Chapter 1
Cloud Computing and Internet of Things: Recent Trends and Directions

Mohammad Goudarzi, Shashikant Ilager, and Rajkumar Buyya

Contents

1.1 Introduction

The twenty-first-century digital infrastructure and applications are driven by Cloud and Internet of Things (IoT) technologies. Applications built using IoT and Cloud computing technologies have become ubiquitous and influence our modern digital

---

M. Goudarzi · R. Buyya (✉)
CLOUDS Lab, School of Computing and Information Systems, The University of Melbourne, Parkville, VIC, Australia
e-mail: rbuyya@unimelb.edu.au

S. Ilager
Technische Universität (TU) Wien, Vienna, Austria

R. Buyya et al. (eds.), *New Frontiers in Cloud Computing and Internet of Things*,
Internet of Things, https://doi.org/10.1007/978-3-031-05528-7_1

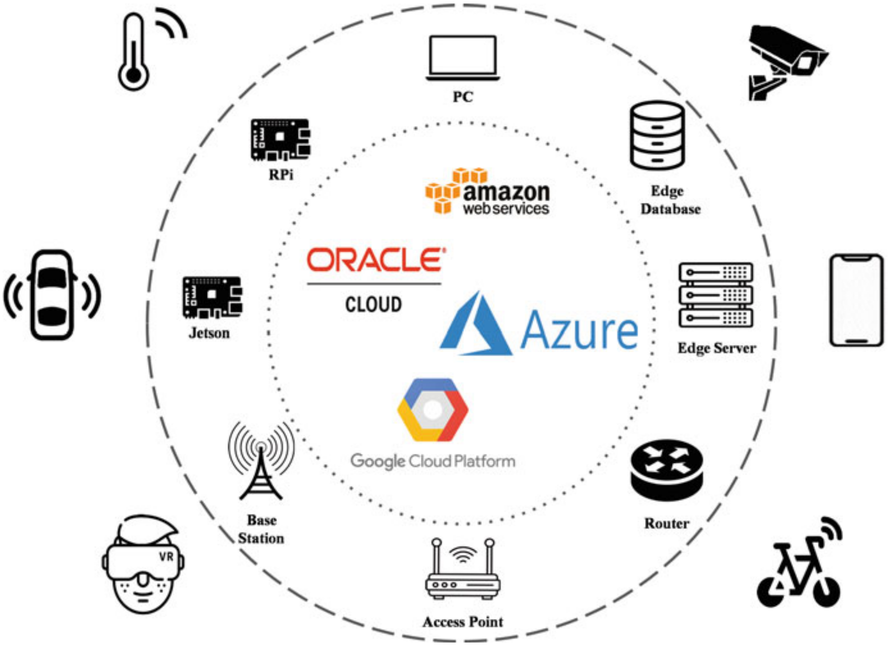

Fig. 1.1 A high-level view of contemporary distributed system involving IoT, Cloud and Edge/Fog components enabling various applications

society in every aspect. The advancement in high-speed networking technologies, pervasive computing devices, and IoT-based sensor networks has resulted in unprecedented growth in data generation. Cloud computing and emerging paradigms such as Edge/Fog computing provide infrastructures and tools required to store, process, and analyze this massive amount of data.

The development in computing and networking technologies over the last many decades are attributed to underlying technologies of Cloud computing and IoT. Cloud and Edge/Fog computing enable IoT-based applications such as smart cities, emergency healthcare applications, and autonomous vehicles, as illustrated in Fig. 1.1. Therefore, a fundamental understanding of technologies associated with Cloud and IoT and the capabilities and limitations are crucial. Therefore, in this chapter, we discuss Cloud and IoT technologies and open challenges associated with them.

1.1.1 Cloud Computing

Cloud computing has seen tremendous growth in recent years. The transition from ownership-based on-premises IT Infrastructure to subscription-based Cloud has changed the way computing services are delivered to end-users. Cloud computing's fundamental principle is to provide computing resources as utility services (similar to water and electricity) [1]. Based on actual resource usage, it offers on-demand access to elastic resources with a pay-as-you-go model. This unique and flexible service delivery model ensures that individuals and businesses can easily access required computing services.

Cloud computing services are traditionally categorized into three types. First, the Infrastructure as a Service (IaaS) model offers computing, storage, and networking resources either in the virtual or physical form. Second, the Platform as a Service (PaaS) model offers tools for rapid application development and deployment such as middleware platforms, Application Programming Interfaces (APIs), and Software Development Kits (SDKs). Finally, the Software as a Service (SaaS) model offers direct access to application software to the users, and the Software is developed and managed by service providers completely. In addition to these, recently new service categories have been added in the Cloud computing paradigm such as Container as a Service (CaaS) and Function as a Service (FaaS) to match the technology advancement and modern application requirements.

The rapid growth in digital services, Internet of Things (IoT), Industry 4.0, and 5G-based application scenarios are creating a massive demand for Cloud services. According to Gartner, by 2022, 60% of organizations will use external Cloud service providers [2], and by 2024, Cloud computing alone accounts for 14.2% of total global IT spending [3]. Clouds have become application's back-end infrastructures for essential resources (compute, storage, and network) for these modern IT services. Along with remote Clouds, recently, Cloud services have been delivered from the edge of the network to satisfy Quality of Service (QoS) requirements for latency-sensitive applications such as autonomous vehicles, emergency healthcare services [4]. Cloud computing uses massive network-based infrastructures, mainly Data Centers (DCs), the core and backbone infrastructure in this networked system. By default, Cloud workloads are always-on, required to provide 24 × 7 access to deployed services. For instance, the Google search engine is expected to achieve an almost 100% availability rate [5]. Similarly, Amazon Web Services (AWS) delivers thousands of Elastic Compute (EC2) instances created in a day through its automated APIs [6].

To cater to Cloud services' demand, major Cloud service providers such as AWS, Microsoft Azure, Google Cloud, Oracle Cloud, and IBM Clouds are deploying many hyper-scale DCs in multiple regions worldwide. Many Cloud service providers are increasingly adopting multi-cloud solutions to increase the reliability and availability of Cloud services they offer. DCs have seen huge growth both in number and size. There are over eight million DCs globally, from private small-scale to hyper-scale DCs, and they are estimated to grow rapidly at 12% annually [7]. As their numbers and size grow, it creates new challenges in managing security and providing uninterrupted power to these Cloud DCs [8].

The emergence of the Internet of Things (IoT), diverse mobile applications, smart grids, innovative industries, and smart cities has resulted in massive data generation. Thus, increasing the demand for computing resources to process this data and derive valuable insights for users and businesses [9, 10]. Besides, new application and execution models like microservices and serverless or FaaS computing [11] are becoming mainstream, significantly reducing the complexities in the design and deployment of software components. On the other hand, this increased connectivity and heterogeneous workloads demand distinct QoS levels to satisfy their application requirements [12, 13]. These developments have led to the building of hyper-scale DCs and complex multi-tier computing infrastructures.

While the fundamental infrastructure capabilities and the number of Cloud services offered by providers are rapidly increasing, new challenges are also emerging that should be addressed diligently for the continued success of Cloud computing technology. Recent advancements have also witnessed Cloud-like services extending to the edge of the network and embedding computing services within network infrastructures, namely, Edge/Fog Computing [14]. The IoT-based latency-sensitive applications mainly drive this need for computing services at the edge of the network since remote Clouds have significant latency due to their geographical distance to the remote Cloud DCs. These new paradigms significantly benefit a few special classes of applications, especially supporting latency-sensitive IoT-based applications. However, we should simultaneously address issues with both Edge/Fog and Cloud infrastructures, including increasing infrastructure efficiency, avoiding resource wastage, reducing power consumption, new pricing and economic models, and managing security.

1.1.2 Internet of Things (IoT)

The Internet of Things (IoT) has become an integral basis of the digital world, thanks to the continuous development of super-cheap and tiny-sized computer chips and ubiquitous access to the Internet. In IoT, “Things” refers to any entities (e.g., smart devices, sensors, human beings) that are context-aware and able to communicate with other entities without any temporal and spatial constraints [15]. Small devices and sensors act as distributed data aggregators with Internet access that forward collected data to the larger computer platforms for processing and permanent storage [16]. This low-cost and distributed paradigm draws a promising future and provides a great opportunity for developers and businesses to form/transition their functionalities to smarter ones.

