Next Generation Content Delivery Infrastructures:

Emerging Paradigms and Technologies

Giancarlo Fortino
University of Calabria, Italy

Carlos E. Palau
Universitat Politecnica de Valencia, Spain

Information Science
REFERENCE

Managing Director:	Lindsay Johnston
Senior Editorial Director:	Heather A. Probst
Book Production Manager:	Sean Woznicki
Development Manager:	Joel Gamon
Development Editor:	Myla Merkel
Acquisitions Editor:	Erika Gallagher
Cover Design:	Nick Newcomer

Published in the United States of America by
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com

Library of Congress Cataloging-in-Publication Data

Next generation content delivery infrastructures : emerging paradigms and technologies / Giancarlo Fortino and Carlos E. Palau, editors.
p. cm.
Includes bibliographical references and index.
Summary: “This book delivers state-of-the-art research on current and future Internet-based content delivery networking topics, bringing to the forefront novel problems that demand investigation”-- Provided by publisher.
ISBN 978-1-4666-1794-0 (hardcover) -- ISBN 978-1-4666-1795-7 (ebook) -- ISBN 978-1-4666-1796-4 (print & perpetual access) 1. Web site development. 2. Information storage and retrieval systems. I. Fortino, Giancarlo, 1971- II. Palau, Carlos E., 1970-
TK5105.888.N4825 2012
004.67’8--dc23
2012002872

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

All work contributed to this book is new, previously-unpublished material. The views expressed in this book are those of the authors, but not necessarily of the publisher.

Chapter 2

On the Performance of Content Delivery Clouds

Mukaddim Pathan
Commonwealth Scientific and Industrial Research Organization (CSIRO), Australia

James Broberg
University of Melbourne, Australia

Rajkumar Buyya
University of Melbourne, Australia

ABSTRACT

Extending the traditional Content Delivery Network (CDN) model to use Cloud Computing is highly appealing. It allows developing a truly on-demand CDN architecture based upon standards designed to ease interoperability, scalability, performance, and flexibility. To better understand the system model, necessity, and perceived advantages of Cloud-based CDNs, this chapter provides an extensive coverage and comparative analysis of the state of the art. It also provides a case study on the MetaCDN Content Delivery Cloud, along with highlights of empirical performance observations from its world-wide distributed platform.

INTRODUCTION

Content Delivery Networks (CDNs) (Buyya, et al., 2008; Pallis & Vakali, 2006) are designed to improve Web access performance, in terms of *response time* and *system throughput*, while delivering content to Internet end-users through multiple, geographically distributed replica servers. The CDN industry, i.e. content delivery, consumption and monetization, has been undergoing rapid changes. The multi-dimensional surge in content delivery from end-users has lead to an explosion of new content, formats as well as an exponential increase in the size and complexity of the digital content supply chain. These changes have been accelerated by economic downturn in that the content providers are under increasing pressure to reduce costs while increasing revenue.

With the traditional model of content delivery, a content provider is locked-in for a particular period of time under specific Service Level Agreements (SLAs) with a high monthly/yearly fees and excess

DOI: 10.4018/978-1-4666-1794-0.ch002

data charges (Hosanagar, et al., 2008). Thus, far from democratizing content delivery, most CDN services are often priced out of reach for all but large enterprise customers (Rayburn, 2009). On the other hand, a commercial CDN provider realizes high operational cost and even monetary penalization if it fails to meet the SLA-bound commitments to provide high quality of service to end-users. Thus, it suffers from—spiraling ownership costs; resource wastage for maintaining infrastructure; inability to grow or to profit from economics of scale; inability to fully monetize new or long tail content—to leave lucrative business deals on the table and forfeit profits.

Furthermore, the main value proposition for CDN services has shifted over time. Initially, the focus was on improving end-user perceived experience by decreasing response time, especially when the customer Web site experiences unexpected traffic surges. Nowadays, CDN services are treated by content providers as a way to use a shared infrastructure to handle their peak capacity requirements, thus allowing reduced investment cost in their own Web site infrastructure. Moreover, recent trends in CDNs indicate a large paradigm shift towards a utility computing model (Canali, et al., 2004), which allows customers to exploit advanced content delivery services without having to build a dedicated infrastructure (Gayek, et al., 2004; Subramanya & Yi, 2005). To break through these barriers, a more efficient content delivery solution is required—a truly on-demand architecture based upon standards designed to ease interoperability, scalability, performance, and flexibility.

One approach to address these issues is to exploit the recent emergence of "Cloud Computing" (Buyya, et al., 2009), a recent technology trend that moves computing and data away from desktop and portable PCs into computational resources such as large Data Centers ("Computing") and make them accessible as scalable, on-demand services over a network (the "Cloud"). The main technical underpinnings of Cloud Computing infrastructures and services include virtualization, service-orientation, elasticity, multi-tenancy, power efficiency, and economics of scale. The perceived advantages for Cloud-service clients include the ability to add more capacity at peak demand, reduce cost, experiment with new services, and to remove unneeded capacity.

Extending the traditional CDN model to use clouds for content delivery, i.e. a Content Delivery Cloud (Cohen, 2008), is highly appealing as cloud providers, e.g. Amazon Simple Storage Service (S3), Mosso Cloud Files, and Nirvanix Storage Delivery Network (SDN), charge customers for their utilization of storage and transfer of content (*pay-as-you-go*), typically in order of cents per gigabyte. Cloud providers, on the face value, offer SLA-backed performance and uptime guarantees for their services. Moreover, they can rapidly and cheaply scale-out during flash crowds (Arlitt & Jin, 2000) and anticipated increases in demand. By exploiting the power of Cloud computing, CDN providers endeavor to improve cost efficiency, accelerate innovations, attain faster time-to-market, and achieve application scalability (Leighton, 2009). There are a number of major players in this domain that are providing cloud-based content delivery services on a commercial basis, either by themselves or by partnering with an existing CDN, such as Amazon CloudFront, VoxCAST CDN, and Akamai Cloud Optimizer.

An example research initiative in this context is MetaCDN (Broberg, et al., 2009; Pathan, et al., 2009), an integrated overlay network that leverages resources from existing storage clouds to provide content delivery services. The main goals of the MetaCDN system is to provide economics of scale and high content delivery performance through its simple yet general purpose, reusable, and reliable geographically distributed framework. MetaCDN delivers high performance content delivery via an on-demand cloud service, eliminating costly capital expenditures or infrastructure upgrades. MetaCDN can be deployed as a fully outsourced, end-to-end services platform or as a complement to a CDN provider's existing infrastructure. Thus, it provides flexibility to CDN providers and their customers (content providers) to tailor a solution to meet their unique needs.

A vital component for MetaCDN is a request-redirection technique for directing end-user requests to optimal replica servers according to performance requirements. A suitable request-redirection mechanism extends the system's reach and scale and can alleviate the problems with overloaded servers and congested networks to maintain high accessibility (Barbir, et al., 2003). Therefore, it is desired to devise a redirection mechanism that exhibit the following properties—scalability, transparency, geographic load sharing, and high user perceived performance, to name a few. Towards this end, this chapter addresses the problem of designing request-redirection mechanisms for MetaCDN. It also presents empirical results from a *proof-of-concept* study to evaluate candidate redirection techniques that are implemented within the MetaCDN Content Delivery Cloud.

THE MetaCDN OVERLAY

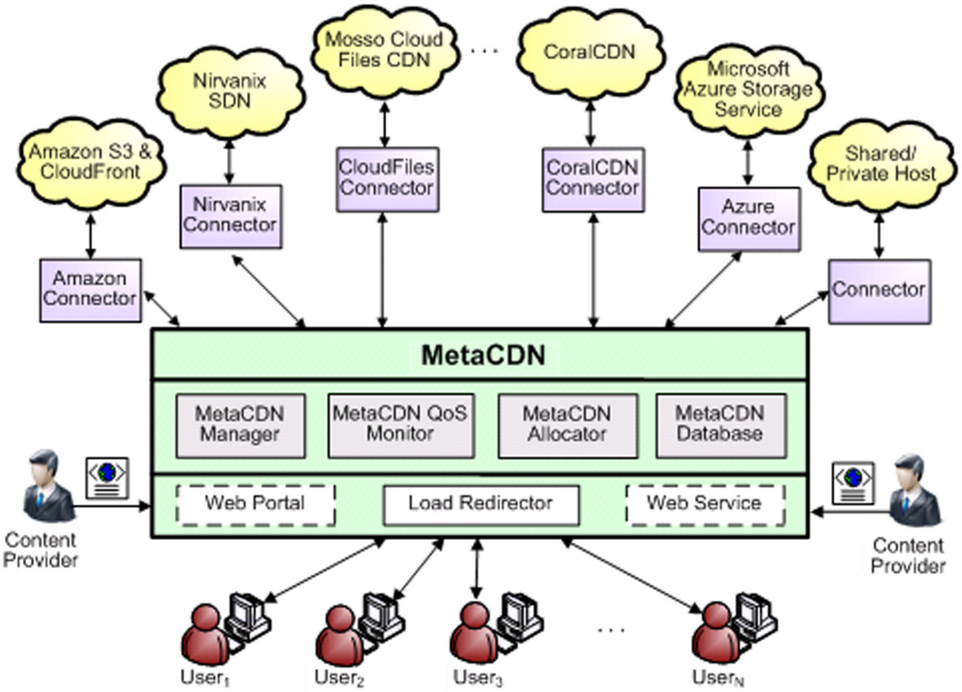
MetaCDN is developed as a simple, general purpose, and reusable overlay network in the face of daunting challenges faced by content providers to exploit multiple cloud providers' resources. It provides a platform to harness content delivery services, by hiding the complexity of using unique Web services or programmer APIs coupled with each cloud provider. End-users experience little of the complex technologies associated with MetaCDN. Content providers interact with the service in a limited number of ways, such as enabling their content to be served, viewing traffic reports, and receiving usage-based billing.

