Université de Toulouse

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le *15/01/2015* **par :**

Saif ul ISLAM

Energy Management in Content Distribution Network Servers.

JURY

Wilfrid Lefer	Professeur d'Université	Rapporteur
Jean-Marc Nicod	Professeur d'Université	Rapporteur
Georges Da Costa	Maître de Conférence	Examinateur
Jean-Marc Pierson	Professeur d'Université	Directeur de Thèse

École doctorale et spécialité :
MITT : Domaine STIC : Réseaux, Télécoms, Systèmes et Architecture
Unité de Recherche :
Institut de Recherche en Informatique de Toulouse (UMR 5505)
Directeur(s) de Thèse :
Jean-Marc Pierson
Rapporteurs :
Wilfrid Lefer et *Jean-Marc Nicod*

Abstract

Explosive increase in Internet infrastructure and installation of energy hungry devices because of huge increase in Internet users and competition of efficient Internet services causing a great increase in energy consumption. Energy management in large scale distributed systems has an important role to minimize the contribution of Information and Communication Technology (ICT) industry in global CO_2 (Carbon Dioxide) footprint and to decrease the energy cost of a product or service. Content Distribution Networks (CDNs) are one of the popular large scale distributed systems, in which client requests are forwarded towards servers and are fulfilled either by surrogate servers or by origin server, depending on contents availability and CDN redirection policy.

Our main goal is therefore, to propose and to develop simulation-based principled mechanisms for the design of CDN redirection policies which will do and carry out dynamic decisions to reduce CDN energy consumption and then to analyze its impact on user experience constraints to provide services.

We started from modeling surrogate server utilization and derived surrogate server energy consumption model based on its utilization. We targeted CDN redirection policies by proposing and developing load-balance and load-unbalance policies using Zipfian distribution, to redirect client requests to servers. We took into account two energy reduction techniques, Dynamic Voltage Frequency Scaling (DVFS) and server consolidation. We applied these energy reduction techniques in the context of a CDN at surrogate server level and injected them in load-balance and load-unbalance policies to have energy savings.

In order to evaluate our proposed policies and mechanisms, we have emphasized, how efficiently the CDN resources are utilized, at what energy cost, its impact on user experience and on quality of infrastructure management. For that purpose, we have considered surrogate server's utilization, energy consumption, energy per request, mean response time, hit ratio and failed requests as evaluation metrics. In order to analyze energy reduction and its impact on user experience, energy consumption, mean response time and failed requests are considered more important parameters.

We have transformed a discrete event simulator CDNsim into Green CDNsim and evaluated our proposed work in different scenarios of a CDN by changing: CDN surrogate infrastructure (number of surrogate servers), traffic load (number of client requests) and traffic intensity (client requests frequency) by taking into account previously discussed evaluation metrics.

We are the first who proposed DVFS and the combination of DVFS and consolidation in a CDN simulation environment, considering load-balance and load-unbalance policies. We have concluded that energy reduction techniques offer considerable energy savings while user experience is degraded. We have exhibited that server consolidation technique performs better in energy reduction while surrogate servers are lightly loaded. While, DVFS impact is more considerable for energy gains when surrogate servers are well loaded. Impact of DVFS on user experience is lesser than that of server consolidation. Combination of both (DVFS and server consolidation) presents more energy savings at higher cost of user experience degradation in comparison when both are used individually.

Keywords: Content Distribution Networks, Energy Management, User Experience

Acknowledgments

At the end of my thesis I would like to thank everyone who made this thesis possible for me.

First of all, I would like to express my deepest, sincere and greatest sense of Appreciation and Gratitude to my supervisor Professor Jean-Marc Pierson for all his enthusiasm, kindness, patience, and outstanding support without which this thesis would not have been possible. I found him always smiling, encouraging and helping a lot whenever I needed him. Briefly, I don't have words to thank you Jean-Marc for being the excellent person you are. It is a great honor for me to work with you.

I acknowledge my gratitude to Konstantinos Stamos for his great help to initialize my work and for his continuous support during my PhD. I would like to express my sincere gratitude to François Thiebolt for his unreserved technical help since the start of my thesis. I am indebted to Thomas Zilio for his great support during my experimental and result oriented phase. It was a great pleasure to work with nice people like you.

I also would like to thank SEPIA team members (George Da Costa, Amal Sayah, Patricia Stolf and Daniel Hagimont) for their fruitful discussions, comments and advices during my stay at IRIT. I am glad to be part of this great team.

I am delighted to pass an excellent time with colleagues and friends. Thank you for unforgettable moments and great company.

I would like to express my gratitude to reporters of my thesis Professor Jean-Marc Nicod and Professor Wilfrid Lefer for making me honor to review my work. I also thank to George Da Costa for taking his precious time to evaluate my work.

I am very thankful to Martine Labruyere and Chantal Morand who contributed to the warm atmosphere which I have experienced during my stay in France.

The Higher Education Commission (HEC) Pakistan, Université Toulouse III - Paul Sabatier and Institut National Polytechnique de Toulouse (INPT) are gratefully acknowledged for their financial support to this study.

I would also pay my great respect and bundle of thanks to all my teachers

throughout my academic career who have a great role for this day.

At a personal level, I wish to especially thank my parents, my brother and my sisters for everything they have done and even sacrificed to help me reach this point.

Finally, and above all my gratitude towards God, who is faithful beyond human comprehension, cannot be expressed by words. Without the help of God, I am sure that this would not have seen the light of this day.

Contents

List of Figures

List of Tables

Part I

Complete dissertation: English version

Chapter 1

Introduction

Contents

Since technological evolution, human life is based on the utilization of machines, sometimes we feel that our life is fully mechanized by machines that are around us. A simple scenario of our daily life starts with a sound of clock alarm to wake up in the morning. A means of transportation to get to work, a coffee machine for drinking coffee, using elevator to go to office etc. Each of these machines consumes energy and the production of this energy emits CO_2 (Carbon Dioxide) and nuclear waste. In the last two decades, we began to feel a large effect of CO_2 and a wide energy consumption. Due to the vast exploitation of machinery, energy prices are becoming more and more expensive. To deal with such an explosive use of machines, and by that the large energy consumption, different research axes, focusing on reduction of energy consumption have been launched. The ultimate goal of this intensive field of research is to provide new energy-aware and environment friendly technologies and mechanisms.

1.1 Why Do We Conserve Energy?

- *To Protect Environment:* It is our social responsibility to change and improve our life style to consume less energy. There are many challenges to our environment e.g. global warming. When we consume more energy, more heat will be produced while consuming less energy, less heat will be ejected to the outdoor environment, ensuring cooler ambient temperature. As a result of that our environment remains green which is a need of today and also requirement for a secure future. Always these types of world issues take greater attention of the communities. Global warming has become a world issue, the efforts and contributions of the corporations are judged and appreciated by communities. The governments and environmental agencies are active to face the challenge of global warming [The Climate Group 2008]. If energy conservation becomes a top priority of any organization then it contributes towards greener environment.

- *Economic Benefit:* Energy cost is one of the major costs of an organization, enterprise and even for a household. The less energy consumption will reduce the overall cost of a product or service. So, the energy conservation helps organizations to reduce operating cost and to be more competitive.

  With the passage of the time, the behavior of the market is changing according to the changes in the needs. Many new business opportunities are being created accordingly. The idea of the reducing energy consumption has also effected the trend of the market. The green products have attracted the manufacturers and customers attention. New products with the green tag are coming into the markets which have created new business opportunities.

- *Equipment Lifespan Prolongation:* Efficient consumption of energy prolongs the lifespan of equipment. For energy conservation, equipment has to be maintained well to operate efficiently. Equipment that consumes less energy will experience less wear and tear. By using energy conservation, equipment operate efficiently with a longer lifespan.

- *Solution to Energy Shortage:* With the new inventions and increasing comfort needs of the communities, energy needs are increasing. Every year usage of energy is increasing. In many parts of the world, energy needs are more than the energy production. These communities are suffering with the problem of energy shortage. If the energy efficient solutions are adopted, it can help to overcome the energy shortage problem and with the same consumption of energy more services can be gained. So, energy conservation provides the economical and environment friendly solution to energy shortages problem.

1.2 Who Does Energy Conservation?

It is not one profession of energy conservation or efficiency, many sectors contribute to make the world more efficient. Computer scientists, chemists, mechanical engineers, electrical engineers, mathematicians etc can contribute to provide smart solutions to minimize energy consumption.

- Economy experts can compare the different products or services with or without energy conservation or efficiency mechanisms and can provide the idea that which product or service is least expensive. They can also provide analysis of the price vs. life of the product. It can guide the customers or organizations to make a choice.

- The energy-aware products and services provide opportunities to the business and commercial communities to contribute towards energy reduction.

- Lawyer community can also guide to the concept of green world by providing their services to make the license process easy for energy-aware solutions.

- Designers can play an important role to make the world green. The systems and the controls can be designed to provide green solutions in a huge amount instead of individual technologies.

- Energy auditors consider how the energy is used in the system. They can recommend the energy-aware techniques and technologies.

- Awareness can be developed in the educational institutions to have a green environment.

- Every individual can contribute to opt for smart mechanisms or to use energy-aware technologies to have a green world. In other words, everyone can do energy efficiency.

1.3 What is Green ICT?

ICT (Information and Communication Technology) contributes more than 2% in the global CO_2 footprint that is increasing day by day [Christensen 2009]. In a business as usual scenario (BAU), CO_2 emissions by the ICT sector are expected to increase from 0.53 billion tonnes (Gt (Gigatonnes)) carbon dioxide equivalent (CO_2e) in 2002 to $1.43GtCO2e$ in 2020 [The Climate Group 2008]. The ICT sector has also been directed to evaluate its impact on environmental changes. It is important to estimate CO_2 emissions from ICT industry and to create opportunities for ICT contributions to enable the low carbon economy. There is intensive need to push the world in this direction.

No doubt ICT has provided a lot of opportunities to sustain the environment but at the same time, it has changed the life style to more energy consuming society. ICT discoveries and developments are directing towards the more resource consuming behaviors which are contributing to change the environment. Hence ICT itself is the part of the problem but also a part of the solution. The term Green ICT is used to address this problematic role of ICT. The aim is to make it environment friendly by changing its overall impact. Basically, the topic Green ICT was used to evaluate the direct affects of ICT on the environment for example its production, use and services but nowadays it is used as term to use the ICT to make the things environment friendly in other domains as well. This is because of the expanded use of the ICT everywhere.

1.4 Area of Interest, Objectives and Contributions

Internet has made the communication faster and easy. The basic idea of Internet was to serve as a heterogeneous network of networks to connect different entities for communicating in a best effort way. Today's Internet has got mature to the point that web browsing and e-mail are no longer the main features. Content providers and e-commerce organizations view the Internet as a tool to make available their rich contents to their widely dispersed customers (as shown in Figure 1.1). Internet users are also increasing significantly every year (as presented in Figure 1.2). Therefore, there is very fast increase in the Internet infrastructure. Today's Internet is comprised of more than $13,000$ autonomous networks [Sitaraman 2014]. This change caused a trend towards the grand, geographically distributed systems [Qureshi 2009]. These systems can have a huge amount of servers and many data centers. Millions of watts of power may be required to run a large data center [Katz 2009] as shown in Figure 1.3. Internet providers are installing more energy consuming devices in order to provide better services. Energy cost of the Internet infrastructure is increasing every year, that affects the both organizations and their customers as well. According to [Gyarmati 2010], 15% of data centers' cost is in terms of energy consumption. One of a popular large scale distributed systems for content delivery Akamai, is estimated to spend a $10M (Millions) of electricity cost every year [Qureshi 2009]. Reduction in energy consumption may play an important role to decrease over all cost [Vasić 2010].

To explore the energy consumption and its reduction in large scale distributed systems is a hot research area [Orgerie 2013], [Chiaraviglio 2010], [Pierson 2013], [Da Costa 2009], [Hlavacs 2009] with important practical applications. A popular type of such a network is the Content Distribution Network (CDN) [Pallis 2006]. CDN is an overlay network (as illustrated in Figure 1.4). A CDN is responsible for managing the large amount of content traffic originating from the Web users. A CDN consists of a set of surrogate servers geographically distributed in the Web, which contain copies (replicas) of content belonging to the origin server (according to a specific storage capacity). Taking example of Akamai, a large scale content

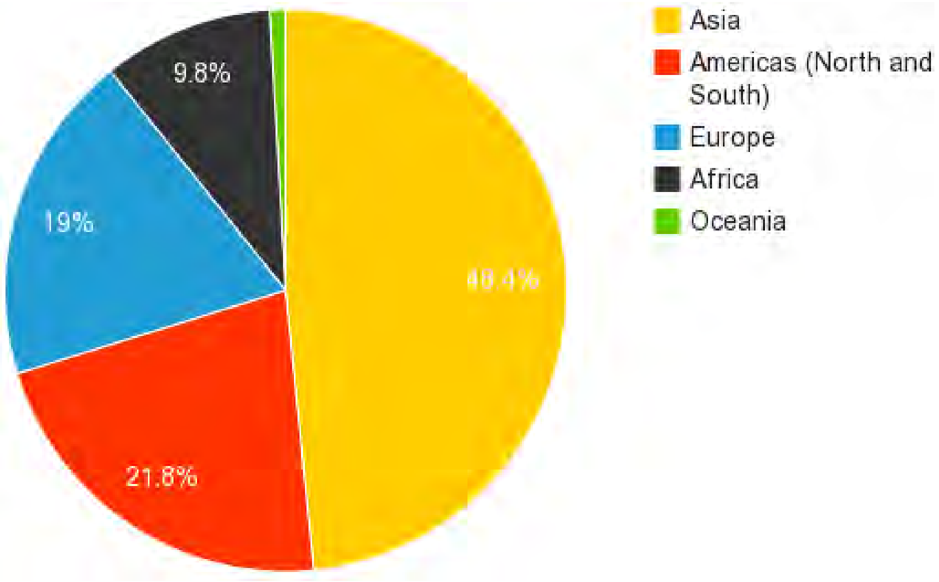


Figure 1.1: Worldwide Internet users by region [live stats 2014].

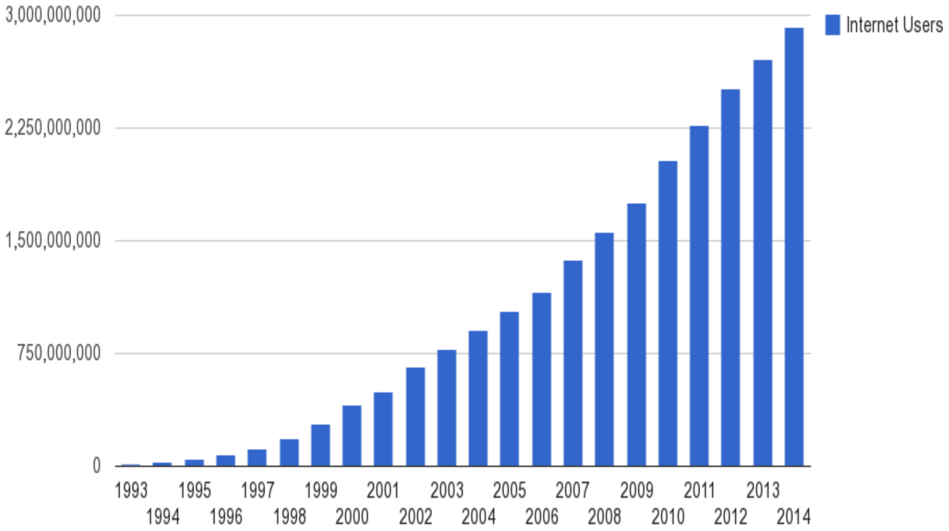
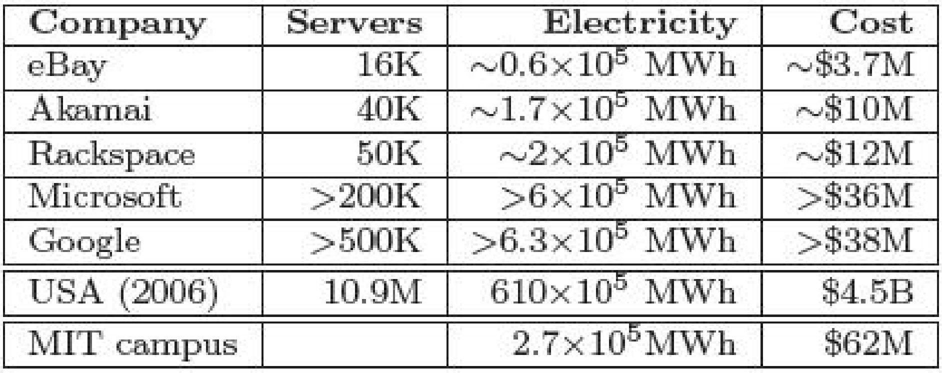


Figure 1.2: Worldwide Internet users [live stats 2014].