IoT applications span almost all vital aspects of today’s life, such as healthcare, security, entertainment, transportation, and industrial systems [15, 17, 18]. According to the report from Cisco [19], Norton [20], and Business Insider [21], 15 billion IoT devices will be connected to the Internet by 2023, 21 billion by 2025, and 41 billion by 2027, which will create substantial economic opportunities. Bain & Company [22] estimates the size of the IoT market in 2021 (including hardware, software, systems integrations, and data services) to be around 520 billion U.S. dollars, while Statista [23] and Business Insider [21] expect the size of IoT market will reach to 1 trillion U.S. dollars by 2022 and over 2 trillion U.S. dollars by 2027, accordingly. Among different categories of IoT applications, the fastest growing one is expected to be the connected vehicles (e.g., navigation, diagnostics) by roughly one-third of market size, according to Cisco [19] and Statista [23]. Also, the second major category is expected to be smart cities (e.g., smart homes) [19].

Considering the ever-increasing number of IoT devices and IoT applications, a tremendous amount of data has been being generated by IoT devices. The statistics depict that 18.3 ZB of data was produced by IoT devices in 2019 while International Data Corporation (IDC) [24] predicts about a 400% increase in upcoming years, which hits 73 ZB of data by 2025. The real power of IoT resides in collecting and analyzing the data circulating in the environment [15]. The processing, storage, and transmission of this gigantic amount of data require special considerations. Inevitably, the Big Data mechanisms and techniques (e.g., data acquisition, filtering, transmission, and analysis) should be adapted and deployed to meet the requirements of the IoT.

Cloud computing is one of the main enablers of IoT that offers on-demand services to process, analyze, and store the data generated from IoT devices in a simplified, scalable, and affordable manner [15, 25]. Recent advances in Cloud computing services such as serverless platforms, FaaS, transactional databases, Machine Learning (ML), and autonomous data warehouses open new horizons in the field of data acquisition and analysis of IoT data [26]. Besides, IoT businesses and companies that deploy their applications and systems on the Cloud can reduce their expenses (e.g., infrastructural and operational costs), which leads to more affordable services and products for the end-users. To illustrate, let's consider the IoT in the healthcare industry. According to Help Net Security [27], the healthcare industry experienced a 14.5% growth in IoT spending in 2020. Also, the healthcare industry is expected to remain as one of the major categories of IoT applications in the future. The e-Health and remote patient monitoring are among the most promising applications of healthcare [26]. These types of applications are directly in touch with the wellness, safety, and convenience of users, especially chronic and elderly patients, that improve their confidence, autonomy, and self-management. Thus, these applications require resources with high availability, scalability, security, and affordability for smooth and efficient execution. Cloud computing satisfies such requirements by providing elastic resources with 24/7 availability and monitoring. Consequently, considering the new advances in Cloud computing and the IoT-enabled healthcare industry, Appinventiv [28] reports that the estimated waiting time of patients has been reduced by 50%, and workforce productivity in the healthcare industry has increased by 57%. In addition, new business models in the healthcare industry have increased by 36%.

Latency-critical and real-time applications (e.g., smart traffic flow control) have necessitated bringing Cloud-like services (e.g., processing, storage) to the edge of the network, called Edge/Fog computing paradigm. In the Edge/Fog computing paradigm, the constituent parts of latency-critical IoT applications can be completely or partially deployed on the Edge/Fog servers, distributed in the vicinity of end-users [14, 29]. Edge/Fog servers are accessible with lower latency and higher data rate compared to Cloud services. Gartner [30] predicts that 75% of enterprise-generated data will be processed at the edge by 2025, which is a big jump from 10% in 2018. However, the resources of Edge/Fog servers are usually limited compared to Cloud resources. Consequently, Edge/Fog computing does not compete with Cloud computing, but they complement each other to satisfy the diverse requirements of heterogeneous IoT applications and systems. According to the Forrester analyst [31], the Edge-Cloud market had a 50% growth in 2020, proving the dominance of applications requiring different levels of QoS.

While new IoT-enabled business models and applications have been exponentially increasing in recent years, new challenges are also emerging that require constant and precise research and analysis to guarantee the continued success of IoT. With the highly heterogeneous nature of IoT devices, applications, and their required QoS level, hybrid computing platforms are required for satisfying such requirements. Hence, different Cloud service providers have started planning for efficient multi-cloud services alongside extending their services to the edge of the network [32]. Besides, big-tech companies are bringing more computational power to the low-level Edge devices (e.g., Nvidia Jetson Platform, and Google Coral Edge TPU) to empower these devices for running more resource-hungry applications and analysis tools. Simultaneously, softwares such as machine-learning-based analysis tools and container orchestration frameworks are being adapted to run on Edge devices. Although these novel advances open new horizons for IoT, such a highly heterogeneous environment requires fast adaptation of resource management techniques and protocols, considering new hardwares and softwares (e.g., scheduling mechanisms, scalability procedures, efficient power management techniques (especially for the Edge devices)), software frameworks and programming models suiting the IoT and Edge/Fog computing environment's characteristics. In addition, open standards, regulatory policies, and pricing mechanisms for collaboration among Edge service providers and privacy-preserving mechanisms, among others.

1.2 Key Cloud Technologies and Services

The advancement in multiple technologies has enabled the realization of Cloud computing, especially virtualization. Virtualization enables service providers to offer customized virtual resources (compute, network, and storage) on a shared physical machine, simultaneously providing an isolation environment for the user applications [33]. This has enabled service providers to reduce resource fragmentation and increase infrastructure utilization to achieve economic sustainability in offering Cloud services. Virtualization, improved network connectivity and speed, and standardized web services and REST APIs have helped Cloud computing become successful. The AWS is the first public Cloud service provider started with offering Virtual Machines (VMs), now the AWS itself has more than thousands of different services in their offering [6]. Moreover, the public Cloud service providers have rapidly increased, including Microsoft Azure, Google Cloud, Oracle Cloud, Alibaba Cloud, and many others, capturing significant Cloud computing market share.

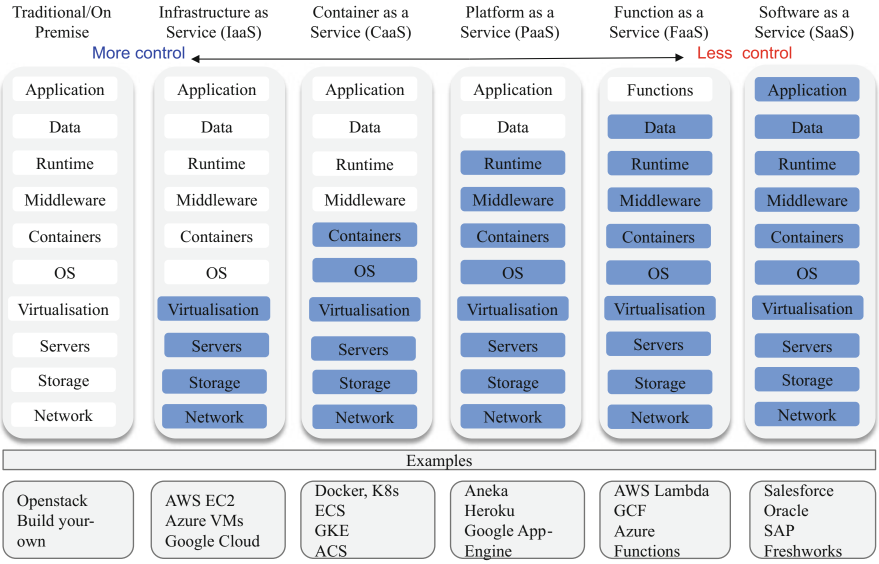

Fig. 1.2 Cloud computing services and their level of manageability

While the main aim of Cloud computing is to provide managed services to users, different services provide different levels of managed services suiting the user and application requirements. Figure 1.2 illustrates the critical services with their respective level of managed services offered by Cloud service providers. In this section, we explore key Cloud technologies categorized into different service models and discuss their characteristics and example technologies associated with them.