Overview

MetaCDN has opened up opportunities for content providers and end-users to reap rewards through low-cost, high performance and easy to use distributed CDN. Figure 1 provides an illustration of the MetaCDN system. It is coupled with each storage cloud via *connectors*, which provide an abstraction an abstraction to conceal different access methodologies to heterogeneous providers. These connectors (cloud provider specific; and FTP, SSH/SCP or WebDAV for shared or private hosts) provide basic operations for creation, deletion, rename and listing of replicated content. End-users can access the MetaCDN overlay either through a Web portal or via RESTful Web services. In the first case, the Web portal acts as an entry point to the system and performs application level load balancing for end-users who intend to download content that has been deployed through MetaCDN. Content providers can sign up for an account on the MetaCDN system and enter credentials for any storage cloud providers that have an account with. Upon authentication, they can utilize MetaCDN functionalities to intelligently deploy content over geographically spanned replicas from multiple storage clouds, according to their performance requirements and budget limitations.

A distributed MetaCDN gateway (middleware entity) provides the logic and management required to encapsulate the functionality of upstream storage cloud providers with a number of core components. The *MetaCDN allocator* performs optimal provider selection and physical content deployment using four options, namely, *maximize-coverage*, *geolocation-based*, *cost-optimized*, and *QoS-optimized* deployment. The *MetaCDN QoS monitor* tracks the current and historical performance of participating storage providers. The *MetaCDN Manager* has authority on each user's current deployment and performs various housekeeping tasks. The *MetaCDN Database* stores crucial information, such as user accounts and deployments, and the capabilities, pricing and historical performance of providers. Finally, the *MetaCDN Load Redirector* is charged with different redirection policies and is responsible for directing end-users to the most appropriate rep-

Figure 1. Components of the MetaCDN overlay system



lica according to performance requirements. Further details on the critical functionalities of MetaCDN along with full architectural description and development methodology can be found in a prior work (Broberg, et al., 2009).

System Characteristics

MetaCDN is a smart, agile and flexible approach for content delivery that is willing to break with tradition. Specifically, the following set of attributes can be used to characterize it:

- ***Multi-tenancy.*** MetaCDN provides content delivery services for many content providers and end-users on the same distributed infrastructure for different content types. With a cloud-based model, all resources and costs are shared among a large pool of users, enabling genuine savings and economics of scale.
- ***Elasticity.*** It is able to support diverse range of performance requirements from content providers and end-users. This characteristic allows it to quickly and gracefully respond to high request rates at reasonable response time.
- ***Scalability.*** MetaCDN resources are dynamically scalable to handle workload variations with growing number of content providers and end-users, thus enabling optimum resource utilization.
- ***Load sharing.*** It offers automatic and totally transparent load balancing on end-user requests. It enables faster absorption of load spikes with the aid of different load balancing and redirection policies.
- ***Global availability and reliability.*** MetaCDN is a truely on-demand service to provide content delivery functionalities to all authorized users from any-where on the Internet natively. It has the ability to automatically avoid failed replicas or replicas

without desired content. In particular, its tolerance to high failure rate ensures that end-users suffer from little to no outages (i.e. rare in-frequent downtime), such as server or network failures.

- ***Ease of use/operability.*** It can be accessed through a simple Web interface that mimics the look and feel of familiar consumer Web applications, making it extremely intuitive and easy to operate.
- ***Reusability and cost of development.*** The low development cost of using sto-rage clouds for MetaCDN ensures significantly reduced upfront costs. It is implemented by means of reusable simple APIs exposed by the storage cloud providers, while avoiding too many parameters that must be tuned in order to perceive good performance for diverse content providers and their end-users.
- ***Metered services.*** By using the third-party content delivery services of the MetaCDN system, content providers have to pay only for the capacity that they use from upstream cloud providers. Usage information for each replica (e.g. download count and last access) is recorded in order to track the cost in-curred for specific content from a content provider.
- ***Security.*** It addresses crucial security concerns, as content providers use their own credentials for any cloud storage or other provider they have an account with. Thus, it allows content providers to entrust their content to MetaCDN for processing, rest assuring that it will be protected from theft, loss or corruption.

A COMPARATIVE ANALYSIS

Interconnecting multi-provider content delivery services, i.e. CDN peering or CDN internetworking (Amini, et al., 2004; Buyya, et al., 2006; Day, et al., 2003; Pathan, et al., 2008; Pathan & Buyya, 2009b), is a new, flexible and effective way to harness multi-provider capabilities. The aims are to improve performance for end-users, and to achieve pervasive geographical coverage and increased capacity for a provider. These aims are achieved through the deployment of proper request-redirection policies. MetaCDN complements such initiatives by providing an end-to-end cloud-based solution, coupled with on-demand intelligent request-redirection. In this section, we first ascertain MetaCDN's feasibility and position it as a distributed CDN by presenting a comparative study with related systems. Then we study existing redirection mechanisms available in literature and used in practice to endorse MetaCDN's novelty and uniqueness.

MetaCDN and Related Systems

The Content Distribution Internetworking (CDI) (Day, et al., 2003) model lays the foundation for interconnecting providers. Following the footsteps of the CDI initiative, several research efforts explore the benefits of internetworking/peering of CDN providers, content providers, Peer-to-Peer (P2P) networks, and overlays with main focus on offering increased capacity, intelligent server selection, reduced cost, and improved fault tolerance. Examples include CDI protocol architecture (Turrini, 2004; Turrini & Panzieri, 2002), multi-provider peering (Amini, et al., 2004), Synergy overlay internetworking (Kwon & Fahmy, 2005), peer-assisted content delivery (Tran & Tavanapong, 2005), group-based content delivery (Lloret, et al., 2009), provisioning content delivery over shared infrastructure (Nguyen, et al., 2003), use of emerging technologies for the development of enhanced content delivery service (Fortino & Russo, 2008), resource management in a Grid-based CDN (Di Stefano & Santoro, 2008), capacity provisioning networks (Geng, et al., 2003), open CDN implementation (Molina, et al., 2006), and CDN peering (Pathan & Buyya, 2009a, 2009b).

In contrast, MetaCDN assumes no cooperation or peering. Rather it follows a brokering-based approach as in CDN brokering (Biliris, et al., 2002), which is a content delivery brokerage system deployed on the Internet on a provisional basis. MetaCDN differs in that it functions as a Content Delivery Cloud (Cohen, 2008; Pathan, 2010; Pathan, et al., 2009), replicating content over its distributed infrastructure spanning multiple continents, and providing content delivery services to far flung end-users. It has demonstrated improved content delivery performance, and enumerate its content-serving utility and content provider's benefits from using it (Broberg, et al., 2009; Pathan, et al., 2009). While MetaCDN is comparable to the collaborative CDNs, such as CoDeeN (Wang, et al., 2004), CoralCDN (M. Freedman, 2010; M. J. Freedman, et al., 2004), and Globule (Pierre & van Steen, 2001, 2006), it is significantly different as it integrates storage cloud resources spanning the globe to provide content delivery services.

Many Websites have utilized individual storage clouds to deliver some or all of their content (Elson & Howell, 2008), most notably the New York Times (Gottfrid, 2007) and SmugMug (MacAskill, 2007). On the contrary, MetaCDN provides general purpose reusable content delivery services by interacting and leveraging multiple cloud providers. MetaCDN is positioned as a logical fit in the industry initiatives to couple content delivery capabilities with existing cloud deployments, such as Amazon S3 and CloudFront; Silverlining and VoxCAST CDN; Mosso Cloud Files; Nirvanix SDN, which partners with CDNet-works for content delivery; TinyCDN, which leverages Amazon Web services and cloud computing; and Edge Content Network (ECN) from Microsoft, which is re-ported to partner with Limelight Networks for content delivery (Miller, 2008). However, as these systems use centralized or a small number of datacenters, they may suffer from deteriorated end-user experience due to network congestions, peering point congestion, routing inefficiencies, and other bottlenecks of the Internet middle mile (Leighton, 2009). On the contrary, MetaCDN is attributed with a distributed CDN infrastructure to overcome the challenges posed by the Internet's middle mile and ensure that end-user performance does not fall short of expectations. The MetaCDN approach is analogous to the Akamai cloud computing initiative (Leighton, 2009), which provides cloud optimization services for its highly distributed EdgePlatform. However, unlike Akamai it endeavors to achieve true economics of scale by exploiting the pay-as-you-go model of upstream cloud providers.

Recent innovations such as P4P (Xie, et al., 2008) and its companion traffic engineering models (Jiang, et al., 2008) enable P2P to communicate with network providers through a portal for cooperative content delivery. Such proactive network provider participation optimizes global peer-to-peer connections as it saves significant user costs, and by using local connections also speeds up download times for P2P downloaders by 45%. MetaCDN endorses them in the sense that it assists toward a systematic understanding and practical realization of the interactions between storage clouds, which provide an operational storage network and content delivery resources, and content providers, who generate and distribute content.

Table 1 summarizes the comparative analysis between MetaCDN and other related systems in terms of distinctive features and system characteristics. This analysis of existing cloud-based content delivery services assists to separate the performance-wise superiority of representative systems.