Company	Servers	Electricity	Cost
eBay	16K	~0.6×10^5 MWh	~$3.7M
Akamai	40K	~1.7×10^5 MWh	~$10M
Rackspace	50K	~2×10^5 MWh	~$12M
Microsoft	>200K	>6×10^5 MWh	>$36M
Google	>500K	>6.3×10^5 MWh	>$38M
USA (2006)	10.9M	610×10^5 MWh	$4.5B
MIT campus		2.7×10^5MWh	$62M

Figure 1.3: Estimation of annual electricity cost for companies with large infrastructure [Qureshi 2009].

provider has deployed more than 100, 000 surrogate servers in over 1150 networks around the world (in over 80 countries) [Sitaraman 2014]. The main idea is to bring content replicas closer to the user. Therefore, CDNs act as a network layer between the origin server and the user, for handling their requests. With this approach, content is located nearer to the user, yielding low response time and high content availability since many replicas are distributed. The origin server is relieved from the requests since the majority of them are handled by the CDN servers. A typical

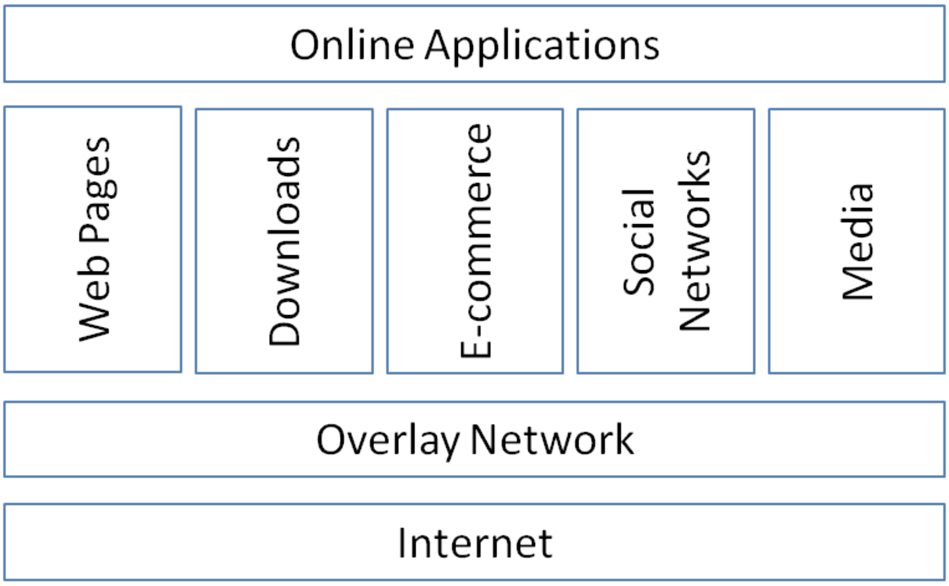


Figure 1.4: An overlay network is built on the top of Internet to work as a bridge between modern application requirements and the Internet basic services.

CDN (as shown in Figure 1.5) includes following functions:

- *Content Distribution Services:* Geographically distributed set of surrogate servers, which store the data on behalf of the origin server or using data replication.
- *Request Redirection and Content Delivery Services:* Client requests are directed towards the closest servers either in terms of proximity or load.
- *Cooperation Services:* Surrogate servers may cooperate with each other in order to fulfill the client requests. If a client sends content request to a server, the server checks for the content in its cache: If the content is not available in its cache it forwards the request to the neighboring surrogates or to the origin server.
- *Management Services:* Services to control utilization of the contents, managing copy rights etc.
- Content adaptation services, e.g. format conversion.

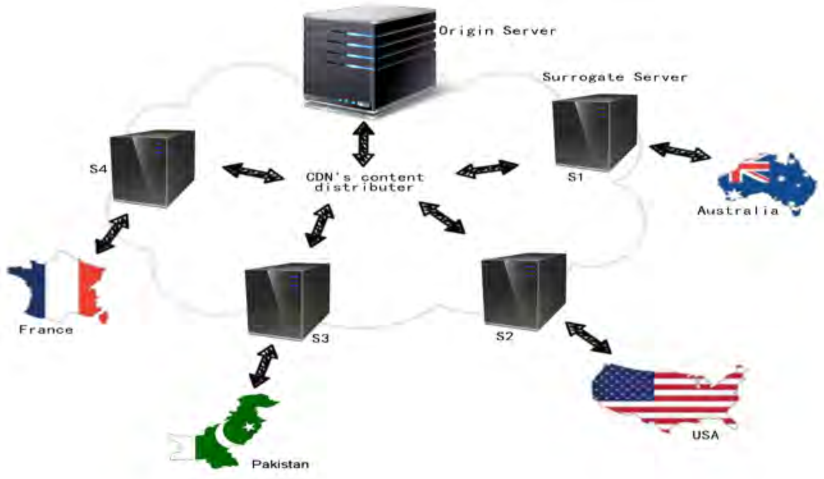


Figure 1.5: A typical CDN architecture.

However, improvement in users' satisfaction comes at the cost of increased energy consumption mainly originated from the surrogate servers activity. The cited works

[Feldmann 2010], [Lee 2010], [Chiaraviglio 2010], [Blackburn 2009] address how the underlying network activity, in conjunction with content delivery, interacts with the energy consumption, but there is not much focus on the CDN redirection policies in particular. Therefore, our key motivation lies on finding a delicate balance between users' satisfaction and reduction in CDN infrastructure energy consumption. We aim at defining an energy-aware forwarding strategy that enhance previous work [Stamos 2009], including energy savings, relying on utilization model of the surrogate servers. We focus on energy reduction in CDN at surrogate server level by working on CDN redirection policies. Our work doesn't include to analyze the energy conservation at network level and is considered for future work. So, this research lies to explore, propose and to develop energy-aware mechanisms and techniques in CDN and analyzing their impact on the user experience.

1.4.1 Objectives

The objectives of our research include:

- Identifying the right CDN metric to find the energy consumption in CDN.
- Identifying and analyzing the energy consumption in a CDN.
- Modeling the energy consumption in a CDN.
- Exploring energy conservation opportunities in CDN systems.
- Identifying and implementing the techniques of energy conservation which can be applicable to implement in a CDN environment.

1.4.2 Contributions

The main advances of this research are the following:

- Deriving an energy consumption model from the surrogate servers' utilization in a CDN.

- Proposing and developing Energy-aware CDN redirection policies by applying the energy conservation techniques (Consolidation and DVFS (Dynamic Voltage Frequency Scaling)) in traditional CDN redirection policies.

- Developing a simulator to integrate energy concerns and energy conservation techniques to evaluate our proposed concepts.

- Studying the impact of CDN infrastructure size i.e. number of surrogate servers participating in a CDN, traffic size i.e. number of client requests and load intensity i.e. frequency of client requests on energy consumption and user experience, to evaluate the behavior of our proposed policies in different CDN scenarios. And then to compare these policies, to help a CDN owner to make the choice of appropriate CDN redirection policy in a particular CDN scenario (depending on the user's and his own requirements).

- Deriving from these studies, some perspectives as potential energy saving techniques that preserve energy while respecting a certain level of quality of services.

1.5 Thesis Organization

Thesis report is organized as follows,

- In the next Chapter, literature review is presented, in which energy consumption models and energy conservation techniques, developments and mechanisms are discussed. Starting from the basic concepts, energy measurements and profiling techniques are discussed which describe different energy measurement techniques and various types of component or system level energy and power consumption models. Later on, Dynamic power management i.e. DVFS, its applications, benefits and drawbacks are discussed. After that, energy conservation methods in a pool of servers are discussed. The chapter is ended with a detailed review of energy reduction in content delivery mechanism.

- Chapter 3 presents the CDN simulation environment. A simulation test-bed CDNsim is discussed in detail. Different input/output parameters (existing and newly proposed), existing CDN redirection policies, evaluation metrics (existing and newly proposed (including their justification)) and our contributions to the CDNsim are exhibited in detail. Also, it shows how the different evaluation metrics are affected by different input parameters. It ends with the presentation of data set we used to perform our simulations which are presented in the next chapters.

- In Chapter 4, CDN redirection policies (traditional and energy-aware) are proposed. CDN utilization, CDN utilization and energy consumption models (with and without DVFS) are exhibited. Also, energy-aware techniques (DVFS, server consolidation) and mechanisms along with their implementation in CDN are illustrated and evaluated. Moreover, CDN redirection policies are evaluated presenting the impact of CDN infrastructure and traffic size on different evaluation parameters.

- Chapter 5 compares the proposed CDN redirection policies (Traditional and/or energy-aware) impact on different evaluation metrics.

- Chapter 6 states, at what extent the frequency of the client requests traffic affects the CDN operations. It compares the proposed policies based on the evaluation metrics in the previously discussed context.

- Chapter 7 concludes the thesis adding some future potential research prospects.

Chapter 2

State of the Art

Contents

This chapter describes some research work done in the context of our area of interest. The main purpose of this chapter is to provide the background of our work to have better understanding before presenting the more explanatory and result oriented sections. In order to be precise, we start from some basic concepts leading towards some related aspects. We have defined some basic terms used in our proposed context. We discussed the models used to calculate energy consumption on component and system level. Then we describe the techniques used for the energy reduction in computer systems. We have discussed the previous work done to explore the energy conservation in cluster of servers. At the end, before concluding the chapter, we have investigated the mechanisms used for energy conservation in content delivery architecture. We have explored the previous work linking with our work and provided necessary comparisons and discussions as well.

2.1 Basic Terms

It is important to describe some basic terminologies before proceeding towards more details.

- *Energy:* In computer systems, it is referred as the electricity resource that can power the hardware devices for doing computation during a certain time. Energy is measured in Joules. In research, the term energy is mostly used in the mobile and data center platforms. In case of mobile devices, energy is referred to the battery lifetime while in case of data centers, it is mostly concerned with the electricity cost.

- *Power:* Power is the rate at which work is done, or energy is dissipated. Power is measured in Watts. Power is used to present the current delivery and voltage regulator of the circuits. In case of system research, power can also be referred for the abstract concepts e.g. process and operating system. For example, if we state that the power of a process "A" is 1W (Watt), it means that the execution of "A" causes the hardware circuits to use 1W of power.

- *Static Power:* This is the power produced when the transistors are not completely turned-off. It is specified as the power that is needed by a device when it is inactive.

- *Dynamic Power:* It is referred to the power that is needed for the working of a device. It is occurred because of switching of the capacitance voltage states.

2.2 Energy Measurement and Profiling Techniques

In this section different existing energy measurement and profiling techniques are investigated. [Benedict 2012], [Chen] and [Trobec 2013] described the recent techniques to measure power consumption in large scale distributed systems. According to them, energy measurement and profiling can be majorly categorized into hardware and software. According to hardware-based energy measurement approaches, different instruments can be used to measure the energy consumption, communication or storage segments e.g. CPU (Central Processing Unit), racks of data centers, motherboards. These instruments can be implemented as meters, special hardware devices usually embedded in hardware platforms or as power sensors put on the hardware. However, hardware-based techniques use expensive sensors to measure the energy consumption, also a good knowledge of the hardware design is required. According to the software-based energy measurement techniques, energy models are developed to evaluate the energy consumption. These energy models are used to estimate the energy consumption at different levels i.e. hardware component, program block, process, instruction or system level etc.

[Rivoire 2008] presents a constant power model. It doesn't take into account the resource utilization and predicts the constant power. This model provides the base for the utilization-based models. Moreover, linear CPU dependent model was also proposed [Fan 2007]. According to them, power consumption is predicted according to the utilization of the CPU of a node. Some works like [Heath 2005] uses the linear model to measure the CPU and disk energy consumption according to their usage. Number of disk transfers is used as the parameter by the disk transfer models to estimate the dynamic energy consumption of the disk. Moreover, [Vereecken 2010] proposed CPU, hard disk and network interface card utilization model where utilization of the different components is mapped to the energy consumption. [Economou 2006] presents the performance counter model where the energy consumption of a system is measured according to the system's performance counters such as utilization of floating point unit, amount of instruction level parallelism or the activity of the cache hierarchy. Additionally some models were proposed to estimate the energy consumption of different architecture styles i.e. peer-to-peer, client/server, publisher/subscriber [Seo 2009].

Load models concentrate on the running hosts only and are sufficient to predict the energy consumption for the servers which are active 24/7[Berl 2011]. Load models take into account the utilization of the servers. After turning the servers on, their active energy consumption is function of their load. We adopted this form of model which fits our case.

2.3 Reducing Energy Consumption in Computing Systems

Energy conservation has a key importance in computing systems because lower power results in lower operating costs, lower fan noise and lower cooling needs. Increase in energy consumption in large-scale distributed systems (e.g. clusters, CDNs (Content Distribution Networks), grids and clouds) raises economical and environmental issues. Proposing and designing new energy-efficient techniques and methods at all levels of distributed architectures, to minimize energy consumption is an issue of high importance [Shuja 2012] [Beloglazov 2011] [Da Costa 2009] [Chang 2003] [Contreras 2005]. Energy can be decreased at different levels of distributed system architecture e.g. on hardware level, at the network level, at the middle-ware level or at the application level etc. [Shuja 2012] [Beloglazov 2011] [Da Costa 2009] [Chang 2003] [Contreras 2005] [Orgerie 2013]. Moreover, energy consumption of computing systems can be defined as the summation of static and dynamic energy consumption. In order to reduce energy consumption of a device, a system or a system of systems: It is necessary to reduce its static, dynamic or static and dynamic energy consumption. By keeping in view previously discussed concepts and to be precise, in this section we have presented related existing work describing energy reduction techniques used at device (particularly CPU), cluster of servers and content distribution level to reduce static, dynamic or static and dynamic energy consumption.

2.3.1 Dynamic Voltage Frequency Scaling (DVFS)

In order to get the higher system performance, we need to increase the operating frequency, or using the powerful ICs (Integrated Circuits). When voltage of a CPU is minimized, its power consumption is substantially reduced. Performance is affected by this process. Frequency of a CPU is changed approximately proportionally according to the change in its voltage ($f \propto V$). Hence, $P \propto V^2 f$, where P denotes power consumption of a CPU. When power is divided by frequency, we get energy per cycle. Energy is proportional to square of frequency $E \propto f^2$. So a CPU can consume less energy when running on lower speed [Lorch 2001].

Hence, increase in performance level inevitably causes the increase in energy consumption. We can have the energy savings at system level by using the low static power consumption devices or by controlling the system operation according to the processing load.

One of the popular energy reduction techniques is known as DVFS (Dynamic Voltage Frequency Scaling), also called dynamic speed scaling or dynamic power scaling. The voltage supply to the different components like CPU, main memories, local buses etc is controllable. The power consumption of a processor depends on the supply voltage and frequency. Power-aware mechanisms try to reduce energy consumption according to the appropriate moments and enhances the energy savings keeping in view the system performance. Energy saving algorithms are proposed for dynamically varying processor clock speed or reducing the supply voltage and hence saving energy as described in [Venkatachalam 2005a], [Flautner 2002], [Venkatachalam 2005b], [Flinn 2000], [Govil 1995], [Grunwald 2000], [Lorch 2001], [Mochocki 2006], [Pettis 2004] and [Pouwelse 2001].