1.2.1 Infrastructure as a Service (IaaS)

Infrastructure as a Service (IaaS) provides computing resources (compute, network, and storage) for users. Service providers manage the physical infrastructure in this service while Operating Systems (OSs) to application management are left to users. In addition to this, Cloud service providers offer many tools for monitoring and managing subscribed resources, such as autoscaling, failover management, and cost management. Resource virtualization is a primary underlying technology that enables IaaS by providing customized resources to end-users. Hypervisors allow spin up new VMs based on user requirements. Along with computing virtualization, storage and network resources are also virtualized using storage area network and Network Function Virtualization (NFV) technologies [34].

AWS was the first public Cloud service provider to offer IaaS with their AWS Elastic Compute (EC2) service. As Cloud adoption has gained rapid growth in recent years, Cloud service providers have also increased. Microsoft Azure, Google Cloud, Oracle, IBM, Alibaba, and others provide different IaaS services for end-users.

Due to privacy and policy requirements, many organizations prefer to build their own private Cloud infrastructures. The open-source IaaS frameworks such as OpenStack provide middleware platforms to build private Cloud [35]. It provides virtualization, networking, storage, DNS, and other essential services to create and manage private Cloud infrastructure easily. Moreover, to accommodate growing business needs and elasticity in Cloud infrastructures, it is common to establish a Hybrid Cloud where confidential data and partial application services are hosted on private Cloud and other components of applications are hosted into public Clouds [36, 37]. The public Cloud service providers offer new IaaS tools to quickly create hybrid Cloud environments that seamlessly deploy and manage applications across private and public Clouds. For instance, *Azure arc* can run and manage Azure data services, and *Azure Stack* helps to develop, deploy, and run Cloud-compatible applications across hybrid Cloud infrastructures.

1.2.2 Container as a Service (CaaS)

Containers as a Service (CaaS) aims to provide applications with a virtualized, lightweight, isolated execution environment. Although the VM-based resource allocation provides complete isolation, each VM consumes many resources and increases the boot-up time of new VMs [38, 39]. While VMs enable virtualization of physical machines having their own OSs, the container provides OS-level virtualization and is isolated from one another. A container bundles its software, libraries, dependency packages, and configuration files; they can communicate with each other through well-defined APIs. The container technology has been present for a long time in Linux environments; the recent advancements have made it easier to use and allowed software applications to be broken down into smaller services, known as microservices.

Containers are lightweight compared to VMs since they do not need their own entire OSs. As a result, they have a faster boot-up time compared to VMs. More importantly, the cost of deploying an application with container frameworks significantly reduces since they consume fewer resources compared to VM, and Cloud service providers can serve more users using containers with the same amount of physical resources as opposed to VMs. However, it is essential to note that not all applications would benefit from containers [40]. An enterprise application requiring a significant amount of resources and complete OS functionalities might benefit from VMs. Nevertheless, a large class of applications can be easily containerized and deployed using CaaS offerings from Cloud service providers.

Docker [41] is a popular container management open-source framework, and it implements a Docker engine that enables the composition and creation of containers on Linux environments. Docker also supports Docker swarm for composing and orchestrating multiple containers for application deployment. Kubernetes [42] is another popular open-source framework developed by Google to enable container orchestration in large clusters and provide auto-scaling and application deployment mechanisms. Kubernetes has become a standard container management framework for containerized applications on private and public Cloud clusters.

Many public Cloud service providers offer managed CaaS. Amazon Elastic Container Service (Amazon ECS) is a fully managed container orchestration service that provides a secure, reliable, and scalable way to run containerized application services. Microsoft Azure also provides a fully managed Kubernetes container orchestration service that integrates with Azure Active Directory. It allows building microservice applications and managing containers at scale. Google Kubernetes engine also has managed Kubernetes services. Similarly, many other Cloud providers provide a variety of managed container services for users. This lightweight virtualization technology has enabled rapid application development and deployment at a scale using managed services.

1.2.3 Platform as a Service (PaaS)

The Platform as a Service (PaaS) service model provides frameworks, tools, and SDKs to rapidly build and deploy the applications. In this model, physical infrastructures, Oss, and users leverage the Cloud Application Platforms (CAPs) APIs to deploy their application on the Cloud [1]. Users are relieved from procuring and managing the computing infrastructure. For instance, Heroku enables developers to build, run, and operate applications entirely in the Cloud. Google App engine provides fully managed APIs to deploy applications without scaling and managing the underlying resources.

Aneka [43] is another example of a Platform-as-a-Service framework supporting multiple programming models for the rapid development of applications and their deployment on distributed heterogeneous computing resources. Aneka provides a rich set of .NET-based APIs to developers for exploiting resources transparently available in Cloud infrastructure and expressing the business logic of applications by using the preferred programming abstractions. Moreover, system administrators can leverage a collection of tools to monitor and control the deployed infrastructure. The infrastructure can be built upon a public Cloud available to anyone through the Internet or a private Cloud constituted by a set of nodes with restricted access. More importantly, the Aneka can also be used to set up a hybrid Cloud that includes computing nodes from both private and public Clouds. It supports multiple programming models, and developers can choose a suitable model to build the Cloud-native applications according to the application needs; the programming models include task programming, thread programming, and map-reduce programming models.

The PaaS tools are not only limited to application management; they can also extend their services into data management by providing managed data stores and databases where users can leverage APIs and directly create and manage their data without worrying about storage infrastructures. Similarly, it applies to networking resources where PaaS tools help to create secure and dynamic networks for user applications.

1.2.4 Function as a Service (FaaS)

The emergence of the microservice application paradigm has introduced a new type of FaaS Cloud service model. The FaaS models enable applications and developers to design software as a set of functions that communicate with well-defined APIs [44]. This has led to a new computing paradigm popularly known as serverless computing, an architectural pattern of designing Cloud applications without needing to provision explicit resources by users. For instance, users need not lease VMs to deploy their applications. Instead, each function is ephemeral in this model, and once the request is generated to a particular function, the FaaS platform creates a new containerized function that terminates itself after it executes. Serverless computing and FaaS are often interchangeably used. However, we can say that FaaS implements serverless computing, a paradigm in the Cloud where developers can decompose application business logic into a set of functions that are executed in Linux containers fully managed by a FaaS platform.

In serverless computing, the resources (servers) required to execute an application are fully managed by service providers [45, 46]. It is important to note that the function execution state can be returned to the caller or managed externally by saving the results to a persistent database. The resources are temporarily provisioned for the execution of functions whenever they are invoked, and often these functions are containerized and executed to increase boot-up time and reduce resource usages. Due to their unique architectural pattern, the service provider charges the application execution based on the number of invocations to a function and also the duration of that Function. Since each Function in this paradigm is stateless and ephemeral, service providers have a number of restrictions on function characteristics, including its size and maximum execution duration. Hence, decomposing application into meaningful functions suiting the FaaS platforms is necessary. However, not all applications can be converted or designed based on the FaaS model due to application complexities and limitations of FaaS models. Many issues need to be addressed, such as efficient FaaS middlewares and cold-start bottlenecks of functions [44].

The FaaS model enables Continuous Integration (CI) and Continuous Delivery (CD) in application development where developers can independently update or change the individual Function and deploy in runtime without affecting the entire application stack, reducing engineering cost, and accelerating innovation rate in organizations. This Cloud service model is supported by many Cloud service providers in the market and supports major programming language environments. AWS offers a Lambda service that is tightly integrated with its other Cloud services. Oracle Cloud Functions, Microsoft Azure Functions, and Google Cloud functions also deliver similar services implementing the FaaS model. The open-source platforms such as OpenFaaS, Kubeless, Nucleo, Fission, and IBM's OpenWhisk platforms enable application implementation using the FaaS model and deploy them on private or on-premises infrastructures.