Request-Redirection Techniques

Request-redirection is an indispensible enabling cornerstone for CDNs. It is generally used to direct end-user requests to replica servers based on various policies and a possible set of metrics, such as network proximity, user perceived latency,

Table 1. Feature comparison

Feature[a]	Amazon (S3 & CloudFront)	Rackspace (Mosso Cloud Files)	Voxel (VoxCAST, Silverlining)	Nirvanix (CloudNAS)	Microsoft (Windows Azure CDN)	Akamai (Cloud Optimizer)	MetaCDN (integrates storage clouds)
Storage & content delivery	S3 Storage services; CloudFront content delivery	Mosso storage services; content delivery via Limelight	Silverlining cloud services; VoxCAST CDN	Storage services; content delivery via CDNetworks	Azure storage services; content delivery via Limelight	NetStorage services; EdgePlatform content delivery	**Services by leveraging upstream cloud providers**
Service type	On-demand storage in multiple datacenters; on-demand content delivery	On-premises storage	Managed hosting; On-demand content delivery	Managed cloud storage services	On-demand managed hosting in datacenters	On-demand storage and content delivery	**Storage in multiple cloud providers; on-demand content delivery**
Performance	Comparable latency with customer-owned data centers. Sparsely reported performance problem due to outages	Twice more latency than S3 & CloudFront. Reported stability and performance issues for increased traffic	Reported consistent performance on par with competitors such as Akamai and Limelight	Storage functions 222% faster and 2 MB sample file transfer is nearly 300% faster than Amazon S3	Best performance obtained from CDN edge caching by delivering blobs less than 10 GB in size	Up to 400% improvement and at least twice faster application response time than Amazon EC2	**Comparable perceived latency and throughput with upstream providers with little overhead due to load redirection**
Availability & reliability	Availability zones to enable resiliency in case of single location failure, and redundancy	Subject to single point of failure	All time availability as it fails safe against origin server outages	Customizable availability against unplanned outages and redundancy	Service deployment, update and failure management to maintain availability	No single point of failure, automatic failover and redundancy	**Harness the state-of-the-art availability and reliability features of cloud providers**
Geographic distribution	Datacenters at 14 edge locations in three continents (North America, Europe & Asia)	Partnership with Limelight Networks for coverage at 60 locations	POPs at 17 locations in Asia, North America, and Europe	Storage nodes at 5 locations in North America, Europe & Asia	22 physical nodes available globally	48000 servers in 1000 networks world-wide	**Footprint in six continents (Asia, North & South America, Europe, Australia, Africa)**
Multi-tenancy	Yes	Yes	Yes (also dedicated mode)	Yes	Yes	Yes	**Yes**
Load balancing	Listed in future investments	Apache as load balancer	Yes (server switching)	Yes (global and dynamic)	Yes (built-in hardware)	Yes (global and dynamic)	**Yes (automatic and transparent)**
On-demand scalability	Yes	No	Yes	Yes	Yes	Yes	**Partial (work in progress)**
Accessibility	Amazon Web Services API or management console	Browser-based control panel or programmatic API	VoxCAST Web-based portal	Web-based Nirvanix management portal	Azure Services Management Tools	Akamai EdgeControl	**Yes (Web interface)**

continued on following page

Table 1. Continued

Feature[a]	**Amazon (S3 & CloudFront)**	**Rackspace (Mosso Cloud Files)**	**Voxel (VoxCAST, Silverlining)**	**Nirvanix (CloudNAS)**	**Microsoft (Windows Azure CDN)**	**Akamai (Cloud Optimizer)**	**MetaCDN (integrates storage clouds)**
Automatic replication	S3: No; CloudFront: Yes	Yes	Yes	Yes	No	Yes	**Yes**
SLA (%)	99-99.9	99.9	100	99.9	99.95	100	**Provider specific**
Developer API	Yes (Amazon Web services)	Yes (Cloud Servers API)	Yes (Hosting API)	Yes (Web services API)	Yes (Azure SDK API)	Yes (EdgeScape API)	**Connectors for integration**
Economic model and pricing	Pay-as-you-go	Pay-as-you-go	Progressive universal scale billing upon usage	Pay-as-you-go	Consumption-based pricing model	Volume-based pricing; pay-par-use model for NetStorage	**Built on pay-as-you-go model**
Security	Protection for DDoS attacks, access control list and firewalls	Data protection, DDoS migration services, firewalls	Secure authentication, firewalls	Secure authentication, transmission via SSL	Intrusion prevention, .net security, firewalls	Protection for DDoS attacks and application firewall	**Secure authentication to reap provider's security measures**

[a] The facts presented in this table are based on existing literature including industry-specific Website, data sheet, whitepaper, and professional news blogs.

bandwidth, content availability and replica server load. There exist multiple request-redirection mechanisms, which can be categorized in a number of ways according to different performance objectives.

Barbir et al. (Barbir, et al., 2003) categorize the known request-redirection techniques in CDNs into *DNS-based*, *transport-layer* and *application-layer* redirection. In DNS-based techniques, a specialized DNS server is augmented in the name resolution process to return different server addresses to end-users. They are the most common due to the ubiquity of the DNS system as a directory service. The performance and effectiveness of DNS-based redirection techniques have been studied in a number of recent studies (Biliris, et al., 2002; Mao, et al., 2002; Shaikh, et al., 2001). Despite its wide usage, DNS-based approaches are found to suffer from the following drawbacks: (a) actual end-user request is not redirected, rather its Local DNS (LDNS), assuming that end-users are near to their LDNS; (b) browser's request is cached due to the hierarchical organization of the DNS service; (c) the DNS system is not designed for very dynamic changes in the mapping between hostnames and IP addresses; and (d) most significantly DNS cannot be relied upon as it can have control over as little as 5% of incoming requests in many instances (Cardellini, et al., 2002). In transport-layer redirection, the information available in the first packet of the end-user request, in combination with user-defined policies and other metrics are used to take redirection decision. Several research (Liston & Zegura, 2001; Pai, et al., 1998; Yang & Luo, 1999) report using this approach for redirection. In general, this approach is used in combination with DNS-based techniques. While this approach is suitable for steering end-users away from overloaded replica servers, the associated overhead limits its usage for long-lived sessions such as FTP and RTSP. Finally, application-layer redirection involves deeper examination of end-user request packet to provide fine-grain redirection. However, this approach may suffer from the lack of transparency and additional latency. URL rewriting and HTTP 302 redirection are the examples of techniques using this approach. In the context of MetaCDN, the system exploits a combination of DNS-based and application-layer techniques for request-redirection. Specifically, name resolution for the base MetaCDN URL is performed using DNS-redirection and end-user request for specific content (Web object) is serviced using application-layer redirection.

With the objective to minimize Web access latency, request-redirection can be partitioned into client and server-side techniques. Client-side redirections in CDNs (Conti, et al., 2001; Kangasharju, et al., 2001; Rangarajan, et al., 2003; Wang, et al., 2002) are based on the premise that the network is the primary bottleneck. They tend not to rely on any centralization as redirections occur independently. Server-side techniques perform URL redirection using HTTP status code. They direct all incoming requests to a set of clustered hosts based on load characteristics. These techniques are mainly application specific and more suited for clustered servers. There also exist significant research (Cardellini, et al., 2000, 2003; Karaul, et al., 2000; Rabinovich, et al., 2003) combining client and server-side redirection. This hybrid approach works well when the bottleneck is not clearly identified or varying over time. According to this categorization, MetaCDN complements the hybrid request-redirection technique; by performing server-side gateway redirection and client-side HTTP 302 redirection for content requests.

In terms of content retrieval, request-redirection techniques can be divided into full and selective (or partial) redirection. In full redirection, the DNS server is modified in such a way that all end-user requests are directed to a replica server. This scheme requires that either replica servers hold all the content from the origin server, or that they act as surrogate proxies for the origin server. On the other hand, in selective redirection, a content provider modifies its content so that links to specific embedded Web objects have host names in a domain for which the CDN provider is authoritative. Thus, the base HTML page is retrieved from the origin server, while embedded objects are retrieved from CDN replica servers. While full replication has dynamic adaptability to new hot-spots, it is not feasible considering the on-going increase in Web objects size. A selective redirection works better in the sense that it reduces load on the origin server and on the Web site's content generation infrastructure. Moreover, if the embedded content changes infrequently, it exhibits better performance. While it is possible to use the MetaCDN replica infrastructure to enable full redirection, we limit our work for selective redirection by storing only embedded Web content into replicas and directing end-user requests to them.

Request-redirection mechanisms are governed by policies that outline the actual redirection algorithm on how to perform server selection in response to an end-user request. These policies can be either adaptive or non-adaptive. Adaptive policies consider the current system condition, whereas non-adaptive policies use some heuristics in order to perform target server selection. The literature on re-quest-redirection policies is too vast to cite here (see the survey by Sivasubramanian et al. (Sivasubramanian, et al., 2004) and the references therein for initial pointers for redirection policies in CDN context). MetaCDN deploys adaptive redirection with the ability to cope with degenerated load situations. In particular, it strives to demonstrate high system robustness in the face of unanticipated events, e.g. flash crowds.

There exist significant research efforts (Amini, et al., 2003; Erçetin & Tassiulas, 2003; Presti, et al., 2005; Ranjan, et al., 2004) that model request-redirection as a mathematical problem. They attempt to find a solution from an operations research perspective by modeling redirection as a graph theory, optimization, delay constrained routing, or server assignment problem. Most of these work use simulations to evaluate the performance of their approach. On the contrary, the MetaCDN redirection is evaluated through a proof-of-concept implementation on its distributed infrastructure.

The request-redirection techniques employed within MetaCDN also draw similarity with those used in the collaborative CDNs, such as CoDeeN (Wang, et al., 2004), CoralCDN (M. Freedman, 2010; M. J. Freedman, et al., 2004), Globule (Pierre & van Steen, 2001, 2006), and PRSync (Shah, et al., 2008), which perform overlay redirection by exploiting request locality, network measurement, topology, and AS-based proximity. Similarly, MetaCDN's request-redirection techniques are based on metrics such as geographic proximity, cost, request traffic, and QoS metrics (response time, throughput, HTTP response code). The uniqueness lies in adding the capability for quantifying traffic activities using a network utility metric within MetaCDN while intelligently redirecting user requests.