2.3.1.1 Preliminary Work.

DVFS is applicable in different scenarios, one of them is the moment when the processor has a lower utilization. Weizer et al. [Weiser 1994] proposed one of the early approach based on general purpose operating systems. They consider a collection of interval based algorithms i.e. OPT, FUTURE and PAST. They evaluated these algorithms by the help of traces which were collected from the systems which lie on the Unix operating system. On these systems, engineering applications were running. According to their approach, time was divided into intervals. These intervals had fix length. For each interval they notice the frequency of the clock. At the start of each interval, the frequency and the voltage of the CPU has been considered. The purpose was to complete majority of the work at the end of the intervals. Their approach considers the CPU utilization ratio to determine the status of CPU under-utilization. When the CPU is underutilized, it requires lower frequency. Among their proposed algorithms, PAST doesn't require the future knowledge to work. But it remained unimplementable because of its requirements of information about the work done in the previous time intervals, hence the work of Weizer et al. has some inconveniences which made it hard to implement on general purpose systems.

In 1995 Yao et al. [Yao 1995] presented a job scheduling DVFS approach for real time systems. They proposed some off-line and on-line algorithms. They considered the single processor with variable speed to conserve energy, where every tasks are accomplished during the start and the end time with the changing speed of the processor.

2.3.1.2 Advanced Work.

The complexity in work of Weizer et al. was tried to be removed later-on by [Pering 1998] and [Grunwald 2000] by modifying the original algorithms.

These basic works motivated the researchers to progress the research in this direction. Now-a-days, most of the existing processors support DVFS e.g. AMD (Advanced Micro Devices) Athlon and Mobile K6 Plus, Intel Xeon, Pentium-III with SpeedStep technology [Intel Corporation 2000] and Strongarm SA-2 [HEEB 2000].

- ***Operating System level Frequency Scaling:*** With the advancements in DVFS supporting technology, some standards are also introduced e.g. ACPI (Advanced Configuration and Power Interface) [Hewlett-Packard 1999] released in 1999. The purpose was to introduce the energy-awareness at operating system level. The approach of ACPI is to determine CPU performance states. A CPU can have different performance states according to ACPI. When a CPU is working, it is considered to be working in one of these determined performance states. These performance states are co-related with the power states. ACPI existence created the opportunity for DVFS oriented algorithms to be compatible with different types of CPU.

- ***Load level Frequency Scaling:*** According to [Don Domingo 2010], the frequency and the voltage of a processor can be varied according to its load. Linux kernal provide five different modes of DVFS. These modes are performance, powersave, ondemand, userspace and conservative. According to performance mode, the frequency of a processor works at the highest rates, set by the governor. In this mode, the processor frequency will not be changed. Though this mode is useful when there are higher loads or the systems are utilized throughout the time. This mode doesn't offer power saving but provides the higher performance. Powersave governor offers the opposite functionality to the performance one. According to this mode, the processor frequency is set to the minimum possible value. Though this mode provides higher power savings but at the cost of lower CPU performance. It also resolves the overheating problem. But this mode is not useful during the period of higher loads where the energy consumption can be higher. So it is better to set this mode in the conditions where the lower load periods are going to happen. Ondemand mode provides the frequency switching function to the processor. During higher loads, frequency of the processor is kept maximum and is switched to minimum when the system has no load. It can offer different possibilities to manage the different problems of the system like power consumption, heating emissions and performance etc by switching the clock frequency according to the lower and higher loads. Userspace governor provides the facility to the user-oriented programs to set the frequency according to their needs. Depending on its better use, this mode can manage the things in a better way to handle the energy concerns. Conservative mode is less aggressive than the ondemand mode and changes the clock frequency according to the load gradually. Some of these techniques were presented by Guerout et al in [Guerout 2013] keeping in view the grid and cloud environment simulation using CloudSim simulator.

- ***Frequency Scaling in Other Components:*** Dynamic Speed Scaling (DSS) is also done in different components e.g. multi-speed disks [Pinheiro 2004], [Gurumurthi 2003] and [Carrera 2003], memory power management via DVFS [David 2011]. Similarly at network level, the idea of adapting rate according to the load is applied for energy conversation. The idea of reducing power consumption in wired network links and network devices was first studied in [Gupta 2003]and [Christensen 2004]. For example Adaptive Link Rate (ALR) addresses the concept of changing the link rate dynamically and adapting it to the utilization of network. This technology allows Ethernet data link to adjust its speed and ultimately power to traffic levels [Gunaratne 2005], [Anand 2006], [Gunaratne 2006].

2.3.1.3 DVFS Limitations.

Energy reduction techniques try to reduce energy but it doesn't mean that there will always be reduction in the energy consumption by applying the energy saving techniques. An execution of a task on reduced speed doesn't always result in reduced energy consumption [Shekar 2010]. Similarly reduced power doesn't necessarily cause reduced energy consumption. The fact is that energy consumption doesn't depend only on the power. Energy consumption includes the execution time also. So, working on the lower power for a long time can result into the higher energy consumption. Similarly, working on the higher power for a smaller time can result into the lower energy consumption. But, execution time of the task may not be inversely proportional to the clock frequency and DVFS may result in non-linearity in the execution time [Buttazzo 2002] e.g if the task is memory or I/O bounded then speed of the processor will not have a dramatic effect on the time of execution. Also, reducing the processor speed may lead to the changes in the order of scheduled tasks [Venkatachalam 2005a].

There is also an other impact of processor frequency change on the performance of processor. According to [Zhu 2004], the use of DVFS degrades the reliability of the processor. In short, we can gain substantial energy savings by using DVFS technique but it is important to apply it carefully, as the results may change for different system architectures and applications loads.

2.3.1.4 Discussion.

We opt for the concept of changing the rate of devices according to the load. In our work, we target the processor as a device to apply the DVFS. As we are concerned with the CDN, so we take the surrogate server processor and apply the DVFS technique according to load of surrogate server which is directly proportional to the number of simultaneous connections. The previously discussed techniques focus on the energy reduction by DVFS in processors generally while we focus on the surrogate servers in a CDN in particular. We approach the problem with the CDN simulation.

Up to our knowledge, there is no work that addresses the DVFS technique in a CDN simulation environment. Moreover, some works are similar to our approach with considerable differences for example [Don Domingo 2010] that presented the different DVFS techniques in Linux Kernel. Their work is more practical in PCs (Personal Computers), for example they presented a userspace technique where the user oriented programs settle the frequency of the processor according to their needs. In CDN, the surrogate servers are not controlled by the users and users can just access the contents from the servers. However some of their approaches can also be practical for the server systems e.g. powersave, ondemand modes.

According to the previously discussed approaches, DVFS modes can be divided into two major categories i.e. static and dynamic. Static modes can be presented as the modes where the processor frequency works on the same scale and doesn't change whether other factors (e.g. load) are static or dynamic. Dynamic mode can be said as the mode where the processor frequency changes according to a factor (e.g. load). Dynamic mode changes can be mapped to the thresholds as well. In previous works, while considering the static mode of the DVFS, mostly two extreme modes are considered i.e. the mode where the processor works at maximum frequency and the mode where the processor works always on the minimum possible frequency e.g. performance and powersave modes in [Don Domingo 2010]. It can be an interesting idea to be moderate between these two extreme modes and analyzing its impact on the power-saving. That motivated us to propose a DVFS mode where the frequency of the processor always work on medium scale.

One of the interesting DVFS approach is presented in [Don Domingo 2010] i.e. ondemand approach: where the processor frequency works on minimum scale when the system is idle or at maximum when system has some load. No doubt, this approach minimizes the static energy consumption but it doesn't give the opportunities to conserve dynamic energy consumption because it doesn't consider frequency changes when system has different loads. In contrast to this approach, we proposed the technique of frequency adaptation according to thresholds of the system load that makes our work more practical and useful to enhance the static as well as dynamic energy saving. Additionally, we focus not only on the DVFS to reduce the energy consumption but also we present how the DVFS affects the user experience that makes our approach more practical for a CDN environment where the user experience is an important factor.

2.3.2 Energy Reduction in Cluster of Servers

Researchers are doing work on minimizing the energy consumption in clusters of database and web servers. The purpose is to minimize the power consumption and analyzing its impact on user experience. Ultimate goal is to find different ways for the services, which are acceptable for both customers and the service providers in terms of energy cost and quality of service. Energy consumption in such type of systems, mostly depends on the utilization of CPU along with the part of power consumption of memory, network devices, hard-disks etc. A server without any load can still consume 60% of its peak power [Chase 2001]. So, in order to have maximum possible energy reduction in such systems, it is important to power-off or down the servers as a function of their utilization. Different works have been proposed to attain this goal.

Chase et al. [Chase 2001] proposed energy-aware mechanisms. According to their policies, in a large cluster of servers, resources are allocated on the base of economic criteria by keeping in view the energy consumption and user experience. Resource allocation is managed by a request dispatcher which focuses on the minimum number of active servers for incoming requests. The rest of the servers are kept in a low power idle mode.

Sharma et al. [Sharma 2003] exhibited feedback control mechanism to control application level service demands. They proposed adaptive algorithms for Dynamic Voltage Scaling (DVS) in QoS-enabled web servers by keeping in view the service delay problem while minimizing energy consumption. Dovrolis et al. [Dovrolis 1999] proposed a feedback controller to handle the response time in cluster of servers. Sharma et al. [Sharma 2005] proposed a mechanism for thermal load-balancing using load monitoring and dynamic workload provisioning. [Nedevschi 2008] compares the utilization of hot data centers that should be cooled with cool peers.

Chen et al. [Chen 2008] discussed related dynamic provisioning problem. They concentrated on long-lived TCP (Transmission Control Protocol) connections e.g. in case of Skype, MS (Microsoft) Messenger, online gaming etc. They proposed techniques which include dynamic provisioning and load dispatching. Their technique proposes to turn on minimum number of servers which can be enough to satisfy the user requests while keeping in view the user experience. They proposed a CPU load-oriented power model. They also proposed a model to measure the performance according to the number of connections and log in rate. Moreover, they used a model for the prediction of the load. They use all these models to develop the provisioning techniques. [Urgaonkar 2008] proposed a dynamic provisioning algorithm. They focus on a platform which can host many applications. They use a queuing model for analyzing the system. The provisioning algorithm exhibits a predictive strategy and analytics model which focuses on the minimum set of servers. They also take into account the user experience. Their technique predicts the future load based on the long time intervals. In order to avoid the complexities in their technique which can be caused by the sudden load variations, they also used a reactive provisioning at short time intervals.

Goiri et al. [Goiri 2010] [Comellas 2010] presented that reduction in the online systems decreases the energy consumption and they tried to find the compromise between power savings and online machines. According to them, they used two thresholds i.e. minimum working nodes threshold and maximum working nodes threshold. Minimum working nodes threshold determines when the service owner can power-off the nodes. Maximum working nodes threshold determines when the service provider should power-on the new nodes. By evaluating the energy consumption and the user experience, different analysis can be obtained.

On the basis of previous work, [Berral 2010] presented a mechanism which proposes an intelligent consolidation framework by applying different techniques e.g. turning on/off nodes, power-aware consolidation policies and machine learning techniques in order to optimize the previous technique. [Kamitsos 2010] also exhibited the turning on/off technique and proposed to put the idle machines in a low power consuming mode to gain energy reduction. In order to get this functionality, they used Bellman's function to consider when to set the unnecessary machines into sleeping mode while managing rest of the jobs in the active machines.

[Lu 2013] tried to explore the energy saving by proposing the decentralized online dynamic provisioning algorithms. They did dynamic provisioning with and without future knowledge about the workload and found that full size window future workload knowledge will not enhance the performance of dynamic provisioning. They suggested that without or with partial future workload information, there is possibility to reduce the energy consumption in cluster of servers.

[Pinheiro 2001] proposed power-aware algorithm at application and operating system level. Their proposed algorithm is able to do load-balancing and load-unbalancing at cluster level. They used energy-aware technique of powering-off servers during lower loads and keeping them on while higher loads. They focus on clusters of the PCs or workstations in the context of clusters of web server and compute server. According to their method, the nodes of clusters are turned-on and turned-off dynamically. They use the load-balancing technique to use idle servers while the load-unbalancing technique was used to concentrate on some nodes and making the other nodes idle to consider them for powering-off. Their approach is closer to our work in the sense that we both consider the load balancing to use the idle servers and load-unbalancing to have the under-utilization in servers to consider them for turning-off. They exhibit the power and performance trade-off. For the performance they took throughput and execution time as the metrics. But the major difference in our work is that we emphasized on CDN servers. Similarly, we use DVFS but they didn't consider this important technique to have the maximum energy savings even doing load-balancing. Though they focused on the systems performance vs energy gains but we consider the user experience which is very important in CDN case. Their work was then extended by Elnozahy et al.

[Elnozahy 2003] who proposed the global (cluster level) and individual (server level) energy conservation policies. They proposed five different algorithms to manage the power reduction in cluster of servers. These five algorithms are the combination of two popular energy reduction techniques 1) Dynamically power-on and power-off servers 2) Dynamically changing the voltage of processor. They are the one who first time combine these two energy reduction techniques in cluster of servers. Their policies are: Individual voltage scaling (IVS), coordinated voltage scaling (CVS), vary-on vary-off (VOVO), combination of IVS and VOVO, and fifth algorithm combination of VOVO and CVS. The first policy IVS, uses the concept of frequency and voltage changes according to the workload of a cluster node. CVS policy also does the frequency scaling like IVS but CVS does this operation in a coordinated way. According to CVS, the nodes coordinate for voltage scaling and work close to average frequency setting in the cluster. According to them, for this purpose a monitor can be used that computes the frequency setting of all the nodes in a cluster and then broadcasts it to restrict all the nodes to work closer to that average frequency for a given interval of time. Though according to the author, CVS is supposed to do better in energy savings but it is more complex to implement. VOVO policy is actually the same policy as investigated by [Pinheiro 2001] where the nodes of the clusters are powered-on and powered-off dynamically to reduce the energy consumption of system. VOVO-IVS combines the functionality of IVS and VOVOS to gain maximum energy saving by taking into account the local and global measures. VOVO-CVS algorithm is combination of VOVO and CVS policies for energy gains but it is considered the most complicated policy to implement. Their work is similar to our work and we use both powering-off servers and DVFS techniques and their combinations. Along with these similarities, there are considerable differences. Our focus is on the energy reduction in CDN environment particularly in the CDN servers or surrogate servers. We use the load-balancing and load-unbalancing using Zipfian distribution. We applied DVFS technique while balancing the load among CDN servers and combined it with powering-off servers while doing load-unbalancing. Their global DVFS technique i.e. CVS is more complicated to implement and also increase the overhead of installing load monitor.

CVS computes the average load of all the nodes and then restrict them to work on corresponding frequency. As a result of that it also affects performance and causes delay process while computing the load. In comparison, our global DVFS technique doesn't require all this overhead and complications and is very practical because of its simplicity. In our global DVFS approach, for a CP (Content Provider), there is only a single requirement to determine the different load periods and then according to the algorithm, all the nodes work on the appropriate frequencies i.e. Highest, medium or lowest. It also prevent the burden of frequent fluctuations of the processor frequencies as well. Moreover, our two local adapting DVFS policies provide the opportunity to save the energy on the individual server level by using three thresholds which helps to decide to change the processor frequency according to the server load.

2.3.2.1 Discussion

In previous works, two of the main approaches or their combination is adopted to achieve energy minimization in cluster of servers i.e. Global (sleeping/turning-off servers) and local or server level (DVFS). Most of the previous techniques target the different types of cluster of servers while our emphasis is particularly on the CDN. Though some of these works also discuss the trade-offs between energy saving and some aspects of quality of experience but mainly it concentrates on the energy reduction and system performance. We analyze the energy saving and its affects on the user experience and performance of the CDN. Our work combines the different approaches of the several previously discussed works and adds the new techniques to reduce the energy consumption in distributed systems. Some of the approaches prefer to keep some of the servers idle while lower load periods but to have the maximum savings, we determine to turn-off the servers as the servers in idle mode also consumes a considerable amount of energy. We also consider to minimize the cooling cost by adjusting a threshold of the servers by avoiding them to be fully loaded to overcome the problem of over heating which also increases the cooling cost.