1.2.5 Software as a Service (SaaS)

Software as a Service (SaaS) offers on-demand subscription-based software solutions based on a pay-as-you-go basis from a Cloud service provider. In this service, users remotely access the required services hosted in centralized Cloud DCs, and service providers fully manage the computing stack from hardware to application management [47]. These services are accessed directly from the Internet, either through web applications or mobile applications. The underlying physical infrastructure, middleware platforms, applications software, and data repository are hosted in the Cloud service provider's DC. The Cloud service provider ensures QoS through Service Level Agreements (SLAs) [Citation error]. The SLAs will guarantee the availability and the security of the applications and user data. This service delivery model enables users and organizations to reduce any upfront cost of getting access to required software solutions since all the required elements in the application and computing stack are fully managed by service providers.

All major Cloud service providers, including AWS, Google, Oracle, and Microsoft Azure, offer different SaaS applications. However, the software-centric nature of this service and the ability to leverage infrastructure from existing large Cloud service providers have allowed many small to medium-scale companies to develop and offer specialized SaaS services for users. Generally, SaaS applications can be broadly categorized into two types: Business to Business (B2B) SaaS and Business to Customers (B2C) SaaS.

B2B SaaS: These software systems are broadly developed to serve the business needs of organizations and enterprises. Software systems such as Customer Relation Management (CRMs) and enterprise data management fall into this category. Examples of SaaS service providers include Salesforce, Zoho, Freshworks, and SAP, among many others. For instance, Salesforce is one of the largest SaaS providers that offers CRM and enterprise software focusing on customer service, marketing, and analytics. Similarly, Zoho provides various project management and business automation tools, and Freshworks provides SaaS tools for businesses to support customers through email, phone, website, and social networks. The B2B SaaS applications enable business organizations to adopt digital services rapidly without having expensive in-house technical capabilities to build sophisticated software systems.

B2C SaaS: These software systems are developed to serve the end users' requirements directly. Most major Cloud service providers offer many SaaS applications for users; example services include emails, calendars, offices, social media applications, and personal Cloud storage systems such as Dropbox, Google Drive, and Microsoft OneDrive. Due to the rapid adoption of Cloud technologies from most Internet-based service providers, directly or indirectly, these digital services are part of the SaaS Cloud service delivery model.

1.3 Key IoT Technologies and Applications

In the IoT paradigm, many hardware and software components work closely together to satisfy the requirements of various types of IoT applications. These components are responsible for the contextual perception of the environment as raw data, transferring raw data, management and scheduling of incoming data, processing and storage of raw data, and the interaction with the target environment based on the information (i.e., processed data). According to the data flow, the continuum of the IoT paradigm can be defined as things, communication protocols, frameworks, deployment models, and applications. Figure 1.3 represents an overview of the IoT continuum.

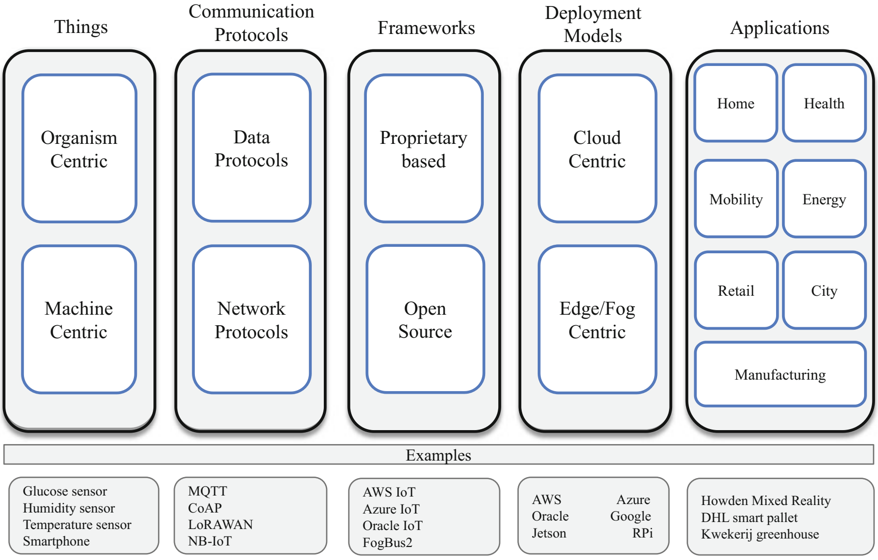

Fig. 1.3 Internet of Things (IoT) technology continuum

1.3.1 Things

There are several definitions for Things in the literature. Here, we consider Things as the most generic form and refer to it as any physical objects/entities that can perceive the change in the environment (i.e., embedded with sensor), has a unique identifier, and embedded with a communication module to communicate with other objects/entities without any constraints [15]. Based on the source of data, Things can be categorized into two different categories: (a) organism-centric and (b) machine-centric.

- *Organism-centric*: This category contains any living organisms (i.e., living plants, animals, and humans) that are either embedded with biochips or interact with some types of sensors while they are the main source of data. Some examples of this category include, but are not limited to, human beings embedded with any implantable sensors (e.g., blood pressure, heart rate), animals with tracking sensors, and the sap flow sensors for plants (used as an indicator of transpiration).
- *Machine-centric*: This category contains any artifacts that are not physically integrated with living organisms, such as smart cars, smart buildings, robots, industrial machines, etc.

Once the Things (and their sensors) are set up and connected, they perceive their target environment and forward the obtained data for processing and storage using the integrated communication modules (wired or wireless). Also, Things may receive feedback to act on their target environment based on the processed data.

1.3.2 Communication Protocols

In the IoT paradigm, communication protocols are a fundamental part of the IoT technology stack, enabling a large number of IoT devices to communicate and interact together. One of the main differentiating factors between an ordinary and smart device is the capability of communication and interaction among smart devices. Hence, a medium (i.e., a common language) is required for each interaction, provided by the IoT communication protocols. Such protocols play a principal role in the evaluation of the cost and specification of IoT applications based on supporting communication features such as quality, credibility, power consumption, connectivity, and communication range. IoT's communication protocols and standards can be broadly classified into (a) IoT data protocols and (b) Network protocols, described in what follows.

1.3.2.1 IoT Data Protocols

Data protocols can provide end-to-end communication among IoT devices, while the connectivity can be obtained using wired or wireless networks. These protocols sit in the presentation and application layers of the standard OSI network architecture. Some examples of the popular IoT data protocols include:

- Message Queuing Telemetry Transport (MQTT): It is a lightweight protocol that works based on the publish-subscribe messaging model and provides a reliable and low power consumption for devices. The MQTT is widely adapted for IoT devices, and it is proved as one of the best choices for wireless networks with occasional bandwidth constraints and unreliable connections. Facebook has used the MQTT for the online chat of Facebook Messenger.
- Constrained Application Protocol (CoAP): It is a specialized web transfer protocol that works based on the request-response messaging model for constrained nodes and networks in the IoT. The CoAP is designed for machine-to-machine applications such as smart energy and building automation.
- Advanced Message Queuing Protocol (AMQP): It is an open standard protocol for business message passing between applications or organizations that work based on the publish-subscribe messaging model. Despite the security and reliability, the AMQP is not widely used in the IoT, especially for devices with limited memory, due to its heaviness. For financial institutions, the AMQP is used for Business to Business (B2B) transactions.
- Data Distribution Service (DDS): It is considered as the first open international middleware standard directly addressing publish-subscribe communications for real-time and embedded systems. The DDS is a scalable protocol that can be used for a diverse range of devices, from Cloud servers to tiny IoT devices.
- Extensible Messaging and Presence Protocol (XMPP): It is a set of open technologies for instant messaging, presence, multi-party chat, voice and video calls, collaboration, lightweight middleware, content syndication, and generalized routing of XML data, that works based on publish-subscribe and request-response messaging models. It supports near real-time and low-latency communication while it does not provide any QoS guarantee.
- Zero Message Queuing (ZMQ): It is a lightweight messaging library that works based on the request-reply and publish-subscribe messaging model. It extends the standard socket interfaces and provides abstractions of asynchronous message queues, message filtering, etc.

While these IoT data protocols are available to use for any IoT system, selecting the suitable protocol in the development phase is a complex design choice that requires a comprehensive understanding of the requirements of the target IoT system and the features provided by each protocol. Table 1.1 summarizes several important features of these data protocols.

