. [161] shows the dependence of power with frequency but the algorithm does not scale for large data centers and SLA violations are not taken into account.

Our work provides a novel DVFS-aware consolidation algorithm that helps to reduce the energy consumption of data centers under dynamic workload conditions. The proposed strategy considers the trade-offs between energy consumption and performance degradation thus maintaining QoS. The work presented in this chapter outperforms previous contributions by allowing the optimization of Cloud data centers from a proactive perspective in terms of IT energy consumption and ensuring the QoS of Cloud-based services.

9.3 Frequency-Aware VM consolidation

The major challenge that we face in this work is to reduce the energy consumption of the IT infrastructure of data centers, while maintaining QoS, and under dynamic workload conditions. In our previous work [33], Chapter 6 of this dissertation, we derived a complete accurate model to calculate the total energy consumption of a server $E_{\text{host}}(m, k)$ in $kW \cdot h$ that can be seen in Equation 9.1:

$$
\begin{aligned}
E_{\text{host}}(m,k,w) &= P_{\text{host}}(m,k,w) \cdot \Delta t = (P_{\text{dyn}}(m,k,w) + P_{\text{stat}}(m)) \cdot \Delta t && (9.1)\\
T &= \{t_1, ..., t_i, ..., t_T\} && (9.2)\\
\Delta t &= t_{i+1} - t_i && (9.3)\\
P_{\text{dyn}}(m,k,w) &= \alpha(m) \cdot V_{\text{DD}}^2(m,k) \cdot f_{\text{op}}(m,k) \cdot u_{\text{cpu}}(m,k,w) && (9.4)\\
P_{\text{stat}}(m) &= \beta(m) \cdot T_{\text{mem}}^2(m) + \gamma(m) \cdot FS^3(m) && (9.5)
\end{aligned}
$$

where Δt is the time along which the energy is calculated. In this research we assume a discrete set of times T in which the algorithm is evaluated in order to optimize the power performance of the system. We define each time t_i as the particular instant in which the system evaluates an incoming batch of workload. Our proposed model estimates the instantaneous electric power of a server in t_i so, the energy is computed for the time interval Δt between two workload evaluations, considering that the power is stable in this time period. For practical reasons, we have selected Δt to be $300s$ in our experiments, which is a realistic assumption for our setup.

$P_{\text{host}}(m,k,w)$, $P_{\text{dyn}}(m,k,w)$ and $P_{\text{stat}}(m)$ represent total, dynamic and static contributions of the power consumption in Watts of the physical machine m operating in a specific k DVFS mode for a specific workload w. In the research conducted in this chapter, we define $P_{\text{host}}(m,k,w)$ as the optimized expression obtained in Chapter 6 for modeling the server's power consumption that can be seen in Equation 6.22. In Equation 9.1, we split the expression into its dynamic and static contributions and rename the technological constants $\alpha_4(m)$ and $\beta_2(m)$ as $\alpha(m)$ and $\beta(m)$.

Our proposed model consists of 5 different variables: $u_{\text{cpu}}(m,k,w)$ is the averaged CPU percentage utilization of the specific server m and is proportional to the number of CPU cycles defined in Millions of Instructions Per Second (MIPS) in the range [0,1]. V_{DD} is the CPU supply voltage and f_{op} is the operating frequency in GHz. T_{mem} defines the averaged temperature of the main memory in Kelvin and FS represents the averaged fan speed in RPM. Depending on the target architecture, some factors might have higher impact than others. This model has been tested for Intel architectures achieving accuracy results of about 95% as can be seen in Chapter 6. Our model allows to obtain power estimations during run-time facilitating the integration of proactive strategies in real scenarios. Power consumption is measured with a current clamp, so we can validate our approach comparing our estimations with real values, obtaining a testing error of 4.87%.

As shown in Equation 9.4, the energy consumption due to the dynamic power consumption $P_{\text{dyn}}(m,k,w)$ depends on the workload profile executed in the server. So, the lower the $u_{\text{cpu}}(m,k,w)$, the lower the dynamic energy contribution. On the other hand, the static consumption represents the energy consumed due to power dissipation of a powered-on server, even if it is idle. This energy represents around 70% of the total server consumption. In this context, we can see some observations about the dynamic consolidation problem:

- **DVFS vs SLA.** DVFS can be used to achieve power savings because reducing the frequency and voltage of the CPU ($f_{op}(m,k)$ and $V_{DD}(m,k)$) slows its switching activity. However, it also impacts on the performance of the system by extending tasks duration (*t*), which can lead to the appearance of SLA violations and to the increase of energy consumption.

- **Underloaded servers.** If the workload is spread over a larger number of servers, the CPU utilization in each server will be lower, so the dynamic power contribution in each server will be also lower. As u_{cpu} is reduced, f_{op} can be scaled down thus decreasing the power contribution due to CPU frequency. However, the global energy consumption will be increased disproportionately due to the impact of static consumption of a higher number of servers.

- **Overloaded servers.** On the other hand, if the incoming workload is concentrated in a smaller set of servers, even though the static consumption is reduced, the QoS may be affected. This situation is intensified due to the dynamic variation of workload and, if the maximum server capacity is exceeded during peak loads, it would lead to performance degradation. To avoid overloaded servers, one or more VMs can be migrated from one server to another. However, VM migration has associated costs in terms of energy consumption and time, which could lead to SLA violations.

In this chapter we propose a strategy to allow the energy optimization of a Cloud under SLA constraints. As opposed to previous approaches, our work offers a DVFS policy that considers the trade-offs between energy consumption and performance degradation explained in subsection 9.3.1. Thus, frequency is managed according to the available states depending on the server architecture while ensuring QoS. On the other hand, in subsection 9.3.2 we provide an energy-aware dynamic placement algorithm that considers the frequency configuration according to the allocation of VMs. Finally, in subsection 9.3.3 we use both strategies combined to proactively optimize a Cloud under dynamic workload conditions.

9.3.1 DVFS-Performance Management

DVFS scales the power of the system varying both CPU frequency and voltage. Reducing the operating frequency and voltage slows the switching activity to achieve energy savings; however, it also impacts negatively on the performance of the system.

The CPU performance of a physical machine m can be characterized by the maximum workload (w) that can be run by its CPU without performance degradation at its maximum frequency ($f_{MAX}(m)$). Moreover, the real workload that may be run by the system depends on the current operating frequency of the CPU ($f(m,k)$). As we present in Equation 9.8, $f(m,k)$ can only take a value from a specific set of valid frequencies where k represents the operating DVFS mode. It is important to note that not all frequencies from 0 to $f_{MAX}(m)$ are available, as the set of states of frequency and voltage of the CPU for each physical machine is limited and it may be different depending on the architecture of the physical machine. We define the equivalent CPU utilization percentage ($u_{cpu_{eq}}(m,k)$) in Equation 9.9 as the maximum CPU utilization that could be used by the workload without performance degradation.

In this research we propose a DVFS policy that selects the minimum operating frequency that ensures that the equivalent utilization $u_{cpu_{eq}}(m,k)$ is greater or equal than the utilization required by all the workload allocated on the host at maximum frequency $u_{cpu}(m, f_{MAX}(m), w)$. This statement, which can be seen is Equation 9.10, helps to provide a frequency that does not degrade the performance of the workload, thus ensuring the SLA for CPU-bounded workloads. Finally, the utilization of the CPU at the system level for the operating frequency ($u_{cpu}(m,k,w)$) is shown in Equation 9.12.

$$\begin{aligned}
u_{cpu}(m,k,w) &\in [0,1] && (9.6)\\
& \quad where \quad u_{cpu_{MAX}} = 1 && (9.7)\\
f(m,k) &\in \{f_1(m), f_2(m), \cdots, f_k(m), \cdots, f_{MAX}(m)\} && (9.8)\\
u_{cpu_{eq}}(m,k) &= \frac{f(m,k)}{f_{MAX}(m)} \cdot u_{cpu_{MAX}} && (9.9)\\
f_{host}(m,k,w) &= min\{f(m,k)\} \quad that \quad ensures && (9.10)\\
& \quad u_{cpu_{eq}}(m,k) \geq u_{cpu}(m, f_{MAX}(m), w) && (9.11)\\
u_{cpu}(m,k,w) &= \frac{u_{cpu}(m, f_{MAX}, w)}{u_{cpu_{eq}}(m,k)} = u_{cpu}(m, f_{MAX}, w) \cdot \frac{f_{MAX}(m)}{f(m,k) \cdot u_{cpu_{MAX}}} && (9.12)
\end{aligned}$$

In order to motivate these metrics we provide the following case of use for the Fujitsu RX300 S6 server. The maximum frequency for this type of physical machine is $f_{MAX}(Fujitsu) = 2.4GHz$. For this example we assume that this server is able to run, at $2.4GHz$, a CPU-bounded workload w_{100} with a rate of requests of 100 requests per second without losing performance, while the system detects a CPU usage of $u_{cpu}(Fujitsu, f_{MAX}(Fujitsu), w_{100}) = 1$. One of the available operating frequencies for this server is $f(Fujitsu, 1) = f_1(Fujitsu) = 1.73GHz$ so, according to Equation 9.9, the equivalent CPU utilization percentage takes the value $u_{cpu_{eq}}(Fujitsu, 1) = 0.72$. Thus, if the utilization of the Fujitsu server, running at $1.73GHz$, exceeds the 72% of its total capacity, the required requests per second to be executed will exceed the limit rate that can be provided for this frequency, provoking a delay. On the other hand, when the utilization is kept below this threshold, no performance degradation occurs due to DVFS. These quality and performance metrics will be considered by the proposed energy optimization algorithm, so that they are not degraded (as it will be confirmed by the experimental results).

Our proposed DVFS management policy (DVFS-perf), presented in Algorithm 1 takes into account the previous relationships in order to improve energy efficiency, avoiding performance degradation. As inputs, we consider the complete set of hosts (*hostList*), and the set of valid frequencies (*frequenciesList*). For each host, the current value of CPU *utilization* is acquired in step 4 by using Equation 9.12 applied to the current utilization at the current frequency. This variable, which depends on the workload that is already hosted and running in the server, is monitored during runtime by using calls to the system utility (e.g. *Linux ps*). Then, $u_{cpu_{eq}}(m, k)$ (*eqUtilization*) is calculated for the different frequencies in *frequenciesList* in steps 5 to 9. The algorithm selects the minimum frequency that offers a suitable $u_{cpu_{eq}}(m, k)$ value that is greater or equal to the current *utilization* in the host in step 7. Finally, the DVFS configuration for the entire set of hosts is provided by *frequencyConfiguration*.

Algorithm 1 DVFS-perf configuration

```
Input: hostsList, frequenciesList
Output: frequencyConfiguration
 1: frequenciesList.sortIncreasingFrequency()
 2: maxFrequency ← frequenciesList.getMax()
 3: foreach host in hostList do
 4:    utilization ← host.getFmaxUtilization()
 5:    foreach frequency in frequenciesList do
 6:       eqUtilization ← frequency / maxFrequency
 7:       if eqUtilization ≥ utilization then
 8:          frequencyConfiguration.add(frequency)
 9:          break
10: return frequencyConfiguration
```

As dynamic power is reduced with frequency, our algorithm sets the operating frequency of each host to the lowest available value that provides sufficient CPU resources according to Equation 9.9. This ensures that the server offers sufficient resources based on the amount demanded by the allocated workload satisfying QoS.

We motivate these metrics by providing a case of use based on a Fujitsu RX300 S6 server, whose *maxFrequency* is $2.4GHz$. The operating frequencies set (*frequenciesList*) in GHz is $f(Fujitsu, k) = \{1.73, 1.86, 2.13, 2.26, 2.39, 2.40\}$. Our aim is to find the best *frequencyConfiguration* for a current *utilization* of the server of 80% at maximum frequency ($u_{cpu}(Fujitsu, f_{MAX}(Fujitsu), w_{80}) = 0.8$). First, we calculate the *eqUtilization* for the minimum frequency according to Equation 9.9 obtaining $u_{cpu_{eq}}(Fujitsu, 1) = 1.73/2.4 = 0.72$. As 72% is lower than 80%, this frequency is discarded so the algorithm check the next one in an increasing order. $u_{cpu_{eq}}(Fujitsu, 2) = 1.86/2.4 = 0.775$ is also lower so the next frequency is evaluated. For $f(Fujitsu, 3) = 2.13GHz$, we calculate $u_{cpu_{eq}}(Fujitsu, 3) = 2.13/2.4 = 0.887$ obtaining an equivalent CPU utilization of 88.7%, which is higher than the 80% required by the workload allocated in it. Thus, our algorithm sets the frequency of the Fujitsu server to 2.13GHz, as it is the minimum frequency that allows running the workload without performance degradation due to DVFS.

This policy allows servers to execute the workload in a more efficient way in terms of energy as frequency is scaled depending on the CPU requirements of the workload, while maintaining QoS.

9.3.2 Frequency-Aware Dynamic Consolidation

As an alternative to previous approaches, in this research we provide an energy-aware consolidation strategy that considers the frequency configuration according to the allocation of VMs. We use this approach to proactively optimize a Cloud under dynamic workload conditions.

Dynamic Consolidation Outlook

In this context, the dynamic consolidation problem can be split into four different phases, as proposed by Beloglazov et al. [162]. Each phase considers (i) detection of overloaded and (ii) underloaded hosts, (iii) selection of VMs to be migrated from these hosts, and (iv) VM placement after migrations respectively. Their research also present different algorithms for optimizing phases (i)-(iii) that we use during performance evaluation (see Subsection 9.4.4). So our work will be focused on finding new placements to host VMs after their migration from underloaded and overloaded hosts. In this work, we aim to optimize VM placement taking into account the frequency variations caused by the workload allocation together with the estimation of its impact in the overall consumption. This premise is incorporated in our policy, and evaluated lately in terms of energy efficiency and performance.

Algorithm Considerations

One of the main challenges when designing data center optimizations is to implement fast algorithms that can be evaluated for each workload batch during run-time. For this reason, the present research is focused on the design of an optimization algorithm that is simple in terms of computational requirements, in which both decision making and its implementation in a real infrastructure are fast. Instead of developing an algorithm for searching the optimal solution, we propose a sequential heuristic approach because it requires lower computational complexity. Our solution scales properly in accordance with large numbers of servers as explained in Subsection 9.3.2.

Minimizing the overall IT power consumption of the data center as a whole by only considering the consumption of each server separately may drive to some inefficiencies. The dynamic power of a host depends linearly on the CPU utilization, while the static remains constant (see Equation 9.1). So, when the reduction in consumption is performed individually, server by server, it results in the allocation of less workload on each physical machine, leading to the *underloaded server*-issue. This situation increases the number of active servers, which become underutilized, regardless the increase in the global static consumption. Otherwise, if the total energy consumed by the infrastructure is considered to be optimized, increasing the CPU utilization will reduce the number of servers required to execute the workload thus decreasing the overall static consumption but leading to an *overloaded server*-scenario. Therefore, both QoS and energy consumption could be affected as a consequence of VM migrations.

The proposed power and performance considerations, in Equations 9.1-9.5 and 9.6-9.12 respectively, provide a better understanding on how the system's behavior varies depending on frequency and utilization simultaneously. According to this, a more proactive allocation policy could be devised using DVFS to dynamically constrain aggressive consolidation scenarios to preserve QoS. To this purpose, the trade-offs between CPU utilization and frequency have to be analyzed in terms of energy. An increase in the resource demand of a host in terms of CPU utilization could represent an increment in its frequency depending on the available set of frequencies to maintain QoS. If frequency needs to be risen, the power consumption will be increased due to the frequency contribution (see Equation 9.8). So, we propose a VM placement policy that estimates the frequency increment during workload consolidation. Our strategy decides to allocate workload in those servers that have a higher utilization (but still have resources left to accommodate the incoming VM) and that impact less on the frequency contribution. Consequently, the policy uses more efficiently the ranges of utilization in which the frequency is not increased.

DVFS-Aware Dynamic Placement

The policy proposed in this research is not only aware of the utilization of the incoming workload to be assigned, but also is conscious of the impact of its allocation on servers working at different frequencies. DVFS-awareness allows to predict operating frequencies depending on VM allocation, thus helping to estimate future energy contributions. The presented approach takes advantage of this knowledge to optimize VM placement within the Cloud infrastructure under QoS and energy constraints.

Our algorithm is based on the bin packing problem, where servers are represented as bins with variable sizes due to the frequency scaling. To solve this NP-hard problem we use a Best Fit Decreasing (BFD)-based algorithm as BFDs are shown to use no more than $11/9 \cdot OPT + 1$ bins [163], being OPT the bins provided by the optimal solution. The bin packing approach under similar conditions has been proved to work well for this type of problems with large server sets of 800 hosts [162].

The allocation of a VM in a specific host provokes an increase in its CPU utilization and, according to our proposed *DVFS-perf configuration* algorithm, may increase or not its operating frequency. According to our previous considerations, a trade-off between servers' utilization and frequency may be inferred to reduce the energy consumption of dynamic workload scenarios. Typically, the frequency span in which a CPU ranges is of about 1GHz. So, the difference between a frequency of the set of valid frequencies and the next one is in the order of magnitude of 10^{-1}, being more common steps of about 0.1-0.5 GHz. On the other hand, average Cloud workload utilization ranges from 16%-59% [164]. As we define utilization of CPU as a value that ranges from 0 to 1, average Cloud workload utilization would be in the range 0.16-0.59. Thus utilization and frequency increments originated by the allocation of VMs have the same orders of magnitude. So, in order to maximize servers' utilization while minimizing frequency increment, we propose to maximize the difference between these two parameters as can be seen in Equation 9.13. We avoid the use of normalization, providing a light algorithm. We mean that our proposed algorithm is light because, compared with tests that we have conducted with metaheuristics as Simulated Annealing and Grammatical Evolution, we achieve simulation times that are about 160 times lower.

$$Placement_{host,vm} = u_{host,vm} - \Delta f_{host,vm} \tag{9.13}$$

$$u_{host,vm} = u_{host} + u_{vm} \tag{9.14}$$

$$\Delta f_{host,vm} = f_{host,vm} - f_{host} \tag{9.15}$$

$u_{host,vm}$ is the estimated CPU utilization resulting from adding both host and vm utilizations (u_{host} and u_{vm}). $\Delta f_{host,vm}$ provides the estimated difference between the host frequency after ($f_{host,vm}$) and before (f_{host}) the VM allocation calculated for the new estimated utilization. Algorithm 2 presents our DVFS-Aware Dynamic Placement proposal.

The input *vmList* represents the VMs that have to be migrated according to the stages (i), (ii) and (iii) of the consolidation process, defined in Subsection 9.3.2, while *hostsList* is the entire set of servers in the data center that are not considered overutilized. First, VMs are sorted in a decreasing order of their CPU requirements. Steps 3 and 4 initialize *bestPlacement* and *bestHost*, which are the best placement value for each iteration and the best host to allocate the VM respectively. Then, each VM in *vmList* will be allocated in a server that belongs to the list of hosts that are not overutilized (*hostList*) and have enough resources to host it.

Algorithm 2 Frequency-Aware Placement

```
Input: hostsList, vmList
Output: FreqAwarePlacement of VMs
 1: vmList.sortDecreasingUtilization()
 2: foreach vm in vmList do
 3:    bestPlacement $\leftarrow$ MIN
 4:    bestHost $\leftarrow$ NULL
 5:    foreach host in hostList do
 6:       if host has enough resources for vm then
 7:          utilization $\leftarrow$ estimateUtilization(host, vm)
 8:          frequencyIncrement $\leftarrow$ estimateFrequencyIncrement(host,vm)
 9:          placement $\leftarrow$ utilization - frequencyIncrement
10:          if placement $>$ bestPlacement then
11:             bestHost $\leftarrow$ host
12:             bestPlacement $\leftarrow$ placement
13:    if bestHost $\neq$ NULL then
14:       FreqAwarePlacement.add(vm, bestHost)
15: return FreqAwarePlacement
```

In steps 7 and 8, the algorithm calculates the value of the estimated CPU *utilization* ($u_{host,vm}$) and *freqIncrement* ($\Delta f_{host,vm}$) after vm allocation using Equations 9.14 and 9.15. According to our allocation strategy, derived from the above considerations, the *placement* value ($Placement_{host,vm}$) obtained when a VM is allocated in a specific host is calculated in step 9 using Equation 9.13.

As can be seen in steps 10, 11 and 12, the VM is allocated in the host that has a higher *placement* value, which means a high CPU utilization but, on the contrary, it represents a low increase in frequency due to the utilization increment. This approach minimizes the number of bins used by this combinatorial NP-hard problem while taking full advantage of the range of the equivalent CPU utilizations for each frequency. The output of this algorithm is the frequency-aware placement (*FreqAwarePlacement*) of the VMs that have to be mapped according to the under/overloaded detection and VM selection policies.

9.3.3 Frequency-Aware Optimization

Our Frequency-Aware Optimization combining the DVFS-perf policy with the Freq-Aware Placement algorithm is shown in listing of Algorithm 3. First, it finds the optimized placement of the VMs (*optimizedPlacement*) that have to be migrated due to dynamic workload variations. This is calculated in Algorithm 2, taking care of the frequency requirements. In step 2, the function *consolidateVM* allocates the VMs according to this mapping, performing VM migrations and updating utilization requirements for each host. Then in steps 3 and 4, the DVFS-perf configuration is obtained using Algorithm 1 with current utilization values. Finally the data center status is updated according to the optimized allocation and frequency configuration. Our DVFS-Aware strategy provides an elastic scale out that is adapted to the varying demand of resources. Also, the algorithm is light, making it suitable for quickly adaptation to workload fluctuations in the data center and run-time execution.

9.4 Simulation Configuration

In this section, we present the impact of our frequency-aware policies in energy consumption due to the improved management of the workload and the frequency assignment in servers. However, large-scale experiments and their evaluations are difficult to replicate in a real data center infrastructure because it is difficult to maintain the same experimental system

Algorithm 3 Frequency-Aware Optimization

```
Input: hostsList, vmList, frequenciesList
Output: optimizedConfiguration of the data center
optimizedPlacement ← frequencyAwarePlacement (hostList, vmList)
consolidateVM (optimizedPlacement)
optimizedFrequencies ← DVFS-perfConfiguration (hostList, frequenciesList)
setFrequencyConfiguration (optimizedFrequencies)
optimizedConfiguration ← configureDC (optimizedAllocation, optimizedFrequencies)

return optimizedConfiguration
```

conditions that are necessary for comparing different user and application scenarios. This can be achieved in simulation environment as simulators helps in setting up repeatable and controllable experiments.