REQUEST-REDIRECTION DESIGN

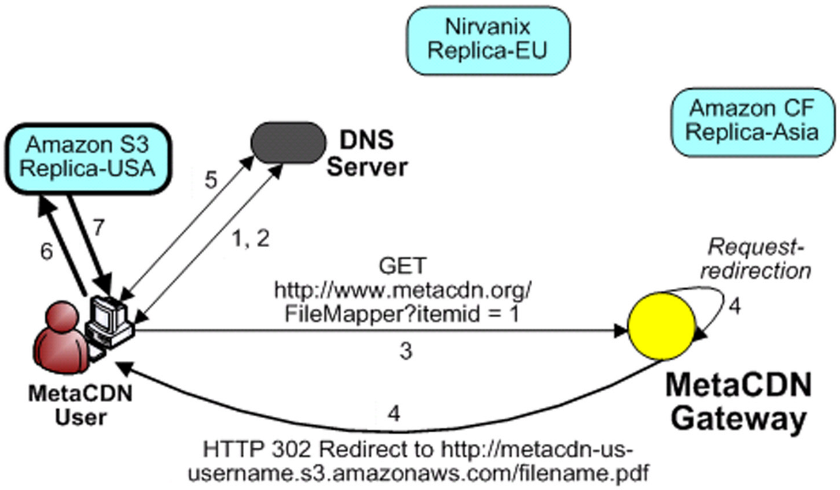
An efficient request-redirection technique is vital to extend the reach and scale of MetaCDN. In this section, we analyze the design space of competent request-redirection techniques and describe MetaCDN redirection logic along with the candidate techniques.

Design Space

Designing a request-redirection strategy that does not sacrifice the scalability, transparency, availability and performance benefits of a content delivery cloud, i.e. MetaCDN, is a challenging task. A candidate redirection technique should have the following properties:

- ***Scalability.*** It should be responsive to changing circumstances. It should aid the system with the ability to gracefully scale and expand its network reach in order to handle new and large number of data, end-user requests, and transactions without any significant decline in performance.
- ***Load balancing.*** With the aid of the redirection technique, MetaCDN as a service provider should be able to effectively react to overload conditions by selecting least loaded optimal server(s) for serving content requests. The load balancing decisions should ensure that end-users experience reasonable con-tent delivery performance.
- ***Distributed redirection.*** It should not rely on any centralization and all redi-rectors (i.e. MetaCDN gateway) should operate independently. It should also accommodate any dynamic changes in network performance and incoming request traffic.
- ***Transparent name resolution.*** DNS mapping during redirection should be transparent to end-users. In order to transparently contact a replica server for desired content, redirection should ensure a one-to-many mapping from the hostname to one of the IP addresses of distributed replicas.
- ***Fault transparency.*** It should ensure that unresponsive replicas are detected, bypassed and end-users are unaware of the redirection to other replicas. Moreover, previously failed replicas that become available again should be incorporated quickly.
- ***Flexibility.*** There should be provision to accommodate different request-redir- ection techniques to provide options to content providers and its users with varied objectives. In addition, a candidate request-redirection technique should improve the usefulness of distributed replicas.
- ***Server decoupling.*** The redirection logic should be implemented without any change of the existing client or server code, conforming to existing standards. It should also be possible to deploy the devised redirection scheme easily, pre-ferrably as a

Figure 2. MetaCDN request-redirection



plug-in to the server, with minimum effort. Thus, it should be ensured that the implementation overhead of a given request-redirection technique is minimal.

MetaCDN Request-Redirection Logic

Request-redirection in MetaCDN takes place under the governance of the MetaCDN gateways, which resemble distributed request-redirectors to forward end-user content requests to appropriate replica server. The MetaCDN gateway is capable of utilizing any request-redirection technique that is plugged into the MetaCDN Load Redirection module. Integrating a new request-redirection scheme does not require any changes to the server or client-side.

As shown in Figure 2, the sequence of steps for an end-user in the East Coast of the USA to retrieve content through MetaCDN is as follows:

1. The end-user issues an HTTP request for a content that has been deployed by the MetaCDN Allocator using one of the content deployment options available. The browser attempts to resolve the base hostname (http://www.metacdn.org) for the MetaCDN URL http://www.metacdn.org/*FileMapper?itemid=XX*, where XX in the URL format is a unique key associated with the deployed content.
2. The Local DNS (LDNS) of the end-user contacts the authoritative DNS (ADNS) for that domain to resolve this request to the IP address of the closest MetaCDN gateway, e.g. http://us.metacdn.org.
3. The end-user (or its browser) then makes an HTTP GET request for the desired content on the MetaCDN gateway.
4. Depending on the utilized request-redirection scheme, the MetaCDN Load Redirector is triggered to select the optimal replica that conforms to the specified service requirements. At this point, the MetaCDN gateway returns an HTTP redirect request with the URL of the selected replica.
5. Upon receiving the URL of the selected replica, the DNS resolves its domain name and returns the associated IP address to the end-user.
6. The user sends request for the content to the selected replica.
7. The selected replica satisfies the user request by serving the desired content.

In order to ensure that the best replica is selected for serving user requests, the following tests are performed during the request-redirection process:

- Is there a content replica available within required response time threshold?
- Is the throughput of the target replica within tolerance?
- Is the end-user located in the same geographical region as the target replica?
- Is the replica utility the highest among all target sites?
- Is one of the target replicas preferred, according to user requirements or any administrative settings?

If it is assumed that all candidate replicas are available and have capacity, i.e. response time and throughput thresholds are met, the MetaCDN system checks for the continent/geographic location and administrative preference (an indicative flag used by MetaCDN manager to manually prefer or avoid a replica). MetaCDN achieves transparency as end-user browsers automatically access the redirection service, being redirected by the MetaCDN gateway. End-users have least possible to do to take benefit of request-redirection. They see only MetaCDN URL and they have no way for discovering the address of a replica when using the redirection service and accessing the replica server directly. Thus, an end-user is prevented to keep an explicit reference to a replica, which may cause dangling pointers during the downtime of the replica.

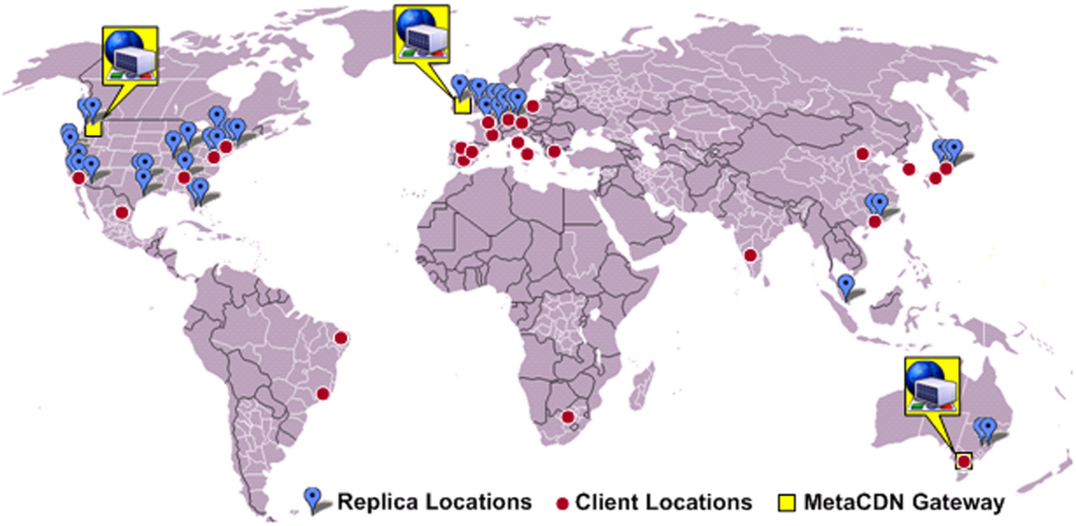
While MetaCDN Load Redirector ensures directing users to the best responding replica, an extra feature is realized through its ability to automatically avoid failed replicas or replicas without the desired content. Bypassing occurs in the following two ways. Firstly, if a replica has the desired content, but shows limited serving capacity due to network congestions, it is reflected in its measured network utility metric, exhibiting a low value. As a consequence, the replica is not considered as a candidate for redirection. Secondly, if the replica does not have the desired content, it can not serve end-user requests and thus leads to an insignificant utility value. Hence, it is automatically discarded to be considered as a candidate replica. In addition, a secondary level of internal redirection enabled by an individual cloud provider ensures that request-redirection does not overload any particular replica.

Candidate Techniques

Representative request-redirection techniques used for experimentation and evaluation of the MetaCDN system are:

- ***Random redirection.*** It is a simple baseline policy where each content request is sent to a randomly picked replica. This scheme can be used for comparison purpose to determine a reasonable level of performance, since an effective redirection technique is expected to scale with the increasing number of clients and MetaCDN replicas, and to not exhibit any pathological behavior due to the assignment patterns. The drawback of this approach is to often increase latency by not picking up the most appropriate replica. Moreover, adding more servers does not reduce the working set of each server.
- ***Geolocation-based redirection.*** It exploits the request locality by taking into account end-user preferences and directing the user to the closest physical replica in the specified region(s). For this purpose, a geolocation service is utilized that finds the geographic location (latitude and longitude) of the end-user and measures their distance from each matching replica using a simple spherical law of cosines, or a more accurate approach such as the Vincenty formula for distance between two latitude/longitude points (Vincenty, 1975), to find the clos-

Figure 3. Experiment testbed



est replica. Although there exists a strong correlation between the performance experienced by end-users and their locality to replicas (Broberg, et al., 2009), there is no guarantee that the closest replica is always the best choice, due to cyclical and transient load fluctuations in the network path.