Table 1.1 IoT data protocols comparisons

Protocol name	Transport protocol	Messaging model	Security	QoS support	Sample use case
MQTT	TCP	Publish/subscribe	✓ Medium optional	✓ Medium 3 levels	IoT messaging
CoAP	UDP	Request/reply	✓ Medium optional	✓ Medium 2 levels	Utility field
AMQP	TCP	Publish/subscribe	✓ High optional	✓ Medium 3 levels	Enterprise integration
DDS	TCP, UDP	Publish/subscribe	✓ High optional	✓ High 23 levels	Military, big organizations
XMPP	TCP	Publish/subscribe, request/reply	✓ High mandatory	x	Remote management
ZMQ	UDP	Publish/subscribe, request/reply	✓ High optional	✓ Medium 3 levels	Big organization, IoT messaging

1.3.2.2 IoT Network Protocols

The network protocols belong to lower layers of standard OSI network architecture and are typically used over the Internet to connect devices over the network. Some examples of the popular IoT network protocols include:

- ZigBee: It is one of the most effective low-power communication protocols for IoT which provides high scalability and security. It is widely used for home automation products.
- Z-wave: It is an increasingly popular IoT protocol based on low-power radio frequency communication technology. It provides high scalability, durability, and security, while its price is higher than ZigBee. Similar to ZigBee, it is widely used for home automation products.
- Bluetooth: It is one of the most important communication protocols for short-range communications, which is highly suitable for mobile devices. Bluetooth Low Energy (BLE), with reduced power consumption, is a version of Bluetooth which is used by many IoT companies because of its low latency, responsiveness, scalability, and cross-vendor interoperability.
- Long Range Wide Area Networking (LoRAWAN): This protocol is a low-power and noncellular protocol for long-range wireless communication of IoT devices. LoRAWAN is strongly adopted and deployed, specifically for IoT products to be deployed in areas without cellular service. Besides, it is widely used for smart city and industrial use cases.
- Narrow Band IoT (NB-IoT): It is categorized as a 5G technology, specifically designed for low energy consumption, supporting massive connection density, security, and low latency.
- WiFi: It is the most popular wireless communication protocol for Local Area Networks (LAN), which provides easy deployments, scalability, and high speed.

Table 1.2 provides a summary of important features of these IoT network protocols.

Table 1.2 IoT network protocols comparisons

Protocol name	Frequency	Range	Data rate	Power draw	Topology
ZigBee	2.4 GHz (US)	~100 ft	250 Kbps	Low	Mesh
Z-wave	915 MHz (US)	~100 ft	40 Kbps	Low	Mesh
BLE	2.4 GHz	~300 ft	1–2 Mbps	Low	Mesh, P2P
LoRAWAN	150 MHz–1 GHz	~20 miles	50 Kbps	Low	Star
NB-IoT	<1 GHz	~10 miles	100 Kbps	Low	Star
WiFi	2.4, 5 GHz	~230 ft	7 Gbps	High	Star

1.3.3 IoT Frameworks

Considering the ever-increasing popularity of the IoT paradigm with new applications, design, and integration technologies, many IoT frameworks have been developed to satisfy the requirements of different IoT use cases. We consider frameworks as a suite of software that provides a high-level abstraction, including management, security, interoperability, and flexibility of services, systems, and devices [48]. Some of these frameworks are proprietary-based such as AWS IoT and Microsoft Azure IoT, while other frameworks are fully open-source such as Kaa [49], ThingSpeak [50], and FogBus2 [51]. In this section, we study some of these frameworks and briefly discuss some of their key features.

1.3.3.1 Proprietary-Based

The proprietary frameworks work on specific platforms or require a subscription fee to use the partial or complete set of services. Some examples of proprietary-based IoT frameworks include AWS IoT, Microsoft Azure IoT, Oracle IoT, IBM Watson IoT, Siemens MindSphere, SAP Leonardo IoT, Alibaba Cloud IoT, Braincube, and Hitachi Lumada, just to mention a few. These frameworks are mostly Cloud service providers that optimize some/full suite of their services while considering different requirements of IoT devices. To illustrate, AWS IoT provides several Cloud-based services for IoT scenarios. AWS IoT provides software that helps integrate IoT devices into AWS IoT-based solutions. When IoT devices connect to AWS IoT, it connects them to the Cloud services that AWS provides (e.g., analytics, databases, networking).

1.3.3.2 Open Source

The open-source frameworks usually do not depend on specific platforms, and they can be deployed on different hardware configurations. Besides, due to their open-source nature, developers can extend these frameworks based on their requirements. Some examples of open-source IoT frameworks include:

- Kaa IoT [49]: It is one of the highly flexible and multi-purpose frameworks for implementing end-to-end IoT solutions. It enables information exchange among devices, real-time monitoring of devices, Cloud integration, and visualization of the system.
- ThingSpeak [50]: It is an IoT analysis platform that allows aggregation, visualization, and analysis of live IoT data streams in the Cloud and sending monitoring alarms and feedback. It uses the power of MATLAB to make sense of the IoT data.

- Zetta IoT [52]: It is built on Node.js for creating IoT servers that run across distributed computers and the Cloud. It combines REST APIs, WebSockets, and reactive programming, which suits assembling many devices into data-intensive and real-time applications.
- DeviceHive IoT [53]: It is an IoT platform distributed under Apache 2.0 license and provides deployment options for Docker and Kubernetes. It supports Android and iOS and can run batch analysis and ML techniques.
- FogBus2 [51]: It is a containerized framework for the integration of heterogeneous IoT devices with the multi-cloud platform and Edge servers. It supports dynamic resource scheduling, scalability, profiling, resource discovery, multi-database, and blockchain.

1.3.4 Deployment Models

Recent advancements in hardware and software technologies provide new opportunities for the deployment of IoT systems and applications. The majority of IoT systems and applications were previously deployed in a centralized manner using the resources and services provided by Cloud service providers. However, the emergence of the Edge/Fog computing paradigm alongside advancements in communication technologies make the efficient distributed deployment of IoT systems and applications practically possible. Selecting the optimal deployment type for IoT systems and applications is an important design choice, requiring precise analysis of important contributing factors such as the requirements of IoT systems and applications, the availability of resources, scalability, and pricing, just to mention a few.

1.3.4.1 Cloud-Centric IoT

Cloud service providers offer numerous types of on-demand and pay-as-you-go Cloud services with different levels of QoS for their customers. Generally, the resources at the Cloud are richer and more powerful than distributed resources at the edge of the network. Besides, the Cloud service providers offer high availability services with a diverse range of flexible pricing schemes for their customers. Hence, many large organizations and companies integrate their IoT systems with these Cloud service providers and send their resource-hungry computations or streams of data to the Cloud for processing and storage. To illustrate, ML services of AWS and Microsoft Azure are being used by a wide range of startups and large organizations for market prediction, finding target outliers, and processing of large and high-quality video files and streams. Moreover, Oracle Cloud offers a range of database solutions such as the Autonomous Database to enable optimized Big Data analysis on very large datasets. It is important to note that IoT systems can also be integrated with multiple Cloud platforms to become beneficiaries of the optimized suite of services that are provided by individual Cloud service providers.

Alongside the suite of centralized services that the majority of Cloud service providers are currently offering, some Cloud service providers such as AWS and Microsoft Azure started expanding their resources at the edge of the network to offer edge-optimized services. While the complete suite of services is yet to be optimized for use at the edge of the network, it shows the future plans of large Cloud service providers such as AWS and Microsoft Azure to be predominant on the Edge/Fog computing platforms [36].

1.3.4.2 Edge/Fog-Centric IoT

The majority of Cloud service providers offer centralized services on Cloud servers; however, many emerging IoT applications such as online gaming, healthcare, and online transportation systems require significantly low latency and real-time services. Proposed solutions that only rely on Cloud computing for processing and storage are not scalable and cannot satisfy the latency requirements of real-time applications. Besides, forwarding the huge amount of data generated by IoT devices may result in congestion at the Cloud servers and interrupt their services [14, 16].