For that reason, we have chosen the CloudSim toolkit [132] to simulate a Infrastructure as a Service (IaaS) Cloud computing environment. In contrast to other simulators, CloudSim provides the management of on-demand resource provisioning, representing accurately the models of virtualized data centers. The software version 2.0 that we have chosen supports the energy consumption accounting as well as the execution of service applications with workloads that vary along time [162]. For this work, we have provided frequency-awareness to the CloudSim simulator, also incorporating the ability to modify the frequency of servers. This frequency management policy allows us to evaluate the performance of the algorithms proposed in Sections 9.3.1 and 9.3.2. Our code also supports switching the VM placement policy to compare our strategy with other approaches.

Our simulations have been executed in a 64-bit Windows 7 OS running on an Intel Core i5-2400 3.10GHz Dell Workstation with four cores and 4 GB of RAM. Moreover, the simulations are configured according to the following considerations:

9.4.1 Workload

We conduct our experiments using real data from PlanetLab, which comprises more than a thousand servers located at 645 sites around the world. The workload consists of 5 days of data with different resource demand profiles obtained from the CoMon monitoring project [165]. The data traces are available and fully operative in CloudSim as this workload is commonly used by researchers using this simulator. By using these traces we can compare our approach with published and future research works.

The main features of each of the 5 sets, as the number of VMs and both the mean and standard deviation values of the CPU utilization, are shown in Table 9.1. Each of the five data sets includes CPU utilization values of around a thousand VMs with a monitoring interval of 300 seconds. We have chosen this collection because each independent workload can be executed for the same data center's initial size. Also, the usage of traces from a real system makes our simulation-based analysis applicable to real scenarios.

Table 9.1: PlanetLab workload main features

Date	VMs	CPU mean utilization	CPU utilization SD
2011.03.03	1052	12.31 %	17.09 %
2011.03.06	898	11.44 %	16.83 %
2011.03.09	1061	10.70 %	15.57 %
2011.04.12	1054	11.54 %	15.15 %
2011.04.20	1033	10.43 %	15.21 %

9.4.2 Physical Nodes

The simulation consists of a set of 400 hosts conforming a data center. This is the minimum amount of resources required by the CloudSim initial provisioning policy to manage the number of VMs for the different workloads that we have selected. During simulations, the number of servers will be significantly reduced as oversubscription is enabled. Hosts are modeled (as in Part II of this dissertation) as a Fujitsu RX300 S6 server based on an Intel Xeon E5620 Quad Core processor @2.4GHz, RAM memory of 16GB and storage of 1GB, running a 64bit CentOS 6.4 OS virtualized by the QEMU-KVM hypervisor.

DVFS Governors

The DVFS system of our Fujitsu server operates at 1.73, 1.86, 2.13, 2.26, 2.39 and 2.40 GHz respectively. For our experiments, we define two different governors to dynamically manage the CPU frequency. Both of them are fully available in our CloudSim modified version. For this work, we have provided frequency-awareness to the CloudSim simulator, also incorporating the ability to modify the frequency of servers according to our new DVFS-perf policy.

- **Performance**. The CPUfreq governor *performance*[1] is a typical governor available in the Linux Kernel. It sets the CPU to the highest frequency of the system.

- **DVFS-perf**. This governor dynamically modifies the CPU frequency according to Algorithm 1 so, it is set to the minimum frequency that ensures QoS depending on the workload.

Server Power Modeling

The power model used to estimate the energy consumed by these servers was proposed in our previous work in Chapter 6 (Equation 6.22) and can be seen in Equation 9.16. Then the energy consumption is obtained using Equation 9.17 where t determines the time in which the energy value is required. The operating frequencies set (in GHz) is provided in 9.18.

$$\begin{aligned} P_{\text{Fujitsu,k,w}} &= 3.32 \cdot V_{\text{DD}}^2(k) \cdot f_{op}(k) \cdot u_{\text{cpu}}(Fujitsu, k, w) + \\ &= 1.63 \cdot 10^{-3} \cdot T_{\text{mem}}^2 + 4.88 \cdot 10^{-11} \cdot FS^3 \qquad (9.16) \\ E_{\text{Fujitsu}} &= P_{\text{Fujitsu}} \cdot t \qquad (9.17) \\ f_{op}(k) &= \{1.73, 1.86, 2.13, 2.26, 2.39, 2.40\}(GHz) \qquad (9.18) \end{aligned}$$

This model presents a testing error of 4.87% when comparing power estimation to real measurements of the actual power. We used applications that can be commonly found in nowadays' Cloud data centers (including web search engines, and intensive applications) for training and testing stages. We assume a thermal management that allows memory temperature and fan speed to remain constant as we are interested in analyzing the power variations only due to utilization and DVFS management provided by our Freq-Aware optimization. The temperature of the memory T_{mem} and the fan speed FS are considered constant at 308 K and 5370 RPM respectively. Both parameters take their average values from the exhaustive experimental evaluation for this type of server that has been performed in our aforementioned previous work in Part II. This approach is valid since current models usually take into account only the variation of the dynamic consumption, as seen in Section 9.2. By including our power model in the CloudSim toolkit we are able to evaluate the power consumption in a more accurate way, as both the dynamic (depending on CPU utilization and frequency) and the static contributions are now considered. Thus, the impact of DVFS and consolidation-aware optimizations on the data center IT energy consumption is more likely to be measured by including our proposed models.

[1] www.kernel.org/doc/Documentation/cpu-freq/governors.txt

Active Server Set

In this work we assume that a server is switched off when it is idle, so no power is consumed when there is not any running workload. Also, servers are turned on when needed, if the system is overloaded. We take into account the booting energy consumption required by a server to be fully operative as seen in Equation 9.21.

$$\begin{aligned} P_{boot} &= 1.63 \cdot 10^{-3} \cdot 308^2 + 4.88 \cdot 10^{-11} \cdot 5370^3 = 162.1768W && (9.19) \\ t_{boot} &= 300s && (9.20) \\ E_{boot} &= P_{boot} \cdot t_{boot} = 13.514 \cdot 10^{-3} kW \cdot h && (9.21) \end{aligned}$$

where P_{boot} is the server booting power working at 308 K and 5370 RPM as defined above and t_{boot} is the booting time obtained experimentally.

9.4.3 Virtual Machines

VM types

The simulation uses heterogeneous VM instances that correspond to existing types of the Amazon EC2 Cloud provider. The Extra Large Instance (2000 MIPS, 1.7 GB RAM), the Small Instance (1000 MIPS, 1.7 GB RAM) and the Micro Instance (500 MIPS, 613 MB RAM) are available for all the scenarios. All the VM are forced to be single-core to meet the PlanetLab data set requirements.

Migration policy

In all our scenarios we allow online migration, where VMs follow a straightforward load migration policy already deployed on CloudSim 2.0. During migration, another VM, which has the same configuration as the one that is going to be migrated, is created in the target server. Then the cloudlets are migrated from the source VM to the target VM. Finally, when the migration is finished the source VM is removed. Live migration has two different overheads that affect to energy consumption and performance degradation. Therefore, it is crucial to minimize the number migrations in order to optimize energy efficiency while maintaining QoS.

Energy overhead. A migration takes a time known as *migration time* ($t_{migration}$), which is defined in Equation 9.22. The migration delay depends on the network bandwidth (*BW*) and the *RAM* memory used by the VM. We consider that only the half of the bandwidth is used for migration purposes, as the other half is for communication. Thus, migrations have an energy overhead because, during migration time, two identical VMs are running, consuming the same power in both servers.

$$t_{migration} = \frac{RAM}{BW/2} \tag{9.22}$$

Performance overhead. Performance degradation occurs when the workload demand in a host exceeds its resource capacity. In this work we model that oversubscription is enabled in all servers. So, if the VMs hosted in one physical machine simultaneously request their maximum CPU performance, the total CPU demand could exceed its available capacity. This situation may lead to performance degradation due to host overloading. The impact on SLA can be calculated as the SLA violation time per active host (SLA_{TAH}) that can be seen in Equation 9.23.

On the other hand, when overloading situations are detected, VMs are migrated to better placements, thus provoking performance degradation due to migration (PDM) as seen in Equation 9.24. The metric used in this work to determine the SLA violation ($SLA_{violation}$) [162] combines SLA_{TAH} and PDM as shown in Equation 9.26:

$$SLA_{TAH} = \frac{1}{M}\sum_{i=1} M \frac{t_{100\%_i}}{t_{active_i}} \tag{9.23}$$

$$PDM = \frac{1}{V}\sum_{j=1} V \frac{pdm_j}{C_{demand_j}} \tag{9.24}$$

$$pdm_j = 0.1 \cdot \int_{t_0}^{t_0+t_{migration}} u_j dt \tag{9.25}$$

$$SLA_{violation} = SLA_{TAH} \cdot PDM \tag{9.26}$$

where M is the number of servers; $t_{100\%_i}$ and t_{active_i} are the time in which the CPU utilization of the host i is 100% and the total time in which it is active respectively. V is the number of VMs and C_{demand_j} represents the CPU demand of the VM during its lifetime. pdm_j defines the performance degradation per VM during $t_{migration}$. In our experiments it is estimated as the 10% of the CPU utilization in MIPS during the migration time of VM j. Finally, t_0 is the time in which the migration starts and u_j is the CPU utilization of VM j.

9.4.4 Dynamic Consolidation Configuration

The present work aims to evaluate the performance of DVFS-aware dynamic consolidation. Consolidation phases (i), (ii) and (iii), defined in Subsection 9.3.2, are able to use the algorithms for the detection of overloaded or underloaded hosts and for the selection of VMs to be migrated that are available in CloudSim 2.0 [162]. We have simulated all the possible combinations for both types of algorithms with the default configuration of internal parameters, resulting in 15 different tests. The internal parameters for each option are set to those values that provide better performance according to Beloglazov et al [162]. Finally, consolidation phase (iv) is able to use two different power-aware placement algorithms.

Over/Underloading detection algorithms

We consider the detection of overloaded or underloaded hosts using five specific policies that belong to three different detection strategies.

- **Adaptive Utilization Threshold Methods**. Include the *Interquartile Range (IQR)* and the *Median Absolute Deviation (MAD)* algorithms, and offer an adaptive threshold based on the workload utilization to detect overloaded or underloaded hosts. The internal safety parameters take the value 1.5 and 2.5 respectively, and define how aggressively the consolidation is considered in this stage.

- **Regression Methods**. Both the *Local Regression (LR)* and the *Local Regression Robust (LRR)* are regression methods based on the Loess method and have the same internal parameter of 1.2.

- **Static Threshold Method**. The *Static threshold (THR)* sets a fixed value to consider when a host is overloaded or underloaded. The internal parameter is 0.8.

VM Selection Algorithms

The selection of the VMs that have to be migrated from overloaded or underloaded hosts is performed by three different algorithms.

- **Maximum correlation (MC)**. The system migrates the VM that presents a higher correlation of CPU utilization with other VMs so, the peak loads would occur at the same time.

- **Minimum migration time (MMT)**. The algorithm selects the VM that takes less time to be migrated when compared with the rest of VMs hosted in the same server.

- **Random choice (RS)**. The VM is randomly selected.

VM Placement Algorithms

- **Power Aware Best Fit Decreasing (PABFD)**. This placement policy for Cloud infrastructures takes into account the power consumption of the servers when finding an optimal placement under dynamic workload conditions [162]. It works well for SLA-constrained systems, maintaining QoS while reducing energy consumption. This state-of-the-art solution does not take into account frequency increments due to workload allocation and will serve us as a baseline consolidation policy.

- **Frequency-Aware Placement** (Freq-Aware Placement). This is the DVFS-aware placement policy that we propose in Algorithm 2. This solution allows a dynamic consolidation that is aware of both power and frequency also taking into account QoS.

9.4.5 Scenarios

We provide three different scenarios to evaluate the performance of our frequency-aware optimization. For this purpose, we will compare our work with two different approaches. All the proposed scenarios are able to power on/off servers when needed as can be seen in section 9.4.2

- The **Baseline** scenario represents the default performance of CloudSim. The performance governor is active so, the servers always operate at the maximum frequency. PABFD placement is used to perform VM allocation.

- The **DVFS-only** scenario uses our DVFS-perf governor combined with PABFD placement. Thus, the frequency of each server is reduced to the lowest value that allows the system to meet QoS. However, the mapping is not aware of the allocation impact on CPU frequency that also impacts on the power consumption.

- The **Freq-Aware Optimization** scenario combines our DVFS-perf governor with our Freq-Aware Placement as shown in Algorithm 3. Both utilization and frequency estimations are considered to find the optimized allocation. It aims to evaluate our proposed optimization strategy.

9.5 Experimental Results

We have simulated the 3 different scenarios for each of the 5 different PlanetLab workloads presented in Table 9.1, and tested the 15 different combinations of the algorithms for overloading detection and VM selection aforementioned. Therefore, for each of the daily workloads, we are able to present the following results per test (under/overload detection-VM selection) and per scenario, in order to compare our Freq-Aware optimization with the other two baseline alternatives.

9.5.1 Performance Analysis

We consider the following metrics to analyze the obtained results. The number of VM migrations is considered as a figure of merit because migrations may cause SLA violations due to performance degradation, also impacting on energy consumption. Additionally we have included the overall SLA violations provided by the metric $SLA_{violation}$ to simultaneously verify if our policies meet QoS requirements. As CloudSim allows turning machines on when needed, we have included the additional booting energy consumption of the servers to the simulation. The number of *Power on events* is our proposed metric to evaluate its impact because, reducing the number of these events would decrease the overall data center energy. Service outages are experienced when the power density exceeds the maximum capacity of the grid. So, we evaluate the peak power during the simulation in order to analyze the system's performance under critic situations in terms of electricity supply. Finally, the IT energy signature is obtained in order to evaluate the efficiency introduced by our strategy.

Table 9.2 shows the average values of these metrics when comparing the baseline with the DVFS-only policy and with our Freq-Aware optimization. For each PlanetLab workload (represented as the date when it was obtained) the table shows the averaged values that result from their execution under every possible combination of the overloading detection and the VM selection algorithms. An average of 3.35% energy savings is achieved just including the DVFS capabilities to the simulation infrastructure for all the workloads. The savings in energy consumption come from the combined reduction of the VM migrations and the Power on events. In this scenario, QoS is maintained but the peak power is not improved when compared with the baseline.

Table 9.2: Average values per day for baseline comparison

Optim. Policy	Date (yy.mm.dd)	VM migrations reduction	Power on events reduction	SLA violations reduction	Peak power reduction	Energy savings
DVFS-only	2011.03.03	4.40 %	13.63 %	0 %	-6.08 %	4.64 %
	2011.03.06	4.81 %	9.60 %	0.01 %	-2.87 %	3.45 %
	2011.03.09	3.63 %	5.16 %	0 %	-7.27 %	3.44 %
	2011.04.12	1.44 %	1.49 %	0 %	0.1 %	2.36 %
	2011.04.20	1.82 %	-3.72 %	0.01 %	5.81 %	2.59 %
Freq-Aware	2011.03.03	23.44 %	86.10 %	0 %	68.16 %	34.82 %
	2011.03.06	19.38 %	79.16 %	0.01 %	64.29 %	34.64 %
	2011.03.09	19.53 %	85.41 %	0 %	64.34 %	39.14 %
	2011.04.12	26.77 %	88.03 %	0.01 %	66.19 %	38.88 %
	2011.04.20	19.55 %	85.81 %	0 %	69.09 %	41.62 %

The proposed Frequency-Aware Placement combined with the DVFS management significantly reduces both the number of power on events and VM migrations. The minimization of the times that a server is powered on has several benefits, not only reducing the energy consumption but also extending its lifetime. However, its impact on the total energy consumption represents only about 5.31%. So, the energy savings are obtained mainly due to the reduction of the VM migrations as, during each migration, an identical VM is simultaneously running in the source and in the target hosts. Our proposed Freq-Aware optimization policy outperforms the baseline obtaining average energy savings of 37.86% significantly reducing peak power consumption around 66.14% while maintaining the QoS, as can be seen in the peak power reduction column and in the SLA violations reduction column respectively.

The different tests, each of them representing a specific combination of overloading detection and VM selection algorithms, perform differently. However, the performance pattern for each test is repeated for every considered PlanetLab workload in Table 9.1. Thus, we are able to analyze the system's performance for every test, as can be seen in Figure 9.1, which presents the averaged values of each metric for all the workloads. As shown in 9.1.e, both policies achieve energy savings for each test but the Freq-Aware optimization reduces the data center energy consumption to an average value of 69.16 kWh for all the workloads regardless the combination of algorithms. This means an average savings of 37.86%. In 9.1.d we obtain a similar pattern in the overall peak power of the IT infrastructure, achieving a reduction of about 66.14%.

The same occurs in 9.1.c for the number of power on events that is reduced to about 76.53 events, showing average savings of 86.03%. However, not every test performs the same in terms of SLA violations. Overall SLA violation for local regression methods combined with MC and MMT algorithms present better values of about 0.05% as can be seen in 9.1.b. Also in 9.1.a, average VM migrations vary considerably from one test to another. So, the SLA violations and VM migrations metrics may be determining factors when selecting a combination of overloading detection and VM selection algorithms.

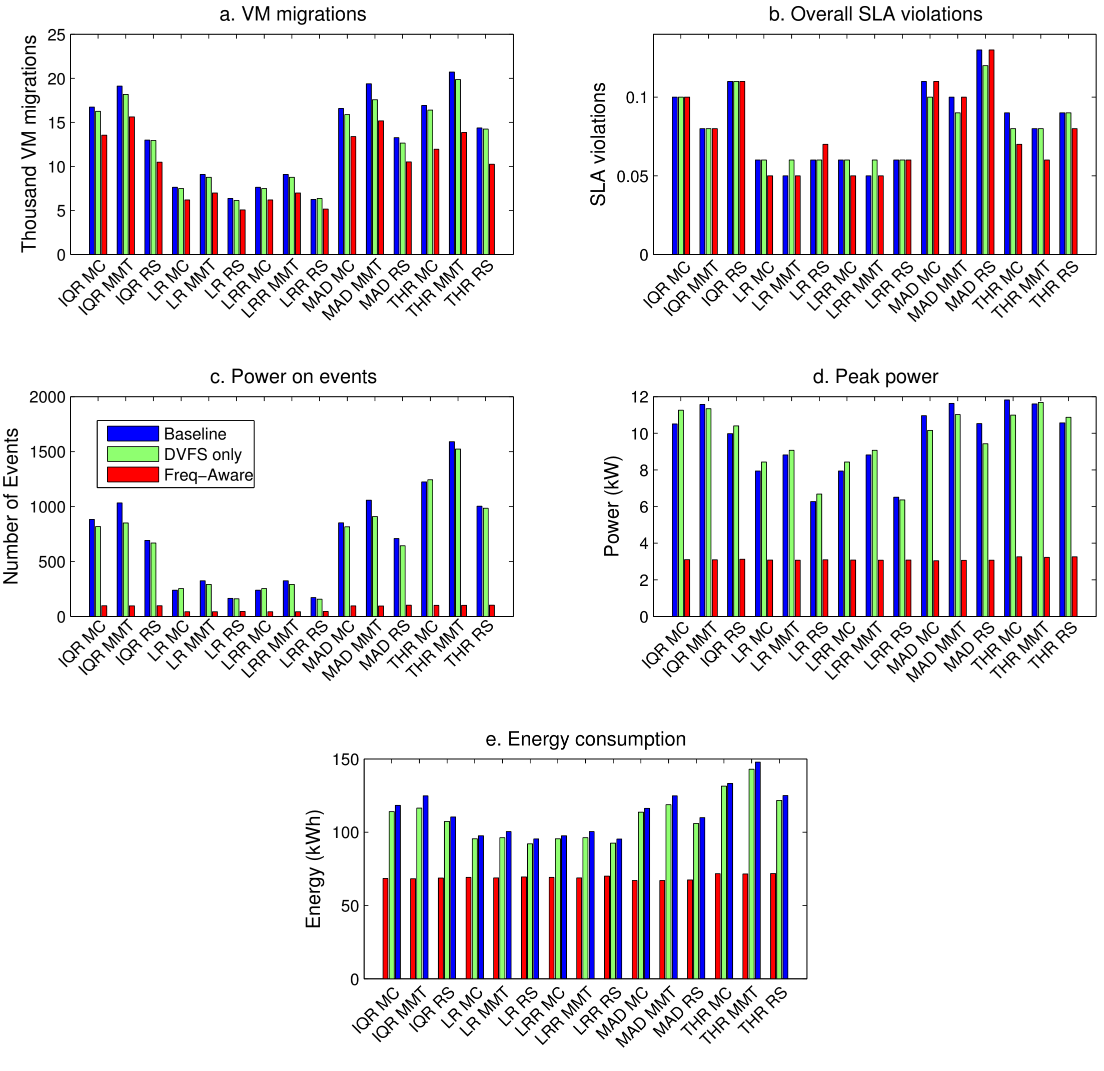


Figure 9.1: Average metrics per test

9.5.2 Run-time Evaluation

Moreover, to deeply understand the performance of the Freq-Aware optimization during run-time, we have selected one of the workloads for its simulated execution under the conditions of a specific test. Figure 9.2 presents the temporal evolution of the test that combines the MAD and MMT algorithms as it achieved the lowest total energy consumption. The test runs the 1052 VMs of the workload dated on 2011.03.03 because it achieves the highest CPU utilization and standard deviation (see Table 9.1) thus presenting the most variable working conditions.

In this framework, we evaluate additional metrics to compare both baseline and Freq-Aware scenarios. Figure 9.2.a shows the global resource demand of this workload in terms of MIPS. The global utilization represents the average CPU utilization of all the servers in the data center. The number of active hosts within the total facility is also analyzed because, as this value increases, the global utilization will be reduced. Finally the cumulative energy consumption of the IT infrastructure is presented to study its deviation between both scenarios during a 24 hours-workload.