2.3.3 Reducing Energy Consumption of Content Delivery

Different content delivery architectures are proposed and studied by researchers. Reducing energy consumption of content delivery has become an area of high importance because of its explosive increase. Also, trade-offs among content delivery architectures were analyzed inside several works.

2.3.3.1 Content Centric Networking(CCN).

Jacobson et al. [Jacobson 2009] [Jacobson 2012] proposed a new architecture, Content Centric Networking (CCN), also called as named data networking, content-based networking or data-oriented networking. According to them "Accessing content and services requires mapping from the what that users care about to the network's where." Its main objective is that a communication network must permit an end user to concentrate on the required data rather than giving a reference from where that data is to be accessed. The purpose is to change the current networking communication model by replacing the machines with the content as shown in Figure 2.1.

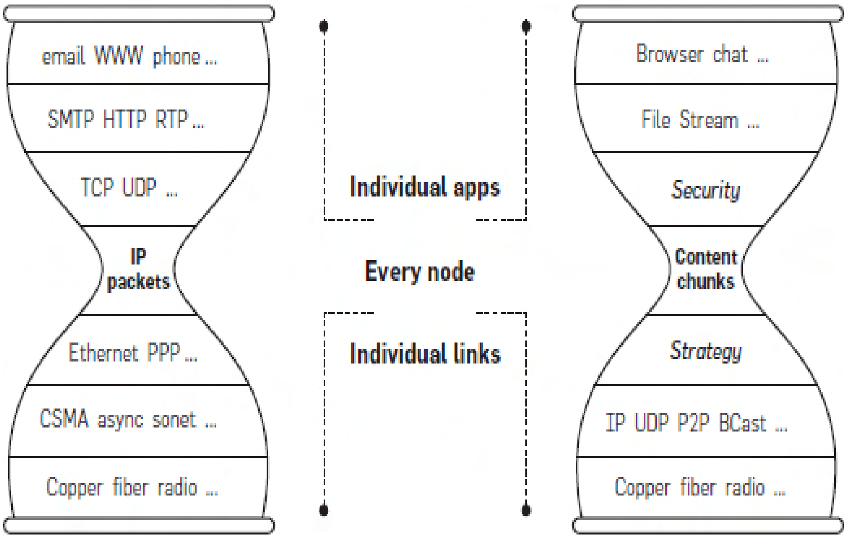


Figure 2.1: CCN network communication model [Jacobson 2012].

In [Lee 2010], a network architecture problem is investigated. Authors proposed an architecture based on CCN to reduce the power consumption. They tried to find that change in network architecture from host-oriented to CCN can open new possibilities for energy-efficient content dissemination. CCN concentrate on networking devices i.e core/edge routers and optical multiplexers to find the opportunities for energy reduction instead of content servers(in data centers i.e. CDN or in user promises i.e. P2P). In this paper, authors present an energy-efficient content router architecture ranging from core routers to home gateways. In this work, authors didn't take into account to apply energy efficient techniques which can be interesting to get more energy savings e.g. routers are not enabled to dynamically turn on/off. Sleeping of routers or some ports of the routers can be switched-off to gain more energy savings. Though the CCN technology provides a novel architecture with some handsome advantages like energy conversation, network load reduction and low latency etc but also creates some doubts while thinking about the current technology. In order to analyze this complication of CCN, Perino et al. [Perino 2011] investigated CCN according to the today's technology. They provide an abstract model of a generic content router component. They concluded that the current hardware and software don't support the CCN to be implemented on the Internet scale. However it is possible to implement it on smaller scales e.g. ISP (Internet Service Provider) level or smaller CDN scales.

2.3.3.2 Nano Data Centers (Nada).

Additionally, in [Valancius 2009] an architecture based on home gateways forming a distributed data center infrastructure managed by the ISP, is proposed and evaluated. Nano Data centers (NaDa) are one of the new steps for providing the content distribution. This concept utilizes smart home devices. These smart devices can be controlled by the ISP e.g. smart gateways etc. The purpose is to build managed peer-to-peer network to distribute the contents in more economic way than the traditional farms of servers. Moreover, this type of content distribution reduces the energy consumption through different mechanisms, e.g. limiting the number of hops between clients and servers, avoiding cost of cooling and reusing the powered-on devices. A high level NaDa architecture is shown in Figure 2.2.

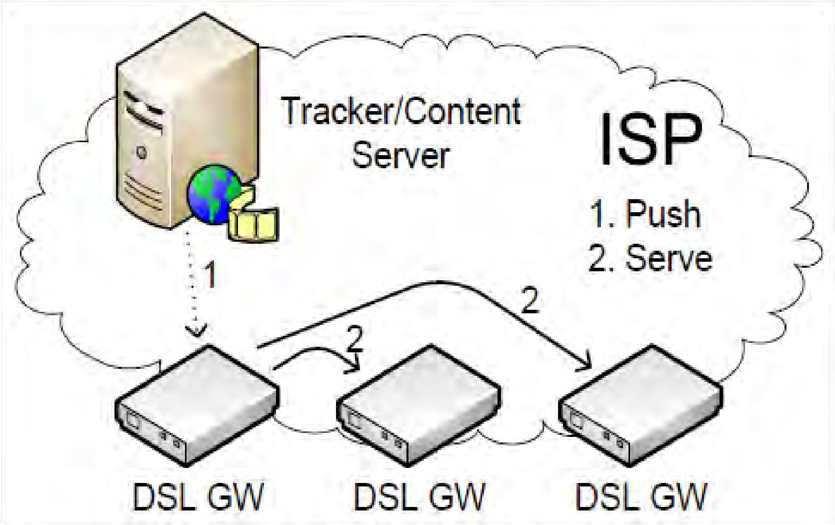


Figure 2.2: High level NaDa architecture. Content can be served by home gateway as well [Valancius 2009].

2.3.3.3 Green Cooperation (GreenCoop).

Researchers also studied that how to reduce the power consumption in a backbone network by moving the contents accessed by users to adequate CDN servers or in-network caches [Chiaraviglio 2010]. They studied the opportunities to save energy when content provider and the Internet service provider cooperate to minimize the total power consumption. They discussed a network design problem in which a

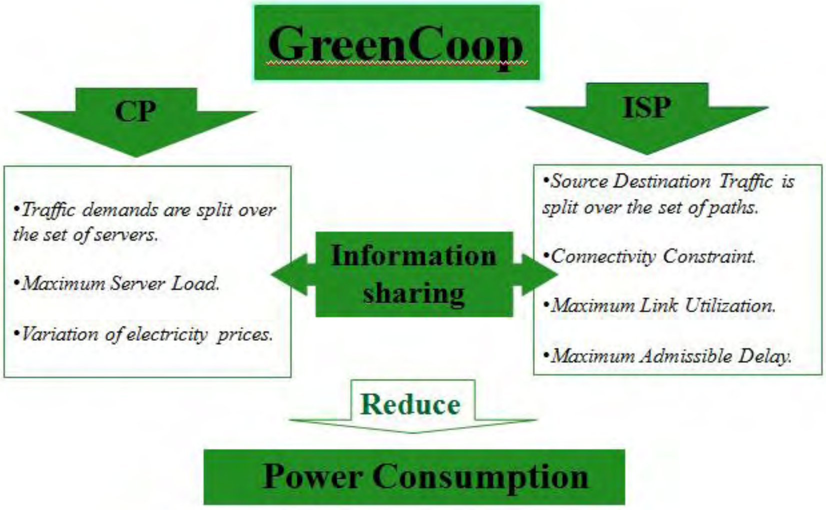

GreenCoop
CP
ISP
•Traffic demands are split over the set of servers.
•Maximum Server Load.
•Variation of electricity prices.
Information sharing
•Source Destination Traffic is split over the set of paths.
•Connectivity Constraint.
•Maximum Link Utilization.
•Maximum Admissible Delay.
Reduce
Power Consumption

Figure 2.3: GreenCoop model

Content Provider(CP) and an ISP cooperate to reduce the total power consumption as presented in Figure 2.3. The paper shows how the degree of cooperation impacts overall power consumption. They assumed that the ISP is the owner of network infrastructure who manages the network topology i.e. set of nodes and links. CP is composed of a number of servers connected to the ISP. CP infrastructure is composed of 15 servers, placed in the largest cities. Here the case of one CP and one ISP is considered. When a user asks for a CP resources, they assumed that resource is replicated over the CP infrastructure so that user can be served by any of servers of the CP. They show their model provide power reductions. But this model relates the complexities like privacy and security problem: as ISP and CP are not willing to share sensible data and user delay also degrades the user experience. Their focus was mostly on power reduction but they didn't concentrate much on quality of service aspect. These complexities can be improved by limiting the shared information between CP and ISP and by considering the user experience.

2.3.3.4 Content Distribution Network (CDN).

CDN is one of the major contributors in the Internet traffic. A.Feldmann et al. [Feldmann 2010] took three content distribution architectures (data centers, P2P (Peer-to-peer) and CDN). They proposed an energy consumption model in the context of IPTV (Internet Protocol Television). They conclude that CDNs are clear winner in terms of total energy costs.

Mathew et al. [Mathew 2012] explore energy reduction possibilities by performing load-balancing in the CDN. They use the energy reduction technique of turning-off CDN servers during the lighter load conditions. They took into account the service availability while reducing energy consumption. To achieve this goal, they did local (within a cluster of servers) and global load-balancing (among the cluster of servers) and exhibited online and offline algorithms. They also presented that shifting the workload to the nearby data centers can improve the service availability but has a lower impact on power reduction. Later-on Mathew et al. also proposed an energy-aware technique where a cluster of CDN server is considered to be turned-off [Mathew 2013]. They call this technique as cluster shutdown. They use the concept of global load balancer (GBL) in order to shift the load of a cluster to an other cluster. According to their technique the overall cluster is shutdown but the servers within a cluster can not be turned-off individually. Obvious advantage of this technique is providing the energy reduction and cooling cost reduction as well. Our work is different to their work in the case that they only focused on reducing the static energy consumption in a CDN by applying the technique of powering-off servers when they are idle while we considered static as well dynamic energy reduction in CDN by applying DVFS, powering-off servers and the combination of DVFS and the powering-off techniques to exploit the maximum energy reduction opportunities. Moreover, we also do the load-balancing to use the idle servers to maximize the CDN performance and service availability in an energy-aware way by applying the dynamic energy saving technique (such as DVFS). Though their work is interesting in case of energy savings in a CDN but their main focus remained on energy reduction and they were just interested towards the availability of services and they didn't determine the other important user experience and CDN performance metrics e.g. response time or mean response time which is crucial factor in case of CDN services otherwise financial threats can be faced by the company providing the services. Similarly they missed to exhibit the hit ratio which shows how the contents are replicated intelligently and has an impact on the response time and on energy as well as described by Xu et al. [Xu 2010]. Also, failed requests were not taken into account.

Xu et al. in [Xu 2010], proposed to find the energy saving in video CDN by the use of intelligent coordination among edge video servers. They focused on the impact of hit ratio and energy proportionality. According to them, CDN in normal conditions is less energy efficient than the central network systems. Though CDN can be less effective in energy consumption than the central networks systems but it is obvious that in the current era of content explosion central network are not effective regarding the delay, bandwidth and other requirements of the current applications and the user experience needs. Authors also proposed that improvement in hit ratio by using intelligent caching algorithms and cooperation among the servers can play an important role to reduce energy consumption in video CDN. In our work, according to all proposed policies, the surrogate servers cooperate with each other to satisfy client requests. Though, exploiting the cache policies for energy consumption don't remain our main focus and can be considered in future. In our experiments, we tested with different cache sizes and fixed a cache size which helps to evaluate CDN operations.

In order to have maximum energy savings, it is important to consider the maximum number of devices for the application of energy efficiency techniques. The efforts were also done to improve the energy reduction on the network level in CDN. Mandal et al. [Mandal 2011] proposes the energy efficient routing technique in CDN. They present the in-network caches and CDN cooperation to achieve energy reduction. They propose the green routing technique in the CDN, keeping in view the bandwidth and network service availability. They determine the different parameters i.e. cache size, popularity of contents etc. They show that energy consumption can be reduced by placing the cache on the backbone routers in order to cache the popular contents and then by choosing the appropriate CDN content provider for each of the CDN request. Their work find the energy reduction opportunities at the network layer of CDN. They considered the cache problem to gain energy savings that can be important in CDN. But in case of placing the caches to the routers can increase the routing overhead and also the complications to implement it. Also, they need to have the information about the server and the traffic for the communication of these information which can cause the problems for bandwidth and delay in network.

Moreover, the geographic distribution criteria and variation of the prices across the different locations also plays an important role for the cost of electricity and for CO_2 (Carbon Dioxide) footprint. CO_2 emissions and electricity prices per watt are location dependent. Gao et al. in [Gao 2012] concentrate on this aspect. They presented flow optimization based framework for request routing and traffic engineering. They focus on three aspects i.e. electricity cost, CO_2 emissions and latency. Their algorithm finds the trade-offs between CO_2 footprint, electricity cost and latency. The purpose is to route the user requests to a server keeping in view the previously described three factors.

Utilization-aware redirection policies for energy conservation for CDNs were also proposed [ul Islam 2011] [ul Islam 2012].

2.4 Conclusion

Energy conservation has attracted the researchers to propose and to implement the environment friendly and cost effective solutions. There is considerable advancement in this direction from machine component level to large scale distributed systems. Different techniques have been proposed to attain this objective. One thing is clear that most of the work in this domain concentrates on the energy reduction but there is not much focus on the user experience constraints which are gained as penalty at the cost of energy reduction advantages, particularly in the field of CDN where the user experience is one of the crucial element to be considered. Also the existing research in this field has more focus on the data centers, computing grids etc but there is less concentration on the CDN while the current trend of the Internet applications is emphasizing the expansion of the CDNs. In our work, we targeted CDN to optimize the operations, exploiting local and global measures, to have energy savings and analyze energy/user experience trad-offs. It is concluded that CPU is the source of major energy consumption at a machine/server. Even a machine/server is idle, consumes a considerable amount of power. So, it is important to minimize the dynamic and static energy consumption to gain the maximum energy savings. For that purpose different energy saving methodologies has been proposed. Among these different energy conservation techniques, two of the most popular techniques are used by most of the researchers i.e. turning-off devices (servers, routers etc) and changing the operating scale of the devices (processors, links etc). We adopted these two mechanisms in our work in the context of CDN and targeted CDN redirection policies and surrogate servers. Though the turning-off servers technique is proposed in the CDN to have the energy savings but there was a need to reduce the dynamic energy as well. We proposed turning-off servers and DVFS techniques and also the combination of both techniques to achieve maximum energy reduction. Finally, Table 2.1 presents a comparative analysis of some

Table 2.1: Characteristics of some existing work. Con = Consolidation; LB = Load-Balance; LUB = Load-Unbalance; UE = User Experience

Reference	DVFS	Con	LB	LUB	UE	CDN
[Weiser 1994] [Yao 1995] [Pering 1998] [Grunwald 2000] [HEEB 2000] [Don Domingo 2010] [Shekar 2010] [Guerout 2013]	+	-	-	-	-	-
[Berral 2010] [Kamitsos 2010]	-	+	-	-	-	-
[Sharma 2005]	-	-	+	-	-	-
[Dovrolis 1999]	-	-	-	-	+	-
[Sharma 2003]	+	-	-	-	+	-
[Chase 2001] [Chen 2008] [Urgaonkar 2008] [Goiri 2010] [Comellas 2010]	-	+	-	-	+	-
[Pinheiro 2001]	-	+	+	+	-	-
[Elnozahy 2003]	+	+	+	-	+	-
[Xu 2010] [Xu 2010]	-	-	-	-	-	+
[ul Islam 2011] [ul Islam 2012]	-	-	+	+	+	+
[Mathew 2012], [Mathew 2013]	-	+	+	-	+	+
Our Approach (Chapter 4)	+	+	+	+	+	+

research works.