In Edge/Fog computing paradigm, geographically distributed servers (from backbone gateways and routers to racks of Jetson, RPis, and small servers) are being used for pre-processing, processing, and storage of generated traffic from distributed IoT devices with low latency and high access bandwidth. Not only does this distributed computing paradigm bring about the possibility of new IoT systems and applications, but it also helps re-considering and re-designing previous centralized deployments for the more efficient distributed or hybrid ones. Besides, several edge accelerators such as Google Coral Edge TPU Coprocessor have been introduced that further strengthen the computing power of on-premises Edge resources. Alongside the advancement of hardwares, software companies have been optimizing their libraries and products to be efficiently deployed on the Edge resources. Some examples of these software frameworks are Tensorflow Lite, PyTorch Mobile, and the Nvidia Jetson framework.

Although each Edge server may solely not be sufficient for processing and storage of large amounts of incoming requests, the Edge/Fog deployment model enables federated processing and storage of incoming data on several physically or virtually clustered Edge servers. Moreover, the Edge/Fog deployment model enables the possibility of integration of Edge servers with one or multiple Cloud service providers to further solidify the services it offers and extend the range of services. While the Edge/Fog deployment model provides services with less latency and higher scalability, performing optimized resource management and security mechanisms for its heterogeneous distributed resources is more complex compared to the Cloud-centric deployment model.

1.3.5 Applications

The number of IoT systems and applications has been rapidly increasing due to recent advancements in hardware and software. In this section, we describe the main scopes of IoT applications [9, 12, 15].

- Smart Home: It is one of the most practical applications of IoT to bring higher levels of convenience and security for users. Some examples of smart home applications include metering daily electricity and water usage using sensors, automatic illumination systems, and automatic surveillance systems for homes.
- Healthcare: It is one of the most critical applications of the IoT to improve human wellness. To illustrate, IoT enables real-time data acquisition to monitor the overall condition of patients with critical diseases through implantable or attachable body areal sensors.
- Mobility: It is usually among the largest IoT applications area in recent years. Some IoT-based mobility and transportation applications include fleet management solutions and connected cars.
- Retail: It offers many digitals solutions, including, but not limited to, in-store digital signage, cashier-less payment systems, customer tracking, real-time goods monitoring, and warehouse management.
- Energy: IoT is significantly transforming the energy industry from generation to interaction process with end-users. To illustrate, the companies in this area are providing some real-time analysis on the usage patterns of customers, predicting their periodic usage, and providing their customers with periodic feedback and energy management advice.
- Smart City: There are many projects to combine connected technologies with infrastructural requirements such as smart parking, traffic management, smart waste, public safety, and environmental monitoring.
- Manufacturing: Big technology companies such as Microsoft, AWS, and Siemens are among the pioneers of digital transformation in manufacturing and industrial industries. These applications support a wide range of heterogeneous connected devices inside and outside the companies for remotely controlling the connected machinery and equipment monitoring, just to mention a few.

1.4 Modeling and Simulation Toolkits

The most effective approach to study, analyze, and evaluate the performance of any services, IoT applications, and resource management techniques in real-world deployment is the ability to perform inexpensive and repetitive experiments. The real implementation and building of a prototype system in heterogeneous computing environments (consisting of many servers in tiered or flatted order, different resource management techniques, different services, and IoT applications) is challenging and tedious work. Besides, the scalable experiments in such environments are infeasible since heterogeneous IoT devices and Edge/Fog servers are required for conducting scalability tests. Also, the real deployment with tuning and adjusting the configuration of each device leads to significantly high implementation time. To address these constraints, modeling and simulation environments can be used. Simulation helps design and evaluate customized and scalable experiments while allowing the opportunity to repeat experiments under different settings. In the following, we briefly discuss several modeling and simulation toolkits relevant to Cloud and IoT technologies.

- CloudSim [54]: CloudSim is a widely adopted modeling and simulation toolkit for the evaluation of Cloud computing resource management and application scheduling algorithms. It is based on Java programming and provides tools for modeling data center elements such as virtual machines and physical machines. The ability to extend different application scenarios has made CloudSim a popular toolkit to evaluate resource management algorithms in Cloud computing.
- iCanCloud [55]: It is a C++-based simulator based on OMNET++ [56], which can simulate different physical layer entities. It allows the modeling of hypervisor, virtual machine, and physical machine infrastructure to evaluate the management of Cloud applications.
- ContainerCloudSim [57]: This is an extension of core CloudSim designed to evaluate the performance of scheduling and allocation policies in containerized Cloud DCs. It provides several use cases demonstrating methods using and comparing container scheduling and provisioning policies in terms of energy efficiency and SLA compliance.
- CloudSimSDN [58]: It is another extension of CloudSim designed for modeling and evaluation of SDN-enabled cloud environments. It is a lightweight and scalable simulation environment that allows the evaluation of the network allocation capacity policies.

Along with the above-mentioned important simulation frameworks, there have been many extensions of CloudSim such as CloudAnalyst [59], NetworkCloudSim [60], and CloudSimDisk [61] for different use cases and adding extended functionalities. Similar to Cloud Computing, the following are the important modeling and simulation toolkits for IoT and Edge/Fog computing domains.

- iFogSim [62, 63]: It is a simulation environment to imitate the different scenarios in IoT and Cloud/Fog/Edge computing environments. It is built on top of the CloudSim simulator [54] and extends those features to imitate characteristics of IoT devices and Edge/Fog computing environments. The new release of iFogSim [62] further extends its capabilities by supporting several real and random mobility models, service migration, microservice orchestration techniques, and the dynamic formation of clusters.
- EdgeCloudSim [64]: It is a simulation environment, working based on the CloudSim [54], to conduct experiments that consider both computational and networking resources at the edge of the network. It considers the static deployment and coverage area for the gateway nodes and assumes the link quality between IoT and gateway nodes remains always the same despite their distance.

- FogNetSim++ [65]: A toolkit to simulate a distributed Fog computing environment, working based on OMNeT++ [56]. It can imitate different mobility models for the IoT devices, such as random waypoint and linear mobility while supporting customization of different mobility models.
- MobFogSim [66]: It extends iFogSim to enable modeling of device mobility and service migration in Fog computing. However, the mobility support system of MobFogSim only deals with the IoT gateways and Cloud DCs instead of tiered Edge/Fog infrastructure and limits the scope for creating clusters in Edge/Fog computing environments.
- PureEdgeSim [67]: It is a simulator for Edge/Fog and Mist computing environments. It allows evaluating the performance of resources management strategies in terms of network usage, latency, resources utilization, and energy consumption. Besides, it supports mobility-aware application management while its default policy is complex and difficult to customize. It also has limitations in forming node clusters and augmenting microservice management techniques.
- STEP-ONE [68]: This simulator imitates the operations of fog-based opportunistic network environments, which is working based on the ONE simulator [69] with advanced mobility and messaging interfaces. Although it supports the real-world dataset, it lacks default policies for mobility management, node clustering, and microservice orchestration.
- IoTNetSim [70]: It simulates end-to-end IoT services, their granular details, and networking characteristics. While it supports the mobility of IoT devices, it lacks benchmark policies for mobility-driven service management and dynamic clustering.

1.5 Open Challenges

Distributed Resource Management: Due to the highly distributed nature of the IoT paradigm and Edge/Fog computing, contemporary centralized resource management techniques cannot be efficiently adapted. As a result, lightweight and distributed resource management techniques such as distributed scheduling and orchestration techniques are required to be deployed at the Edge devices to enable low-latency access and reduce the service ready time. As the IoT applications generate different workloads (e.g., streaming and batch), such resource management techniques should be able to manage hybrid workloads.

Privacy-Aware Computing: The emerging concerns about providing confidentiality and privacy for users' data directly affect the design and implementation of digital applications. The application deployment should comply with the new privacy policies such as General Data Protection Regulation (GDPR) and California Consumer Privacy Act (CCPA). Hence, it is requisite to move the computational units where the data resides to comply with such policies.

Security: As the number of connected devices with remote access increases, security challenges become more prevalent. The distributed hardwares are openly exposed to everyone without proper security policies and they can be a potential target for attackers to break the system. Also, the IoT data may transmit over unreliable and insecure networks which are open to attackers. Moreover, there are many vulnerabilities in software frameworks and applications that manage the IoT data while these frameworks and applications are not continuously being updated and patched to address security concerns. Furthermore, while there are many tutorials and solutions to connect IoT devices together, there is a lack of IoT and security skills for those who design and develop IoT systems and applications.