For the baseline policy, the number of active hosts is highly increased during peaks of workload demand, consequently reducing the data center global utilization, as can be seen in Figures 9.2.b and 9.2.c respectively. The decrease on the overall utilization also reduces

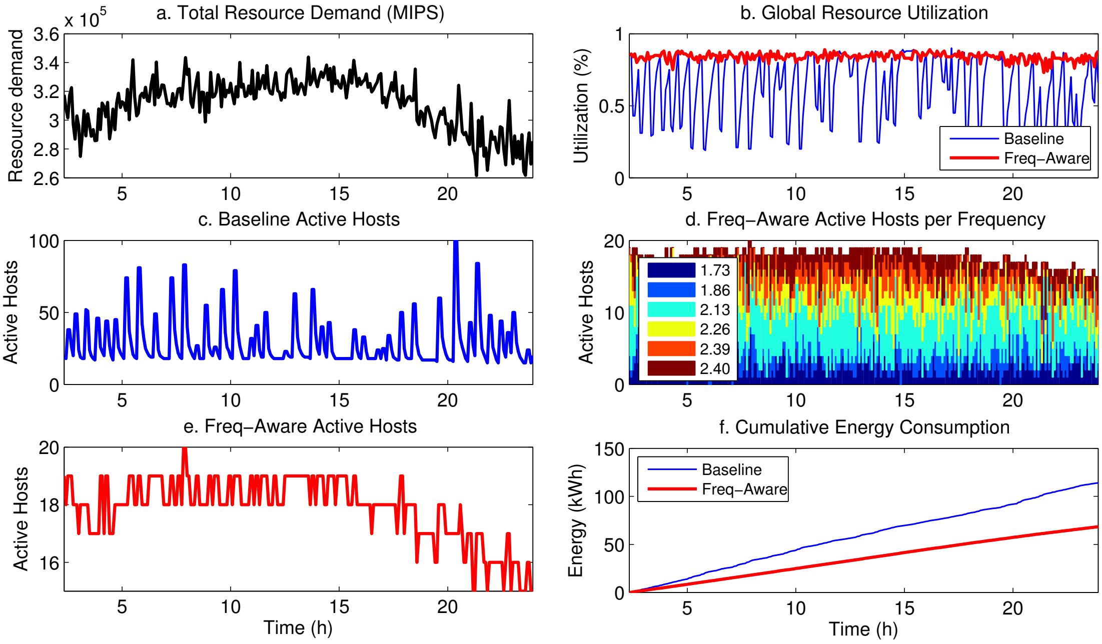


Figure 9.2: Temporal evolution for MAD-MMT test running workload 2011.03.03

each server energy consumption, as its power depends linearly on CPU demand. However, the static consumption (which accounts for about 70% of total consumption in each physical machine) due to the additional servers that are required to execute the workload with this utilization, highly increases the total energy budget. On the other hand, for the Freq-Aware optimization policy, both values remain more constant, as shown in Figures 9.2.b and 9.2.e respectively.

The DVFS configuration of the active server set during run-time can be seen in Figure 9.2.d. The DVFS mode operating at 2.13GHz is the most selected, as it offers a wider range of utilizations in which the frequency remains constant. This frequency allows a sufficiently high utilization (from 77.5% to 88.75%) that helps to minimize the number of servers. The rest of DVFS modes are also used but mainly to absorb load peaks as dynamic workload fluctuates during run-time.

Our algorithm, when compared with the baseline, speeds up both the consolidation into a lower number of active servers and the elastic scale out of the IT infrastructure, increasing the global utilization in a 23.46% while reducing the number of active hosts around a 44.91%. Table 9.3 presents the averaged values for these results. Figure 9.2.f shows how this behavior impacts on the energy usage of the data center where the baseline consumption grows at a higher rate during dynamic workload variations than for the optimized scenario, achieving total energy savings of 45.76%.

Table 9.3: Average results for MAD-MMT test running workload 2011.03.03.

Scenario	Global Utilization	Active Hosts	Total Energy
Baseline	60 %	35.49	125.45 kWh
Freq-Aware	83 %	19.55	76.72 kWh

9.6 Summary

The work presented in this chapter makes relevant contributions on the optimization of Cloud data centers from a proactive perspective. In this work we present the Freq-Aware optimization that combines a novel reactive DVFS policy with our proactive Frequency-aware Consolidation technique. We have achieved competitive energy savings of up to 45.76%, when compared with the PABFD baseline, for the IT infrastructure while maintaining QoS, even improving slightly the SLA violations around 0.01%, for real workload traces in a realistic Cloud scenario. According to our results, our algorithm enhances the consolidation process and speeds up the elastic scale out, reducing the global peak power demand about a 66.14% while improving the energy efficiency by increasing global server utilization to 83% in average.

The following chapter extends this research to provide not only DVFS awareness but also thermal awareness to the dynamic management of the data center. So, novel dynamic and proactive consolidation strategies are provided to improve energy savings in both IT and cooling infrastructures from a holistic perspective.

10. Power and Thermal Aware VM Allocation Strategies for Energy-Efficient Clouds

> *"Whatever it is you seek, you have to put in the time, the practice, the effort. You must give up a lot to get it. It has to be very important to you. And once you have attained it, it is your power."*
>
> — Michael Crichton, *Jurassic Park*

The cooling needed to keep the servers within reliable thermal operating conditions has a significant impact on the thermal distribution of the data room, thus affecting servers' power leakage. The energy efficiency of novel cooling technologies, as water-based cooling, heat reusing and free cooling approaches outperform traditional CRAC units. However, the implantation rate of these new techniques is still low for typical data centers. Therefore, optimizing the energy consumption of both IT and cooling infrastructures is a major challenge to place data centers on a more scalable scenario. Also, many Cloud applications expect services to be delivered as per SLA, so power consumption in data centers may be minimized without violating these requirements whenever it is feasible. Thus, understanding the relationships between power, temperature, consolidation and performance is crucial to enable energy-efficient management at the data center level.

In this chapter, we propose novel power and thermal-aware strategies to model global optimizations from a local perspective based on the global energy consumption of metaheuristic-based optimizations. For this purpose, thermal models, which accurately describe the behavior of the CPU and the memory devices, are provided. They have been tested using a real infrastructure running real Cloud applications resulting in an average temperature estimation error of 0.85% and 0.5049% respectively. Our results show that the combined awareness from both metaheuristic and BFD algorithms allow us to infer models that describe the global energy behavior into faster and lighter global optimization strategies that may be used during runtime. This approach allows us to improve the energy efficiency of the data center, considering both IT and cooling infrastructures, in a 21.74% while maintaining QoS.

10.1 Introduction

As the connectivity in personal and working environments is gaining importance, an increasing number of services with diverse application-level requirements are offered over the Internet [84]. The integration of application-level strategies together with server consolidation is a major challenge to maximize energy savings [85]. The resource demand required by the VMs is highly variable and is not always known a priori, as it depends mostly on users' activity. So, it is recommendable to have additional knowledge about application-level performance, not only regarding CPU but also in terms of memory and disk among others, to optimize the resource adaptation to the studied variations in the application requirements. This issue is particularly accentuated when servers are overloaded and applications cannot access enough resources to operate efficiently. Therefore it is a great recommendation to use consolidation algorithms that dynamically reallocate VMs on different physical servers throughout the data center in order to optimize resource utilization during runtime.

One of the main challenges within energy-efficient Clouds consist in reaching a compromise between the QoS in terms of SLA and energy consumption so that the performance is not degraded. To achieve this goal, the different components of the system should be analyzed, as well as the interaction between them when they operate as a whole. This work is intended to offer novel optimization strategies that take into account the contributions to power of non-traditional parameters such as temperature and frequency among others. Our research is based on fast and accurate models that are aware of the relationships with power of these parameters, allowing us to combine both energy and thermal-aware strategies. The new holistic paradigm proposed in this Ph.D. thesis, focuses for the first time in literature on considering the energy globally. Hence, all the data center elements are aware of the evolution of the global energy demand and the thermal behavior of the data room. So, our decisions are based on information from all available subsystems to perform different energy and performance optimizations.

Our work makes the following **key contributions**: 1) a set of single and multi-objective BFD-based policies that optimize the energy consumption of the data center considering both IT and cooling parameters; 2) a novel strategy to infer a global optimization from a local perspective based on modeling the global energy consumption of metaheuristic-based optimizations; 3) two thermal models that accurately describe the behavior of the CPU and the memory devices; and 4) a cooling strategy based on the estimated temperature of devices due to VM allocation.

The remainder of this chapter is organized as follows: Section 10.2 gives further information on the related work on this topic. Our proposed algorithms and models are presented in Section 10.3, Section 10.4 and Section 10.5 respectively. Section 10.6 describes profusely the experimental results. Finally, in Section 10.7 the main conclusions are drawn.

10.2 Related Work

Due to the impact of energy-efficient optimizations in an environment that handles so impressive high figures as data centers, many researchers have been motivated to focus their academic work on obtaining solutions for this issue. In this section, we present different approaches of the state-of-the-art from both server and data center perspectives that are aware of thermal and power contributions.

10.2.1 Server Efficiency

Joint thermal and power-aware strategies can be found within the server scope considering fan control together with scheduling in a multi-objective optimization approach [102]. Chan et al. [103] propose a technique that combines both energy and thermal management policies to reduce the server cooling and memory energy costs. They provide a model to estimate temperature that uses electrical analogies to represent the thermal and cooling behavior of components. However, their work does not split the contributions of leakage and cooling power, so their minimization strategy is unaware of the leakage-cooling trade-offs.

10.2.2 Data Center Efficiency

By virtualizing a data center, savings in the electricity bill of around 17% can be achieved. However, by combining improvements in power of both computation and cooling devices, savings have the potential to reach about 54% [83]. This is the main challenge to reduce data center energy from a global perspective.

On its own, virtualization has the potential of minimizing the hot-spot issue by migrating VMs. Migration policies allow to distribute the workload during run-time without stopping task execution. Some Cloud computing solutions, such as those introduced in the work presented by Li et al. [104], have taken into account the dependence of power consumption with temperature, due to fan speed and the induced leakage current.

Abbasi et al. [105] propose heuristic algorithms to address this problem. Their work presents the data center as a distributed CPS in which both computational and physical parameters can be measured with the goal of minimizing energy consumption. However, the validation of these works is kept in the simulation space, and solutions are not applied in a real data center scenario.

The current work in the area of joint energy and thermal aware strategies is not addressing the issue of proactive resource management with the goal of total energy reduction. Instead, techniques so far either rely on the data room thermal modeling provided by CFD software, or just focus on measuring inlet temperature of servers. However, the models at the data room level do not monitor the CPU temperature of servers nor adjusting the infrastructure status proactively or performing a joint workload and cooling management during run-time for arbitrary workloads.

10.3 VM Allocation Strategies Description

The different policies presented in this section are based on the knowledge achieved from our power model in Equation 9.3 that considers non-traditional contributions as frequency and temperature of the server physical resources. In this section we provide a taxonomy of candidate optimization algorithms that take into account IT and cooling power of the data center infrastructure under energy and thermal considerations. The mathematical description of these objectives will allow the later optimization by the development of an optimization algorithm.

10.3.1 Single-Objective BFD-based Allocation Policies

These Single-Objective (SO) policies optimize the consolidation of a set of VMs ($vmList$) using the BFD approach. First, VMs are sorted in decreasing order of CPU utilization and then, they are allocated on the set of available hosts ($hostList$) according to the minimization of an optimization objective ($SOvalue$) as can be seen in Algorithm 4. *bestPlacement* and *bestHost* are the best placement value for each iteration and the best host to allocate the VM respectively.

Algorithm 4 SO Placement Policy

```
Input: hostList, vmList
Output: SOPlacement of VMs
vmList.sortDecreasingUtilization()
foreach vm in vmList do
    bestPlacement ← MAX
    bestHost ← NULL
    foreach host in hostList do
        if host has enough resources for vm then
            placement ← SOvalue
            if placement < bestPlacement then
                bestHost ← host
                bestPlacement ← placement
    if bestHost ≠ NULL then
        SOPlacement.add(vm, bestHost)
return SOPlacement
```

Then, each VM in *vmList* will be allocated in a server that belongs to the list of hosts that are not overutilized (*hostList*) and have enough resources to host it. The VM is allocated in the host that has a lower *placement* value. The output of this algorithm is the placement (*SOPlacement*) of the VMs that have to be mapped according to the MAD-MMT detection and VM selection policy. In this subsection we present the different SO objectives proposed in this research.

$SO_1 : min\{\Delta P_{Host}\}$

This policy minimizes the increment of power consumption when a VM is allocated in a host. The algorithm consolidates the VM in the server whose power consumption has the lowest increase in terms of power. The increment is calculated as the difference between the power after and before the allocation of the incoming VM. This approach is used as a baseline, as it has been proposed by Beloglazov et al. [162] and it is included in the open source version of CloudSim 2.0 as PABFD, so it could be easily compared with other state-of-the-art research.

$SO_2 : min\{P_{Host}\}$

The consolidation value is the host's power consumption. The algorithm chooses the server that presents the lowest power consumption when hosting the incoming VM. The main objective of this policy is to show the relevance of understanding the different contributions to power in a data center. The fact is that the global power consumption can not be reduced by minimizing local power, as the static contribution increases globally with the number of active hosts. However, this consolidation technique may be useful in some scenarios in which power capping is a necessity, enforcing a drastic reduction of server power.

$SO_3 : min\{1/(u_{cpu} - \Delta freq)\}$

The consolidation value calculated by this approach has been proposed by the authors in Chapter 9. Our proposed policy is not only aware of the utilization of the incoming VM, but also considers the impact of its allocation in terms of frequency. This approach is interesting from the point of view of combining both static and dynamic contributions to global power consumption from a local perspective. The higher the CPU utilization allowed in servers, the lower the number of active hosts required to execute the incoming workload, thus reducing global static consumption. However, host's dynamic consumption increases with frequency. As frequency increases with CPU utilization demand, we propose a compromise between increasing u_{cpu} after the allocation, while reducing the frequency increment due to the incoming VM. This equation may be devised as both the u_{cpu} and the frequency increment range in the same orders of magnitude.

$SO_4 : min\{T_{mem}\}$

This consolidation approach aims to minimize the temperature of the memory as it has been demonstrated to be a key contribution to static power consumption. Moreover, this parameter also depends on the inlet temperature of the server, which impacts on the cooling power of the data center infrastructure and on the dynamic memory activity. Cooling down the computing infrastructure is needed to avoid failures on servers due to temperature, or even the destruction of components as in the case of thermal cycling and electromigration among others. Therefore, this policy may be helpful for extremely hot conditions in the outside, and also when undergoing cooling failures.

$SO_5 : min\{\Delta freq\}$

This algorithm consolidates the VM in the server whose frequency has a lowest increase in terms of frequency. The policy aims to minimize the increment of power consumption when a VM is allocated in a host. The increment is calculated as the difference between the frequency before and after the allocation of the VM. This consolidation technique aims to minimize the dynamic contribution to power consumption.

$SO_6 : min\{1/u_{cpu}\}$

The consolidation value proposed in this approach maximizes the overall CPU utilization in the active host set, also constraining the number of active servers. In all our proposed approaches, the maximum load that can be allocated in each host is bounded by its available resources. Also, the VMs are migrated from overloaded servers according to previous workload variations. However, the possibilities of violating the SLA are high when using this specific technique. This is because workload variations in a highly loaded server may exceed the total resource capacity of the device, degrading the performance of the applications. This policy may be specially useful in scenarios with low penalties per SLA violations or if performance degradation does not have a great impact on economic or contractual issues.

$SO_7 : min\{P_{Host} + P_{Cooling}\}$

The consolidation value is the aggregation of the host's power consumption and the power dimensioned to cool it down avoiding thermal issues. The algorithm chooses the server that presents the lower power IT and cooling consumption when hosting the incoming VM. The main objective of this policy is to show the relevance of understanding the thermal contributions to power in a data center. The overall power consumption can not be reduced by minimizing local power, as the static IT contribution increases globally with the number of active hosts and the cooling power depends on IT consumption as well as on their inlet temperature needed to keep them safe. However, this consolidation technique may be useful in some scenarios in which power capping is a necessity in both IT and cooling infrastructures and when combined with variable cooling techniques.

$SO_8 : min\{\sum |SO_X|\}$

This approach aims to minimize the total power consumption of the data center by combining the different parameters presented in the SO section in a single metric. In order to make the values comparable we normalize them in the range [1,2] (according to their maximum and minimum in each range) so all the inputs take values in the same orders of magnitude. It is worthwhile to mention that we have performed a variable standardization for every feature in order to ensure the same probability of appearance for all the variables. The algorithm consolidates the VM in the host that minimizes the summation of all the normalized parameters as seen in Equation 10.1. After an exhaustive analysis, the best global power results are shown for the parameter combination presented in Equation 10.2.

$$\begin{aligned} SO_8 = min\{&|\Delta P_{Host}| + |P_{Host}| + |1/(u_{cpu} - \Delta freq)| + \\ &+|T_{mem}| + |\Delta freq| + |1/u_{cpu}| + |P_{Host} + P_{Cooling}|\} \end{aligned} \tag{10.1}$$

$$SO_8 = min\{|1/(u_{cpu} - \Delta freq)| + |\Delta freq| + |1/u_{cpu}|\} \tag{10.2}$$

10.3.2 Multi-Objective BFD-based Allocation Policies

The approaches that we present in this section also aim to minimize global power consumption from a local perspective. However, these policies do not consider one single objective to be optimized, but more than one. Multi-Objective (MO) optimizations try to simultaneously optimize several contradictory objectives. This is also useful due to the fact that, in some cases, the parameters cannot be linearly combined as their units are not comparable without normalization (e.g. their orders of magnitude are very different).

In all our MO policies, the VMs from $VMlist$ are also allocated one by one in the host that minimizes the consolidation value according to the given policy. However, MO techniques offer a multidimensional space of solutions instead of returning a single value. For this kind of problems, single optimal solution does not exist, and some trade-offs need to be considered. The number of dimensions is equal to the number of objectives of the problem. Each objective of our MO strategy consists of each one of the SO consolidation values presented in the previous subsection. Hence, to find the solution for the allocation of a VM in a specific $Host_i$ from $hostList$, we calculate every consolidation value of the SO policies ($SOvalues$) as can be seen in Algorithm 5. Thus, each solution of the algorithm has the following objective vector ($hostVector$):

$$\begin{aligned} solution_{Host_i} = \{ & \Delta P_{Host_i}, P_{Host_i}, 1/(u_{cpu} - \Delta freq)_i, \\ & T_{mem_i}, \Delta freq_i, 1/u_{cpu_i}, P_{Host_i} + P_{Cooling_i} \} \end{aligned} \tag{10.3}$$

Then, we constrain the set of solutions to provide the ones that are in the Pareto-optimal Front ($paretoOptimal$). This optimal subset provides only those solutions that are non-dominated by others in the entire feasible search space. This approach discards solutions that may be the optimum for a SO policy, but are dominated by other solutions that appear in MO problems. *bestPlacement* and *bestHost* are the best placement value for each iteration and the best host to allocate the VM respectively. Then, each VM in *vmList* will be allocated in a server that belongs to the list of hosts whose $hostVector$ are non-dominated and have enough resources to host it. The VM is allocated in the host that has a lower *placement* value. The output of this algorithm is the placement (*MOPlacement*) of the VMs that have to be mapped according to the MAD-MMT detection and VM selection policy. In this section, we present two MO metrics to decide a solution from the Pareto-optimal set of solutions.

Algorithm 5 MO Placement Policy

```
Input: hostList, vmList
Output: MOPlacement of VMs
vmList.sortDecreasingUtilization()
foreach vm in vmList do
    bestPlacement ← MAX
    bestHost ← NULL
    foreach host in hostList do
        if host has enough resources for vm then
            hostVector ← SOvalues
            hostVectorSolutions.add(host, hostVector)
    paretoOptimal ← hostVectorSolutions.getNonDominatedHostVectors()
    foreach host in paretoOptimal do
        placement ← MOvalue
        if placement < bestPlacement then
            bestHost ← host
            bestPlacement ← placement
    if bestHost ≠ NULL then
        MOPlacement.add(vm, bestHost)
return MOPlacement
```

$MO_1 : min\{\sum(P_{host} + P_{Cooling})\}$

To allocate each VM, we consider the solution from the Pareto-optimal set that provides the lowest global IT and cooling power. Thus, for every solution in POF, the algorithm calculates the MO_1 consolidation value as the power consumed by the data center considering the VM placement. Then, the VM is allocated in the host that minimizes this consolidation value.

$MO_2 : min\{|d(solution_{Host_i}, o)|\}$

For each solution in POF, the algorithm calculates the Euclidean distance from the objective vector $solution_{Host_i}$ to the origin. Finally, the VM is allocated in the host that minimizes this distance.

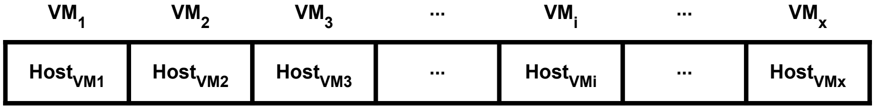
10.3.3 Metaheuristic-based Allocation Policies

In order to compare our algorithms with non-local policies we consider a different approach. Local search methods usually fall in suboptimal regions where many solutions are equally fit. The methods proposed in these sections intend to help the solutions to get out from local minimums finding more optimal solutions. Metaheuristics aim to optimize the global power consumption by simultaneously allocating all the VMs in the available hosts set, instead of in a sequential way as in our BFD-based algorithms. Thus, these algorithms do not only consider local power in each server but the entire IT and cooling data center consumption during the allocation.

$SimulatedAnnealing : min\{\sum(P_{Host} + P_{Cooling})\}$

Simulated Annealing (SA) is a metaheuristic based on the physical annealing procedure used in metallurgy to reduce manufacturing defects. The material is heated and then it is cooled down slowly in a controlled way, so the size of its crystals increases and, in consequence, this minimizes the energy of the system. SA is used for solving problems in a large search space, both unconstrained and bound-constrained, approximating the global optimum of a given function. This algorithm performs well for problems in which an acceptable local optimum is a satisfactory solution, and it is often used when the search space is discrete.