Chapter 3

Content Delivery Networks Simulation

Contents

In order to evaluate the CDN functioning over different configurations, it is crucial to have a testbed that provides us the CDN analytical simulation environment because the CDN real time applications are hard to get for research purposes. We also need a collection of Web traces of the users which access a Web server content through a CDN, furthermore the topology of this Web server content that helps to identify the Web page communities. This environment includes:

- System model simulating the CDN infrastructure

- Network topology generator

- Website generator
- Client request stream generator

A suitable simulation environment for this purpose is CDNsim [Stamos 2010]. Since our simulator environment is based on CDNsim, we will detail here its main characteristics and we will outline the changes and extension we had to include. Also we will explain our choices of parameters and the evaluation metrics we considered and added.

3.1 CDNsim

CDNsim simulates a main CDN infrastructure and it is implemented in C++ programming language. It is based on OMNeT++ library which provides a discrete event simulation environment. It takes into account the specifications of Internet infrastructure. It is robust and scalable to provide a broad range of CDN policies. It has been designed to provide a wide range of CDN services for research purpose. All CDN networking issues, like surrogate server selection, propagation, queuing, bottle-necks and processing delays are computed dynamically via CDNsim, which provides a detailed implementation of the TCP(Transmission Control Protocol)/IP(Internet protocol) protocol, implementing packet switching, packet re-transmission upon misses, freshness, etc. CDNsim allows to add new client redirection policies.

3.1.1 Input Parameters

CDNsim simulation environment considers following input parameters to perform simulation. We have described existing and proposed input parameters.

3.1.1.1 Network Topology.

We used a real Internet topology of AS(Autonomous System) level, having 3037 routers, that consists of routing data collected from $7BGP$ (Border Gateway Protocol) peers dispersed at different locations. The backbone network topology has a set of routers. The other network entities like surrogate servers, origin server and clients are connected randomly to the router backbone. The clients and servers' distributions have an impact on the performance of the CDN.

3.1.1.2 Link Speed.

We set link speed to $16Mbps$ (Megabits Per Second), in order to have meaningful utilization of the surrogate servers without disturbing the generality. According to Akmai International quarterly report [rep 2014], global average connection speed for broadband Internet remained $4.6Mbs$ in the second quarter of 2014. It is increased 18% from the second quarter of 2013. For the streaming of ultra HD (High Definition) content, between 10 to 20 $Mbps$ bandwidth is required [rep 2014].

3.1.1.3 Website Generation.

A synthetic but realistic website having 50000 objects of $1GB$ (Gigabyte) total size, is generated. For the size of the objects, Zipfian distribution [Padmanabhan 2000] is used. Parameter z is used to modify the distribution. The values of z have an impact on the distribution slope. As the values of z increases, it makes the distribution slope steeper. For $z = 0$, website objects are identical in size. In our case $z = 1$, where object size fades exponentially.

3.1.1.4 Requests Stream Generation.

A request stream generator is used that takes the website graph and generates requests stream that shows the access patterns closer to the realistic one. The request stream generator uses random walks [Padmanabhan 2000]. The following parameters are considered for requests generation.

- *Popularity Distribution:* Not all the website objects are requested with the same frequency. Popularity of the objects in a website graph is considered using Zipfian distribution [Padmanabhan 2000]. The higher values of the parameter z cause the handling of most of the requests to the smaller number of objects.

In our case parameter z have the 1 value that means few objects are frequently requested. So caches will have these objects frequently and easily. It means few objects absorb most of the requests.

- *Popularity-size Correlation of Objects:* As different objects in a website can have different popularity and size. There may and may not be a correlation between size and popularity. The correlation between size and popularity is considered to have the values in a range from -1 to 1. Negative values indicate that an object smaller in size will have more popularity than the larger ones and vice-versa. The value 0 lies in between the two extremes where objects popularity is not related to the size of the objects. In our case, we set the popularity-size correlation of objects to 0.

The requests stream presents which clients, when and what they will request from the CDN. It is also called the traffic. In order to proceed gradually, first we tested our methods with a warm-up phase of a 5×10^4 requests (not shown here). In order to have detailed behavior of the method, it is important to test the system with different loads. For that purpose, we tested the system with different number of requests i.e. 10^5, 2×10^5, 3×10^5, 4×10^5, 5×10^5, 6×10^5, 7×10^5, 8×10^5, 9×10^5 and 10^6. In order to simplify the presentation we presented the results with 2×10^5, 4×10^5, 6×10^5, 8×10^5 and 10^6.

3.1.1.5 Cache Size.

Each surrogate server has a cache to store the contents. The surrogate server cache is considered as a hotspot as it is accessed millions of times. The size of the cache is presented as the percentage size of the website size. For content consistency and freshness, on-demand or periodic update is supported by CDNsim. According to the periodic demand, a recent copy of the content is sent to the surrogate server on the base of the prior requests for that content. While in the periodic update, the instructions i.e. what content should be stored in cache, the duration of the content to consider it as fresh and to update the content with origin server are provided to caches by the origin Web servers content, configured by the CDNsim.

Caches are updated regularly in our work, following previous considerations from [Laoutaris 2005], [Stamos 2006]. Cache management algorithms and the respective data structures are the keys for cache performance. In order to manage the storage space of the surrogate servers, cache replacement policies are used. CDNsim considers priority queues for the main cache replacement policies like LRU (Least-Recently Used), LFU (Least-Frequently Used) and SIZE. In our case, we consider that the surrogate server cache is updated by a standard LRU cache replacement policy. It means that the most recently requested objects are retained in the cache and the older are removed to save the space. Each surrogate server has 40% content of the total website size. We tested different cache sizes (5%, 10%, 20%, 40% , 80%) and found that with 40% of the cache size, we have better performance regarding our objectives, without loss of generality.

3.1.1.6 Number of Servers.

CDN infrastructure has two types of servers, i.e. origin server and surrogate server. The origin server stores the original version of resources. It is also called the Web server content. A surrogate server has a replica of a resource and acts as a reference for clients responses. Surrogate servers communicate with origin server to update the contents. CDNsim is capable to support a large number of surrogate and origin servers. The number of surrogate servers presents the size of the CDN infrastructure. In CDNsim, the number of surrogate and origin servers can be selected. Origin server is able to serve the connections which come from the surrogate servers. Where surrogate server is able to perform like a host which takes and sends the requests. In CDNsim, regarding surrogate servers, there are options of the number of connections for consuming services, which are called "outgoing connections" and the number of connections for serving which are called "incoming connections" while origin server has just the option of incoming connections as it only provides the services to the surrogate servers. We consider the case of one origin server that contains the original website. Origin server has the capacity to serve 3500 connections simultaneously. All the surrogate servers are considered to be identical. Each surrogate server is able to serve 500 connections at the same time. Each user request causes a connection to the surrogate server. Similarly, it can send 500 content requests to the origin server. All the surrogate servers are considered to be distributed geographically in the world. Our energy consumption model is based on the utilization of the surrogate servers (see section Utilization model). It is observed that the utilization of the surrogate servers is decreased with the increase in the number of surrogate servers. In case of 10 surrogate servers, we have the best utilization and it decreases with the increase in the number of surrogate servers. We found that from 10 to 50 surrogate servers, a meaningful utilization of the surrogate servers is found while it shows the lower values as the number of surrogate servers is 60 or 70 and it decreases even more respectively while the number of surrogate servers is 80, 90 and 100. So in case of 60,70, 80, 90 and 100 surrogate servers, utilization of the surrogate servers is very low that is not interesting to evaluate given the request distribution we used. So, in order to have a meaningful behavior, to evaluate our method effectively, we kept the number of surrogate servers 10, 20, 30, 40 and 50.

3.1.1.7 Number of Clients.

The clients are divided into 100 groups distributed all over the world. Each group is being linked with one surrogate server. Each client group has 1000 outgoing connections i.e. each client group can send 1000 requests simultaneously to the surrogate servers for the content.

3.1.1.8 CDNsim Policies.

In content distribution network, client requests are redirected to the surrogate servers. In order to manage content distribution and request redirection, different methods can be applied, known as CDN redirection policies. Stamos et al [Stamos 2009] examine the following CDN redirection policies in CDNsim.

- *Closest Surrogate Server With Cooperation:* According to this policy when a client sends the request for an object, its request is forwarded towards the nearest surrogate server s_1 in terms of network topology. If s_1 has the requested object in its cache, it uploads the object to the client. In case, if s_1 doesn't have the requested object in its cache, it forwards the request to its closest surrogate server s_2. If s_2 has the requested object, s_1 pulls the object from s_2, stores it in the cache and then uploads the object to the client. In case, the object request is not satisfied by the CDN, the surrogate server s_1 downloads the object directly from the origin server and it then uploads the object to the client after updating its cache.

- *Closest Surrogate Server Without Cooperation:* This policy is not much different from the "Closest Surrogate With Cooperation". The main difference between two policies is of cooperation between surrogate servers. In this policy, if surrogate server s_1 doesn't have the requested object in its cache then it downloads the object directly from the origin server instead of sending the request to the server s_2. Finally, requested object is uploaded to the client.

- *Random Surrogate Server With Cooperation:* It is also called the load-balance policy. In this policy the requests are redirected towards the surrogate servers evenly (the random probability law follows an uniform distribution). Upon a cache miss, the surrogate server retrieves the object from a random alternative surrogate server that contains the requested object. The object is stored in the cache and then it is served to the client. If any of the surrogate doesn't have the requested object, the object is downloaded from the origin server. In this policy network topology distance doesn't matter, so the objects travel via long paths that increases the network traffic.

- *Surrogate Load-Balance With Cooperation:* The client request is forwarded to the closest surrogate server in terms of network topology. If the load of the surrogate server is 95% then the client request is redirected towards the least loaded surrogate server. If the surrogate server doesn't have the content in its cache, it retrieves the object from the closest alternative surrogate server that contains the requested object. Again if the load is 95%, the surrogate server is redirected to the least loaded surrogate server that contains the object. On retrieving the object, the surrogate server stores it in the cache and then it is served to the client. If the object is not outsourced at all in any surrogate server then the surrogate server retrieves the object from the closest origin server. Surprisingly, this policy is mentioned in the CDNsim but the detailed analysis of the policy like the precedent three policies, is not considered.

CDNsim allows us to add new policies. We have defined other policies, see Chapter 4.

3.1.2 Our Proposed Input Parameters

3.1.2.1 MinLoad and MaxLoad.

We defined the parameters MinLoad and MaxLoad in CDNsim to manage the load of the surrogate servers. These parameters are useful for defining new request redirection policies in CDNsim. These parameters are used to restrict the load of the surrogate server. MinLoad and MaxLoad have the values between 0 and 1. Value 1 presents the 100% load. MinLoad presents the lower bound of the load of the surrogate server. After attaining MinLoad a surrogate server will not receive new incoming client requests. The surrogate server will treat the existing requests and then its load will be equal to 0. Whereas MaxLoad defines the upper bound of the load, after attaining it, a surrogate server will not receive the incoming requests. It can be useful when we don't want to load a surrogate server 100% in order to avoid the heating problem. Also, MinLoad and MaxLoad provide the thresholds to know when a surrogate server can be considered to be powered-off and when the new surrogate servers should be powered-on, respectively.

3.1.2.2 FreqMin, FreqMed and FreqMax.

In CDNsim there was no option to define the frequency of the processor of a surrogate server. All the surrogate servers were supposed to treat the requests on the full processor frequency. We included these parameters in CDNsim in order to manage the frequency of the processor of the surrogate servers. These parameters have the values which are inverse of the real processor frequencies. FreqMin presents the minimum frequency a processor can attain. FreqMed presents the medium processor frequency and FreqMax maps the highest processor frequency when it is working at the peak. These parameters help to add new policies or to modify existing request redirection policies. In our case, we took the specifications of Intel (R) Xeon (R) E5620 processor.

3.1.2.3 DVFS MinLoad, DVFS MedLoad and DVFS MaxLoad.

Load of the surrogate server has an important role to apply new techniques in the simulator. We have defined DVFS MinLoad, DVFS MedLoad and DVFS MaxLoad parameters. It defines the range of the load values when a surrogate server is supposed to have the minimum, medium and maximum load. It makes a bridge between the load of the surrogate serves and the processor frequency of the surrogate server. It permits us to change the processor frequencies dynamically according to the load (utilization) of the surrogate servers.

- *DVFS MinLoad:* Defines the range when a surrogate server is considered to have the minimum utilization.

- *DVFS MedLoad:* Shows the range when a surrogate server utilization is consider to process at medium frequencies.

- *DVFS MaxLoad:* illustrates the range of load when the surrogate server is supposed to utilized at its maximum.

The above classification of the load of the surrogate servers provides the opportunity to apply the technique where the frequency of the processor is changed according to the given conditions. It is discussed in detail in Chapter 4. The difference between MinLoad, MaxLoad and DVFS MinLoad/MedLoad/MaxLoad is: MinLoad and MaxLoad provides the limits where a surrogate server below or above this limit respectively, is not considered to receive client requests. While DVFS MinLoad/MedLoad/MaxLoad provide the logical division of the surrogate server's load which allows to apply different techniques.

3.1.2.4 Load-Unbalancing Parameter (ZipfUnbalance) z.

When client requests are sent to the surrogate servers, the behavior of the requests distribution to the surrogate servers is important for the CDN functions. The client requests can be forwarded to the surrogate servers randomly or the distribution of the requests to the surrogate servers can have an exponential behavior etc. The behavior of the requests distribution has an impact on the surrogate servers performance and on the overall CDN communications. So it is important to define a parameter to control the behavior of the client request redirection to the surrogate servers. We defined a parameter $ZipfUnbalance$ in the CDNsim in order to change the way of requests distribution to the surrogate servers. $ZipfUnbalance$ value is in the range of 0 to ∞. The value 0 of the $ZipfUnbalance$ shows random selection of servers. A higher number causes the zipf distribution. The value 1 shows the exponential behavior. As the value of the $ZifUnbalance$ increases from 0, load-unbalancing behavior of the requests distribution occurs. For example value 1 shows load-unbalancing behavior of the request distribution to the surrogate servers in the CDN system. This variant behavior of the $ZipfUnbalance$ parameter is useful to propose and to implement new client request redirection policies in the CDN.

3.1.2.5 Centralized or CDN Environment (cdnON).

We have included this parameter in CDNsim which allows us to decide whether we are considering the CDN environment or centralized network environment. It can have value 0 or 1. Value 1 shows that CDN environment mode is active where surrogate servers and origin server both work. While value 0 presents the centralized environment (where all the requests are forwarded to the origin server).

3.1.2.6 Redirection Policy (redirectionPolicy).

We have defined this parameter which permits us to choose whether the client requests will be forwarded to closest surrogate server or according to load-balancing/load-unbalancing mechanism. It has two values 0 and 1. Value 1 represents load-balancing/load-unbalancing behavior.

3.1.2.7 Cooperation Among Surrogate Servers (cooperationON).

In a CDN, surrogate servers can work by cooperating with each other or without cooperation. If surrogate server doesn't cooperate with each other, client request is sent to origin server if destination surrogate server doesn't have required content in its cache. We have included this parameter in CDNsim to define cooperation among surrogate servers. It has 0 and 1 values. Value 1 presents surrogate will cooperate with each other to satisfy client requests.

3.1.2.8 Seed.

In order to verify the statistical significance of a result we need to perform a number of tests. In order to do so we need a parameter that generates the randomness in the simulator. We put an additional argument named "seed". Seed values are used for the topology and the traffic. Different values of the seed changes the placement of servers, clients and the distribution of requests. The seed can be any value >= 0. Default value of the seed is 0. Same seed produces same results. In our case, we used 10 and 20 different seeds for each data set and find the average of the results.