- ***Utility-redirection.*** In this scheme, end-users are directed to the highest utility optimal replica that conforms to the specified service requirements. If there is more than one candidate target replica exhibiting the highest utility, the one with the fastest response time is chosen to redirect user requests. For this purpose, utility is measured quantitatively based on MetaCDN's traffic activities. It is expressed with a value in the range [0, 1], quantifying the relation between the number of bytes of the served content against the number of bytes of the replicated content (Pathan, et al., 2009). The measured utility metric represents the usefulness of MetaCDN replicas in terms of data circulation in its distributed network. It is vital as system wellness greatly affects the content delivery performance to end-users. Although utility-based request-redirection outcomes sensible replica selection in terms of response time, it may not provide a high throughput performance to end-users. Nevertheless, being it is focused on maximizing the utility of the MetaCDN system; it results in high utility for content delivery to end-users.

PERFORMANCE EVALUATION

This section presents the outcome of a proof-of-concept testbed experiment to determine the performance of MetaCDN content delivery cloud, by measuring the user perceived response time and throughput. Figure 3 provides a schematic representation of the experimental testbed and Table 2 provides a summary of the conducted experiment. The global MetaCDN testbed spans six continents with distributed clients at different institutions; replicas from multiple storage cloud providers; and MetaCDN gateways, hosted on the Amazon Elastic Computing Cloud (EC2) and a cluster at the University of Melbourne, Australia. All client locations, except in Africa, South America and

Table 2. Summary of the experiment

	Category	Value	Provider	Locations
Experiment Testbed	Number of MetaCDN gateways	3	Amazon EC2 and own cluster	Asia/Australia, Europe, and North America
	Number of replicas	40	Amazon, Mosso and Nirvanix	Asia, Australia, Europe, and North America
	Number of clients (end-user nodes)	26	Voluntary	Asia, Australia, Europe, North and South America, and Africa

	Category	Description
Experiment Details	Total experiment time	48 hours
	Duration of an epoch	2 hours
	Maximum user requests/epoch	30 requests from each client
	Service timeout for each request	30 seconds
	Test file size	1 KB and 5 MB
	Content Deployment	Maximize-coverage deployment
	Request-redirection policies	Random, Geo, and Utility

	Category	Distribution	PMF	Parameters
End-user Request Modeling	Session inter-arrival time (Floyd & Paxson, 2001)	Exponential	$\lambda e^{-\lambda x}$	$\lambda = 0.05$
	Content requests per session (Arlitt & Jin, 2000)	Inverse Gaussian	$\sqrt{\frac{\lambda}{2\pi x^3}} e^{\frac{-\lambda(x-\mu)^2}{2\mu^2 x}}$	$\mu = 3.86$ $\lambda = 9.46$
	User think time (Barford & Crovella, 1999)	Pareto	$\alpha k^{\alpha} x^{-\alpha-1}$	$\alpha = 1.4$, $k = 1$

South Asia, have high speed connectivity to major Internet backbones to minimize the client being the bottleneck during experiments.

Methodology

The experiment was run simultaneously at each client location over a period of 48 hours, during the middle of the week in May 2009. As it spans two days, localized peak times (time-of-day) is experienced in each geographical region. Two test files of size 1KB and 5MB have been deployed by the MetaCDN Allocator module, which was instructed to maximize coverage and performance, and consequently the test files were deployed in all available replica locations of the storage cloud providers integrated to MetaCDN. While these file sizes are appropriate for the conducted experiment, a few constraints restrict the use varied and/or even larger sized files. Firstly, the experiments generate heavy network traffic consuming significant network bandwidth, thus larger file trafficking would impose more strain and network congestions on the voluntary clients, which some clients may not be able to handle. Moreover, at some client locations, e.g. India and South Africa, Internet is at a premium and there are checks regarding Internet traffic so that other users in the client domain accessing the Internet are not affected.

The workload to drive the experiment incorporates recent results on Web characterization

Table 3. List of performance indices

Performance Index	Description
Response time	The time experienced by an end-user to get serviced
Throughput	Transfer speed to download a test file by an end-user
Utility	Content-serving ability, ranges in [0, 1]
Probability(Utility achieved)	The probability or the fraction of time that the system achieves the given utility
Content provider's benefit (Surplus)	Surplus from using MetaCDN, expressed as a percentage

(Arlitt & Jin, 2000; Barford & Crovella, 1999; Floyd & Paxson, 2001). The high variability and self-similar nature of Web access load is modeled through heavy-tailed distributions. The experiment time comprises epochs of 2 hours, with each epoch consisting of a set of user sessions. Each session opens a persistent HTTP connection to MetaCDN and each client generates requests to it to download each test files, with a timeout of 30 seconds. Between two requests, a user waits for a think time before the next request is generated. The mean think time, together with number of users defines the mean request arrival rate to MetaCDN. For statistical significance, each client is bounded to generate a maximum number of 30 requests in each epoch. The files are downloaded using the UNIX utility, wget, with the --no-cache and --no-dns-cache options to ensure that a fresh copy of the content is downloaded each time (not from any intermediary cache) and that the DNS lookup is not cached either.

The *response time* and *throughput* obtained from each client location were measured. The first performance metric captures the end-to-end performance for end-users when downloading a 1 KB test file from MetaCDN. Due to the negligible file size, the response time is dominated by DNS lookup and HTTP connection establishment time. Lower value of response time indicates fast serviced content. The latter metric shows the transfer speed obtained when the 5 MB test file is downloaded by users from the MetaCDN replicas. It provides an indication of consistency and variability of throughput over time.

The *utility* of MetaCDN is measured according to a quantitative expression, capturing the true traffic activities, in terms of the number of bytes transferred during content replication and servicing (Pathan, et al., 2009). A high utility value shows the content-serving ability of the system, and signifies its durability under highly variable traffic activities. To emphasize the impact of request-redirection on the measured utility, the *probability* that MetaCDN achieves a given level of utility as the performance metric. Finally, based on the measured observations, we determine the benefits of a content provider (surplus) from using the MetaCDN system. Table 3 summarizes the performance indices used in the experimental evaluation.

EMPIRICAL RESULTS

To avoid redundancy, we present average of the results from the following eight representative client locations in five continents—Paris (France), Innsbruck (Austria), and Poznan (Poland) in Europe; Beijing (China) and Melbourne (Australia) in Asia/Australia; Atlanta, GA, and Irvine, CA (USA) in North America, and Rio de Janeiro (Brazil) in South America. Detailed results in each locations for the full experiment duration can be found in another work (Pathan, et al., 2009).

Table 4. Average response time observations (in seconds) at client locations

End-user location	Random	Geo	Utility
Paris	0.92	0.78	0.99
Innsbruck	0.75	0.71	0.70
Poznan	1.59	1.53	1.48
Beijing	4.16	4.03	3.61
Melbourne	1.73	1.26	1.52
Atlanta	0.81	0.74	0.72
Irvine	0.90	0.90	0.77
Rio de Janeiro	1.67	1.63	1.20

Response Time

Table 4 shows the end-to-end response time experienced by end-users when downloading the 1 KB test file over a period of 48 hours. The measure of the response time depends on the network proximity, congestions in network path and traffic load on the target replica server. It provides an indication of the responsiveness of the replica infrastructure and the network conditions in the path between the client and the target replica which serves the end-user. A general trend is observed that the clients experience mostly consistent end-to-end response time. For all the request-redirection policies, the average response time in all the client locations except Beijing is just over 1 second, with a few exceptions. Notably the users in Beijing experience close to 4 seconds average response time from the MetaCDN infrastructure. This exception originates as a consequence of firewall policies applied by the Chinese government. Similar observations have been reported in a previous measurement study (Rahul, et al., 2006), which demonstrates that the failure characteristics on the Internet path to the edge nodes in China are remarkably different than the Internet paths to the edge nodes in other part of the world.

At several time instances during the experiment, end-users experience increased response time. The resulting spikes are due to the sudden increases in request traffic, imposing strain on the MetaCDN replicas. Under traffic surges, the MetaCDN Load Redirector module activates to handle peak loads. As a consequence, end-user requests are often redirected to a target replica outside its authoritative domain and/or are served from an optimal distant proximity server, thereby, contributing to the increased response time. However, MetaCDN handles peak loads well to provide satisfactory service responsiveness to end-users. This phenomenon of increased response time is more visible for random-redirection. As it makes a random choice, often the target replica selection is not optimized, thus leading to highly variable response time. Especially, at several occasions, users observe more than 30 seconds response time, thus leading to service timeout. Geo-redirection directs user requests to the closest proximity server, understandably producing low response time. On the contrary, utility-redirection chooses the highest utility replica, which may not be in close proximity to an individual client location. Nevertheless, there is no clear winner between them in terms of response time, as they exhibit changeable performance at different client locations. As for instance, end-users in Paris enjoy better average response time (0.77 seconds) with geo-redirection, due to their close proximity to the Amazon, Mosso and Nirvanix nodes in Frankfurt (Germany), Dublin (Ireland), and London (UK). For Melbourne, the reason behind better performance of geo-redirection is the existence

Table 5. Average throughput observations (in KBs) at client locations

End-user location	Random	Geo	Utility
Paris	1486.46	2146.75	475.39
Innsbruck	2020.76	2178.03	518.67
Poznan	7551.53	9012.28	1795.80
Beijing	229.32	269.15	206.54
Melbourne	3625.26	6519.39	413.15
Atlanta	6137.11	6448.30	3349.39
Irvine	4412.62	2757.73	504.74
Rio de Janeiro	838.94	521.30	1138.14

of the Mosso node in Sydney. For both of these two clients, utility-redirection policy directs requests to a distant replica than the closest one and results in increased response time.

Throughput

Table 5 shows the average throughput obtained per two hours, when downloading content (5MB file) via MetaCDN. At all the client locations, consistent throughput was observed during the experiment. As expected, we observe that in almost all the client locations, geo-redirection results in highest throughput as the users get serviced from the closest proximity replica. However, it performs worse than random-redirection for the Irvine client. The reason is that random-redirection decision in this location most of the time selects close proximity Amazon replica(s) with better network path than that of geo-redirection, which chooses Mosso replica. Moreover, the service capability from these two replicas and the network path between the replica and client also contribute to the observed throughput variations.