Our SA proposed in this research evaluates the power consumption of the data center after the consolidation of the VM set along the infrastructure. We provide the solution structure in Figure 10.1, where each (VM_i) is hosted in $Host_{VM_i}$. The size of the solution is the size of the list of VMs that have to be allocated in the system. The allocation is performed for the solution that minimizes the global contribution to data center power. Also, in the calculation of the SA objective value ($SAvalue$) we have included power penalties in the solution evaluation for those servers that are overutilized after the consolidation process in terms of CPU utilization, RAM memory or I/O Bandwidth.



VM_1	VM_2	VM_3	...	VM_i	...	VM_x
$Host_{VM1}$	$Host_{VM2}$	$Host_{VM3}$	...	$Host_{VMi}$	...	$Host_{VMx}$

Figure 10.1: Solution scheme for the SA algorithm.

Apart from this, we have included an optional optimization of the SA, where the first proposed solution is initialized to the best solution found by the SO set of algorithms. This initialization ensures that the SA finds a feasible solution that is, at least, as good as the one provided by the best SO in each optimization slot.

As can be seen in Algorithm 6 a set of candidate solutions (*solutionList*) are provided by the SA according to the VMs in $vmList$ that have to be placed simultaneously within the $hostList$ set. *bestPlacement* and *bestSolution* are the best placement value when allocating the entire $vmList$ set and the best solution that provides this placement respectively for the minimum $SAvalue$. Then, each VM in *vmList* will be allocated in the server provided by the best solution.

10.3.4 Metaheuristic-based SO Allocation Policies

In this subsection we present a novel strategy to derive global optimizations from a local perspective based on modeling the global energy consumption of metaheuristic-based optimizations. This approach is used to model a new SO policy that combines the different SO consolidation metrics previously presented in order to find a local policy that outperforms their single outcomes. By using this modeling technique, we aim to find a local, fast and light

Algorithm 6 Metaheuristic Placement Policy

```
Input: hostList, vmList
Output: MetaheuristicPlacement of VMs
solutionList ← getSolutionList(vmList, hostList)
bestPlacement ← MAX
bestSolution ← NULL
foreach solution in solutionList do
    placement ← SAvalue
    if placement < bestPlacement then
        bestSolution ← solution
foreach vm in bestSolution do
    bestHost ← vm.getHost()
    MetaheuristicPlacement.add(vm, bestHost)
return MetaheuristicPlacement
```

consolidation algorithm that is aware of the relationships between the contributions to energy, not only during the allocation, but also taking into account further VM migrations.

Using the consolidation values and the energy values obtained during the simulation of the SO experiments we model a function that describes the behavior of the energy consumption of the values obtained for the SA metaheuristic as in Equation 10.4. Then, we use this function to provide a SO_{SA} local consolidation value, which will be used to optimize the system as done before for the regular SO policies. So, we use SO_{SA} as the other SO policies, allocating the VMs of the set, one by one, in the host that offers a lowest consolidation value (as detailed in Algorithm 4). Further details regarding the implementation of this strategy are provided in Section 10.4.

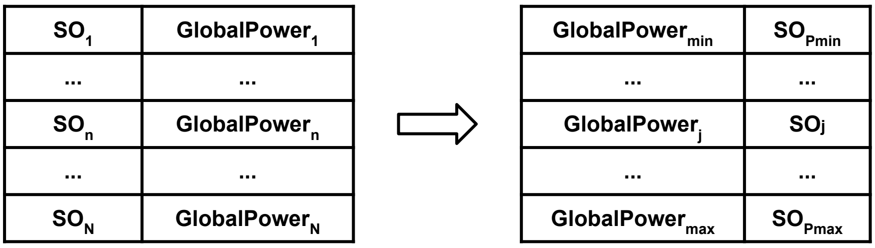
$$SO_{SA} = f(SO_1, E_{SO_1}, SO_2, E_{SO_2}, ..., SO_8, E_{SO_8}) \tag{10.4}$$

10.3.5 BFD-based SO Dynamic selection Allocation Policy

The approach that we present in this section also aims to minimize global power consumption from a local perspective. However, this policy does not consider only one consolidation value, but the complete set of values offered by all the SO approaches in this research. The SO Dynamic Selection approach (*DynSO*) allocates the set of VMs using the SO policy that minimizes the overall IT power in each time slot.

$$SO \in SO_1, ..., SO_n, ..., SO_N \tag{10.5}$$

Algorithm 7 presents the implementation for this allocation policy. First, the algorithm evaluates the final global power consumption ($GlobalPower_n$) provided by the allocation of the entire set of VMs when using the consolidation value of SO_n. Then, the same calculations are done for the rest of SO_{n+1} until all the N-dimensioned set of SOs is covered obtaining the table $PowerSO$ shown in Figure 10.2.



SO_1	$GlobalPower_1$
...	...
SO_n	$GlobalPower_n$
...	...
SO_N	$GlobalPower_N$

$GlobalPower_{min}$	SO_{Pmin}
...	...
$GlobalPower_j$	SOj
...	...
$GlobalPower_{max}$	SO_{Pmax}

Figure 10.2: Dynamic selection of the best SO policy.

Finally, all the VMs of the set are allocated one by one in the host that presents the lowest consolidation value according to the SO policy that offers a lowest global power consumption after the allocation of all the VM set (SO_{Pmin}). The calculation of the $bestSO$ that minimizes the power consumption of the allocation of the VMs' set is performed for every time slot in the system.

Algorithm 7 Dynamic SO Placement Policy

```
Input: hostList, vmList, SOList
Output: DynSOPlacement of VMs
vmList.sortDecreasingUtilization()
foreach SO in SOList do
    foreach vm in vmList do
        bestPlacement ← MAX
        bestHost ← NULL
        foreach host in hostList do
            if host has enough resources for vm then
                placement ← SOvalue
                if placement < bestPlacement then
                    bestHost ← host
                    bestPlacement ← placement
        if bestHost ≠ NULL then
            tentativeSOPlacement.add(vm, bestHost)
    GlobalPowerSO ← getSOGlobalPower(tentativeSOPlacement)
    PowerSO.add(SO,GlobalPowerSO)
bestSO ← PowerSO.getSOMinPowerAfterAlloaction()
DynSOPlacement ← PowerSO.getTentativeSOPlacement(bestSO)
return DynSOPlacement
```

10.4 Modeling Metaheuristic-based SO Allocation Objectives

In this section we aim to obtain an expression that defines the energy behavior of our SA allocation algorithm using local optimizations. For this purpose, we model the global energy consumption of each time slot for the metaheuristic ($Energy_{SA}$) using parameters of the different SO described in Section 10.3. The conducted experiments have the same configuration as the ones described in Section 10.6.1. First, we run the workload using each SO_n algorithm and, after each time slot we monitor: i) the consolidation value normalized in the range [1,2] ($|SO_n|_1^2$), and ii) the total energy consumption of the infrastructure after the consolidation process is completed (E_{SO_n}). Then, the same workload is run using the SA optimization algorithm for VM allocation and also, after each time slot, we collect the global energy consumption E_{SA}. Considering the global energy of the entire data center helps us to incorporate in the optimization the knowledge not only from the IT and cooling contributions but also the contributions of the VM migrations needed to avoid underloaded situations.

In this work we use the SA provided by the HEuRistic Optimization (HERO) library of optimization algorithms[1] configured as in Section 10.6.1. For SA samples, we separate the entire monitored data into a training and a testing data set. The data set used for for this modeling process consist of the samples collected during the simulation of only the first 24 hours of the Workload 1 configured as defined in Section 10.6.2. Also, we only use those samples in which the SA outperforms the SO policies in terms of energy, as the SA not always perform better than the SO strategies because it is able to provide worse final solutions to get out from local minima. We train the models inferring the expressions shown in Equation 10.6, where $|SO_3|_1^2$ and $|SO_6|_1^2$ are the normalized consolidation values obtained for local

[1] github.com/jlrisco/hero

optimizations SO_3 and SO_6 respectively.

$$SO_{SA} \quad = \quad E_{SA} = 0.1603 \cdot |SO_3|_1^2 \cdot E_{SO_3} + 0.7724 \cdot |SO_6|_1^2 \cdot E_{SO_6} + 0.0102 \qquad (10.6)$$

The fitting is shown in Figures 10.3 and 10.4 for training and testing respectively. For the SA energy, E_{SA}, we obtain an average error percentage of 3.05% and 2.87% for training and testing. Finally, we use this expression to calculate the consolidation value used in SO_{SA}.

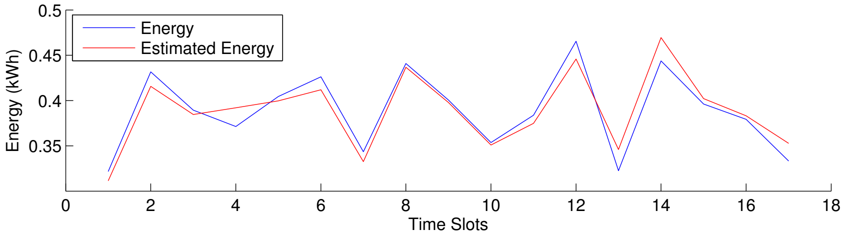


Figure 10.3: Modeling fitting for SO_{SA} using Simulated Annealing samples.

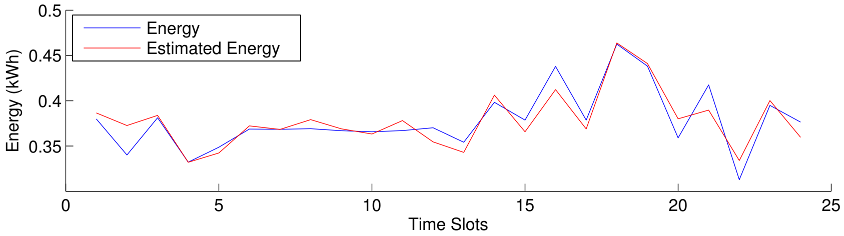


Figure 10.4: Testing modeling for SO_{SA} using Simulated Annealing samples.

For our model we obtain a mean error between the estimated energy and the real trace of $8.84 \cdot 10^{-7}$ kWh and a standard deviation of 0.0144 kWh. Figure 10.5 shows the power error distribution for this model, where it can be seen that the error in terms of power of the 68% of the samples ranges from -0.0144 to 0.0144 kWh.

10.5 Cooling strategy based on VM allocation

The power needed to cool down the servers, thus maintaining a safe temperature, is one of the major contributors to the overall data center budget. Many of the reliability issues and system failures in a data center are given by the adverse effects due to hot spots that may also cause an irreversible damage in the IT infrastructure. However, controlling the set point temperature of the data room is still to be clearly defined and represents a key challenge from the energy perspective. This value is often chosen for the worst case scenario (all devices running consuming maximum power), and based on conservative suggestions provided by the manufacturers of the equipment, resulting in overcooled facilities. In this section we present a novel cooling strategy based on the temperature of the system's devices due to VMs' allocation that can be seen in Algorithm 8.

Our cooling strategy aims to find the highest cooling set point that ensures safe operation for the whole data center infrastructure. Inside the physical machine, the CPU is the component that presents the highest temperatures and this parameter depends on both the inlet temperature and the CPU utilization ($utilization$). So, the CPU temperature will limit the highest value for the inlet temperature of the host ($maxInletTemperature$) in order to operate in a safe range (lower than $maximumSafeCPUTemperature$) avoiding thermal issues. Depending on the VMs distribution and the server location, we define a maximum cooling set point ($maxCoolingSetPoint$) for each host that ensures that its maximum inlet

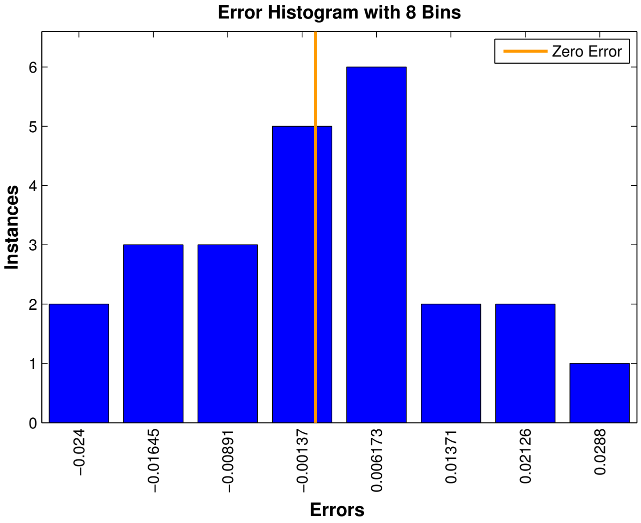


Figure 10.5: Power error distribution for our SO_{SA} model.

temperature is not exceeded so the CPU temperature is safe. Finally the cooling set point is set to the lowest value within the $maxCoolingSetPoint$ for all the servers, thus guaranteeing that the infrastructure operates below the maximum safe CPU temperature defined as $maximumSafeCPUTemperature$.

Algorithm 8 Cooling strategy

```
Input: hostsList maximumSafeCPUTemperature
Output: cooling
foreach host in hostList do
    utilization $\leftarrow$ host.getUtilization()
    maxInletTemperature $\leftarrow$ host.getMaxInletTemperature(utilization, maximumSafeCPUTemperature)
    maxCoolingSetPoint $\leftarrow$ host.getMaxCoolingSetPoint(maxInletTemperature)
    hostCooling.add(maxCoolingSetPoint)
globalMaxCoolingSetPoint $\leftarrow$ hostCooling.getMin()
cooling.setCoolingSetPoint(maxCoolingSetPoint)
return cooling
```

10.6 Performance Evaluation

In this section, we present the impact of our proposed optimization strategies in the energy consumption of the data center, including both IT and cooling contributions. As it is difficult to replicate large-scale experiments in a real data center infrastructure, thus maintaining experimental system conditions, we have chosen the CloudSim 2.0 toolkit [132] to simulate a IaaS Cloud computing environment as in the previous chapter.

Apart from the DVFS management, for this work, we have provided thermal-awareness to the CloudSim simulator. Moreover, the temperature of servers' inlet, memory devices and CPUs vary depending on the workload distribution and on the resource demand. We incorporate these dependence by including different thermal models. Also our frequency and thermal-aware server power model has been included. Finally, in order to obtain temperature and power performance, we have also incorporated memory and disk usage management.

Our simulations run on a 64-bit Ubuntu 14.04.5 Long Term Support (LTS) OS running on an Intel Core i7-4770 CPU @3.40GHz ASUS Workstation with four cores and 8 GB of RAM. Experiments are configured according to the following considerations.

10.6.1 Experimental Setup

We conduct our experiments using real data from the Bitbrains service provider. This workload has the typical characteristics of Cloud computing environments in terms of variability and scalability [166]. Our data set contains performance metrics of 1,127 VMs from a distributed data center from Bitbrains. It includes resource provisioning and resource demand of CPU, RAM and disk as well as the number of cores of each VM with a monitoring interval of 300 seconds. These parameters define the heterogeneous VM instances available for all the simulations. We split the data set into three workloads that provide three scenarios with different CPU variability. Each scenario represents one week of real traces from the Bitbrains Cloud data center. As can be seen in Figure 10.6, Workloads 1 to 3 present decreasing aggregated CPU utilization variability of 568.507%, 284.626% and 143.603% respectively.

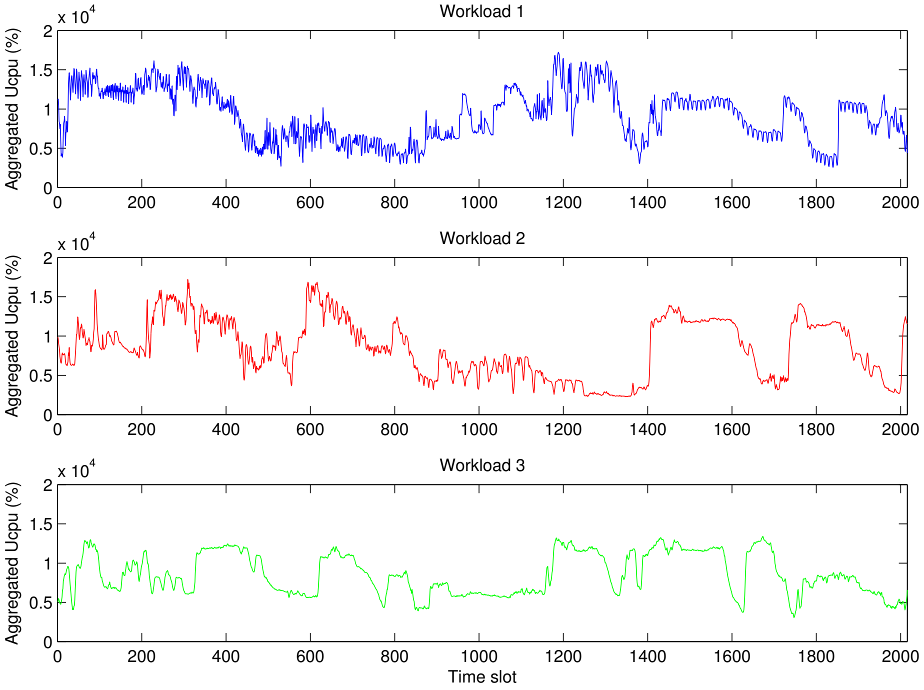


Figure 10.6: One-week workloads with different CPU utilization variability.

The simulation consists of a data center of 1200 hosts modeled as a Fujitsu RX300 S6 server based on an Intel Xeon E5620 Quad Core processor @2.4GHz, RAM memory of 16GB and storage of 1GB, running a 64bit CentOS 6.4 OS virtualized by the QEMU-KVM hypervisor. During the simulations, the number of servers will be significantly reduced as oversubscription is enabled. The proposed Fujitsu server operates at different DVFS modes as seen in Equation 9.18. For optimization purposes, we have simulated all our algorithms under the frequency constraints of our ad-hoc DVFS-performance aware governor proposed in Section 9.3.1, as it has been demonstrated to give further energy improvements without affecting SLA. Moreover, maximum CPU temperature is constrained to take values that are lower or equal to 65° C for reliability purposes. Also server's inlet is limited to a 30° C upper bound, to avoid fan failures.

Power and Thermal Models

In this subsection we present the different models referenced in our work, and also two novel ones derived for this research. To estimate the energy consumed by the IT infrastructure, we use our DVFS and thermal aware server power model defined in Section 9.4. As our power model shown in Equation 9.16 depends on the memory temperature, for this research, we present a novel temperature model for the memory device.

The temperature of the memories in a server depends on several factors both internal and external to the physical machine. The utilization of the memory subsystems, the inlet temperature of the host and the fan speeds are potential contributors to memory temperature that have to be taken into account. In order to gather the real data during runtime, we monitor the system using different hardware and software resources. *collectd* monitoring tool is used to collect the values taken by the system in order to monitor u_{MEM}. Memory and CPU temperatures and fan speed are monitored using on board sensors that are consulted via the software tool *IPMI*. Inlet temperature is collected using external temperature sensors. Finally, room temperature has been modified during run-time in order to find the dependence with the inlet temperature.

In this research, a synthetic workload is used to stress specifically the memory resources, increasing the range of possible values of the considered variables. Therefore, our model may be adapted to estimate different workload characteristics and profiles. We run *RandMem* onto 4 parallel Virtual Machines that have been provisioned to the available computing resources of the server. Then the samples are separated into a training and a testing data set.

After training, we obtain the model shown in Equation 10.7, where $Tmem$ is the memory temperature, $Umem$ the memory utilization, $Tinlet$ the inlet temperature of the server, $k_1 = 0.9965$ and $k_2 = 2.6225$. Then, we evaluate the quality of the thermal model using the testing data set in order to verify the reliability of the estimation. For our data fitting, we obtain an average error percentage of 0.5303% during training and 0.5049% for testing. These values have been obtained using Equation 10.8. In our research, the time slots are defined as each time an optimization is performed in order to consolidate a set of VMs into a set of candidate hosts.

$$T_{mem} = k_1 \cdot T_{inlet} + k_2 \cdot ln(U_{mem}^2) \tag{10.7}$$

$$e_{AVG} = \sqrt{\frac{1}{N} \cdot \sum_n \Big(\frac{|T_{mem}(n) - \widehat{T_{mem}}(n)| \cdot 100}{T_{mem}(n)}\Big)^2}, 1 \leq n \leq N \tag{10.8}$$

$$\tag{10.9}$$

Finally, for our thermal model we obtain a mean error between the estimated temperature and the real measurement of $8.54 \cdot 10^{-4}$ K and a standard deviation of 2.02 K. Figure 10.7 shows the error distribution for this model. According to this, we can conclude that the error in terms of temperature of about the 68% of the samples ranges from -2.02 to 2.02 K. In Figure 10.8, the fitting of our thermal model is provided.

On the other hand, the CPU presents the highest temperatures inside the physical machine, so its temperature will limit the highest value for inlet temperature in order to operate in a safe range, while avoiding thermal issues. Thus, we follow the same approach in order to model the CPU temperature of the server. This parameter depends on both the inlet temperature and the CPU utilization (u_{CPU}).

After training, we obtain the model shown in Equation 10.10, where $Tcpu$ is the CPU temperature, $Ucpu$ its utilization, $k_1 = 1.052$ and $k_2 = 19.845$. Our model presents average error percentages of 0.64% and 0.84% during training and testing respectively.

$$T_{cpu} = k_1 \cdot T_{inlet} + k_2 \cdot U_{cpu} \tag{10.10}$$

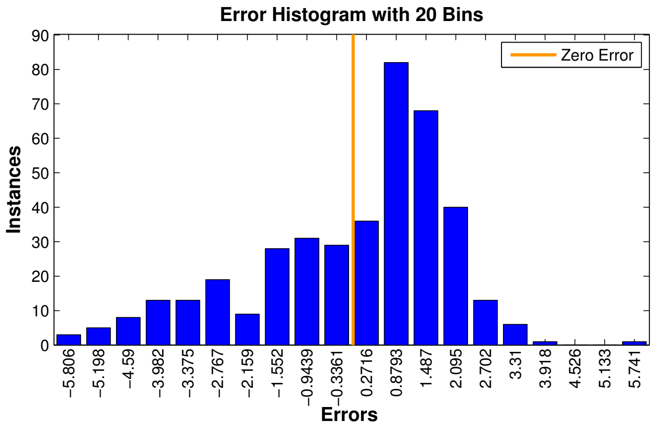


Figure 10.7: Temperature error distribution for our memory model.