3.2 From CDN Utility to CDN Utilization

Net utility is used to identify the performance of the CDNs. It is also called as the CDN utility. CDN utility identifies the traffic activity in the CDN. Different approaches are adopted in order to improve the CDN utility. There are different parameters in the CDN which affect the CDN utility e.g. Network topology, cache size, request distribution pattern and redirection policies etc. Stamos et al [Stamos 2009] evaluated the utility of the CDN surrogate servers and identified some parameters which affect the surrogate server's utility in CDN infrastructure. They defined a metric that measures the utility of CDN surrogate servers, called CDN utility. This metric captures the traffic activity in a CDN, expressing the usefulness of surrogate servers in terms of data circulation in CDN. They defined the net utility as a value that presents the relation between the number of bytes of the served contents (by the surrogate servers) against the number of bytes of the pulled contents (from other surrogate servers or from origin server). They quantify a net utility μ_i of a CDN surrogate server i as

$$\mu_i = \frac{2}{\pi} \times arctan(\xi) \tag{3.1}$$

According to this metric, a surrogate server is considered useful if it has high net utility. It means that the surrogate server uploads the contents more than it downloads. The parameter ξ is the ratio of the uploaded bytes to the downloaded bytes. μ_i has the values [0....1]. Mortazavi et al [Mortazavi 2006] proposed and used a similar net utility metric for a peer-to-peer system. Considering that a CDN has N surrogate servers, the CDN utility can be defined as follows,

$$\mu = \frac{\Sigma_{j=1}^{N} \mu_i}{N} \tag{3.2}$$

They evaluated the CDN utility, mean response time, hit ratio and byte hit ratio to take the measures. They evaluated the CDN utility against network topology, request generation patterns and CDN redirection policies. They concluded that with the increasing cache size there are certain peaks in the CDN utility which is invariant with different network topologies. When the large files are transferred more in the CDN, it increases the CDN utility so it is more useful in case of pricing as well for the content provider. They concluded that a performance peak, in terms of CDN utility has been found. This peak is invariant of the network topology, the traffic model and the Web site model. They showed that a poorly designed redirection policy can't attain the better CDN utility. They proposed and evaluated the following request redirection policies, 1) Closest surrogate server with cooperation, 2) Closest surrogate server without cooperation 3) Random surrogate server with cooperation. They presented that the closest surrogate server with cooperation performs better than the other two policies. The CDN utility in case of closest surrogate with cooperation is quite better than the rest of the two policies. It shows a performance peak in the CDN utility because of the cooperation among the surrogate servers. The closest surrogate server without cooperation doesn't show such a peak as the amount of uploaded content is affected solely by each individual surrogate server performance. In case of random surrogate server with cooperation the CDN utility leads to a plateau after the peak because of the random distribution of the requests in the CDN. They found that closest surrogate server with cooperation performs better in case of mean response time as well. The mean response time in case of random surrogate server with cooperation is very poor because of the high network traffic. In case of random the proximity criteria is not taken into account so the client requests and server responses are sent and got from a distance that increase the response time. But the closest surrogate server without cooperation performs better in case of mean response time than the random surrogate server.

As the authors evaluated these policies considering surrogate utility but we are interested in surrogate utilization. But these policies were not responding well for the surrogate utilization though they were showing considerable values for surrogate utility. It showed very low utilization. When utilization of surrogate servers is augmented, there were lot of failed requests. This behavior couldn't allow us to proceed with these policies to evaluate.

We need a metric in the CDN, that can also lead us to measure the energy consumption in the CDN servers. Surrogate server utility is the upload/download ratio normalized in range $0-1$. It doesn't explain, how a surrogate server is loaded by the client requests. So a surrogate server can have the higher values of the utility but utilization of the surrogate server can be low. So the utility of the surrogate server doesn't reflect the real utilization of the platform. For example, Figures 3.1 and 3.2 show mean surrogate server utility and the average surrogate server utilization (details on utilization will be given in Chapter 4). It presents the higher mean surrogate server utility while the surrogate server average utilization is very low for the same set of simulation parameters. In order to know how a surrogate server is loaded over time, a metric is needed to define. Surrogate server load is used to refer that how many client requests are being served by a server simultaneously to the capacity of the surrogate server. Surrogate server utilization is how the server is loaded over time. So in our case, rather than CDN surrogate utility, we are interested to evaluate the CDN surrogate utilization. A surrogate server utilization permits us to define the energy consumption in the surrogate servers. Furthermore, it allows us to model the energy consumption of the surrogate server. Details are given in Chapter 4.

The CDN infrastructure has an impact on the different important CDN parameters i.e. surrogate server utilization, mean response time etc. The extent of the load also plays a role in the CDN operations. In order to evaluate the impact of the CDN infrastructure size and effect of the load, we took the number of the surrogate servers and number of client requests as the basic parameters to evaluate different client side and server side evaluation parameters. Figure 3.3 and Figure 3.4 show how the number of surrogate servers and the number of client requests affect the utilization (indirectly the energy consumption) of the surrogate servers. It shows the surrogate server utilization has a linear relation with the number of client requests and it has a non-linear relation with the number of surrogate servers. Figures 3.5 and 3.6 present the behavior of mean response with different number of client requests traffic and with different number of surrogate servers.

3.2.1 Factors Affecting CDN Utilization

Surrogate server utilization is affected by the following input simulation parameters.

3.2.1.1 Network Link Speed.

Network link speed plays an important role in the duration of a connection and in mean response time as well. If a network link speed is higher, the process of sending and receiving of requests and contents will be faster so the network connection time will be smaller and contents will be sent in a smaller time interval, if other parameters remain same.

3.2.1.2 Frequency of Content Requests.

If more requests are sent in a smaller amount of time, the servers have more connections and it increases their utilization. So we can say that surrogate server utilization is directly proportional to the frequency of the content requests, if the other parameters remain same. But at the other side, more requests in small inter-

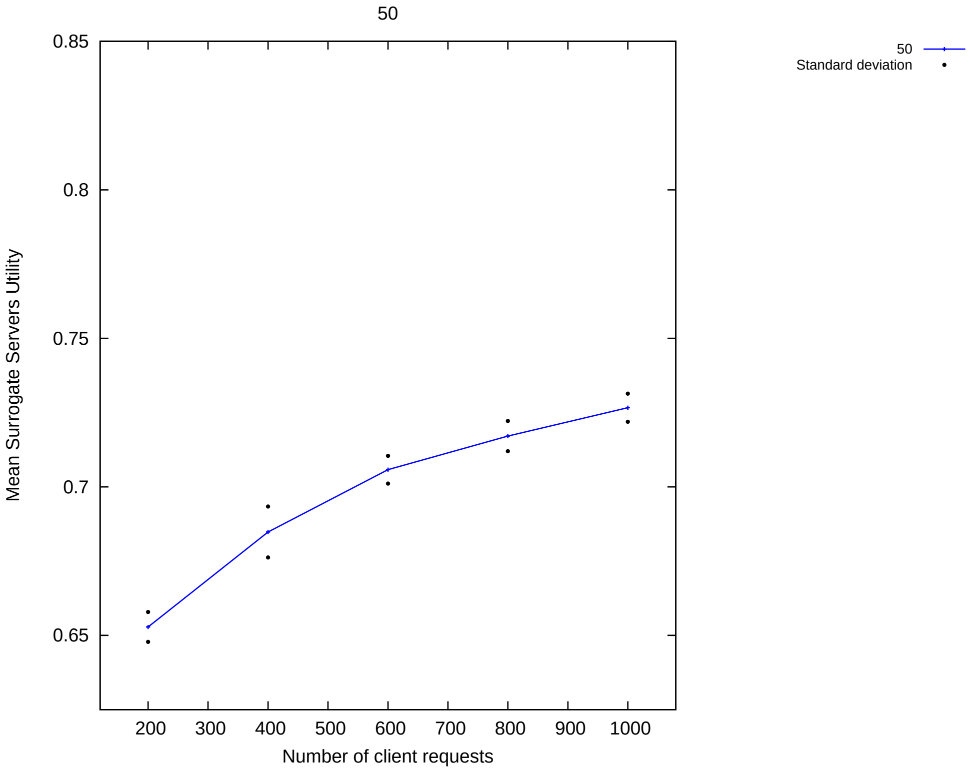


Figure 3.1: Mean surrogate servers utility, for 50 surrogate servers over different number of client requests for Load-Unbalance policy.

val of time increase the congestion at the links and as well as at the network nodes that can cause the increase in response time. Chapter 6 will study the impact of this frequency in our work.

3.2.1.3 Object Size.

The duration of a connection is directly proportional to the object size. If a web site has smaller objects, request completion speed is higher but the duration of the connection is smaller. So the websites with the bigger objects have higher connection duration that increases the time for the completion of a request.

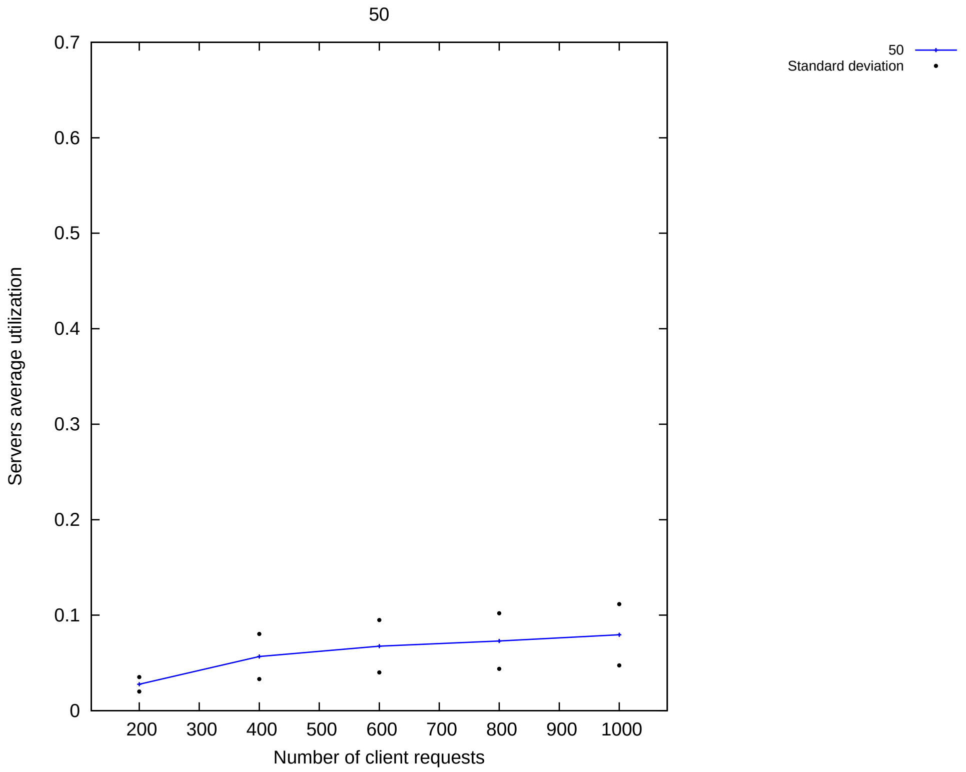


Figure 3.2: Surrogate servers utilization, for 50 surrogate servers over different number of client requests for Load-Unbalance policy.

3.2.1.4 Cache Size.

As the objects are stored in the cache of a surrogate server, if a surrogate server has a smaller cache, it has smaller amount of objects to serve and probability of the completion of the requests decreases. In case of absence of the requested objects the surrogate server is obliged to pull the contents from other surrogate servers or from the origin server. This can augment the connection duration and the response time. If the cache size is bigger it will have more probability to serve the requested contents.

3.2.1.5 Content Popularity.

The contents popularity can also affect the connection duration. If some contents are demanded more frequently, these are considered more popular and surrogate

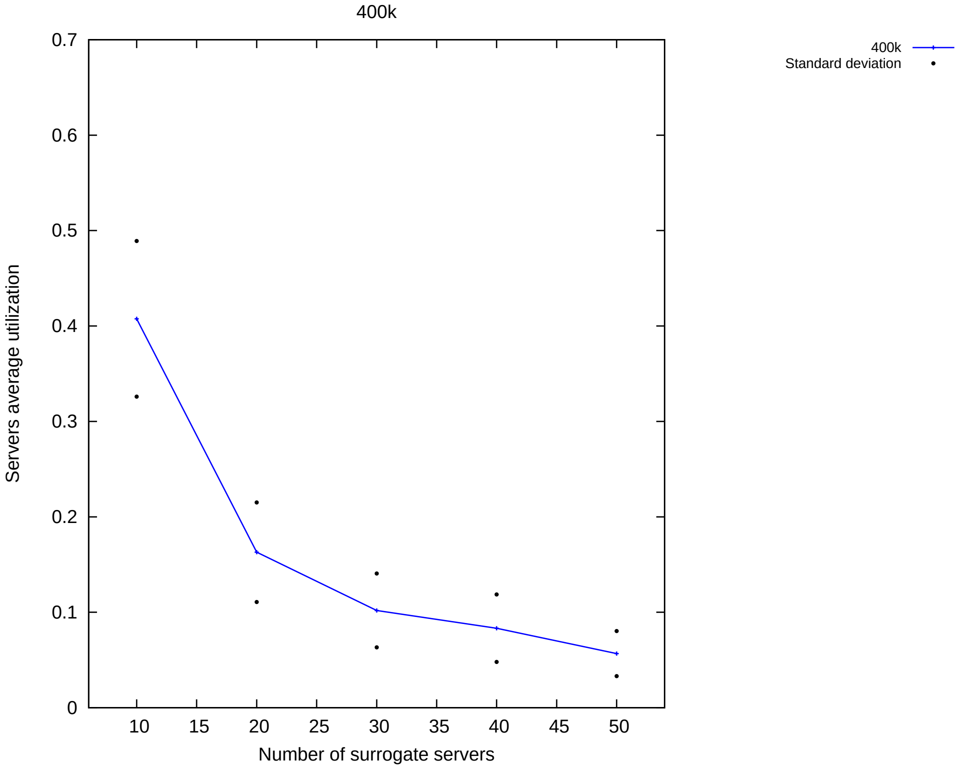


Figure 3.3: Average utilization of the surrogate servers over 400k client requests for Load-Unbalance policy.

servers tries to keep them available in their caches. So the more popular contents have the higher probability to be served in a smaller amount of time than the unpopular objects.

3.2.1.6 Client and Server Location.

The client and surrogate servers' location is also important. If a client sends a request to the nearer surrogate server then the response of the request can be rapid in case of availability of the contents. In this case, the connection made for the request has a higher probability to be shorter in time than the connection made for a request to the server far from the client.

All the above factors affect the surrogate servers' utilization directly or indirectly and therefore their energy consumption. We propose a simple utilization model

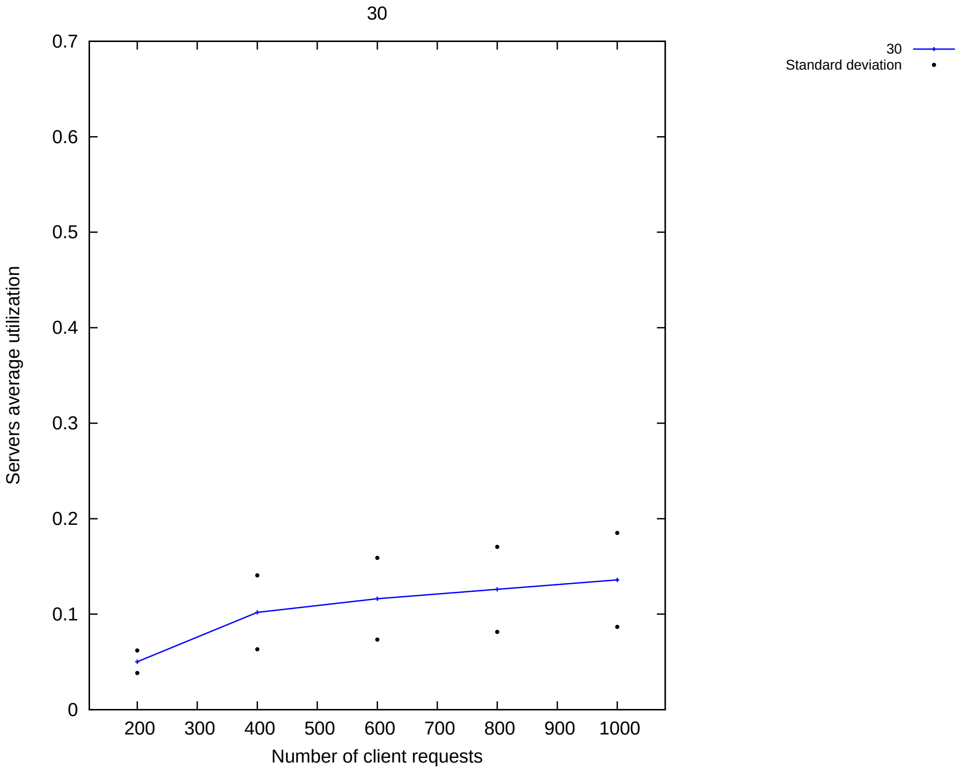


Figure 3.4: Average utilization of 30 surrogate servers over different number of client requests for Load-Unbalance policy.

based on computing the connections duration that reflects the usage of the server over the time.