For most of the clients, except Rio de Janeiro, utility redirection performs much worse than geo-redirection. The reason is understandable, as utility-redirection emphasizes maximizing MetaCDN's utility rather than serving an individual user, thus sacrificing end-user perceived performance. For Rio de Janeiro, geo-redirection leads to the closest Mosso node in the USA, whereas utility-redirection results in more utility-aware replica, which is the Amazon node(s) in the USA. It could be presumed that Amazon node supersedes the Mosso node in terms of its service capability, better network path, internal overlay routing, and less request traffic strain.

It is observed that users in Poznan enjoy the best average throughput, which is 9MB/s for geo-redirection. The reason is that the client machine is in a MAN net-work, which is connected to the country-wide Polish optical network PIONEER with high capacity channels dedicated to the content delivery traffic (Kusmierek, et al., 2007). Another client location with high throughput is Atlanta, which achieves speeds of approximately 6.2 MB/s for geo-redirection and 3.3 MB/s for utility-redirection, due to the existence of better network path between the client and the MetaCDN replica infrastructure. This reasoning is deemed valid, since there are Mosso nodes in the same location.

Alike response time, end-users in China achieves the lowest throughput among all the client locations. The underlying reason is again checks on the request traffic and bandwidth constraints due to firewall policies. We put more emphasis on the results from Melbourne, which is of interest as Australia is not as highly connected as Europe or North America, depending on a small number of expensive international links to major data centers in Europe and the USA. We observe that due to the existence of a nearby Mosso node in Sydney, the users in Melbourne experience 6.5 MB/s of throughput with geo-redirection and 3.6 MB/s for random-redirection. However, for utility-redirection the replica selections result in the Amazon node(s) in the USA, thus leading to a lower but consistent average throughput of 410 KB/s.

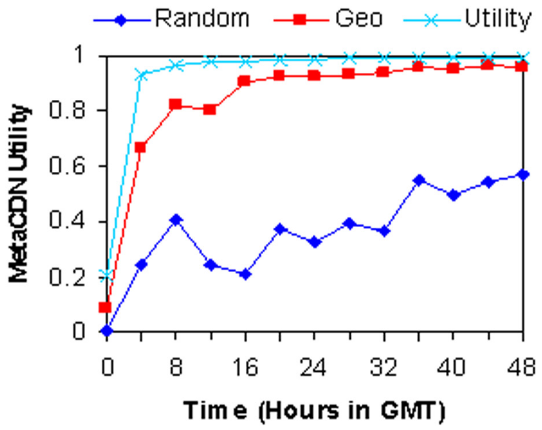
From these observations, the following decisive conclusions can be reached. Although utility-redirection outcomes sensible replica selection in terms of response time, it may not provide a high throughput performance to end-users. Nevertheless, being focused on maximizing the utility of the MetaCDN system; it results in high utility for content delivery. Sufficient results to support this claim are presented in the next section.

MetaCDN Utility

Figure 4 shows how MetaCDN utility is varied during the testbed experiment upon replica selection for incoming content requests. The shown utility values in the figure are averaged over three deployed MetaCDN gateways in Asia/Australia, Europe and North America. It is observed that utility-redirection produces the highest utility in the system by selecting the most active replicas to serve users. It also improves the traffic activities and contributes to uplifting MetaCDN's content-serving ability. It should be noted that there is a warm-up phase at the beginning of the 48 hours experiment during which the replicas are populated with content requests, resulting in low utility values. This is visible during the initial hours for utility and geo-redirection.

To emphasize the content-serving ability of MetaCDN, Figure 4 presents the probability (or the fraction of time) that the system observes a utility above a certain utility level during the experiment. The intention is to show to what extent the system can maximize its own profit. The higher the probability, the more likely it is that the specified utility level could be achieved. From the figure, it is noticeable that utility-redirection outperforms other alternatives, as it often produces over 0.95 utility for MetaCDN with a 0.85 probability. Geo-redirection performs well as it has a 0.77 probability that it can achieve 0.9 utility. Finally, random-redirection performs the worst and it can only achieve close to 0.56 utility for MetaCDN with a probability of 0.23. Therefore, a MetaCDN administrator may utilize a request-redirection policy apart from random, in order to maximize the system's content-serving ability.

Figure 4. Probability of achieving specified utility



FUTURE RESEARCH DIRECTIONS

A number of future research directions in relation to Cloud-based content delivery systems can be devised. In this section, an indicative list is populated, realizing the awaited technological innovations in this area in the coming years. While elaborating on the future research topics, pointers to existing literature are provided so as to lay out a comprehensive research roadmap to the CDN community.

A Cooperative Architecture for Dynamic Replication

The "time-shifted" nature of the dynamic content defies the existing content delivery architectures and increases the overall traffic loads and bandwidth demands by orders of magnitude. To overcome the problems of resource over-provisioning, performance degradation, and adverse business impact, it is required to develop a light-weight cooperative architecture, potentially taking advantage of the Cloud systems, where CDN servers are grouped into clusters of neighbor surrogates, cooperatively replicate and deliver the user-requested content. A solution towards this end can extend the existing architecture (Amini, et al., 2004; Buyya, et al., 2006; Day, et al., 2003; Pathan, et al., 2008) that allow resource sharing among multiple Cloud-based content delivery services.

On the Economics of Cooperation

There is the need to incentivize CDN providers to keep motivated for contributing resources to allow replication in the cooperative domain content delivery clouds. To ensure sustained resource sharing, sufficient incentives should be provided to all parties (Pathan & Buyya, 2007). Use of economics principles in this context represents a dynamic scenario and makes the system more manageable through regulating and analyzing the emergent marketplace behavior. In this context, an economic model can be developed to consider a CDN as an independent economic agent for buying and selling content. It is significant to emphasize the QoS-oriented aspects of provider selection and analyze the sensitivity of different performance metrics such as cost, net benefit, value and popularity of the content, and transport cost. Future research in this direction will focus on the development of dynamic pricing policies for Cloud systems and CDNs (Anandasivam & Premm, 2009; Pueschel, et al., 2009); study of the interaction between different pricing approaches (Hosanagar, et al., 2008); and investigation of the impact of competition in the CDN industry on CDN pricing (Christin & Chuang, 2004, 2005; Christin, et al., 2008).

Replication to Consider Mobility in the Cooperative Domain

CDNs offer an exciting playground to exploit the emerging technological advances of mobile computing. To deliver content to a large number of highly dynamic users, it is required to take into account the mobility notion. The variations in mobile user requests are caused not only by changes in content popularity, but also by user mobility. Each user request is characterized by the requested content, the time of the request, and the location of the user. The concept of caching "hot" content is not new, but in the context of mobility for content delivery in the Cloud-based cooperative domain, there are significant competing considerations. It is required to develop dynamic, scalable, and efficient replication mechanisms that cache content on demand with respect to the locality of requests, focusing on regions where specific content is needed most (Chen, et al., 2003; Fortino, et al., 2009). In this context, developed solutions should include a mobility model, geolocation-oriented services, a monitoring mechanism and a service delivery protocol for CDNs (Loulloudes, et al., 2008). Future research in this direction will focus on potentially considering user location context, navigational behavior, and very high spatial and temporal demand variations to dynamically reconfigure the system, and minimize the total traffic over the network backbone.

Replica Placement, Consistency, and Ranking

There are a number of research issues to be resolved for replica management, such as how many replicas of various objects to have, where in the network to place them, how to manage the replicas, and how they are to be ranked for efficient request distribution (Cameron, et al., 2002; Chen, et al., 2002; Presti, et al., 2005). In this context, existing approaches will be extended for cooperative content delivery in Cloud-based CDNs. It is crucial to decide on the use of static or dynamic approach, granularity of replication and handling of failed replicas. In order to guarantee that the requested users are not serviced with stale objects, a proper replica consistency technique is to be devised. An appropriate technique for ranking replicas can also be developed by using a combination of metrics such as Web server load, latency, geographical proximity and network distance (Bakiras & Loukopoulos, 2005).

Energy-Aware Request-Redirection

Energy-awareness in computing is an emerging research area. Large-scale distributed systems such as CDNs consume huge amount of electricity, thus leading to high energy cost (Qureshi, et al., 2009). Conventionally, the approach to reduce energy cost is to decrease the amount of the consumed energy. Request-redirection to optimal replicas can aid to cut down the energy cost by decreasing the amount of the consumed energy during cooperative content delivery in Cloud-based CDNs. While energy-aware content delivery is economically beneficial for commercial CDNs, there are also benefits for a third-party Cloud-based CDN system, e.g. MetaCDN (Broberg, et al., 2009; Pathan, et al., 2009), which may be interested in attaining social welfare by reducing the environmental impact of high energy consumption. Therefore, it is required to develop schemes to reduce the energy consumption and carbon footprint of CDNs. These energy-aware request-routing techniques will consider end-user's geographical proximity, energy usage and cost, and incoming traffic load for directing users to the most cost-effective replica.

Enhancement for Cloud-based CDNs

Extension of traditional CDNs model to Cloud-based CDNs enhances capabilities to deliver services that are not only limited to Web applications, but also include storage, raw computing or access to any number of specialized services. It initiates potential research that focuses on identifying necessary application requirements, enhancing scalability, system robustness, usability and access performance, low cost, data durability, and support for security and privacy. For instance, as an advancement of previous work with the MetaCDN system, future research can develop active measurement approaches for QoS-based and probabilistic request-redirection, autonomic scaling of infrastructure, and a security framework that spans the integrated storage cloud providers.