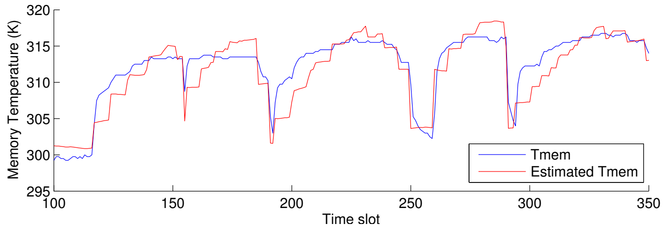


Figure 10.8: Modeling fitting for the memory temperature.

Finally, we obtain a mean error between the estimated temperature and the real measurement of 0.0026 K and a standard deviation of 2.683 K. Figure 10.9 shows the error distribution for this model, where the error in terms of temperature of about the 68% of the samples ranges from -2.68 to 2.68 K. In Figure 10.10, the fitting of our thermal model is provided.

Disk power consumption is modeled according to the work proposed by Lewis et al. [70], as can be seen in Equation 10.11, where $Disk_r$ and $Disk_w$ are the read and write throughputs. We define cooling energy model, shown in Equation 10.12, as in the research presented by Moore et al. [167]. The COP depends on the inlet temperature of the servers' T_{inlet}.

$$\begin{aligned} P_{Disk} &= 3.327 \cdot 10^{-7} \cdot Disk_r + 1.668 \cdot 10^{-7} \cdot Disk_w & (10.11)\\ E_{Cooling} &= E_{IT}/COP = (P_{IT} \cdot t)/COP & (10.12)\\ E_{IT} &= (P_{Fujitsu} + P_{Disk}) \cdot t & (10.13)\\ COP &= 0.0068 \cdot T_{inlet}^2 + 0.0008 \cdot T_{inlet} + 0.458 & (10.14) \end{aligned}$$

Dynamic Consolidation considerations

In all our scenarios we allow online migration, where VMs follow a straightforward load migration policy. Thus, migrations have an energy overhead because, during migration time, two identical VMs are running, consuming the same power in both servers. Performance degradation occurs when the workload demand in a host exceeds its resource capacity.

In this work we allow oversubscription in all the servers so, the total resource demand may exceed their available capacity. If the VMs in a host simultaneously request their maximum performance, this situation may lead to performance degradation due to host overloading. We calculate the impact on SLA as the SLA violation time per active host (SLA_{TAH}) shown in Equation 9.23.

Our dynamic consolidation strategy first chooses which VMs have to be migrated in each server of the data center. For this purpose, we use the adaptive utilization threshold based on

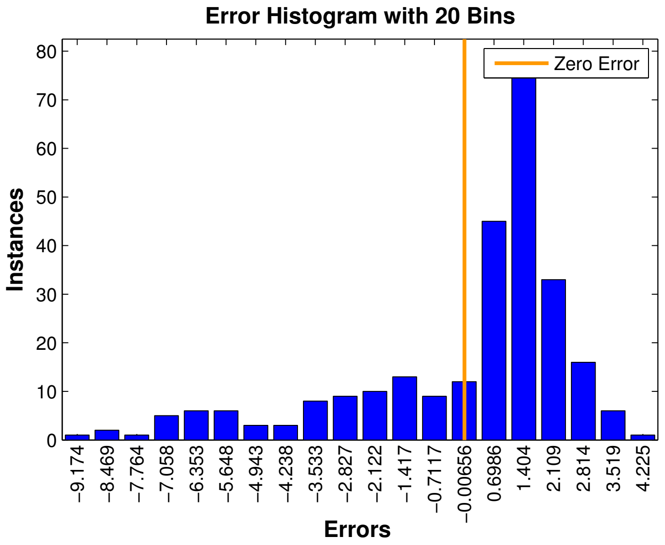


Figure 10.9: Temperature error distribution for our CPU model.

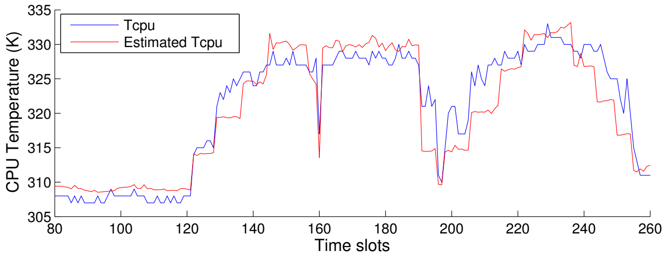


Figure 10.10: Modeling fitting for the CPU temperature.

the *Median Absolute Deviation* of the CPU and the *Minimum migration time* algorithm provided by Beloglazov et al [162]. Then, the VM allocation is performed according to the optimization algorithms provided in this Section 10.3.

When overloading situations are detected using the MAD-MMT technique, VMs are migrated to better placements according to the different proposed algorithms. These migrations provoke Performance Degradation due to Migration (PDM) as seen in Equation 9.24. In this research, to determine SLA violations ($SLA_{violation}$) [162], we use the same metrics that in the previous chapter, which can be seen in Equations 9.22 and 9.23- 9.26.

Metaheuristic-based optimization considerations

In this work we use the SA provided by the HERO library of optimization algorithms configured as single objective for 100,000 iterations and value of k=0.5 for the control of the annealing temperature. Additionally, in the first iteration, the integer variables of the solution are set to the best solution found for the SO policies each time the optimization algorithm is run. This helps to accelerate the algorithm on finding a low-energy valid solution. For the rest of iterations the new set of solutions are provided randomly.

10.6.2 Experimental results

To obtain a preliminary evaluation of the performance of the different VM allocation policies, we have simulated one day of our Workload 1 from Bitbrains using the proposed strategies presented in Section 10.3 for a fixed inlet temperature of 291K. On the one hand, the simulation time when using the SO approaches (SO_{1-8}, SO_{SA}, and $DynSO$) ranges from 8 to 12 minutes. This parameter for MO approaches is around 15 minutes, being in the same order of magnitude. On the other hand, the metaheuristic-based SA has a simulation time that is 60 times higher than the simulation time of SO policies. Also, for SA, there exist optimizations that take more time than the time fixed for the optimization slot (300 s), making it unfeasible to use this metaheuristic during runtime. The energy results provided for the selected algorithms are shown in Figure 10.11. Also, Table 10.1 shows the numerical results of the additional metrics considered.

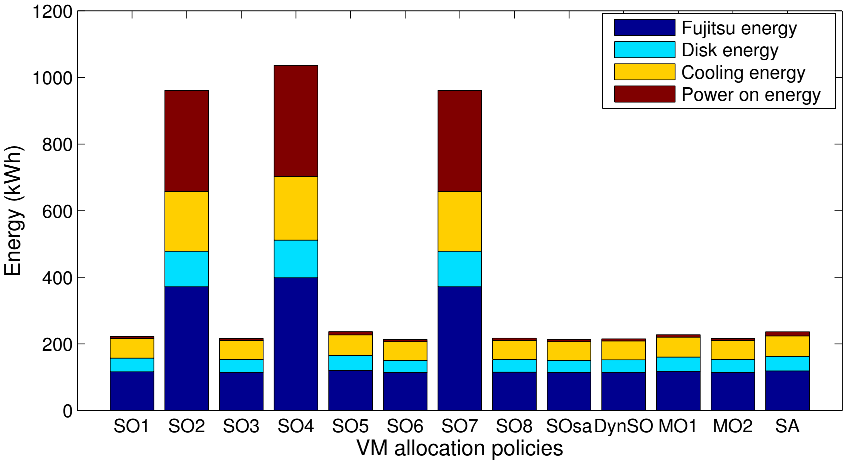


Figure 10.11: Contributions to data center energy per VM allocation strategy for 1 day of Workload 1

Table 10.1: Performance metrics per VM allocation strategy for 1 day of Workload 1

Algorithm	IT Energy (kWh)	Cooling Energy (kWh)	Power-on events	Power-on Energy (kWh)	Migrations events	Average SLA $\cdot 10^{-4}$ (%)	Final Energy (kWh)
SO_1	157.62	58.91	309	5.80	42545	18.43	222.32
SO_2	478.54	178.85	16194	303.74	116044	11.66	961.13
SO_3	153.15	57.24	335	6.28	36711	18.28	216.67
SO_4	511.60	191.21	17792	333.71	125807	11.89	1036.52
SO_5	165.12	61.71	524	9.83	45044	18.36	236.66
SO_6	150.43	56.22	336	6.30	35619	19.29	212.95
SO_7	478.54	178.85	16194	303.74	116044	11.66	961.13
SO_8	153.54	57.38	334	6.26	37376	18.28	217.19
SO_{SA}	150.20	56.14	333	6.25	36983	19.50	212.58
$DynSO$	152.15	56.86	326	6.11	36126	18.71	215.13
MO_1	160.41	59.95	367	6.88	40342	18.15	227.25
MO_2	152.86	57.13	331	6.21	37690	18.62	216.20
SA	162.95	60.90	662	12.42	50540	18.34	236.27

For SO_2, SO_4 and SO_7, the power or temperature of each server is minimized locally resulting in a higher energy consumption due to an increase in the number of active hosts. These algorithms spread the workload as much as possible through the candidate host set as they intend to reduce only the dynamic contribution, which depends on the workload requirements. So, the lower the servers' load, the better for reducing dynamic power consumption locally, thus increasing the global IT contribution as these policies are not aware of their impact on the rest of the infrastructure. Then, after allocating those VMs incoming from overloaded servers, in the next iteration, the algorithm constrains the active server set by migrating VMs from underutilized hosts if possible. So, these algorithms present a higher number of migrations.

Moreover, the energy consumption of both SO_5 and SA policies is above the average due to a higher number of VM migrations and power on events, performed to find the best data center configuration in each time slot. This is due to their trend towards allocating part of the workload in underutilized servers. In the case of SO_5, this situation occurs because, when servers present an utilization below 72% (equivalent utilization for 1.73GHz that is the lowest available frequency), their frequency increment is zero. On the other hand, SA is highly penalized when its solutions provide overloaded servers. So, underutilized servers are preferred when the algorithm does not find a minimum.

Our results show that SO_3 and SO_6 are the best simple-SO optimization policies. This outcome is consistent with the high idle consumption of the Fujitsu's server architecture, so reducing the active servers' set by increasing CPU utilization is a major target to improve energy efficiency. The multi-objective strategies that we present in this research also outperform the baseline, where MO_2 is more competitive in terms of energy.

The SO_{SA} strategy shows the lowest final consumption value, providing energy savings of 4.38% and 10.02% on the global power budget when compared with our baselines SO_1 and SA respectively. This is translated into a reduction of 9.74 kWh and 23.69 kWh as this novel technique also incorporates global information regarding the effect of allocation on future VM migrations. This local approach takes advantage of global knowledge from a holistic viewpoint thus outperforming other strategies for highly variable workloads.

The dynamic selection of the SO consolidation values during runtime ($DynSO$), also reduces power significantly, but do not achieve the best result, even including the best policy. Thus, the best policy (SO_{SA} in this scenario) has not the best consolidation value during local calculations but has the one that best describes the energy behavior of the data center infrastructures as a whole, considering also future migrations. Moreover, the SLA is maintained for all the tests, where a higher increment of $1.07 \cdot 10^{-4}\%$ is detected.

After this proof of concept, we use all the different VM allocation policies to optimize the power consumption of the same infrastructure under several conditions. In this work we propose the optimization of 9 scenarios that combine different cooling strategies and workloads with different load profiles.

First, we optimize Workload 1, which is the one that presents the higher instantaneous variability, for fixed cooling inlets of 291 K and 297 K, and for our variable inlet cooling strategy ($VarInlet$). We obtain the results provided in Table 10.2 in terms of final energy consumption, average SLA and number of migrations.

Table 10.2: Energy, SLA and Migration metrics per inlet temperature and allocation policy for workload 1.

Policy	Energy (kWh)			Average SLA ($\cdot 10^{-4}$ %)			Migrations ($\cdot 10^3$)		
	291K	297K	VarInlet	291K	297K	VarInlet	291K	297K	VarInlet
SO_1	1178.41	1089.27	1034.20	20.76	20.98	21.04	200.8	202.2	203.2
SO_2	5256.76	4718.89	4594.32	11.97	11.94	11.88	647.4	647.7	654.1
SO_3	1159.07	1059.19	1018.19	20.60	20.60	20.60	190.2	190.2	190.2
SO_4	5725.25	5207.93	4990.10	11.87	11.87	11.88	766.6	766.6	772.2
SO_5	1247.97	1139.50	1094.45	20.15	20.15	20.15	217.5	217.5	217.5
SO_6	1157.61	1057.85	1016.91	20.90	20.90	20.90	189.1	189.1	189.1
SO_7	5256.76	4718.89	4594.32	11.97	11.94	11.88	647.4	647.7	654.1
SO_8	1161.44	1061.33	1020.22	20.31	20.31	20.31	192.5	192.5	192.5
SO_{SA}	1152.13	1052.93	1012.30	20.84	20.84	20.84	187.9	187.9	187.9
$DynSO$	1164.30	1055.83	1020.59	20.56	20.61	20.56	190.8	189.0	192.9
MO_1	1199.55	1090.12	1047.56	20.01	20.31	19.94	198.0	199.0	201.0
MO_2	1159.39	1059.45	1018.42	20.62	20.62	20.62	191.7	191.7	191.7
SA	1293.51	1178.35	1130.98	19.94	19.54	19.80	244.9	241.3	244.6

Figure 10.12 shows the different contributions to final energy per VM allocation policy for the different cooling strategies. As inlet temperature rises, the IT power consumption is

increased due to power leakage (see IT 291K, IT 297K and IT VarInlet in the bottom of each stacked column). However, cooling power is reduced with increasing temperatures due to a higher cooling efficiency (shown in Cooling 291K, Cooling 297K and Cooling VarInlet in the middle of each stacked column). The savings performed by higher cooling set points outperform the IT power increments, thus resulting in more efficient scenarios for all the proposed allocation strategies. For this three scenarios, only by applying our $VarInlet$ cooling strategy provides additional energy savings of 3.78% and 12.38% in average when compared with fixed cooling at 297 K and 291 K respectively for all the allocation policies.

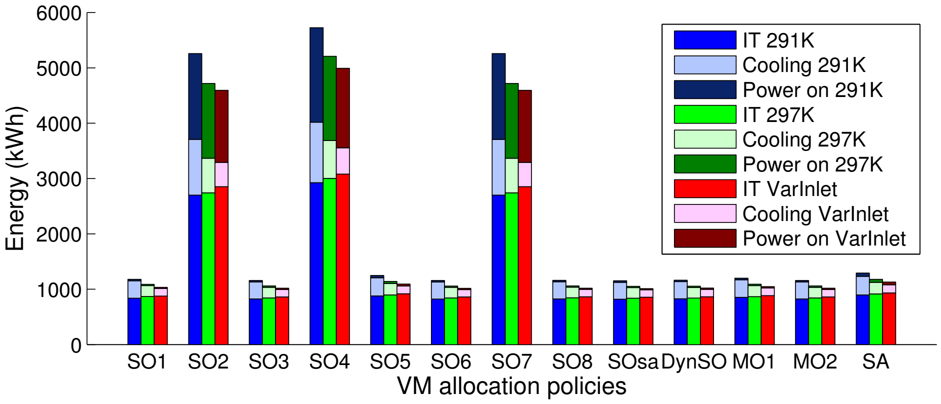


Figure 10.12: Contributions to data center energy per VM allocation strategy for Workload 1

Figure 10.13 shows the energy and average SLA percentage comparison for those strategies that outperform our SO_1 baseline. SO_{SA}, SO_6, SO_3, MO_2 and $DynSO$ allocation policies offer better results, in terms of energy savings, when compared to our global baseline SA and our local baseline SO_1. These policies, when combined with $VarInlet$ strategy, provide savings, of 13.67% and 21.35% in average with respect to SA at 297 K and 291 K respectively. Maximum savings are found of up to 14.09% and 21.74% respectively for our $VarInlet$-SO_{SA} combined strategy. When compared with the local baseline SO_1, these allocation policies provide average savings of 6.61% and 13.67% at 297 K and 291 K respectively, and maximum savings of up to 7.07% and 14.10% respectively for SO_{SA}.

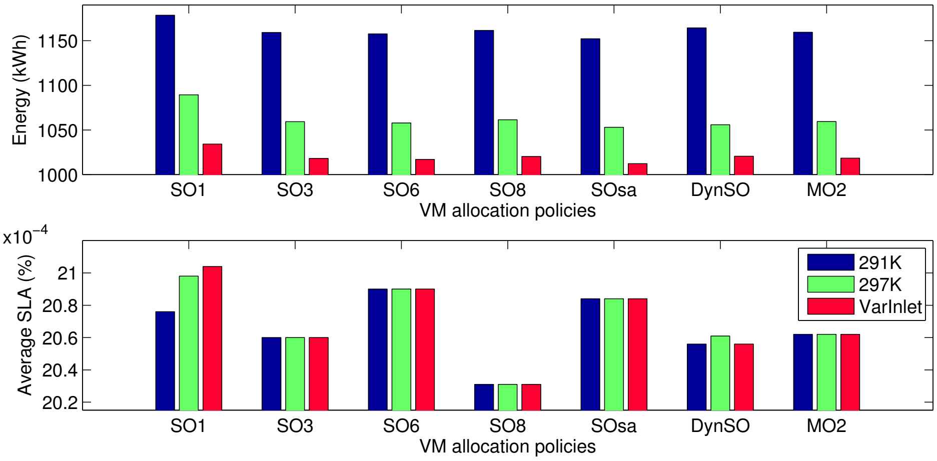


Figure 10.13: Data center energy and SLA per VM allocation strategy for Workload 1

In our three following scenarios we optimize Workload 2, which presents medium instantaneous variability. We obtain the energy consumption, average SLA and migration results provided in Table 10.3.

Table 10.3: Energy, SLA and Migration metrics per inlet temperature and allocation policy for workload 2.

Policy	Energy (kWh)			Average SLA ($\cdot 10^{-4}$ %)			Migrations ($\cdot 10^3$)		
	291K	297K	VarInlet	291K	297K	VarInlet	291K	297K	VarInlet
SO_1	1033.75	949.09	912.18	18.84	18.67	18.58	119.3	126.0	123.0
SO_2	3712.18	3422.65	3278.08	11.15	11.11	11.21	564.2	579.4	570.9
SO_3	1022.49	935.02	899.36	18.43	18.43	18.43	121.7	121.7	121.7
SO_4	4562.67	4142.05	3978.78	11.00	11.00	11.01	751.5	751.5	761.0
SO_5	1081.89	988.79	950.59	18.11	18.11	18.11	136.0	136.0	136.0
SO_6	1015.27	928.55	893.27	18.51	18.51	18.51	120.8	120.8	120.8
SO_7	3712.18	3422.65	3278.08	11.15	11.11	11.21	564.2	579.4	570.9
SO_8	1024.57	936.92	901.17	18.46	18.46	18.46	120.3	120.3	120.3
SO_{SA}	1016.71	929.82	894.48	18.54	18.54	18.54	119.3	119.3	119.3
$DynSO$	1027.06	939.13	903.27	18.24	18.24	18.24	118.9	118.9	118.9
MO_1	1032.37	945.82	913.50	18.21	18.16	18.21	125.0	121.8	122.6
MO_2	1024.80	937.07	901.30	18.28	18.28	18.28	121.5	121.5	121.5
SA	1122.67	1026.40	985.68	18.34	18.12	18.06	160.1	163.3	161.6

In figure 10.14, the same trend towards power and temperature is shown as in Workload 1 scenarios. Savings obtained by increasing cooling set points also outperform IT power increments, thus resulting in more efficient scenarios for all the proposed allocation strategies. For this scenario, only our $VarInlet$ cooling strategy provides additional energy savings of 3.88% and 12.00% in average, when compared with fixed cooling at 297 K and 291 K respectively, for all the allocation policies.

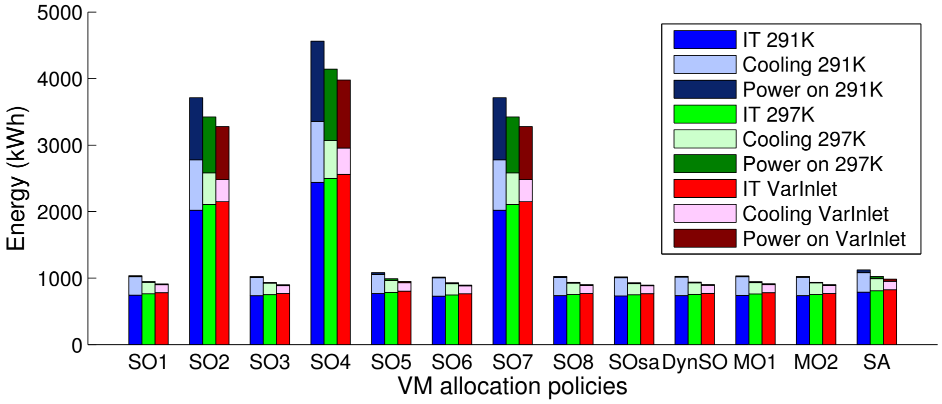


Figure 10.14: Contributions to data center energy per VM allocation strategy for Workload 2

Figure 10.15 shows the energy and average SLA percentage comparison for those strategies that outperform our baseline SO_1, where SLA is maintained. As in the Workload 1 scenarios, SO_{SA}, SO_6, SO_3, MO_2 and $DynSO$ allocation policies offer the best results, in terms of energy savings, when compared to our baselines SA and SO_1. These policies, when combined with $VarInlet$ strategy, provide average savings of 12.48% and 19.99% with respect to SA at 297 K and 291 K respectively, and maximum savings of up to 12.97% and 20.43% respectively for SO_6. These policies, when compared with SO_1 provide average savings of 5.34% and 13.09%

at 297 K and 291 K respectively, and maximum savings of up to 5.88% and 13.59% respectively for SO_6.