Interoperability: There are obviously many heterogeneous connected devices (from things to servers) in the concept of IoT; however, there are few standards, protocols, or regulatory policies to manage interoperability among a wide variety of connected devices. Defining regulatory policies for networking protocols, authentication, and QoS levels are among the important open challenges for interoperability.

Sustainability in Cloud and Edge/Fog Computing: The increasing demand for computing resources has created a new energy challenge. Cloud DCs consume massive amounts of energy, on the other hand, Edge/Fog infrastructures are often powered through batteries or have limited power supply. Hence, energy-efficient resource management is a challenging task that should be addressed.

Systems for ML: Due to advancements in ML techniques and their rapid adoptions across many business and consumer applications, new demand for specialized hardware resources and software frameworks was created. New systems and software frameworks should be built to support the massive computational requirement of these AI workloads.

ML for Systems: While ML systems themselves are becoming mature and adopted into many critical application domains, it is equally important to use these ML techniques to design and operate large-scale systems. Adopting the ML techniques to solve different resource management problems in Edge/Fog and Cloud is crucial to manage these complex infrastructures and workloads.

1.6 Summary

Cloud and Edge/Fog computing have become pervasive technologies transforming the modern economy and digital society. The utility-based service delivery models have truly realized the potential of delivering computing services with ease of access to users by building planet-scale networked computing infrastructure. Cloud computing has matured and continued to evolve and offer different types of new services, influencing the emergence of new architectural patterns such as Serverless computing. Similarly, IoT-based application scenarios have created an emerging Edge/Fog computing paradigm to serve the requirements of latency-sensitive IoT-based applications. In this chapter, we have explored state-of-the-art Cloud and IoT technologies and provided a detailed discussion on different services and key technologies associated with them. In addition, we have discussed open challenges that should be addressed to make Cloud and IoT technologies robust and secure.

References

1. R. Buyya, C.S. Yeo, S. Venugopal, J. Broberg, I. Brandic, Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Gener. Comput. Syst. **25**, 599–616 (2009)
2. Gartner: Gartner forecasts worldwide public cloud revenue to grow 17 percent in 2020 (2019), https://www.gartner.com/en/newsroom/press-releases/2019-11-13-gartner-forecasts-worldwide-public-cloud-revenue-to-grow-17-percent-in-2020
3. Gartner: Gartner forecasts worldwide public cloud end-user spending to grow 18 percent in 2021 (2020), https://www.gartner.com/en/newsroom/press-releases/2020-11-17-gartner-forecasts-worldwide-public-cloud-end-user-spending-to-grow-18-percent-in-2021
4. M. Satyanarayanan, The emergence of edge computing. Computer **50**, 30–39 (2017)
5. S. Brin, L. Page, The anatomy of a large-scale hypertextual web search engine. Comput. Netw. ISDN Syst. **30**, 107–117 (1998)
6. Amazon: Amazon Web Services, https://aws.amazon.com/
7. S. Maybury: How much energy does your data centre use?, https://www.metronode.com.au/energy_usage/
8. S. Ilager, K. Ramamohanarao, R. Buyya, ETAS: Energy and thermal-aware dynamic virtual machine consolidation in cloud data center with proactive hotspot mitigation. Concurr. Comput. **31**, e5221 (2019)
9. J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, Internet of Things (IoT): A vision, architectural elements, and future directions. Future Gener. Comput. Syst. **29**, 1645–1660 (2013)
10. L. Atzori, A. Iera, G. Morabito, The internet of things: A survey. Comput. Netw. **54**, 2787–2805 (2010)
11. I. Baldini, P. Castro, K. Chang, P. Cheng, S. Fink, V. Ishakian, N. Mitchell, V. Muthusamy, R. Rabbah, A. Slominski, et al., Serverless computing: Current trends and open problems, in *Proceedings of the Research Advances in Cloud Computing*, (Springer, 2017), pp. 1–20
12. A.V. Dastjerdi, R. Buyya, Fog computing: Helping the Internet of Things realize its potential. Computer **49**, 112–116 (2016)
13. Y. Gan, Y. Zhang, D. Cheng, A. Shetty, P. Rathi, N. Katarki, A. Bruno, J. Hu, B. Ritchken, B. Jackson, et al., An open source benchmark suite for microservices and their hardware software implications for cloud and edge systems, in *Proceedings of the Twenty Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, (ACM, 2019), pp. 3–18
14. R. Mahmud, R. Kotagiri, R. Buyya, Fog computing: A taxonomy, survey and future directions, in *Internet of Everything*, (Springer, 2018), pp. 103–130
15. F. Khodadadi, A.V. Dastjerdi, R. Buyya, Internet of things: An overview, in *Internet of Things*, (Morgan Kaufmann, 2016)
16. M. Goudarzi, H. Wu, M. Palaniswami, R. Buyya, An application placement technique for concurrent IoT applications in edge and fog computing environments. IEEE Trans. Mob. Comput. **20**, 1298–1311 (2020)

17. R. Ranjan, O. Rana, S. Nepal, M. Yousif, P. James, Z. Wen, S. Barr, P. Watson, P.P. Jayaraman, D. Georgakopoulos, Others: The next grand challenges: Integrating the Internet of Things and data science. IEEE Cloud Comput. **5**, 12–26 (2018)
18. M. Goudarzi, M. Palaniswami, R. Buyya, A distributed application placement and migration management techniques for edge and fog computing environments, in *Proceedings of the 16th Conference on Computer Science and Information Systems (FedCSIS 2021), Sofia, Bulgaria*, (IEEE Press, 2021)
19. Cisco: Cisco Annual Internet Report (2018–2023) White Paper (2020), https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html. Last accessed 2021/10/07
20. Norton: The future of IoT: 10 predictions about the Internet of Things (2019), https://us.norton.com/internetsecurity-iot-5-predictions-for-the-future-of-iot.html
21. Business Insider: The Internet of Things 2020, https://www.businessinsider.com/internet-of-things-report. Last accessed 2021/10/07
22. Bain & Company: https://www.bain.com/about/media-center/press-releases/2018/bain-predicts-the-iot-market-will-more-than-double-by-2021/. Last accessed 2021/10/07
23. Statista: Internet of Things spending worldwide 2023, https://www.statista.com/statistics/668996/worldwide-expenditures-for-the-internet-of-things/. Last accessed 2021/10/07
24. IDC: IoT growth demands rethink of long-term storage strategies, https://www.idc.com/getdoc.jsp?containerId=prAP46737220. Last accessed 2021/10/07
25. M. Goudarzi, Z. Movahedi, M. Nazari, Mobile cloud computing: A multisite computation offloading, in *Proceedings of the 8th International Symposium on Telecommunications (IST)*, (2016), pp. 660–665. ieeexplore.ieee.org
26. E.G.M. Petrakis, S. Sotiriadis, T. Soultanopoulos, P.T. Renta, R. Buyya, N. Bessis, Internet of Things as a Service (iTaaS): Challenges and solutions for management of sensor data on the cloud and the fog. Internet Things **3–4**, 156–174 (2018)
27. Help Net Security: Global IoT spending dropping significantly in 2020, but expected to rebound in 2021, https://www.helpnetsecurity.com/2020/06/22/2020-iot-spending/. Last accessed 2021/10/07
28. Appinventiv: Understanding the impact of IoT in healthcare, https://appinventiv.com/blog/iot-in-healthcare/. Last accessed 2021/10/07
29. M. Goudarzi, M. Palaniswami, R. Buyya, A fog-driven dynamic resource allocation technique in ultra dense femtocell networks. J. Netw. Comput. Appl. **145**, 102407 (2019)
30. Gartner: What edge computing means for infrastructure and operations leaders, https://www.gartner.com/smarterwithgartner/what-edge-computing-means-for-infrastructure-and-operations-leaders. Last accessed 2021/10/07
31. Forrester: Predictions 2020: Edge computing makes the leap, https://www.forrester.com/blogs/predictions-2020-edge-computing/. Last accessed 2021/10/07
32. M. Goudarzi, Q. Deng, R. Buyya, Resource management in edge and fog computing using FogBus2 framework. arXiv preprint arXiv:2108.00591 (2021)
33. A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica et al., Above the clouds: A Berkeley view of cloud computing. University of California, Berkeley, Rep. UCB/EECS-2009-28 (2009)
34. S. Ilager, R. Muralidhar, R. Buyya, Artificial Intelligence (AI)-centric management of resources in modern distributed computing systems, in *2020 IEEE Cloud Summit*, (IEEE, 2020), pp. 1–10
35. O. Sefraoui, M. Aissaoui, M. Eleuldj, OpenStack: Toward an open-source solution for cloud computing. Int. J. Comput. **55**, 38–43 (2012)
36. Y. Perry, Azure hybrid cloud: Azure in your local data center, https://cloud.netapp.com/blog/azure-cvo-blg-azure-hybrid-cloud-in-your-data-center. Last accessed 2021/10/07
37. A.N. Toosi, R.N. Calheiros, R.K. Thulasiram, R. Buyya, Resource provisioning policies to increase IaaS provider's profit in a federated cloud environment, in *Proceedings of the IEEE International Conference on High Performance Computing and Communications*, (IEEE, 2011), pp. 279–287