CONCLUSION

MetaCDN, characterized as a Content Delivery Cloud, provides a cost-effective solution for responsive, scalable, and transparent content delivery services by harnessing the resources of multiple storage cloud providers. It provides sensible performance and availability benefits without requiring the content providers to build or manage complex content delivery infrastructure themselves. This chapter presented a performance study of MetaCDN, based on conducted *proof-of-concept* experiments on a global testbed. An indicative list of future research directions is also presented, including the development of advanced request-redirection techniques and pricing policies for Content Delivery Clouds; and on-demand autonomic management (expansion/contraction) of replica deployment. From the results obtained, it can be concluded that the utility of MetaCDN is maximized by using utility-based request-redirection to provide sensible replica selection and consistent average response time; however, with the cost of lower throughput in comparison to other candidate request-redirection policies. In contrast, a content provider's benefit is enhanced with improvement of the perceived throughput through MetaCDN.

REFERENCES

Amini, L., Shaikh, A., & Schulzrinne, H. (2003). Modeling redirection in geographically diverse server sets. *Proceedings 12th International Conference on World Wide Web (WWW'03)*, (pp. 472-481).

Amini, L., Shaikh, A., Schulzrinne, H., Res, I. B. M., & Hawthorne, N. Y. (2004). Effective peering for multi-provider content delivery services. *Proceedings - IEEE INFOCOM*, *04*, 850–861.

Anandasivam, A., & Premm, M. (2009). Bid price control and dynamic pricing in clouds. *Proceedings 17th European Conference on Information Systems (ECIS'09)*.

Arlitt, M., & Jin, T. (2000). A workload characterization study of the 1998 World Cup Web site. *IEEE Network*, *14*(3), 30–37. doi:10.1109/65.844498

Bakiras, S., & Loukopoulos, T. (2005). Combining replica placement and caching techniques in content distribution networks. *Computer Communications*, *28*(9), 1062–1073. doi:10.1016/j.comcom.2005.01.012

Barbir, A., Cain, B., Nair, R., & Spatscheck, O. (2003). Known content network (CN) request-routing mechanisms. *Internet Engineering Task Force RFC 3568*.

Barford, P., & Crovella, M. (1999). A performance evaluation of hyper text transfer protocols. *ACM SIGMETRICS Performance Evaluation Review*, *27*(1), 188–197. doi:10.1145/301464.301560

Biliris, A., Cranor, C., Douglis, F., Rabinovich, M., Sibal, S., Spatscheck, O., & Sturm, W. (2002). CDN brokering. *Computer Communications*, *25*(4), 393–402. doi:10.1016/S0140-3664(01)00411-X

Broberg, J., Buyya, R., & Tari, Z. (2009). MetaCDN: Harnessing 'storage clouds' for high performance content delivery. *Journal of Network and Computer Applications*, *32*(5), 1012–1022. doi:10.1016/j.jnca.2009.03.004

Buyya, R., Pathan, M., & Vakali (Eds.), A. (2008). *Content delivery networks* (Vol. 9). Springer, Germany.

Buyya, R., Pathan, M., Broberg, J., & Tari, Z. (2006). A case for peering of content delivery networks. *IEEE Distributed Systems Online*, *7*(10), 3. doi:10.1109/MDSO.2006.57

Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, *25*(6), 599–616. doi:10.1016/j.future.2008.12.001

Cameron, C. W., Low, S. H., & Wei, D. X. (2002). High-density model for server allocation and placement. *Proceedings ACM SIGMETRICS*, *02*, 152–159. doi:10.1145/511399.511354

Canali, C., Rabinovich, M., & Xiao, Z. (2004). Utility computing for Internet applications. In Tang, X., Xu, J., & Chanson, S. T. (Eds.), *Web content delivery* (pp. 131–151). Springer.

Cardellini, V., Casalicchio, E., Colajanni, M., & Yu, P. S. (2002). The state of the art in locally distributed Web-server systems. *ACM Computing Surveys*, *34*(2), 263–311. doi:10.1145/508352.508355

Cardellini, V., Colajanni, M., & Yu, P. S. (2000). Geographic load balancing for scalable distributed Web systems. *Proceedings International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'00)*.

Cardellini, V., Colajanni, M., & Yu, P. S. (2003). Request redirection algorithms for distributed Web systems. *IEEE Transactions on Parallel and Distributed Systems*, *14*(4), 355–368. doi:10.1109/TPDS.2003.1195408

Chen, Y., Katz, R. H., & Kubiatowicz, J. (2002). Dynamic replica placement for scalable content delivery. *Lecture Notes in Computer Science, Peer-to-Peer Systems: Revised Papers of 1st International Workshop on Peer-to-Peer Systems (IPTPS'02), 2429*, (pp. 306–318).

Chen, Y., Qiu, L., Chen, W., Nguyen, L., & Katz, R. H. (2003). Efficient and adaptive Web replication using content clustering. *IEEE Journal on Selected Areas in Communications*, *21*(6), 979–994. doi:10.1109/JSAC.2003.814608

Christin, N., & Chuang, J. (2004). On the cost of participating in a peer-to-peer network. *Lecture Notes in Computer Science, Peer-to-Peer Systems III: Revised paper of 4th International Workshop on Peer-to-Peer Systems (IPTPS'04), 3279*, (pp. 22-32).

Christin, N., & Chuang, J. (2005). A cost-based analysis of overlay routing geometrics. *Proceedings IEEE INFOCOM'05, 4*, (pp. 2566-2577).

Christin, N., Chuang, J., & Grossklags, J. (2008). Economics-informed design of CDNs. In Buyya, R., Pathan, A.-M. K., & Vakali, A. (Eds.), *Content delivery networks* (pp. 183–210). Germany: Springer-Verlag. doi:10.1007/978-3-540-77887-5_7

Cohen, R. (2008). Content delivery cloud (CDC). *ElasticVapor: Life in the Cloud*.

Conti, M., Gregori, E., & Panzieri, F. (2001). QoS-based architectures for geographically replicated Web servers. *Cluster Computing*, *4*(2), 109–120. doi:10.1023/A:1011412830658

Day, M., Cain, B., Tomlinson, G., & Rzewski, P. (2003). A model for content internetworking (CDI). *Internet Engineering Task Force RFC 3466*.

Di Stefano, A., & Santoro, C. (2008). An economic model for resource management in a Grid-based content distribution network. *Future Generation Computer Systems*, *24*(3), 202–212. doi:10.1016/j.future.2007.07.014

Elson, J., & Howell, J. (2008). Handling flash crowds from your garage. *Proceedings USENIX 2008 Annual Technical Conference (USENIX'08)*, (pp. 171-184).

Erçetin, O., & Tassiulas, L. (2003). *Request routing in content distribution networks.* (Technical Report). Retrieved from http://digital.sabanciuniv.edu/elitfulltext/3011800000049.pdf

Floyd, S., & Paxson, V. (2001). Difficulties in simulating the Internet. *IEEE/ACM Transactions on Networking*, *9*(4), 392–403. doi:10.1109/90.944338

Fortino, G., Garro, A., Mascillaro, S., Russo, W., & Vaccaro, M. (2009). Distributed architectures for surrogate clustering in CDNs: A simulation-based analysis. *UPGRADE-CN'09, Proceedings 18th IEEE International Symposium on High Performance Distributed Computing (HPDC'09) Workshops*, (pp. 3-10).

Fortino, G., & Russo, W. (2008). Using P2P, GRID and agent technologies for the development of content distribution networks. *Future Generation Computer Systems*, *24*(3), 180–190. doi:10.1016/j.future.2007.06.007

Freedman, M. (2010). Experiences with CoralCDN: A five-year operational view. *Proceedings 7th USENIX Symposium on Network Design and Implementation (NSDI '10)*.

Freedman, M. J., Freudenthal, E., & Mazieres, D. (2004). Democratizing content publication with Coral. *Proceedings 1st USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI'04)*, (pp. 239-252).

Gayek, P., Nesbitt, R., Pearthree, H., Shaikh, A., & Snitzer, B. (2004). A Web content serving utility. *IBM Systems Journal*, *43*(1), 43–63. doi:10.1147/sj.431.0043

Geng, X., Gopal, R. D., Ramesh, R., & Whinston, A. B. (2003). Scaling Web services with capacity provision networks. *IEEE Computer*, *36*(11), 64–72. doi:10.1109/MC.2003.1244537

Gottfrid, D. (2007). Self-service, prorated super computing fun! *The New York Times*.

Hosanagar, K., Chuang, J., Krishnan, R., & Smith, M. D. (2008). Service adoption and pricing of content delivery network (CDN) services. *Management Science*, *54*(9), 1579–1593. doi:10.1287/mnsc.1080.0875

Jiang, W., Zhang-Shen, R., Rexford, J., & Chiang, M. (2008). Cooperative content distribution and traffic engineering. *Proceedings Workshop on the Economics of Networks, Systems, and Computation (NetEcon'08)*, (pp. 7-12).

Kangasharju, J., Ross, K. W., & Roberts, J. W. (2001). Performance evaluation of redirection schemes in content distribution networks. *Computer Communications*, *24*(2), 207–214. doi:10.1016/S0140-3664(00)00316-9

Karaul, M., Korilis, Y. A., & Orda, A. (2000). A market-based architecture for management of geographically dispersed, replicated Web servers. *Decision Support Systems*, *28*(1-2), 191–204. doi:10.1016/S0167-9236(99)00068-8

Kusmierek, E., Czyrnek, M., Mazurek, C., & Stroinski, M. (2007). iTVP: Large-scale content distribution for live and on-demand video services. *Proceedings SPIE'07*.

Kwon, M., & Fahmy, S. (2005). Synergy: An overlay internetworking architecture. *Proceedings 14th International Conference on Computer Communications and Networks (ICCCN'05)*, (pp. 401-406).

Leighton, T. (2009). *Akamai and cloud computing: A perspective from the edge of the cloud* (No. White Paper). Akamai Technologies, Inc. Retrieved from http://www.akamai.com/cloud

Liston, R., & Zegura, E. (2001). Using a proxy to measure client-side web performance. *Proceedings Intlernational Web Content Caching and Distribution Workshop (WCW'01)*.