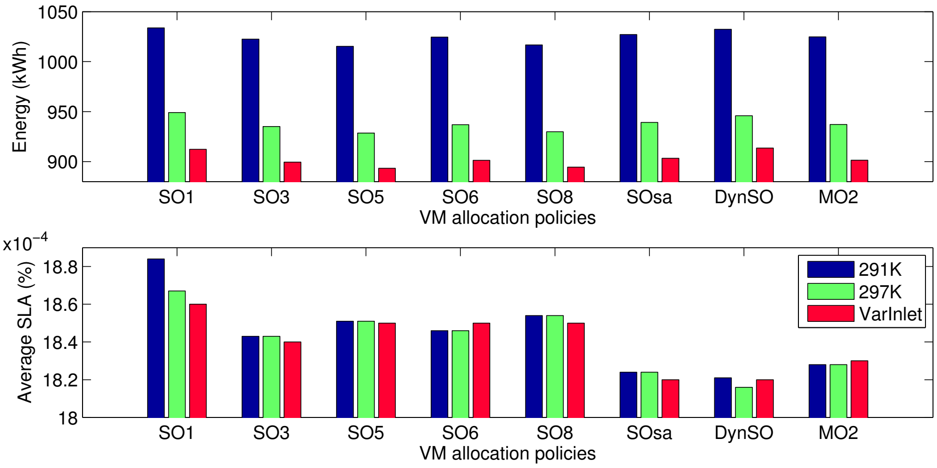


Figure 10.15: Data center energy and SLA per VM allocation strategy for Workload 2

Finally, we optimize Workload 3, which presents the lower instantaneous variability. For this optimization scenarios, we obtain the energy consumption, average SLA and migration results provided in Table 10.4.

Table 10.4: Energy, SLA and Migration metrics per inlet temperature and allocation policy for workload 3.

Policy	Energy (kWh)			Average SLA ($\cdot 10^{-4}$ %)			Migrations ($\cdot 10^{3}$)		
	291K	297K	VarInlet	291K	297K	VarInlet	291K	297K	VarInlet
SO_1	855.52	786.93	759.02	15.68	15.77	15.74	64.4	63.2	67.5
SO_2	2715.34	2505.51	2381.12	10.44	10.49	10.46	467.9	477.7	466.9
SO_3	852.66	781.12	752.55	15.35	15.35	15.35	70.3	70.3	70.3
SO_4	3725.87	3378.07	3232.15	10.37	10.37	10.38	718.1	718.1	721.4
SO_5	884.77	810.29	780.37	15.47	15.47	15.47	73.3	73.3	73.3
SO_6	858.16	785.95	757.06	15.39	15.39	15.39	74.7	74.7	74.7
SO_7	2715.34	2505.51	2381.12	10.44	10.49	10.46	467.9	477.7	466.9
SO_8	855.74	783.85	755.10	15.32	15.32	15.32	72.5	72.5	72.5
SO_{SA}	856.41	84.40	755.58	15.02	15.02	15.02	75.7	75.7	75.7
$DynSO$	853.27	781.65	753.01	15.21	15.21	15.21	72.3	72.3	72.3
MO_1	864.66	787.28	757.73	15.20	15.26	15.41	73.1	71.0	70.1
MO_2	859.39	787.07	758.11	15.21	15.21	15.21	73.8	73.8	73.8
SA	946.98	856.72	827.95	14.87	14.43	14.67	93.9	95.4	96.9

In figure 10.16, the same trend towards power and temperature is presented as in Workload 1 and Workload 2 scenarios. Increasing cooling set points provides savings that outperform IT power increments, thus resulting in more efficient scenarios for all the proposed allocation strategies. For these scenarios, only our $VarInlet$ cooling strategy provides additional energy savings of 3.94% and 11.99% in average when compared with fixed cooling at 297 K and 291 K respectively for all the allocation policies.

Figure 10.15 shows the energy and average SLA percentage comparison for those strategies that outperform our baselines SA and SO_1, where SLA is maintained. For this

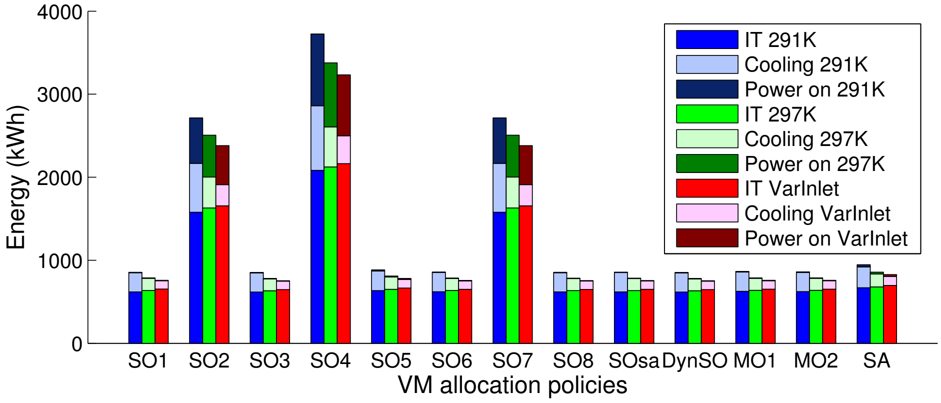


Figure 10.16: Contributions to data center energy per VM allocation strategy for Workload 3

workload, SO_{SA}, SO_6, SO_3, MO_1 and $DynSO$ allocation policies offer the best results, in terms of energy savings, when compared to our baselines. These policies, when combined with $VarInlet$ strategy, provide average savings of 11.78% and 20.19% with respect to SA at 297 K and 291 K respectively, and maximum savings of up to 12.16% and 20.53% respectively for SO_3. These allocation policies, when compared with SO_1 also provide average savings of 4.02% and 11.72% at 297 K and 291 K respectively, and maximum savings of up to 4.37% and 12.04% respectively for SO_3.

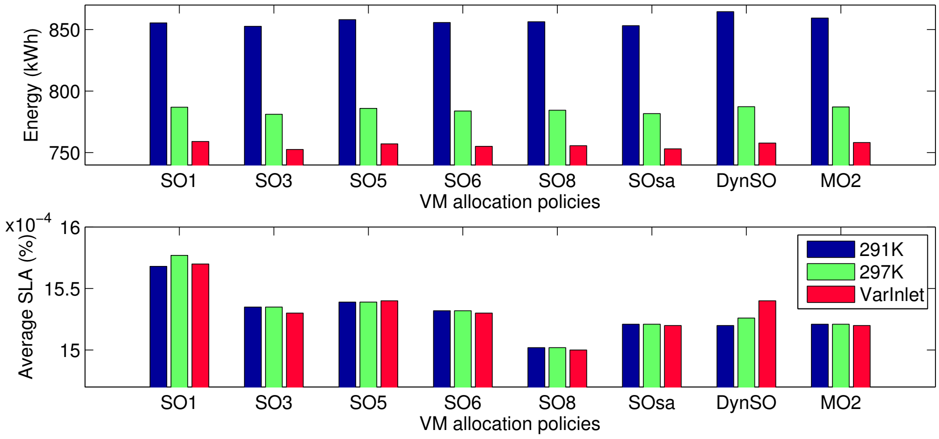


Figure 10.17: Data center energy and SLA per VM allocation strategy for Workload 3

Tables 10.5 and 10.6 provide a summary of the energy savings obtained for the different optimization scenarios for $VarInlet$ cooling strategy, compared with the local SO_1 baseline and the global SA baseline policies respectively. The results show that higher energy savings are provided for increasing instantaneous workload variability for both fixed cooling inlet strategies at 291 K and 297 K.

In the case of Workload 3, as it presents the lower workload variation, the baseline SO_1 presents a better performance so the energy savings are lower. However, the savings provided when compared with the fixed cooling temperature at 297 K cannot been considered as they are below the error obtained for our power model (4.87%). On the other hand, for those workloads with the higher instantaneous variation (Workload 1 and

Table 10.5: Energy savings for $VarInlet$ per VM allocation policy compared with SO_1.

Policy	Energy Savings vs. SO_1 (%)					
	Workload 1		Workload 2		Workload 3	
	297K	291K	297K	291K	297K	291K
SO_1	5.06	12.24	3.89	11.76	3.55	11.28
SO_3	6.53	13.60	5.24	13.00	4.37	12.04
SO_6	6.64	13.71	5.88	13.59	3.80	11.51
SO_{SA}	7.07	14.10	5.75	13.47	3.98	11.68
$DynSO$	6.30	13.39	4.83	12.62	4.31	11.98
MO_2	6.50	13.58	5.04	12.81	3.66	11.39

Table 10.6: Energy savings for $VarInlet$ per VM allocation policy compared with SA.

Policy	Energy Savings vs. SA (%)					
	Workload 1		Workload 2		Workload 3	
	297K	291K	297K	291K	297K	291K
SO_1	12.23	20.05	11.13	18.75	11.40	19.85
SO_3	13.59	21.28	12.38	19.89	12.16	20.53
SO_6	13.70	21.38	12.97	20.43	11.63	20.06
SO_{SA}	14.09	21.74	12.85	20.33	11.81	20.21
$DynSO$	13.39	21.10	12.00	19.54	12.11	20.48
MO_2	13.57	21.27	12.19	19.72	11.51	19.94

Workload 2), our proposed VM allocation algorithms combined with our $VarInlet$ cooling strategy, outperform the baseline for both fixed 291 K and 297 K inlet temperatures achieving significant energy savings.

The proposed VM allocation strategies SO_3, SO_6, SO_{SA}, $DynSO$, MO_2 cannot be compared between them in terms of power savings, as their relative savings fall behind the error of our power model. In terms of SLA, the outcomes show that our algorithms maintain the SLA obtained for the baseline policy. The SLA violations are increased by SO_6 and SO_{SA} for the most variable scenario (Workload 1), and only when compared with the 291 K fixed cooling policy. However, this increment is only of about $0.14 \cdot 10^{-4}$. If the SLA is critical for the data center management, MO_2 provides SLA reductions that are consistent within the 9 scenarios, also offering competitive energy savings.

Globally, our SO_{SA} is the strategy that performs better if selected for all the different scenarios with significantly different workload profiles. This approach presents the best savings for Workload 1, which is the more variable one, and a very high savings value for less variable workloads 2 and 3. For all the scenarios, the SO_{SA} approach outperforms the average savings provided by SO_3, SO_6, $DynSO$ and MO_2 as can be seen in Table 10.7. Our local SO based on SA optimization,SO_{SA}, leverages the information from a global strategy combined with the information of the overall data center infrastructure provided by our holistic approach.

Table 10.7: Energy savings for $VarInlet$ in average and for SO_{SA} strategy.

Policy	Workload 1		Workload 2		Workload 3		
	297K	291K	297K	291K	297K	291K	
Baseline SA	average	13.67%	21.35%	12.48%	19.99%	11.78%	20.19%
	SO_{SA}	14.09%	21.74%	12.85%	20.33%	12.16%	20.50%
Baseline SO_1	average	6.61%	13.67%	5.34%	13.09%	4.02%	11.72%
	SO_{SA}	7.07%	14.10%	5.75%	13.47%	3.98%	11.68%

Finally, the Power Usage Effectiveness (PUE) is a metric that measures the efficiency of a data center in terms of energy usage. Specifically, it provides information of the amount of energy that is used by the computing equipment in contrast to cooling and other overheads. The PUE value works well with cooling optimizations, as reductions on cooling power result in a higher percentage of the power budget consumed only by IT. On the other hand, for IT optimizations that do not impact on cooling consumption, the PUE does not show the efficiency gained by the IT energy reduction, but reports a negative impact, as it results on a higher percentage of the total power used for cooling purposes. Our research reduces both IT and cooling contributions to final power consumption simultaneously. For fixed cooling baseline policies with set point temperatures of 291 K and 297 K, our infrastructure provides PUE values of 1.37 and 1.23 respectively. Our $VarInlet$ approach, together with our proposed dynamic consolidation SO_{SA}, optimizes the PUE in up to 16.05% and 6.5% respectively, providing PUE values between 1.15 and 1.16 that outperforms the state-of-the-art value that is around 1.2.

10.7 Summary

The new holistic paradigm proposed in this work focuses on considering the energy globally. In this way, all the data center elements are aware of the evolution of the global energy demand and the thermal behavior of the room. Our decisions are based on information from all available subsystems to perform energy optimizations from technology impact to data center level.

Metaheuristic algorithms like Simulated Annealing, when used for VM consolidation in data centers, are able to achieve very good results in terms of energy. However, the time they need to perform the optimizations makes them unfeasible to be used during runtime for this purpose. On the other hand, various local BFD-based policies provide good solutions to the energy problem. They constrain the set of active servers, thus reducing the static energy consumption, but this local strategies do not consider the final status of the data center after each optimization, so the number of VM migrations may be increased.

Leveraging the knowledge gathered from both metaheuristic and BFD algorithms helps us to infer models that describe global energy patterns into local strategies, which are faster and lighter to be used to optimize energy consumption during runtime. By using this technique we provide the SO_{SA} VM allocation policy that together with our proposed cooling strategy $VarInlet$, allow us to improve energy efficiency in scenarios with high workload variability. Our local technique achieved energy savings of 7.07% and 14.10% when compared with the local baseline PABFD using two different traditional cooling strategies with fixed set points at 297 K and 291 K respectively. Also, compared with a global SA-based baseline, our local VM allocation policies provided energy savings of up to 14.09% and 21.74%. For all the scenarios proposed in this research, our optimization algorithms maintain QoS when compared with local and global baselines.

Finally, the following chapter concludes this Ph.D thesis, summarizing its main contributions to the state-of-the-art and proposing future research directions derived from this dissertation.

11. Conclusions and Future Directions

> *"I have passed through fire and deep water, since we parted. I have forgotten much that I thought I knew, and learned again much that I had forgotten."*
>
> — J.R.R. Tolkien, *The Lord of the Rings*

This Ph.D. thesis has addressed the energy challenge by proposing proactive power and thermal-aware optimization techniques that contribute to place Cloud data centers on a more scalable curve. In this chapter, we present a synthesis of the conclusions derived from the research fulfilled during this Ph.D. thesis, emphasizing on the contributions to the state-of-the-art provided by this research. Finally, we conclude this dissertation highlighting the open research challenges and future directions derived from this work.

11.1 Summary and Conclusions

As described in the motivation of this Ph.D. thesis (Section 1.1), computational demand in data centers is increasing due to growing popularity of Cloud applications. The contribution of Cloud data centers in the overall consumption of modern cities is growing dramatically, becoming unsustainable in terms of power consumption and growing energy costs. Therefore, minimizing their power consumption is a critic challenge to reduce both economic and environmental impact.

This Ph.D. thesis presents the potential of holistic optimization approaches to improve energy efficiency in Cloud facilities from a higher-level perspective. According to the state-of-the-art in Part I, major challenges in the area have not been yet fulfilled as those concerning combined power and thermal awareness, dynamic-applications consolidation or joint cooling and IT energy minimization.

The main objective of this Ph.D. thesis focuses on addressing the energy challenge in Cloud data centers from a thermal and power-aware perspective using proactive strategies. Our work proposes the design and implementation of models and global optimizations that jointly consider energy consumption of both computing and cooling resources while maintaining QoS.

As presented in Figure 1.3 in Chapter 1, our work proposes a global solution based on the power analysis and optimization for Cloud applications from multiple abstraction layers. We develop power and thermal models that can be used during runtime and use the knowledge about the power demand and the IT and cooling resources available at data center to optimize energy consumption. Moreover, our optimization framework offers a dynamic solution for scenarios running workloads that present high variability while maintaining SLA. This work makes contributions in a complex and multidisciplinary area, of high economic and social impact.

According to the research objectives highlighted in Section 1.6 of Chapter 1, during this Ph.D. thesis we have achieved the following results:

- We have defined a taxonomy that compiles the different levels of abstraction that can be found in data centers, classifying current research and evaluating its impact on energy efficiency.

Part I presents this survey and the problem statement and positioning of our dissertation. We identify new open challenges that have the potential of improving sustainability on data centers significantly by including information at different abstraction levels from a holistic perspective.

- We detect the need of addressing servers' leakage power together with cooling set point temperatures to achieve substantial global savings, evaluating the relationships between temperature and power consumption. We identify the trade-off between leakage and cooling consumption based on empirical research.

  Our results show that increasing the setpoint temperature of the data center in 6° C, reduces cooling power by 11.7%, but also increases application power consumption in about 4.5%.

  These contributions have been presented in Chapter 4.

- We detect those parameters that mainly impact on leading power-aware strategies for improving Cloud efficiency as well as thermal considerations. We derive models that incorporate these contributors that help to find the relationships required to devise global optimizations combining power and thermal-aware strategies.

  For this purpose we analyze and implement novel modeling techniques for the automatic identification of fast and accurate models that help to target enterprise server architectures with no effort for designers. Also the execution of the resulting power models is fast, making them suitable for runtime optimizations.

  Current models, which do not consider both DVFS and thermal-awareness, present power accuracies that range from 7.66% to 5.37%. Our models provide an error, when compared with real measurements, which ranges from 4.87% to 3.98% in average, thus outperforming the state-of-the-art.

  This work has been presented in Chapters 5, 6, 7 and 8.

- Finally, we have developed data center energy optimizations, designing and implementing new policies for dynamic Cloud services that combine leading power-aware strategies while ensuring QoS. For this purpose, we design and implement new approaches that also includes thermal considerations in both cooling and IT consumption.

  First, to evaluate the impact of DVFS on VM dynamic consolidation, we present our Freq-Aware optimization that combines a novel reactive DVFS policy with our proactive Frequency-aware consolidation technique. We have achieved competitive energy savings, compared with a state-of-the-art baseline, of up to 45.76% for the IT infrastructure, also increasing global server utilization to 83% in average, while maintaining QoS.

  Then, we evaluate different dynamic consolidation techniques also taking into account the cooling contribution and the impact of temperature on the IT infrastructure. For this purpose we have also implemented thermal models for the CPU and memory devices with average testing errors of 0.84% and 0.5049% respectively, compared to real measurements. We provide a dynamic cooling strategy, which aims to find the highest cooling set point, ensuring safe operation for the whole data center infrastructure during runtime.

  We also present a novel local optimization that leverages the global knowledge from a holistic viewpoint thus outperforming other strategies for highly variable workloads. Our dynamic consolidation policy SO_{SA}, when combined with our $VarInlet$ cooling strategy, provide maximum savings of up to 14.09% and 21.74% with respect to our state-of-the-art baselines.

  These contributions have been described in Chapters 9 and 10.

The work developed in this Ph.D. thesis has enabled a very close collaboration between the Architecture and Technology of Computing Systems (ArTeCS) group at Universidad Complutense de Madrid and the LSI group at Universidad Politécnica de Madrid. Moreover, a stable collaboration has been established with the CLOUDS Lab. at the University of Melbourne. This collaboration has resulted into a 3-month research stay of the author at the University of Melbourne, one poster, one core-A conference and one JCR journal co-authored papers.

The research presented in this Ph.D. thesis provide realistic models and optimizations that can be used in real data center scenarios, yielding significant savings. All models proposed in this work have been developed and tested in real scenarios. Server models have been validated in presently-shipping enterprise servers, belonging to the Universidad Politécnica de Madrid. For data center room simulations this work has used real traces from PlanetLab and Bitbrains infrastructures that are publicly available. Thus, the work presented has a high applicability, being of high interest to both industry and academic areas, and can potentially obtain important savings in real environments.

11.2 Future Research Directions

The research in this Ph.D. thesis has focused on the development of models and optimization techniques at different abstraction layers: from the server to the data center level, also considering the Cloud application framework. The proposed proactive techniques are aware of the trade-offs between power, temperature, cooling and SLA. However, some interesting points of future research have emerged during the completion of this work. Following, we propose future research directions and improvements of the work presented in this dissertation.

11.2.1 Supporting Thermal Dependencies of Air-Cooling Models

The raised-floor air-cooling model, present in the majority of today's data centers, impacts on the internal cooling devices of servers. As increasing the set point temperature reduces cooling consumption, this increment may accelerate the speed of fans inside the servers, thus increasing IT consumption. This introduces various complexities into the dynamic optimization approaches based on the trade-offs between temperature and consumption of both IT and cooling infrastructures. To overcome these complexities, accurate fan models may be provided to analyze further variations of IT power due to temperature and to offer finer optimizations. As it is hard to find these complex relationships, metaheuristic-based modeling approaches (as our GE-based modeling methodology), could help to automatically infer thermal trade-offs in fan speed modeling.

11.2.2 Supporting Different Cooling Models

While the traditional air-cooled model is a dominant model for many data centers, next-generation cooling techniques, such as oil-based and two-phase immersion, need a different cooling model. This techniques are based on placing the servers in a container filled with a fluid, e.g. oil or Novec, that dissipates the heat. This changes completely the thermal behavior of servers, where new trade-offs arise. In this case, evolutionary computation (as in our GE-based modeling approach) could help to automatically infer the optimal set of features that describe the complex thermal models for both cooling and computing infrastructures.

11.2.3 Supporting Different Application Models

While a service-based model is a dominant model for many Cloud applications in current data centers, there are other applications that could run in a Cloud environment (e.g. Web, HPC, Big Data, enterprise and transactions on mobile applications) that need a different application model. For these application frameworks, disk and network can be a bottleneck and have an impact on the overall energy consumption. This changes completely the dynamic behavior of the applications during runtime and also the power consumption patterns of the different server's subsystems. In this case, applying our modeling techniques to obtain accurate models for disk and network could help to find relationships between application performance, resource contention and power consumption for these application models. Moreover, the design of a local or global proactive optimization that is also aware of the impact of these trade-offs could help to reduce global energy.

11.2.4 Supporting Heterogeneity of Servers

Typically, when upgrading a data center, the economic budget limits the number of new physical machines that may be purchased. So, in the majority of the cases, both new and old architectures coexist. Thus, data centers consist of homogeneous clusters of servers that can be optimized separately. However, the optimal performance of different applications may target different server architectures. In this case, proactive local or global optimization techniques based on resource management and scheduling may leverage heterogeneity of servers. Also, new models may be inferred to describe the power consumption of VM migrations between servers.

Bibliography

[1] Q. Chen and et al., "Profiling energy consumption of vms for green cloud computing", in *DASC*, 2011, pp. 768–775.

[2] N. Engbers and E. Taen, *Green Data Net. Report to IT Room INFRA*, European Commision. FP7 ICT 2013.6.2, Nov. 2014.