3.3 Evaluation Parameters

In order to proceed gradually, first we tested our approach with a warm-up phase of a 50000 requests of traffic (not shown here). After that we performed the experiments with the traffic upto 1 million requests, that is evaluated here. The following measures have been taken into account.

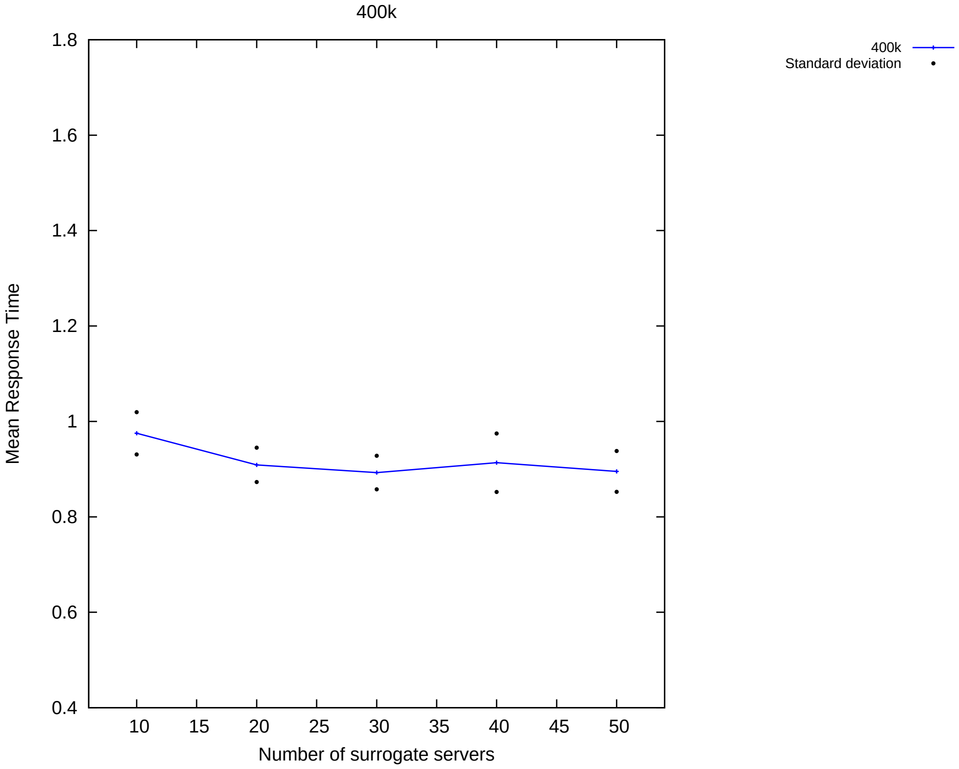


Figure 3.5: Mean response time over $400k$ client requests for Load-Unbalance policy.

3.3.1 Client Side Evaluation Metrics

It presents the client side activities e.g. when a client requests for the contents. The following parameters are taken into account:

- *Response Time:* The response time starts at the time-stamp when the client request begins and ends at the time-stamp when the connection is closed. Smaller values are considered good for user experience.

- *Mean Response Time:* It exhibits the average user experience of the CDN. It shows how fast a client request is fulfilled. It is the ratio of the summation of the time taken to fulfill all client requests to the total number of requests.

- *Completed Requests:* These are the total number of client requests sent to CDN served successfully. Due to DoS (Denial of Service), requests are not

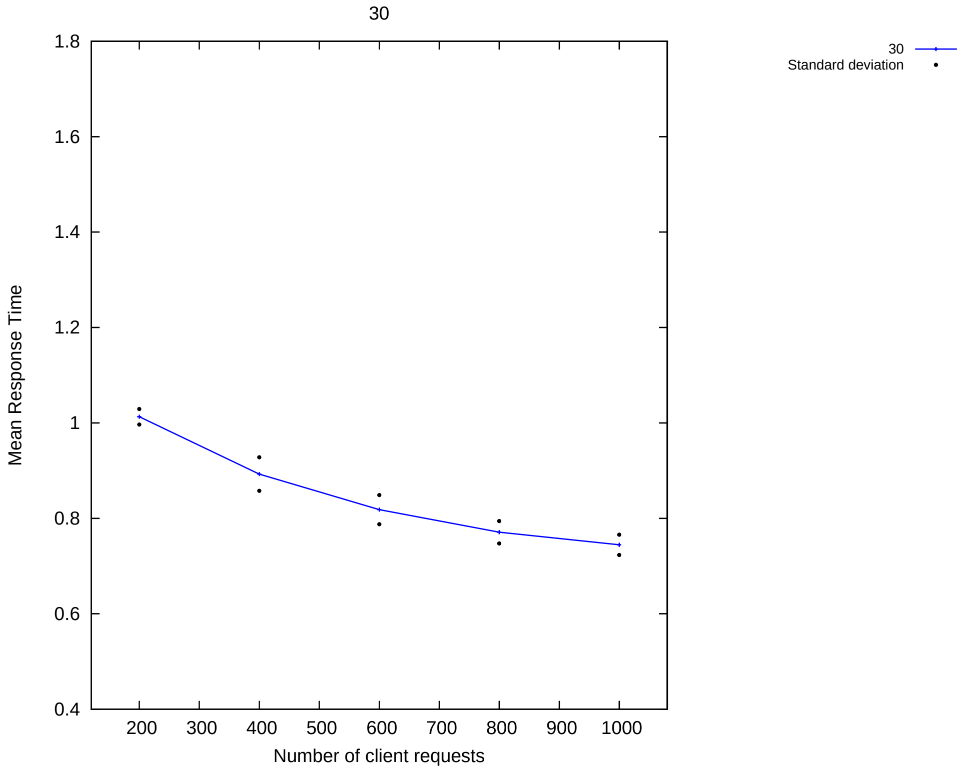


Figure 3.6: Mean response time over 30 surrogate servers over different number of client requests for Load-Unbalance policy.

satisfied. Denial of service is caused when a surrogate server is overloaded e.g. when the number of incoming connections is reaching to the maximum load limit.

- *Failed Requests:* These are the requests made by clients for contents but the requests are not satisfied even after a number of retries. It is shown in percentage. It can be reduced by increasing the number of surrogate server's connections. When a client request for content is redirected towards the surrogate servers, it depends upon different factors whether the request will be completed successfully or will be failed. A client request can be failed due to different reasons e.g. the destination surrogate server has already enough connections to serve the client requests according to its its capacity, congestion on the network nodes, denial of service due to the shutting down of network devices or surrogate servers, unavailability of the contents in the target as well in neighboring server caches (in case of cooperation among servers) and not receiving contents even after maximum number of retries for the contents etc.

3.3.2 Server Side Evaluation Metrics

- *Servers' Utilization:* We compute the average of each surrogate servers' utilization. The values of surrogate servers' utilization range from 0 to 1 (see Chapter 4).

- *Energy Consumption:* It is the power consumed by a surrogate server or a set of surrogates during a time period. We evaluated energy consumption in joules (see Chapter 4).

- *Energy Per Request:* It is average energy consumed in Joules by a request during the simulation process. It is obtained by dividing the total energy consumed during the simulation divided by the total number of requests.

- *Servers Powered-On:* It shows the number of surrogate servers which are available to serve the request during simulation time. The latest discussed metrics are in our proposal to evaluate our work.

- *Hit Ratio:* It represents the ratio of the served requests to the total number of requests which are handled directly by a surrogate server without any cooperation with the other servers. A higher value means that the requests are satisfied quickly if the network state doesn't change. For example, if a surrogate server is able to satisfy most of the incoming requests itself but if it receives a lot of requests then there can be congestion at nearby links and nodes that can slow down the request completion process.

3.4 Summary of Changes Made in Original CDNsim

CDNsim provides a friendly environment. It provides the opportunity to researchers to manipulate the specifications according to their research needs. CDNsim allows to modify the existing parameters or policies. It provides the facility to add new parameters and the new specifications. We have added new parameters and the new specifications to the original CDNsim, which provide the opportunity to define new policies.

- We have added the parameters of MinLoad and MaxLoad that allows to set the upper and lower bound of the surrogate server utilization. Specifically the parameter MinLoad also define the bound of the load after that a surrogate server can't have more requests and can be considered to apply specific techniques i.e. turning-off servers etc.

- The original CDNsim doesn't provide the specification to change the frequency of the processor. In order to have different processor frequencies in the CDNsim, we added the parameters FreqMin, FreqMed and FreqMax that allows to set the frequency of the processor of the surrogate server. The parameters allow to change the processor frequency dynamically, the frequencies can be switched dynamically during the simulation process as well.

- The load of a surrogate server can be classified into the different ranges. This classification can be useful to apply different techniques e.g. changing the frequency of processor according to the load of the surrogate servers. For that purpose, we have defined the parameters DVFS MinLoad, DVFS MedLoad and DVFS MaxLoad which allow to classify the load of the surrogate servers that can be considered to apply the techniques to change the behavior of the CDNsim functionality e.g. changing processor frequency dynamically.

- In CDN environment, the client requests are redirected to the surrogate servers according to the client redirection policy. It defines where to send the client requests. The surrogate servers are loaded according to the CDN redirection policy. We have defined the LoadUnbalancing parameter (ZipfUnbalance) z that permits to add the load-balance or load-unbalance behavior to the CDNsim policy. The parameter z presents the degree of load-unbalance. The value of z parameter plays an important load to decide how the surrogate servers will be utilized. LoadUnbalancing parameter (ZipfUnbalance) z allows us to define redirection policies. On the basis of these parameters and specifications, we have defined and added the CDN redirection policies (described in Chapter 4).

- In original CDNsim, there are few policies presented as discussed earlier. In order to simplify the proposition of more policies and environments, we have proposed *cdnON*, *redirectionPolicy* and *cooperationON*. These parameters permit us to define different environments (centralized or CDN), different modes of an environment (cooperative or non-cooperative) and different policies by making the different combinations of parameter values.

- We also made the changes to the existing CDNsim launching platform. Original CDNsim has a graphical wizard, where input files and parameters for the simulation can be set to prepare and to launch the simulation. In order to set the configuration of each simulation, it was necessary to do every step by the wizard. For each new simulation, it was necessary to redo it through the wizard. It is true that the graphical interface is easy to use and to understand but at the same time, it demands a lot of effort and time when there is need to launch lot of simulations. Moreover, graphical wizard caused the problems of scalability as well (in case when we want to add large number of parameters). For example, all developed policies were presented on a prompt window and you have to select a policy by name. This specification made the problem of scalability for proposing and adding large number of policies. In order to remove this limitation of CDNsim, we have developed the script that automatize the process of setting and launching lot of simulations and permit us to propose large number of policies by the different combinations of parameters. It allows to set and to modify the parameters statically and dynamically as well. It allows to change the values of parameters through passing the command line arguments to the simulator. We have launched extensive simulation experiments at the Cloudmip platform. It allowed us to launch 148 simulations in parallel.

- In original CDNsim, a report can be created having some client and server side output metrics e.g. mean response time, hit ratio etc. We have modified the code and added more output values in CDNsim traces. New version of CDNsim provides response time as well over the simulation time. Number of surrogate servers turned-on over the simulation time is also presented. Utilization of the surrogate servers is also provided at every change of number of active connections for each surrogate server. Moreover, energy related metrics are also taken into account like energy consumption and energy per request. All the process is automatized from launching the simulations till gathering results for all output metrics.

3.5 Conclusion

In Table 3.1, you will find the different parameters used in the following evaluation of the thesis. An exhaustive study would have been needed to test all the combinations of different values, especially the parameters influencing the behavior of the surrogate server (cache size, popularity, link speed, mean inter-arrival time of requests, number of surrogate servers, number of client requests). However, we believe that the proposed evaluation parameters, some of them validated through previous works on CDNsim, are few enough to evaluate the meaningfulness of our approaches to measure and reduce energy consumption in CDN.

Table 3.1: Summary of simulations parameters

Parameter	Experiments set1	Experiments set2
Website size	$1GB$	
Website number of objects	50000	
Website z for size	1	
Size vs. popularity correlation	0	
Number of requests	2×10^5, 4×10^5, 6×10^5, 8×10^5, 10^6	10^6
Mean interval time of requests	0.0033	0.01, 0.005, 0.0033, 0.0025, 0.002, 0.00125
Distribution of the interval time	*exponential*	
Requests stream z	1	
Link speed	$16Mbps$	
Network topology backbone type	AS	
Number of routers in network backbone	3037	
Number of surrogate servers	$10, 20, 30, 40, 50$	40
Processor	Intel (R) Xeon (R) $E5620$	
Processor Minimum Frequency (FreqMin)	$1.6GHz$	
Processor Medium Frequency (FreqMed)	$2.0GHz$	
Processor Maximum Frequency (FreqMax)	$2.4GHz$	
Number of incoming connections per surrogate server	500	
Number of outgoing connections per surrogate server	500	
Surrogate server minimum load (MinLoad)	0, 0.05	
Surrogate server maximum load (MaxLoad)	0.9	
Number of client groups	100	
Number of content providers (Origin server)	1	
Number of incoming connections per origin server	3500	
Cache size percentage of the website's size	40%	
Cache replacement policy	LRU	
Load-unbalancing parameter (ZifUnbalance) z value	0, 1	
Number of seeds	10	20

Chapter 4

Policies for Energy Conservation in Content Distribution Networks (CDNs)

Contents

The purpose of this chapter is to identify the following research question and to introduce the means of exploring possible solutions. How a CDN can redirect the users' requests for content to its surrogate servers in such a way that the energy consumption is minimized while trying to maintain an acceptable Quality of Experience (QoE)? In order to answer this question, a set of discrete milestones have been achieved, starting from theoretical definitions leading to actual implementations. Various techniques are proposed for energy efficiency in networks such as described in Section 2. The geographical distribution of the servers often exposes many opportunities for optimizing energy consumption and costs by intelligently distributing the workload. For designing policies, it is not enough to minimize energy costs, it is also important to keep in mind the performance and availability of services. One of the popular approach is to redirect the traffic towards fewer devices and to shut down the others or to put them in sleeping mode. The concept is based on the fact that the network traffic is not always in the same manner. In normal network conditions, network devices are not utilized according to their full capacity. There is always a need to introduce smart mechanisms which permit to utilize the network devices according to their capacity and to gain the energy savings in an efficient way. Similarly, in normal network conditions, servers work on their higher processor frequencies to serve the contents to clients. The traffic of client requests is not always the same. If the servers processors work on lower frequencies, the energy cost will also be lower. The purpose is to use this idea and adjusting the processor frequencies according to the client traffic conditions which can provide the opportunities to gain energy savings.

4.1 CDN Redirection Policies

In a CDN, a client request is redirected to surrogate server according to the CDN redirection policy. Different CDN redirection policies have been proposed as discussed in previous chapters. We consider the scenario of a CDN and propose two basic policies which redirect the client requests to CDN surrogates. We use the Zipfian distribution in order to define these policies. We consider the Zipfian distribution with the load-unbalancing parameter (zifUnbalance parameter) $z \in \{0, .., 1\}$. For the value 0 we get the uniform distribution and for the value 1 we get an exponential distribution where only a small percentage gathers the majority of the distribution. Then, the client redirection algorithm works like this:

- Sort the surrogate servers by their current utilization
- Set the parameter z
- Pick a random surrogate server according to a probability drawn from the respective Zipfian distribution with slope zipfUnbalance parameter z

Algorithm 1 shows working of the policy. The obvious advantages of proposed method are the generation of under-utilized servers and the ability to smoothly and dynamically balance the energy consumption vs. the surrogate servers availability.