38. M.G. Xavier, M.V. Neves, F.D. Rossi, T.C. Ferreto, T. Lange, C.A.F. De Rose, Performance evaluation of container-based virtualization for high performance computing environments, in *2013 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*, (IEEE, 2013), pp. 233–240
39. C. Pahl, A. Brogi, J. Soldani, P. Jamshidi, Cloud container technologies: A state-of-the-art review. IEEE Trans. Cloud Comput. **7**, 677–692 (2019)
40. Z. Li, M. Kihl, Q. Lu, J.A. Andersson, Performance overhead comparison between hypervisor and container based virtualization, in *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*, (IEEE, 2017), pp. 955–962
41. D. Merkel et al., Docker: Lightweight Linux containers for consistent development and deployment. Linux J. **2014**, 2 (2014)
42. D. Bernstein, Containers and cloud: From LXC to Docker to Kubernetes. IEEE Cloud Comput. **1**, 81–84 (2014)
43. C. Vecchiola, X. Chu, R. Buyya, Aneka: A software platform for .NET-based cloud computing. High Speed Large Scale Sci. Comput. **18**, 267–295 (2009)
44. M. Shahrad, J. Balkind, D. Wentzlaff, Architectural implications of function-as-a-service computing, in *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, (Association for Computing Machinery, New York, 2019), pp. 1063–1075
45. E. Jonas, J. Schleier-Smith, V. Sreekanti, C.-C. Tsai, A. Khandelwal, Q. Pu, V. Shankar, J. Carreira, K. Krauth, N. Yadwadkar, J.E. Gonzalez, R.A. Popa, I. Stoica, D.A. Patterson, Cloud programming simplified: A Berkeley view on serverless computing. arXiv preprint arXiv:1902.03383 (2019), http://arxiv.org/abs/1902.03383
46. P. Castro, V. Ishakian, V. Muthusamy, The rise of serverless computing. Commun. ACM **62**(12), 44–54 (2019)
47. M. Cusumano, Cloud computing and SaaS as new computing platforms. Commun. ACM **53**, 27–29 (2010)
48. C. Paniagua, J. Delsing, Industrial frameworks for internet of things: A survey. IEEE Syst. J. **15**, 1149–1159 (2021)
49. kaa: The most flexible IoT platform for your business, https://www.kaaiot.com. Last accessed 2021/10/07
50. ThingSpeak: https://thingspeak.com/. Last accessed 2021/10/07
51. Q. Deng, M. Goudarzi, R. Buyya, FogBus2: A lightweight and distributed container-based framework for integration of IoT-enabled systems with edge and cloud computing, in *Proceedings of the International Workshop on Big Data in Emergent Distributed Environments*, (ACM, 2021), pp. 1–8
52. Zetta: An API-first Internet of Things (IoT) platform, https://www.zettajs.org/. Last accessed 2021/10/08
53. DeviceHive: https://devicehive.com/. Last accessed 2021/10/08
54. R.N. Calheiros, R. Ranjan, A. Beloglazov, C.A.F. De Rose, R. Buyya, CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Softw. Pract. Exp. **41**, 23–50 (2011)
55. A. Núñez, J.L. Vázquez-Poletti, A.C. Caminero, G.G. Castañé, J. Carretero, I.M. Llorente, iCanCloud: A flexible and scalable cloud infrastructure simulator. Int. J. Grid Util. Comput. **10**, 185–209 (2012)
56. OMNeT++: https://omnetpp.org/. Last accessed 2021/10/09
57. S.F. Piraghaj, A.V. Dastjerdi, R.N. Calheiros, R. Buyya, ContainerCloudSim: An environment for modeling and simulation of containers in cloud data centers. Softw. Pract. Exp. **47**, 505–521 (2017)
58. J. Son, A.V. Dastjerdi, R.N. Calheiros, X. Ji, Y. Yoon, R. Buyya, CloudSimSDN: modeling and simulation of software-defined cloud data centers, in *2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, (ACM, 2015), pp. 475–484

59. B. Wickremasinghe, R.N. Calheiros, R. Buyya, CloudAnalyst: A CloudSim-based visual modeller for analysing cloud computing environments and applications, in *2010 24th IEEE International Conference on Advanced Information Networking and Applications*, (IEEE, 2010), pp. 446–452
60. S.K. Garg, R. Buyya, NetworkCloudSim: Modelling parallel applications in cloud simulations, in *2011 Fourth IEEE International Conference on Utility and Cloud Computing*, (IEEE, 2011), pp. 105–113
61. B. Louis, K. Mitra, S. Saguna, C. Åhlund, CloudSimDisk: Energy-aware storage simulation in CloudSim, in *2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC)*, (IEEE, 2015), pp. 11–15
62. R. Mahmud, S. Pallewatta, M. Goudarzi, R. Buyya, IFogSim2: An extended iFogSim simulator for mobility, clustering, and microservice management in edge and fog computing environments. arXiv preprint arXiv:2109.05636 (2021), http://arxiv.org/abs/2109.05636
63. H. Gupta, A. Vahid Dastjerdi, S.K. Ghosh, R. Buyya, iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, edge and fog computing environments. Softw. Pract. Exp. **47**, 1275–1296 (2017)
64. C. Sonmez, A. Ozgovde, C. Ersoy, Edgecloudsim: An environment for performance evaluation of edge computing systems. Trans. Emerg. Telecommun. Technol. **29**, e3493 (2018)
65. T. Qayyum, A.W. Malik, M.A.K. Khattak, O. Khalid, S.U. Khan, FogNetSim++: A toolkit for modeling and simulation of distributed fog environment. IEEE Access **6**, 63570–63583 (2018)
66. C. Puliafito, D.M. Goncalves, M.M. Lopes, L.L. Martins, E. Madeira, E. Mingozzi, O. Rana, L.F. Bittencourt, MobFogSim: Simulation of mobility and migration for fog computing. Simul. Model. Pract. Theory **101**, 102062 (2020)
67. C. Mechalikh, H. Taktak, F. Moussa, PureEdgeSim: A simulation toolkit for performance evaluation of cloud, fog, and pure edge computing environments, in *Proceedings of the International Conference on High Performance Computing Simulation (HPCS)*, (IEEE, 2019), pp. 700–707
68. J. Mass, S.N. Srirama, C. Chang, STEP-ONE: Simulated testbed for edge-fog processes based on the opportunistic network environment simulator. J. Syst. Softw. **166**, 110587 (2020)
69. A. Keränen, J. Ott, T. Kärkkäinen, The ONE simulator for DTN protocol evaluation, in *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, (ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Brussels, 2009), pp. 1–10
70. M. Salama, Y. Elkhatib, G. Blair, IoTNetSim: A modelling and simulation platform for end-to-end IoT services and networking, in *Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing*, (ACM, New York, 2019), pp. 251–261