[3] A. Donoghue, P. Inglesant, and A. Lawrence, *The EU dreams of renewable-powered datacenters with smart-city addresses*, "https://451research.com", Oct. 2013.

[4] Ponemon Institute, "2013 Study on Data Center Outages", Ponemon Institure sponsored by Emerson Network Power, Tech. Rep., Sep. 2013.

[5] J. Hartley, "The truth about data centre cooling", *eCool Solutions*, 2010.

[6] A. Berl, E. Gelenbe, M. Di Girolamo, G. Giuliani, H. De Meer, M. Q. Dang, and K. Pentikousis, "Energy-efficient cloud computing", *Comput. J.*, vol. 53, no. 7, pp. 1045–1051, Sep. 2010, ISSN: 0010-4620.

[7] P. Niles and P. Donovan, "Virtualization and Cloud Computing: Optimized Power, Cooling, and Management Maximizes Benefits. White paper 118. Revision 3", Schneider Electric, Tech. Rep., 2011.

[8] T. Breen, E. Walsh, J. Punch, A. Shah, and C. Bash, "From chip to cooling tower data center modeling: part i influence of server inlet temperature and temperature rise across cabinet", in *Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm), 2010 12th IEEE Intersociety Conference on*, Jun. 2010, pp. 1–10.

[9] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer", in *Proceedings of the 34th Annual International Symposium on Computer Architecture*, ser. ISCA '07, San Diego, California, USA: ACM, 2007, pp. 13–23, ISBN: 978-1-59593-706-3.

[10] A. W. Services, *Amazon Elastic Compute Cloud (Amazon EC2)*, http://aws.amazon.com/ec2/, [Online; accessed 1-March-2012], 2012.

[11] G. A. for Business, *Top ten advantages of Google's cloud*, http://www.google.com/apps/intl/en/business/cloud.html, [Online; accessed 1-March-2012], 2012.

[12] Microsoft News Center, *Microsoft on Cloud Computing*, http://www.microsoft.com/presspass/presskits/cloud/, [Online; accessed 1-March-2012], 2012.

[13] iCloud, *What is iCloud?*, http://www.apple.com/icloud/what-is.html, [Online; accessed 1-March-2012], 2012.

[14] J. Koomey, "Growth in data center electricity use 2005 to 2010", Analytics Press, Oakland, CA, Tech. Rep., 2011.

[15] R. H. Katz, *Tech Titans Building Boom*, http://spectrum.ieee.org/green-tech/buildings/tech-titans-building-boom/0, [Online; accessed 9-March-2012], February 2009.

[16] Gartner Report, *Forecast Analysis: Public Cloud Services, Worldwide, 4Q16 Update*, 2017.

[17] P. Scheihing, “Creating energy efficient data center”, in *Data Center Facilities and Engineering Conference*, Washington DC, USA, May 2007.

[18] J. Markoff and S. Lohr, “Intel’s huge bet turns iffy”, *New York Times Technology Section*, Sep. 2002.

[19] J. G. Koomey, “Growth in data center electricity use 2005 to 2010”, Tech. Rep., Aug. 2011, p. 24.

[20] J. Hamilton, “Cooperative expendable micro-slice servers (cems): low cost, low power servers for internet-scale services”, in *Proceedings of the 4th Biennial Conf. Innovative Data Systems Research*, ser. CIDR ’09, Asilomar, CA, USA, 2009.

[21] D. Barbagallo, E. Di Nitto, D. Dubois, and R. Mirandola, “A bio-inspired algorithm for energy optimization in a self-organizing data center”, English, in *Self-Organizing Architectures*, ser. Lecture Notes in Computer Science, D. Weyns, S. Malek, R. de Lemos, and J. Andersson, Eds., vol. 6090, Springer Berlin Heidelberg, 2010, pp. 127–151, ISBN: 978-3-642-14411-0.

[22] iTRACS, *Cloud To Cut Energy Consumption 31% by 2020*, ”http://www.itracs.com/cloud-computing/cloud-to-cut-energy-consumption-31-by-2020/”, [Online; accessed 3-March-2012], 2012.

[23] Y. Chen, L. Keys, and R. H. Katz, “Towards energy efficient mapreduce”, EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-109, Aug. 2009.

[24] J. Rabaey, *Low Power Design Essentials*, ser. Engineering (Springer-11647). Springer, 2009, ISBN: 9780387717128.

[25] S. Sanathanamurthy, “Simulated temperature dependency of SEU sensitivity in a 0.5 μm CMOS SRAM”, PhD thesis, Aug. 2008.

[26] W. Wang, J. Tao, and P. Fang, “Dependence of hci mechanism on temperature for 0.18 mu;m technology and beyond”, in *Integrated Reliability Workshop Final Report, 1999. IEEE International*, 1999, pp. 66 –68.

[27] T. Breen, E. Walsh, J. Punch, A. Shah, and C. Bash, “From chip to cooling tower data center modeling: part i influence of server inlet temperature and temperature rise across cabinet”, in *Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm), 2010 12th IEEE Intersociety Conference on*, 2010, pp. 1–10.

[28] P. Arroba, J. M. Moya, J. L. Ayala, and R. Buyya, “Dynamic voltage and frequency scaling-aware dynamic consolidation of virtual machines for energy efficient cloud data centers”, *Concurrency and Computation: Practice and Experience*, vol. 29, no. 10, 2017, ISSN: 1532-0634.

[29] P. Arroba, J. L. Risco-Martín, M. Zapater, J. M. Moya, and J. L. Ayala, “Enhancing regression models for complex systems using evolutionary techniques for feature engineering”, *J. Grid Comput.*, vol. 13, no. 3, pp. 409–423, Sep. 2015, ISSN: 1570-7873.

[30] P. Arroba, J. L. Risco-Martín, M. Zapater, J. M. Moya, J. L. Ayala, and K. Olcoz, “Server power modeling for run-time energy optimization of cloud computing facilities”, *Energy Procedia*, vol. 62, pp. 401 –410, 2014, ISSN: 1876-6102.

[31] P. Arroba, J. M. Moya, J. L. Ayala, and R. Buyya, “Proactive power and thermal aware optimizations for energy-efficient cloud computing”, in *Design Automation and Test in Europe. DATE 2016, Dresden, Germany*, Ph.D Forum, Mar. 2016.

[32] ——, “Dvfs-aware consolidation for energy-efficient clouds”, in *2015 International Conference on Parallel Architecture and Compilation, PACT 2015, San Francisco, CA, USA, 2015*, 2015, pp. 494–495.

[33] P. Arroba, J. L. Risco-Martín, M. Zapater, J. M. Moya, J. L. Ayala, and K. Olcoz, "Server power modeling for run-time energy optimization of cloud computing facilities", in *2014 International Conference on Sustainability in Energy and Buildings, Cardiff, Wales, UK*, Jun. 2014.

[34] P. Arroba, M. Zapater, J. L. Ayala, J. M. Moya, K. Olcoz, and R. Hermida, "On the Leakage-Power modeling for optimal server operation", in *Innovative architecture for future generation high-performance processors and systems (IWIA 2014), Hawaii, USA*, 2014.

[35] P. Arroba, J. L. Risco-Martín, M. Zapater, J. M. Moya, J. L. Ayala, and K. Olcoz, "Evolutionary power modeling for high-end servers in cloud data centers", in *Mathematical Modelling in Engineering & Human Behaviour, Valencia, Spain*, 2014.

[36] M. T. Higuera-Toledano, J. L. Risco-Martin, P. Arroba, and J. L. Ayala, "Green adaptation of real-time web services for industrial cps within a cloud environment", *IEEE Transactions on Industrial Informatics*, vol. PP, no. 99, pp. 1–1, 2017, ISSN: 1551-3203.

[37] M. Zapater, J. L. Risco-Martín, P. Arroba, J. L. Ayala, J. M. Moya, and R. Hermida, "Runtime data center temperature prediction using grammatical evolution techniques", *Applied Soft Computing*, Aug. 2016, ISSN: 15684946.

[38] I. Aransay, M. Zapater, P. Arroba, and J. M. Moya, "A trust and reputation system for energy optimization in cloud data centers", in *2015 IEEE 8th International Conference on Cloud Computing*, 2015, pp. 138–145.

[39] M. Zapater, P. Arroba, J. L. Ayala, K. Olcoz, and J. M. Moya, "Energy-Aware policies in ubiquitous computing facilities", in *Cloud Computing with e-Science Applications*, CRC Press, Jan. 2015, pp. 267–286, ISBN: 978-1-4665-9115-8.

[40] M. Zapater, P. Arroba, J. L. Ayala, J. M. Moya, and K. Olcoz, "A novel energy-driven computing paradigm for e-health scenarios", *Future Generation Computer Systems*, vol. 34, pp. 138 –154, 2014, Special Section: Distributed Solutions for Ubiquitous Computing and Ambient Intelligence, ISSN: 0167-739X.

[41] J. Pagán, M. Zapater, O. Cubo, P. Arroba, V. Martín, and J. M. Moya, "A Cyber-Physical approach to combined HW-SW monitoring for improving energy efficiency in data centers", in *Conference on Design of Circuits and Integrated Systems*, Nov. 2013.

[42] M. Zapater, P. Arroba, J. M. Moya, and Z. Banković, "A State-of-the-Art on energy efficiency in today's datacentres: researcher's contributions and practical approaches", *UPGRADE*, vol. 12, no. 4, pp. 67–74, 2011, Awarded best paper of the year 2011, ISSN: 1684-5285.

[43] R. G. Dreslinski, M Wieckowski, D Blaauw, D Sylvester, and T Mudge, "Near-threshold computing: reclaiming moore's law through energy efficient integrated circuits", *Proceedings of the IEEE*, vol. 98, no. 2, pp. 253–266, 2010.

[44] D. Bol, R. Ambroise, D. Flandre, and J.-D. Legat, "Impact of technology scaling on digital subthreshold circuits", in *Proceedings of the 2008 IEEE Computer Society Annual Symposium on VLSI*, ser. ISVLSI '08, Washington, DC, USA: IEEE Computer Society, 2008, pp. 179–184, ISBN: 978-0-7695-3170-0.

[45] M. de Kruijf, S. Nomura, and K. Sankaralingam, "A unified model for timing speculation: evaluating the impact of technology scaling, cmos design style, and fault recovery mechanism", in *Dependable Systems and Networks (DSN), 2010 IEEE/IFIP International Conference on*, Jul. 2010, pp. 487 –496.

[46] A. Muttreja, P. Mishra, and N. K. Jha, "Threshold voltage control through multiple supply voltages for power-efficient finfet interconnects", in *Proceedings of the 21st International Conference on VLSI Design*, ser. VLSID '08, Washington, DC, USA: IEEE Computer Society, 2008, pp. 220–227, ISBN: 0-7695-3083-4.

[47] L. Benini and G. De Micheli, “Logic synthesis and verification”, in, S. Hassoun and T. Sasao, Eds., Norwell, MA, USA: Kluwer Academic Publishers, 2002, ch. Logic synthesis for low power, pp. 197–223, ISBN: 0-7923-7606-4.

[48] National Research Council (U.S.). Committee on Electric Power for the Dismounted Soldier, *Energy-Efficient Technologies for the Dismounted Soldier*. National Academy Press, 1997, ISBN: 9780309059343.

[49] M. Annavaram, “A case for guarded power gating for multi-core processors”, in *High Performance Computer Architecture (HPCA), 2011 IEEE 17th International Symposium on*, Feb. 2011, pp. 291 –300.

[50] J. Leverich, M. Monchiero, V. Talwar, P. Ranganathan, and C. Kozyrakis, “Power management of datacenter workloads using per-core power gating”, *IEEE Comput. Archit. Lett.*, vol. 8, no. 2, pp. 48–51, Jul. 2009, ISSN: 1556-6056.

[51] D. Meisner, B. T. Gold, and T. F. Wenisch, “Powernap: eliminating server idle power”, *SIGPLAN Not.*, vol. 44, no. 3, pp. 205–216, Mar. 2009, ISSN: 0362-1340.

[52] M. B. Henry, “Emerging power-gating techniques for low power digital circuits”, PhD thesis, Nov. 2011.

[53] M. Seok, D. Jeon, C. Chakrabarti, D. Blaauw, and D. Sylvester, “Pipeline strategy for improving optimal energy efficiency in ultra-low voltage design”, in *Proceedings of the 48th Design Automation Conference*, ser. DAC ’11, San Diego, California: ACM, 2011, pp. 990–995, ISBN: 978-1-4503-0636-2.

[54] D. Jeon, M. Seok, C. Chakrabarti, D. Blaauw, and D. Sylvester, “A super-pipelined energy efficient subthreshold 240 ms/s fft core in 65 nm cmos”, *Solid-State Circuits, IEEE Journal of*, vol. 47, no. 1, pp. 23 –34, Jan. 2012, ISSN: 0018-9200.

[55] A. R. Brahmbhatt, J. Zhang, Q. Qiu, and Q. Wu, “Adaptive lowpower bus encoding based on weighted code mapping”, in *Proc. of IEEE International Symposium on Circuits and Systems*, 2006.

[56] J.-s. Seo, D. Sylvester, D. Blaauw, H. Kaul, and R. Krishnamurthy, “A robust edge encoding technique for energy-efficient multi-cycle interconnect”, in *Proceedings of the 2007 international symposium on Low power electronics and design*, ser. ISLPED ’07, Portland, OR, USA: ACM, 2007, pp. 68–73, ISBN: 978-1-59593-709-4.

[57] E. Pakbaznia, F. Fallah, and M. Pedram, “Charge recycling in mtcmos circuits: concept and analysis”, in *Proceedings of the 43rd annual Design Automation Conference*, ser. DAC ’06, San Francisco, CA, USA: ACM, 2006, pp. 97–102, ISBN: 1-59593-381-6.

[58] E. Le Sueur and G. Heiser, “Dynamic voltage and frequency scaling: the laws of diminishing returns”, in *Proceedings of the 2010 international conference on Power aware computing and systems*, ser. HotPower’10, Vancouver, BC, Canada: USENIX Association, 2010, pp. 1–8.

[59] Q. Deng, D. Meisner, A. Bhattacharjee, T. F. Wenisch, and R. Bianchini, “Multiscale: memory system dvfs with multiple memory controllers”, in *Proceedings of the 2012 ACM/IEEE international symposium on Low power electronics and design*, ser. ISLPED ’12, Redondo Beach, California, USA: ACM, 2012, pp. 297–302, ISBN: 978-1-4503-1249-3.

[60] S. Lee and J. Kim, “Using dynamic voltage scaling for energy-efficient flash-based storage devices”, in *SoC Design Conference (ISOCC), 2010 International*, Nov. 2010, pp. 63 –66.

[61] J. Heo, D. Henriksson, X. Liu, and T. Abdelzaher, “Integrating adaptive components: an emerging challenge in performance-adaptive systems and a server farm case-study”, in *28th IEEE International Real-Time Systems Symposium (RTSS 2007)*, 2007, pp. 227–238.

[62] K. Swaminathan, E. Kultursay, V. Saripalli, V. Narayanan, M. Kandemir, and S. Datta, "Improving energy efficiency of multi-threaded applications using heterogeneous cmos-tfet multicores", in *Proceedings of the 17th IEEE/ACM international symposium on Low-power electronics and design*, ser. ISLPED '11, Fukuoka, Japan: IEEE Press, 2011, pp. 247–252, ISBN: 978-1-61284-660-6.

[63] J. S. Seng and D. M. Tullsen, "Exploring the potential of architecture-level power optimizations", in *PACS*, 2003, pp. 132–147.

[64] L. Minas and B. Ellison, *Energy Efficiency for Information Technology: How to Reduce Power Consumption in Servers and Data Centers*. Intel Press, 2009, ISBN: 9781934053201.

[65] Q. Zhu, F. M. David, C. F. Devaraj, Z. Li, Y. Zhou, and P. Cao, "Reducing energy consumption of disk storage using power-aware cache management", in *Proceedings of the 10th International Symposium on High Performance Computer Architecture*, ser. HPCA '04, Washington, DC, USA: IEEE Computer Society, 2004, pp. 118–, ISBN: 0-7695-2053-7.

[66] E. V. Carrera, E. Pinheiro, and R. Bianchini, "Conserving disk energy in network servers", in *Proceedings of the 17th annual international conference on Supercomputing*, ser. ICS '03, San Francisco, CA, USA: ACM, 2003, pp. 86–97, ISBN: 1-58113-733-8.

[67] T. M. Jones, S. Bartolini, B. De Bus, J. Cavazos, and M. F. P. O'Boyle, "Instruction cache energy saving through compiler way-placement", in *Proceedings of the conference on Design, automation and test in Europe*, ser. DATE '08, Munich, Germany: ACM, 2008, pp. 1196–1201, ISBN: 978-3-9810801-3-1.

[68] Y. Fei, S. Ravi, A. Raghunathan, and N. Jha, "Energy-optimizing source code transformations for os-driven embedded software", in *VLSI Design, 2004. Proceedings. 17th International Conference on*, 2004, pp. 261–266.

[69] W. T. Shiue, "Energy-efficient backend compiler design for embedded systems", in *IEEE Region 10 International Conference on Electrical and Electronic Technology*. 2001, pp. 103–109.

[70] A. Lewis, S. Ghosh, and N.-F. Tzeng, "Run-time energy consumption estimation based on workload in server systems", in *Proceedings of the 2008 conference on Power aware computing and systems*, ser. HotPower'08, San Diego, California: USENIX Association, 2008, pp. 4–4.

[71] G. Dhiman, K. Mihic, and T. Rosing, "A system for online power prediction in virtualized environments using gaussian mixture models", in *Proceedings of the 47th Design Automation Conference*, ser. DAC '10, Anaheim, California: ACM, 2010, pp. 807–812, ISBN: 978-1-4503-0002-5.

[72] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, "Reducing network energy consumption via sleeping and rate-adaptation", in *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, ser. NSDI'08, San Francisco, California: USENIX Association, 2008, pp. 323–336, ISBN: 111-999-5555-22-1.

[73] S. W. Son and M. Kandemir, "Runtime system support for software-guided disk power management", in *Proceedings of the 2007 IEEE International Conference on Cluster Computing*, ser. CLUSTER '07, Washington, DC, USA: IEEE Computer Society, 2007, pp. 139–148, ISBN: 978-1-4244-1387-4.

[74] M. Curtis-Maury, A. Shah, F. Blagojevic, D. S. Nikolopoulos, B. R. de Supinski, and M. Schulz, "Prediction models for multi-dimensional power-performance optimization on many cores", in *Proceedings of the 17th international conference on Parallel architectures and compilation techniques*, ser. PACT '08, Toronto, Ontario, Canada: ACM, 2008, pp. 250–259, ISBN: 978-1-60558-282-5.

[75] P. Yang, C. Wong, P. Marchal, F. Catthoor, D. Desmet, D. Verkest, and R. Lauwereins, "Energy-aware runtime scheduling for embedded-multiprocessor socs", *IEEE Des. Test*, vol. 18, no. 5, pp. 46–58, Sep. 2001, ISSN: 0740-7475.

[76] Y. Lee and A. Zomaya, “Energy efficient utilization of resources in cloud computing systems”, *The Journal of Supercomputing*, vol. 60, pp. 268–280, 2 2012, ISSN: 0920-8542.

[77] J. Torres, “Middleware research for green data centers”, in *Proceedings of e-InfraNet Workshop on Green and Environmental Computing*, ser. CSC - IT, Center for Science Espoo, Finland, Oct. 2010.

[78] A. Corradi, M. Fanelli, and L. Foschini, “Increasing cloud power efficiency through consolidation techniques”, in *Computers and Communications (ISCC), 2011 IEEE Symposium on*, Jul. 2011, pp. 129–134.

[79] L. Lefèvre and A.-C. Orgerie, “Designing and evaluating an energy efficient cloud”, *J. Supercomput.*, vol. 51, no. 3, pp. 352–373, Mar. 2010, ISSN: 0920-8542.

[80] G. Chen, W. He, J. Liu, S. Nath, L. Rigas, L. Xiao, and F. Zhao, “Energy-aware server provisioning and load dispatching for connection-intensive internet services”, in *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, ser. NSDI’08, San Francisco, California: USENIX Association, 2008, pp. 337–350, ISBN: 111-999-5555-22-1.

[81] D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy, and G. Jiang, “Power and performance management of virtualized computing environments via lookahead control”, in *Proceedings of the 2008 International Conference on Autonomic Computing*, ser. ICAC ’08, Washington, DC, USA: IEEE Computer Society, 2008, pp. 3–12, ISBN: 978-0-7695-3175-5.

[82] A. Beloglazov and R. Buyya, “Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers”, in *Proceedings of the 8th International Workshop on Middleware for Grids, Clouds and e-Science*, ser. MGC ’10, Bangalore, India: ACM, 2010, 4:1–4:6, ISBN: 978-1-4503-0453-5.

[83] S. Niles, *Virtualization: optimized power and cooling to maximize benefits*, White paper, APC by Schneider Electric, Dec. 2010.

[84] L. Barroso and U. Hölzle, *The datacenter as a computer: an introduction to the design of warehouse-scale machines*, ser. Synthesis lectures in computer architecture. Morgan & Claypool, 2009, ISBN: 9781598295566.

[85] Y. Wang and X. Wang, “Power optimization with performance assurance for multi-tier applications in virtualized data centers”, in *Proceedings of the 2010 39th International Conference on Parallel Processing Workshops*, ser. ICPPW ’10, Washington, DC, USA: IEEE Computer Society, 2010, pp. 512–519, ISBN: 978-0-7695-4157-0.

[86] MapReduce.org, *What is MapReduce?*, ”http://www.mapreduce.org/what-is-mapreduce.php”, [Online; accessed 9-March-2012], 2011.

[87] J. Polo, Y. Becerra, D. Carrera, V. Beltran, J. Torres, and E. AyguadÃ©, “Towards energy-efficient management of mapreduce workloads”, in *Poster session. 1st Int. Conf. on Energy-Efficient Computing and Networking*, University of Passau, Germany, Apr. 2010.

[88] T. Wirtz and R. Ge, “Improving mapreduce energy efficiency for computation intensive workloads”, in *Green Computing Conference and Workshops (IGCC), 2011 International*, Jul. 2011, pp. 1 –8.