Lloret, J., Garcia, M., Bri, D., & Diaz, J. R. (2009). Study and performance of a group-based content delivery network. *Journal of Network and Computer Applications*, *32*(5), 991–999. doi:10.1016/j.jnca.2009.03.008

Loulloudes, N., Pallis, G., & Dikaiakos, M. D. (2008). Information dissemination in mobile CDNs. In Buyya, R., Pathan, A.-M. K., & Vakali, A. (Eds.), *Content delivery networks* (pp. 343–366). Germany: Springer-Verlag. doi:10.1007/978-3-540-77887-5_14

MacAskill, D. (2007). *Scalability: Set Amazon's servers on fire, not yours*. O'Reilly Emerging Technology Conference (ETech'07).

Mao, Z. M., Cranor, C. D., Douglis, F., Rabinovich, M., Spatscheck, O., & Wang, J. (2002). A precise and efficient evaluation of the proximity between Web clients and their local DNS servers. *Proceedings USENIX 2002 Annual Technical Conference (USENIX'02)*, (pp. 229-242).

Miller, R. (2008). *Microsoft building own CDN network*. Data Center Knowledge.

Molina, B., Palau Salvador, C. E., Esteve Domingo, M., Alonso Peña, I., & Ruiz Extremera, V. (2006). On content delivery network implementation. *Computer Communications*, *29*(12), 2396–2412. doi:10.1016/j.comcom.2006.02.016

Nguyen, T. V., Chou, C. T., & Boustead, P. (2003). Provisioning content distribution networks over shared infrastructure. *Proceedings 11th IEEE International Conference on Networks (ICON'03)*, (pp. 119-124).

Pai, V. S., Aron, M., Banga, G., Svendsen, M., Druschel, P., Zwaenepoel, W., & Nahum, E. (1998). Locality-aware request distribution in cluster-based network servers. *ACM SIGPLAN Notices*, *33*(11), 205–216. doi:10.1145/291006.291048

Pallis, G., & Vakali, A. (2006). Insight and perspectives for content delivery networks. *Communications of the ACM*, *49*(1), 101–106. doi:10.1145/1107458.1107462

Pathan, M. (2010). *Content delivery networks (CDNs) research directory.*

Pathan, M., Broberg, J., & Buyya, R. (2008). Internetworking of CDNs. In Buyya, R., Pathan, A.-M. K., & Vakali, A. (Eds.), *Content delivery networks* (pp. 389–413). Germany: Springer-Verlag. doi:10.1007/978-3-540-77887-5_16

Pathan, M., Broberg, J., & Buyya, R. (2009). Maximizing utility for content delivery clouds. *Lecture Notes in Computer Science, Proceedings 10th International Conference on Web Information Systems Engineering (WISE'09), 5802*, (pp. 13-28).

Pathan, M., & Buyya, R. (2007). Economy-based content replication for peering content delivery networks. *TCSC Doctoral Symposium, Proceedings 7th IEEE International Symposium on Cluster Computing and the Grid (CCGrid'07)*, (pp. 887-892).

Pathan, M., & Buyya, R. (2009a). Architecture and performance models for QoS-driven effective peering of content delivery networks. *Multiagent and Grid Systems*, *5*(2), 165–195.

Pathan, M., & Buyya, R. (2009b). Resource discovery and request-redirection for dynamic load sharing in multi-provider peering content delivery networks. *Journal of Network and Computer Applications*, *32*(5), 976–990. doi:10.1016/j.jnca.2009.03.003

Pierre, G., & van Steen, M. (2001). Globule: A platform for self-replicating Web documents. *Lecture Notes in Computer Science, Proceedings 6th International Conference on Protocols for Multimedia Systems (PROMS'01), 2213*, (pp. 1-11).

Pierre, G., & van Steen, M. (2006). Globule: A collaborative content delivery network. *IEEE Communications Magazine*, *44*(8), 127–133. doi:10.1109/MCOM.2006.1678120

Presti, F. L., Bartolini, N., & Petrioli, C. (2005). Dynamic replica placement and user request redirection in content delivery networks. *Proceedings International Conference on Communications (ICC'05)*, (pp. 1495-1501).

Pueschel, T., Anandasivam, A., Buschek, S., & Neumann, D. (2009). Making money with clouds: Revenue optimization through automated policy decisions. *Proceedings 17th European Conference on Information Systems (ECIS'09)*.

Qureshi, A., Weber, R., Balakrishnan, H., Guttag, J., & Maggs, B. (2009). Cutting the electric bill for Internet-scale systems. *Proceedings ACM SIGCOMM'09*.

Rabinovich, M., Xiao, Z., & Aggarwal, A. (2003). Computing on the edge: A platform for replicating internet applications. *Proceedings 8th International Workshop on Web Content Caching and Distribution (WCW'03)*.

Rahul, H., Kasbekar, M., Sitaraman, R., & Berger, A. (2006). Towards realizing the performance and availability benefits of a global overlay network *Proceedings 7th International Conference on Passive and Active Network Measurement (PAM'06)*.

Rangarajan, S., Mukherjee, S., & Rodriguez, P. (2003). A technique for user specific request redirection in a content delivery network. *Proceedings 8th International Web Content Caching and Distribution Workshop (WCW'03)*.

Ranjan, S., Karrer, R., & Knightly, E. (2004). Wide area redirection of dynamic content by internet data centers. *Proceedings IEEE INFOCOM'04*.

Rayburn, D. (2009). CDN research data: Market sizing and pricing trends. *Streaming Media West: The Business and Technology of Online Video*.

Shah, P., Pâris, J. F., Morgan, J., Schettino, J., & Venkatraman, C. (2008). A P2P-based architecture for secure software delivery using volunteer assistance. *Proceedings International Conference on Peer-to-Peer Networks (P2P'08)*.

Shaikh, A., Tewari, R., Agrawal, M., Center, I., & Heights, Y. (2001). On the effectiveness of DNS-based server selection. *Proceedings - IEEE INFOCOM*, *01*, 1801–1810.

Sivasubramanian, S., Szymaniak, M., Pierre, G., & Van Steen, M. (2004). Replication for Web hosting systems. *ACM Computing Surveys*, *36*(3), 291–334. doi:10.1145/1035570.1035573

Subramanya, S. R., & Yi, B. K. (2005). Utility model for on-demand digital content. *IEEE Computer*, *38*(6), 95–98. doi:10.1109/MC.2005.206

Tran, M., & Tavanapong, W. (2005). Peers-assisted dynamic content distribution networks. *Proceedings 30th IEEE International Conference on Local Computer Networks (LCN'05)*, (pp. 123-131).

Turrini, E. (2004). *An architecture for content distribution internetworking.* (Technical Report UBLCS-2004-2). University of Bologna, Italy.

Turrini, E., & Panzieri, F. (2002). Using P2P techniques for content distribution internetworking: A research proposal. *Proceedings International Conference on Peer-to-Peer Computing (P2P'02)*.

Vincenty, T. (1975). Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. *Survey Review*, *22*(176), 88–93.

Wang, L., Pai, V., & Peterson, L. (2002). The effectiveness of request redirection on CDN robustness. *ACM SIGOPS Operating Systems Review*, *36*, 345–360. doi:10.1145/844128.844160

Wang, L., Park, K. S., Pang, R., Pai, V., & Peterson, L. (2004). Reliability and security in the CoDeeN content distribution network. *Proceedings USENIX 2004 Annual Technical Conference (USENIX'04)*.

Xie, H., Yang, Y. R., Krishnamurthy, A., Liu, Y., & Silberschatz, A. (2008). P4P: Provider portal for (P2P) applications. *Proceedings ACM SIGCOMM'08*.

Yang, C.-S., & Luo, M.-Y. (1999). An effective mechanism for supporting content-based routing in scalable Web server clusters. *Proceedings International Workshop on Parallel Processing (ICPP'99)*, (pp. 240-245).

KEY TERMS AND DEFINITIONS

Cloud Computing: It is a recent technology trend that moves computing and data away from desktop and portable PCs into computational resources such as large Data Centers ("Computing") and make them accessible as scalable, on-demand services over a network (the "Cloud"). The main technical underpinnings of Cloud Computing infrastructures and services include virtualization, service-orientation, elasticity, multi-tenancy, power efficiency, and economics of scale.

Content Delivery Cloud: It extends the traditional CDN model to harness the power of Cloud computing to deliver cost-effective and high performance content delivery to Internet end-users. Alike the Cloud computing paradigm, content delivery cloud follows a pay-per-usage model to charge the customers for using the storage and bandwidth used to deliver content.

Content Delivery Network (CDN): Content Delivery Networks (CDN), evolved first in 1998, replicate contents over several mirrored web servers (i.e. surrogate servers) strategically placed at various locations to deal with the flash crowds. Geographically distributing the web servers' facilities is a method commonly used by service providers to improve performance and scalability. A CDN has some combination of a content-delivery infrastructure, a request-routing infrastructure, a distribution infrastructure and an accounting infrastructure.

Overlay: An overlay network is built on top of another network. Overlay network nodes can be considered as being connected by virtual or logical links, each of which corresponds to a path, likely through many physical links, in the underlying computer network. Distributed systems such as Content Delivery Network, Content Delivery Cloud, Cloud computing infrastructure, Peer-to-Peer (P2P) networks are examples of overlay networks because their nodes run on top of the Internet.

Request-Redirection: It is a technique commonly used in the World Wide Web (WWW) and in particular in CDNs to direct end-user requests to surrogate replica servers in the face of peak loads. Request-redirection mechanisms are governed by policies that outline the actual redirection algorithm on how to perform server selection in response to an end-user request

Response Time: It refers to the time required for a system to react on a given input. In CDN context, response time is associated with the time for an end-user to be serviced, i.e. receive the requested content.

Throughput: It refers to the average message delivery over a communication channel. In CDN context, it is interpreted as the transfer speed to download/receive content from a CDN replica server.