[89] The Apache Software Foundation, *Welcome to Apache Hadoop!*, ”http://hadoop.apache.org/”, [Online; accessed 9-March-2012], 2012.

[90] J. Leverich and C. Kozyrakis, “On the energy (in)efficiency of hadoop clusters”, *SIGOPS Oper. Syst. Rev.*, vol. 44, no. 1, pp. 61–65, Mar. 2010, ISSN: 0163-5980.

[91] N. Maheshwari, R. Nanduri, and V. Varma, “Dynamic energy efficient data placement and cluster reconfiguration algorithm for mapreduce framework.”, *Future Generation Comp. Syst.*, vol. 28, no. 1, pp. 119–127, 2012.

[92] R. Ge, X. Feng, S. Song, H.-C. Chang, D. Li, and K. W. Cameron, “Powerpack: energy profiling and analysis of high-performance systems and applications”, *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 5, pp. 658–671, May 2010, ISSN: 1045-9219.

[93] A. Beloglazov, J. Abawajy, and R. Buyya, “Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing”, *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755 –768, 2012, ISSN: 0167-739X.

[94] B. Li, J. Li, J. Huai, T. Wo, Q. Li, and L. Zhong, “Enacloud: an energy-saving application live placement approach for cloud computing environments”, in *Proceedings of the 2009 IEEE International Conference on Cloud Computing*, ser. CLOUD ’09, Washington, DC, USA: IEEE Computer Society, 2009, pp. 17–24, ISBN: 978-0-7695-3840-2.

[95] L. Liu, H. Wang, X. Liu, X. Jin, W. B. He, Q. B. Wang, and Y. Chen, “Greencloud: a new architecture for green data center”, in *Proceedings of the 6th international conference industry session on Autonomic computing and communications industry session*, ser. ICAC-INDST ’09, Barcelona, Spain: ACM, 2009, pp. 29–38, ISBN: 978-1-60558-612-0.

[96] S. Mitra, N. Seifert, M. Zhang, Q. Shi, and K. S. Kim, “Robust system design with built-in soft-error resilience”, *Computer*, vol. 38, no. 2, pp. 43–52, Feb. 2005, ISSN: 0018-9162.

[97] J. Lee and A. Shrivastava, “Compiler-managed register file protection for energy-efficient soft error reduction”, in *Proceedings of the 2009 Asia and South Pacific Design Automation Conference*, ser. ASP-DAC ’09, Yokohama, Japan: IEEE Press, 2009, pp. 618–623, ISBN: 978-1-4244-2748-2.

[98] L. Wang, G. von Laszewski, J. Dayal, X. He, A. J. Younge, and T. R. Furlani, “Towards thermal aware workload scheduling in a data center”, in *Proceedings of the 2009 10th International Symposium on Pervasive Systems, Algorithms, and Networks*, ser. ISPAN ’09, Washington, DC, USA: IEEE Computer Society, 2009, pp. 116–122, ISBN: 978-0-7695-3908-9.

[99] T. Mukherjee, A. Banerjee, G. Varsamopoulos, S. K. S. Gupta, and S. Rungta, “Spatio-temporal thermal-aware job scheduling to minimize energy consumption in virtualized heterogeneous data centers”, *Comput. Netw.*, vol. 53, no. 17, pp. 2888–2904, Dec. 2009, ISSN: 1389-1286.

[100] Q. Tang, S. K. S. Gupta, and G. Varsamopoulos, “Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: a cyber-physical approach”, *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, no. 11, pp. 1458–1472, Nov. 2008, ISSN: 1045-9219.

[101] A. Beloglazov and R. Buyya, “Energy efficient resource management in virtualized cloud data centers”, in *Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, ser. CCGRID ’10, Washington, DC, USA: IEEE Computer Society, 2010, pp. 826–831, ISBN: 978-0-7695-4039-9.

[102] R. Ayoub, R. Nath, and T. Rosing, “Jetc: joint energy thermal and cooling management for memory and cpu subsystems in servers”, in *High Performance Computer Architecture (HPCA), 2012 IEEE 18th International Symposium on*, Feb. 2012, pp. 1–12.

[103] C. S. Chan, Y. Jin, Y.-K. Wu, K. Gross, K. Vaidyanathan, and T. i. Rosing, “Fan-speed-aware scheduling of data intensive jobs”, in *Proceedings of the 2012 ACM/IEEE International Symposium on Low Power Electronics and Design*, ser. ISLPED ’12, Redondo Beach, California, USA: ACM, 2012, pp. 409–414, ISBN: 978-1-4503-1249-3.

[104] S. Li, T. Abdelzaher, and M. Yuan, “Tapa: temperature aware power allocation in data center with map-reduce”, in *Green Computing Conference and Workshops (IGCC), 2011 International*, Jul. 2011, pp. 1 –8.

[105] Z. Abbasi, G. Varsamopoulos, and S. K. S. Gupta, “Thermal aware server provisioning and workload distribution for internet data centers”, in *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, ser. HPDC ’10, Chicago, Illinois: ACM, 2010, pp. 130–141, ISBN: 978-1-60558-942-8.

[106] K. Ye, D. Huang, X. Jiang, H. Chen, and S. Wu, “Virtual machine based energy-efficient data center architecture for cloud computing: a performance perspective”, in *Proceedings of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing*, ser. GREENCOM-CPSCOM '10, Washington, DC, USA: IEEE Computer Society, 2010, pp. 171–178, ISBN: 978-0-7695-4331-4.

[107] Y. Bar-Yam, *Dynamics of Complex Systems*, ser. Addison-Wesley stydies in nonlinearity. Westview Press, 1997, ISBN: 9780813341217.

[108] N. El-Sayed, I. A. Stefanovici, G. Amvrosiadis, A. A. Hwang, and B. Schroeder, “Temperature management in data centers: why some (might) like it hot”, in *Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE joint international conference on Measurement and Modeling of Computer Systems*, ser. SIGMETRICS '12, New York, NY, USA: ACM, 2012, pp. 163–174.

[109] J. Brandon, “Going green in the data center: practical steps for your SME to become more environmentally friendly”, *Processor*, no. 29, 2007.

[110] R. Miller, *Google: raise your data center temperature.* `http://www.datacenterknowledge.com/archives/2008/10/14/google-raise-your-data-center-temperature/`, 2008.

[111] *Summer time energy-saving tips.*

[112] S. Narendra and A. Chandrakasan, *Leakage in Nanometer CMOS Technologies*, ser. Integrated Circuits and Systems. Springer, 2010, ISBN: 9781441938268.

[113] Daikin AC (Americas), Inc., *Engineering data split, ftxs-l series*, 2010.

[114] SPEC CPU Subcommittee and John L. Henning, *SPEC CPU 2006 benchmark descriptions*, `http://www.spec.org/cpu2006/`.

[115] N. Adiga and et al, “An Overview of the BlueGene/L Supercomputer”, in *Supercomputing, ACM/IEEE 2002 Conference*, 2002, pp. 60–60.

[116] M. Warren and et al., “High-density computing: a 240-processor beowulf in one cubic meter”, in *Supercomputing Conference*, 2002, pp. 61–61.

[117] R. Ge and et al., “Performance-constrained distributed dvs scheduling for scientific applications on power-aware clusters”, in *Supercomputing Conference*, ser. SC '05, Washington, DC, USA: IEEE Computer Society, 2005, pp. 34–34.

[118] C.-H. Hsu and W.-C. Feng, “A power-aware run-time system for high-performance computing”, in *Supercomputing Conference*, 2005, pp. 1–1.

[119] G. Contreras and M. Martonosi, “Power prediction for intel xscale processors using performance monitoring unit events”, in *ISLPED*, San Diego, CA, USA, 2005, pp. 221–226, ISBN: 1-59593-137-6.

[120] A. Lewis and et al., “Run-time energy consumption estimation based on workload in server systems”, in *HotPower*, San Diego, California, 2008, pp. 4–4.

[121] S. Pelley and et al., “Understanding and abstracting total data center power”, in *WEED*, Jun. 2009.

[122] F. Bellosa, “The benefits of event: driven energy accounting in power-sensitive systems”, in *ACM SIGOPS*, Kolding, Denmark, 2000, pp. 37–42.

[123] X. Fan and et al., “Power provisioning for a warehouse-sized computer”, in *ISCA*, San Diego, California, USA, 2007, pp. 13–23.

[124] D. Meisner and et al., “Peak power modeling for data center servers with switched-mode power supplies”, in *ISLPED*, Austin, Texas, USA, 2010, pp. 319–324.

[125] G. Warkozek and et al., “A new approach to model energy consumption of servers in data centers”, in *ICIT*, 2012, pp. 211–216.

[126] A. Bohra and V. Chaudhary, “Vmeter: power modelling for virtualized clouds”, in *IPDPSW*, 2010, pp. 1–8.

[127] Google Data Centers, *Efficiency: How we do it. Temperature control*, `http://www.google.com/intl/en_ALL/about/datacenters/efficiency/internal/#temperature`, Jan. 2014.

[128] N. El-Sayed, I. A. Stefanovici, G. Amvrosiadis, A. A. Hwang, and B. Schroeder, “Temperature management in data centers: why some (might) like it hot”, *SIGMETRICS Perform. Eval. Rev.*, vol. 40, no. 1, pp. 163–174, Jun. 2012.

[129] N. Boccara, *Modeling Complex Systems*, ser. Graduate Texts in Physics. Springer, 2010, ISBN: 9781441965622.

[130] J. C. Salinas-Hilburg, M. Zapater, J. L. Risco-MartÃn, J. M. Moya, and J. L. Ayala, “Unsupervised power modeling of co-allocated workloads for energy efficiency in data centers”, in *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*, Mar. 2016, pp. 1345–1350.

[131] J. L. Henning, “Spec cpu2006 benchmark descriptions”, *SIGARCH Comput. Archit. News*, vol. 34, no. 4, pp. 1–17, Sep. 2006, ISSN: 0163-5964.

[132] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, “Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms”, *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, Jan. 2011, ISSN: 0038-0644.

[133] M. Zapater, J. L. Ayala, J. M. Moya, K. Vaidyanathan, K. Gross, and A. K. Coskun, “Leakage and temperature aware server control for improving energy efficiency in data centers”, in *Proceedings of the Conference on Design, Automation and Test in Europe*, ser. DATE ’13, Grenoble, France: EDA Consortium, 2013, pp. 266–269, ISBN: 978-1-4503-2153-2.

[134] A. L. Anthony and H. K. Watson, “Techniques for developing analytic models.”, *IBM Systems Journal*, vol. 11, no. 4, pp. 316–328, 1972.

[135] L. Bianchi, M. Dorigo, L. M. Gambardella, and W. J. Gutjahr, “A survey on metaheuristics for stochastic combinatorial optimization”, *Natural Computing: An international journal*, vol. 8, no. 2, pp. 239–287, Jun. 2009, ISSN: 1567-7818.

[136] C. Blum and A. Roli, “Metaheuristics in combinatorial optimization: overview and conceptual comparison”, *ACM Comput. Surv.*, vol. 35, no. 3, pp. 268–308, Sep. 2003, ISSN: 0360-0300.

[137] M. Bao, A. Andrei, P. Eles, and Z. Peng, “Temperature-aware idle time distribution for leakage energy optimization”, *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 20, no. 7, pp. 1187–1200, 2012, ISSN: 1063-8210.

[138] C. Reyes-Sierra and C. A. C. Coello, “Multi-objective particle swarm optimizers: a survey of the state-of-the-art”, *International Journal of Computational Intelligence Research*, vol. 2, pp. 287–308, 2006.

[139] R. Poli, J. Kennedy, and T. Blackwell, “Particle swarm optimization”, *Swarm Intelligence*, vol. 1, no. 1, pp. 33–57, 2007.

[140] M. R. Sierra and C. A. Coello Coello, “Improving pso-based multi-objective optimization using crowding, mutation and dominance”, in *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization*, ser. EMO’05, Guanajuato, Mexico: Springer-Verlag, 2005, pp. 505–519, ISBN: 3-540-24983-4, 978-3-540-24983-2.

[141] C. R. Turner, A. Fuggetta, L. Lavazza, and A. L. Wolf, “A conceptual basis for feature engineering”, *Journal of Systems and Software*, vol. 49, no. 1, pp. 3 –15, 1999, ISSN: 0164-1212.

[142] C. Ryan, J. Collins, and M. Neill, “Grammatical evolution: evolving programs for an arbitrary language”, in *Genetic Programming*, ser. Lecture Notes in Computer Science, W. Banzhaf, R. Poli, M. Schoenauer, and T. Fogarty, Eds., vol. 1391, Springer Berlin Heidelberg, 1998, pp. 83–96, ISBN: 978-3-540-64360-9.

[143] E. Vladislavleva, G. Smits, and D. den Hertog, “Order of nonlinearity as a complexity measure for models generated by symbolic regression via pareto genetic programming”, *Evolutionary Computation, IEEE Transactions on*, vol. 13, no. 2, pp. 333–349, Apr. 2009, ISSN: 1089-778X.

[144] M. O’Neill and C. Ryan, “Grammatical evolution”, *Evolutionary Computation, IEEE Transactions on*, vol. 5, no. 4, pp. 349–358, Aug. 2001, ISSN: 1089-778X.

[145] T. Back, U. Hammel, and H.-P. Schwefel, “Evolutionary computation: comments on the history and current state”, *Evolutionary Computation, IEEE Transactions on*, vol. 1, no. 1, pp. 3–17, Apr. 1997, ISSN: 1089-778X.

[146] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional, 1989.

[147] E. Hemberg, L. Ho, M. O’Neill, and H. Claussen, “A comparison of grammatical genetic programming grammars for controlling femtocell network coverage”, English, *Genetic Programming and Evolvable Machines*, vol. 14, no. 1, pp. 65–93, 2013, ISSN: 1389-2576.

[148] R. Tibshirani, “Regression shrinkage and selection via the lasso”, *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996, ISSN: 00359246.

[149] A. Gandhi, M. Harchol-Balter, R. Das, and C. Lefurgy, “Optimal power allocation in server farms”, *ACM SIGMETRICS Performance Evaluation Review*, vol. 37, no. 1, pp. 157–168, Jun. 2009, ISSN: 0163-5999.

[150] M. M. Rafique, N. Ravi, S. Cadambi, A. R. Butt, and S. Chakradhar, “Power management for heterogeneous clusters: an experimental study”, in *Proceedings of the 2011 International Green Computing Conference and Workshops*, ser. IGCC ’11, Washington, DC, USA: IEEE Computer Society, 2011, pp. 1–8, ISBN: 978-1-4577-1222-7.

[151] C.-M. Wu, R.-S. Chang, and H.-Y. Chan, “A green energy-efficient scheduling algorithm using the dvfs technique for cloud datacenters”, *Future Generation Computer Systems*, vol. 37, no. 0, pp. 141 –147, 2014, ISSN: 0167-739X.

[152] L. Wang, G. von Laszewski, J. Dayal, and F. Wang, “Towards energy aware scheduling for precedence constrained parallel tasks in a cluster with dvfs”, in *Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, ser. CCGRID ’10, May 2010, pp. 368–377.

[153] P. Huang, P. Kumar, G. Giannopoulou, and L. Thiele, “Energy efficient dvfs scheduling for mixed-criticality systems”, in *Proceedings of the 14th International Conference on Embedded Software*, ser. EMSOFT ’14, New Delhi, India: ACM, 2014, 11:1–11:10, ISBN: 978-1-4503-3052-7.

[154] N. Rizvandi, J. Taheri, A. Zomaya, and Y. C. Lee, “Linear combinations of dvfs-enabled processor frequencies to modify the energy-aware scheduling algorithms”, in *Proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid2010)*, May 2010, pp. 388–397.

[155] R. Buyya, A. Beloglazov, and J. Abawajy, “Energy-efficient management of data center resources for cloud computing: a vision, architectural elements, and open challenges”, in *Proceedings of the 2010 International Conference on Parallel and Distributed Processing Techniques and Applications, Las Vegas, USA*, ser. PDPTA 2010, CSREA Press, Jul. 2010.

[156] M. Ferdaus, M. Murshed, R. Calheiros, and R. Buyya, "Virtual machine consolidation in cloud data centers using aco metaheuristic", English, in *Proceedings of the 20th International European Conference on Parallel Processing (Euro-Par'14)*, ser. Lecture Notes in Computer Science, F. Silva, I. Dutra, and V. Santos Costa, Eds., vol. 8632, Springer, 2014, pp. 306–317, ISBN: 978-3-319-09872-2.

[157] F. Hermenier, X. Lorca, J.-M. Menaud, G. Muller, and J. Lawall, "Entropy: a consolidation manager for clusters", in *Proceedings of the 2009 ACM SIGPLAN/SIGOPS VEE'09*, Washington, DC, USA: ACM, 2009, pp. 41–50, ISBN: 978-1-60558-375-4.

[158] H. Liu, C.-Z. Xu, H. Jin, J. Gong, and X. Liao, "Performance and energy modeling for live migration of virtual machines", in *Proceedings of the 20th International Symposium on High Performance Distributed Computing*, ser. HPDC '11, San Jose, California, USA: ACM, 2011, pp. 171–182, ISBN: 978-1-4503-0552-5.

[159] V. De Maio, R. Prodan, S. Benedict, and G. Kecskemeti, "Modelling energy consumption of network transfers and virtual machine migration", *Future Generation Computer Systems*, vol. 56, pp. 388 –406, 2016, ISSN: 0167-739X.

[160] Y. Wang and X. Wang, "Performance-controlled server consolidation for virtualized data centers with multi-tier applications", *Sustainable Computing: Informatics and Systems*, vol. 4, no. 1, pp. 52 –65, 2014, ISSN: 2210-5379.

[161] V. Petrucci, O. Loques, and D. Mossé, "A dynamic optimization model for power and performance management of virtualized clusters", in *Proceedings of the 1st International Conference e-Energy '10*, Passau, Germany: ACM, 2010, pp. 225–233, ISBN: 978-1-4503-0042-1.

[162] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers", *Concurrency and Computation: Practice & Experience*, vol. 24, no. 13, pp. 1397–1420, Sep. 2012, ISSN: 1532-0626.

[163] Y. Minyi, "A simple proof of the inequality FFD(L) < 11/9 OPT (L) + 1, for all L for the FFD bin-packing algorithm ", *Acta Mathematicae Applicatae Sinica (English Series)*, vol. 7, no. 4, pp. 321 –331, 1991.

[164] S. Frey, *Conformance Checking and Simulation-based Evolutionary Optimization for Deployment and Reconfiguration of Software in the Cloud*. Books on Demand GmbH University of Kiel, 2014, pp. 0–636, ISBN: 9783735715357.

[165] K. Park and V. S. Pai, "Comon: a mostly-scalable monitoring system for planetlab", *ACM SIGOPS Operating Systems Review*, vol. 40, no. 1, pp. 65–74, Jan. 2006, ISSN: 0163-5980.

[166] S. Shen, V. van Beek, and A. Iosup, "Statistical characterization of business-critical workloads hosted in cloud datacenters", in *15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGrid'15, Shenzhen, China, 2015*, 2015, pp. 465–474.

[167] J. Moore, J. Chase, P. Ranganathan, and R. Sharma, "Making scheduling "cool": temperature-aware workload placement in data centers", in *Proceedings of the annual conference on USENIX Annual Technical Conference*, ser. ATEC '05, Anaheim, CA: USENIX Association, 2005, pp. 5–5.

Acronyms

Acronyms

ArTeCS Architecture and Technology of Computing Systems

ASHRAE American Society of Heating Refrigerating and Air-Conditioning Engineers

BFD Best Fit Decreasing

BNF Backus Naur Form

CDTI Centro para el Desarrollo Tecnológico e Industrial

CFD Computational Fluid Dynamics

CLOUDS Cloud Computing and Distributed Systems

CMOS Complementary Metal-Oxide-Semiconductor

COP Coefficient of Performance

CPS Cyber Physical System

CPU Central Processing Unit

CRAC computer room air conditioning

CV Coefficient of Variation

DIBL Drain-Induced barrier lowering

DVFS Dynamic Voltage and Frequency Scaling

EM Electromigration

FE Feature Engineering

FinFETs fin-type field-effect transistors

GA Genetic Algorithm

GE Grammatical Evolution

GP Genetic Programming

HCI Hot carrier injection

HERO HEuRistic Optimization

HiPEAC European Network of Excellence on High Performance and Embedded Architecture and Compilation

HPC High Performance Computing

IaaS Infrastructure as a Service

IPMI Intelligent Platform Management Interface

IQR Interquartile Range

ISNs index serving nodes

IT Information Technology

KVM Kernel-based Virtual Machine

lasso least absolute shrinkage and selection operator

LR Local Regression

LRR Local Regression Robust

LTS Long Term Support

MAD Median Absolute Deviation

MC Maximum correlation

MINECO Spanish Ministry of Economy and Competitiveness

MIPS Millions of Instructions Per Second

MMT Minimum migration time

MO Multi-Objective

MOS Metal-Oxide-Semiconductor

MOSFET Metal-Oxide-Semiconductor Field-Effect Transistor

MSE Mean Squared Error

NBTI Negative bias temperature instability

OMOPSO Multi-Objective Particle Swarm Optimization

OS Operating System

PABFD Power Aware Best Fit Decreasing

PCPG Per-core power gating

PDUs Power Distribution Units

POF Pareto-Optimal Front

POS Pareto-Optimal Set

PSO Particle Swarm Optimization

PUE Power Usage Effectiveness

QEMU Quick Emulator

QoS Quality of Service

RMSD Root Mean Square Deviation

ROI Return on Investment

RPM revolutions per minute

RS Random choice

SA Simulated Annealing

SEU Single event upset

SLA Service Level Agreement

SM Stress migration

SO Single-Objective

SPEC Standard Performance Evaluation Corporation

SR Symbolic Regression

TASA Thermal Aware Scheduling Algorithm

TC Thermal cycling

TCMS control through multiple supply voltages

TCO Total Cost of Ownership

TDDB Time-dependent dielectric-breakdown

TFET Tunnel field-effect transistor

THR Static threshold

UPS Uninterrupted Power Supply

VM Virtual Machines

"I am your father."

— Darth Vader, *The Empire Strikes Back. Star Wars*