4.1.1 Load-Balance

For the first policy, that is called Load-Balance policy, the requests are sent to the surrogates randomly. For load-balance policy zipfUnbalance parameter z is set to 0. The value 0 of the zipfUnbalance parameter z creates the uniform distribution. According to this policy all the surrogate servers have equal probability to serve the client requests. So, the content requests from the clients to the CDN servers can be redirected to any of them. Therefore all the surrogates have the equal chance to get the client demands throughout the execution time. The advantage of the policy is to improve the performance by balancing the workload intelligently. Instead of concentrating on fewer servers, the requests are distributed to all of the available servers. The availability of the surrogate servers is important in case of better Quality of Experience (QoE). This policy is better in the case when high availability of the servers is required. This policy provides the opportunities for energy savings as well. The changes in processor frequency according to the load is considered to minimize energy consumption in surrogate servers.

4.1.2 Load-Unbalance

For this policy we set the value of Zipfian parameter z to 1. In this policy the requests are distributed in an exponential fashion. According to this policy, most of the client requests are redirected towards a certain number of surrogate servers while the other surrogates have less priority to get the requests. A smaller number of the surrogates capture most of the client requests traffic. It is important to utilize the network resources effectively and to minimize the wastage of the network resources. The obvious advantage of the load-unbalance policy is to utilize the CDN resources in a better way. As compared to normal network traffic conditions, some surrogate servers are utilized according to their capacity by applying this policy. But a number of surrogate servers receive less client requests. Apparently, it shows the bad way of utilizing a number of surrogate servers in the CDN functioning. But actually this under-utilization of surrogate servers is got purposely. These underutilized surrogate servers provide the opportunities to save energy consumption. So, the underutilized surrogate servers can be considered to use the techniques for energy savings. A threshold is arbitrarily set to limit the maximum load of each surrogate server of its full capacity (this is set to 90% in our experiments), to avoid hot spots on one server.

4.2 Surrogate Server Utilization and Energy Consumption

4.2.1 Surrogate Server Utilization

In a CDN, when a client sends a request for some particular contents, the request is forwarded to a surrogate server according to the redirection policy. When a surrogate server s_1 receives a request for an object from client c, s_1 locks a resource. It checks for the demanded object in its cache. If s_1 has the requested object in the cache, it sends the contents to the client c and unlocks the resource. This process of completion of the request is faster. In other case, if s_1 doesn't have the contents in its cache, required by c, it can get the object from another surrogate server

Algorithm 1 This algorithm acts as a redirection policy of a request from a client or surrogate server(in case CDN is used and CDN cooperation is active) to another surrogate server(if CDN is used) or origin server if there are no surrogate servers available

Input: List of surrogate servers $s = 1 \ldots S$, List of origin servers $o = 1 \ldots O$, minLoad, maxLoad, surrogateLoad (L_s), originLoad(L_o).
Output: Assigned appropriate candidate server.

```
Set $found = false$
for ($s \in S$) do
    if ($minLoad <= L_s <= maxLoad$) then
        add $s$ to the $candidate\_list$
        $found = true$
if ($found = false$) then
    for ($o \in O$) do
        if ($minLoad <= L_o <= maxLoad$) then
            add $o$ to the $candidate\_list$
            $found = true$
if ($found = false$) then
    Choose randomly $o$ from $O$; Return $o$
if ($found = true$) then
    Sort the $candidate\_list$ in decreasing order of their load
    Let $sum = 0$
    Let $index_c$ be the index of server $c$ in $candidate\_list$
    for ($c \in candidate\_list$) do
        Let $i_c = index_c^{-\alpha}$ // Calculate $c$'s importance value using Zipfian distribution with slope $-\alpha$
        Let $sum+ = i_c$
    Let $v_{pc} = 0$
    for ($c \in candidate\_list$) do
        Let $v_c = v_{pc} + i_c$ // Calculate cumulative distribution of $c$'s importance value
        Let $v_{pc} = v_c$
        Let $v_c = v_c/sum$
    Let r = random number from [0..1]
    for ($c \in candidate\_list$) do
        if ($v_c >= r$) then
            Return $c$
```

s_2 or from origin server (depending upon the redirection policy). This activity is called the cooperation among the servers and it will cause a lock and an unlock of a resource in another server as well. At the reception of the requested object, surrogate server s_1 stores the object in its cache and sends it to the client c. So, a connection to the surrogate server s_1 is established:

- when a client c makes the request for contents to s_1
- when another surrogate server s_2 requests the contents from s_1 in case of cooperation

When a surrogate server has no request to serve then it is considered as idle. When it gets a connection (lock) in the form of client request or other surrogate servers' redirected request in case of cooperation, it is said to be in utilization. A surrogate server can have multiple connections at the same time, depending upon its capacity. Its utilization is directly proportional to the number of connections it has, at a given time interval. If the ratio of the number of current connections to the maximum number of possible connections is bigger, the surrogate server is said to be better utilized and vice versa. The connection duration is important to calculate the utilization of a surrogate server since some requests may take different duration to be served (depending typically on the load of the server but also of the size of the data being requested).

4.2.2 Surrogate Server Utilization Model

Here, we present a CDN server's utilization model. We first compute the utilization ratio of the server s during the time interval $[t_1, t_2]$ as such:

$$UR_{s_{[t_1,t_2]}} = \frac{Conn_{s_{[t_1,t_2]}}}{ConnMax_s} \tag{4.1}$$

where $Conn_{s_{[t_1,t_2]}}$ is the actual number of connections the surrogate server s handles between time t_1 to time t_2 (considered as constant between t_1 and t_2). $ConnMax_s$ represents the maximum number of connections allowed on the server s i.e. the maximum content requests a surrogate server s can have at the same time. It shows the capacity of a CDN server.

During the lifetime of a server, its utilization ratio will increase and decrease over time, as shown in Figure4.1. In this Figure, if we consider $ConnMax_s = 5$, we have $UR_{s_{[6,7]}} = 3/5$. The duration of this utilization ratio is 2.

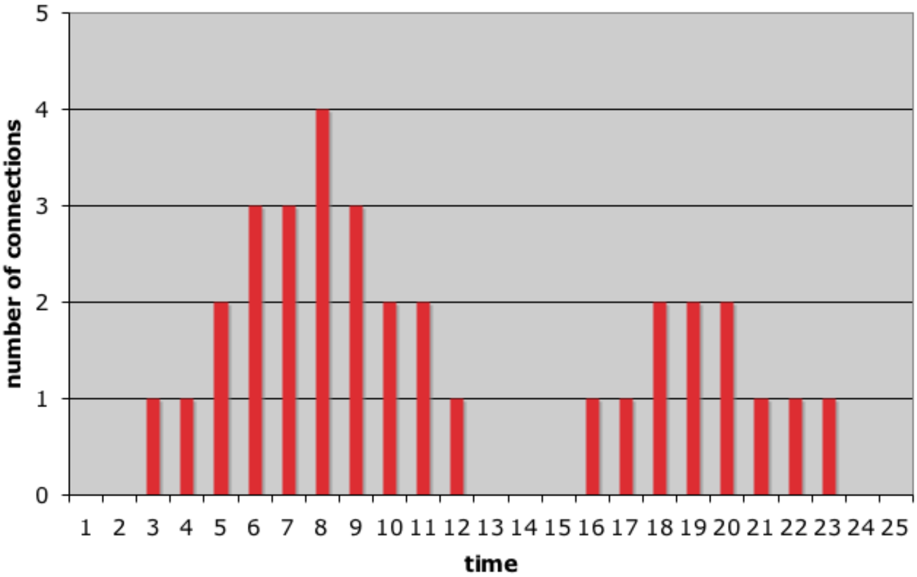


Figure 4.1: Number of connections over time

Hence, we can compute the utilization of a server s between t_i and t_j $(t_j > t_i)$ as:

$$U_{s_{[t_i,t_j]}} = \frac{\sum_{k=i}^{j-1} UR_{s_{[t_k,t_{k+1}]}}(t_{k+1} - t_k)}{t_j - t_i} \tag{4.2}$$

$$U_{s_{[t_i,t_j]}} = \frac{1}{(t_j - t_i) * ConnMax_s}(\sum_{k=i}^{j-1} Conn_{s_{[t_k,t_{k+1}]}} * (t_{k+1} - t_k)) \tag{4.3}$$

In the same example, we have thus $U_{s_{[0,12]}} = (1*2+2*1+3*2+4*1+3*1+2*2+1*1)/(12*5) = 0.37$, meaning that during this period the server is used at 37% of its capacity in average.

Finally, the utilization of the server s during an experiment with a duration T is:

$$U_s = U_{s_{[0,T]}} \tag{4.4}$$

Figures 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8 and 4.9 describe the difference between load-balance and load-unbalance policies by the utilization of the surrogate servers. These figures show the impact of the policies on the utilization of different number of surrogate servers in different traffic patterns. Here the case of 30 and 50 surrogate servers is described with $400k$ and $1000k$ client requests. In figures, x-axis shows the number of surrogate servers where all the surrogate servers are considered to have the same specifications while y-axis presents the utilization of the surrogate servers. The utilization of the surrogate servers is presented in percentage under two request redirection policies i.e. load-balance and load-unbalance. The value of the utilization of the surrogate servers is from 0 and 100. These Figures show how the two different values of the load-unbalancing parameter z affect the utilization of the surrogate servers. The load-balance policy shows the uniform distribution as shown in Figures 4.2, 4.4, 4.6 and 4.8. In this case, all surrogate servers have the equal probability to get the client requests. Load-balance policy shows no peaks and most of the utilization values reside almost in the same region. The change in the value of parameter z affects the requests redirection pattern which affects the surrogate servers' utilization ultimately. As the value of z increases, load-unbalancing occurs gradually. Figures 4.3, 4.5, 4.7 and 4.9 exhibits the load-unbalancing behavior where the value of the load-unbalancing parameter z is set to 1. The greater value of z shows that a group of servers have the more probability than the others to receive the requests. Load-unbalance shows the opposite pattern of the load-balance, where only a small number of surrogate servers get most of the requests as shown the high peaks in the start of Figures 4.3, 4.5, 4.7 and 4.9. The surrogate servers with more load have the more probability to get the requests and they become the bottle necks.

In case of less surrogates, the utilization is better than with more surrogate servers. This trend of utilization can be seen in case of both redirection policies where decrease in the average utilization curve is gradual with the increase in the number of surrogate servers, if all the other simulation parameters are constant, as shown in Figures 4.4, 4.5, 4.8 and 4.9.

The number of requests has an impact on the utilization of the surrogate servers. In case of smaller number of requests, the surrogate servers utilization is lower and it increases with the increase in the number of requests, if the number of surrogate servers and the other simulation parameters are constant. For example, in case of 30 surrogate servers with $400K$ requests, the utilization of the surrogate servers is lower as shown in Figures 4.2 and 4.3. In case of $1000k$ requests, it shows the higher utilization behavior as shown in Figures 4.4 and 4.5 as compared to $400k$ requests. The same trend of the utilization to the number of requests is followed in case of 50 surrogate servers as shown in Figures 4.6, 4.7, 4.8 and 4.9. It shows that utilization of the surrogate servers has a linear relation to the number of the requests.

4.2.3 Energy Consumption in CDN Servers

Each surrogate server consumes a constant quantity of energy just by being turned on. The rest can be considered proportional to the utilization. In this context, we assume energy consumption to be proportional to the ratio of active connections against the maximum simultaneous connections each surrogate server is able to handle. These number of active connections accounts for the work being done at the server side to retrieve the data (handling of the index), the disk IO (Input Output) to fetch the data, the network connection and the cache management policies. An extended model could be used in order to derive power consumption based on actual load on IO, networks and CPU (Central Processing Unit), and could be considered in future. However, even using such a basic assumption is sufficient to compare the energy consumption of different configurations of number of servers and traffic requests. It must be clear that our main aim is not to estimate exactly the energy consumption, but rather to get a metric for comparing several scenarios.

4.2.4 Surrogate Server's Energy Consumption Model

On the basis of above described definition for energy consumption in CDN servers, we propose a model of energy consumption in surrogate servers. Surrogate server's utilization is used as a parameter in order to measure its energy consumption. First, we calculate the power consumed by the surrogate servers while serving the contents to the clients or to the neighboring surrogate servers (in case of cooperation). The power consumed by the surrogate server at a given time can be calculated as follows:

$$P_{s_{[t_1,t_2]}} = P_{idle_s} + \frac{Conn_{s_{[t_1,t_2]}}}{ConnMax_s}(P_{Max_s} - P_{idle_s}) \tag{4.5}$$

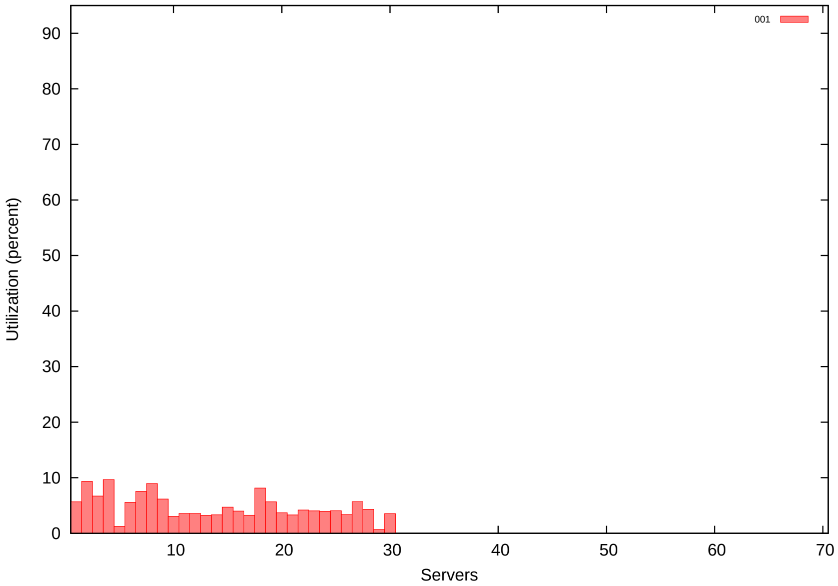


Figure 4.2: Utilization (%) of 30 surrogate servers for serving 400k client requests (Load-Balance).

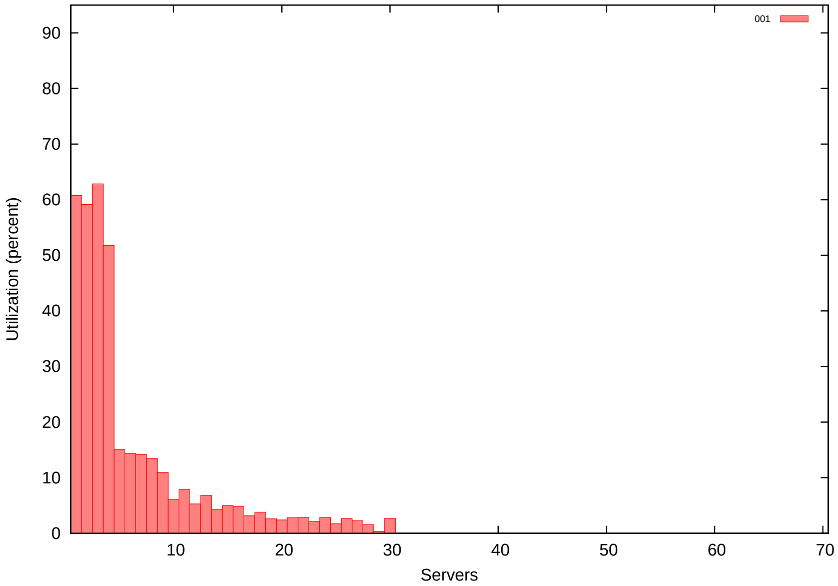


Figure 4.3: Utilization (%) of 30 surrogate servers for serving 400k client requests (Load-Unbalance).

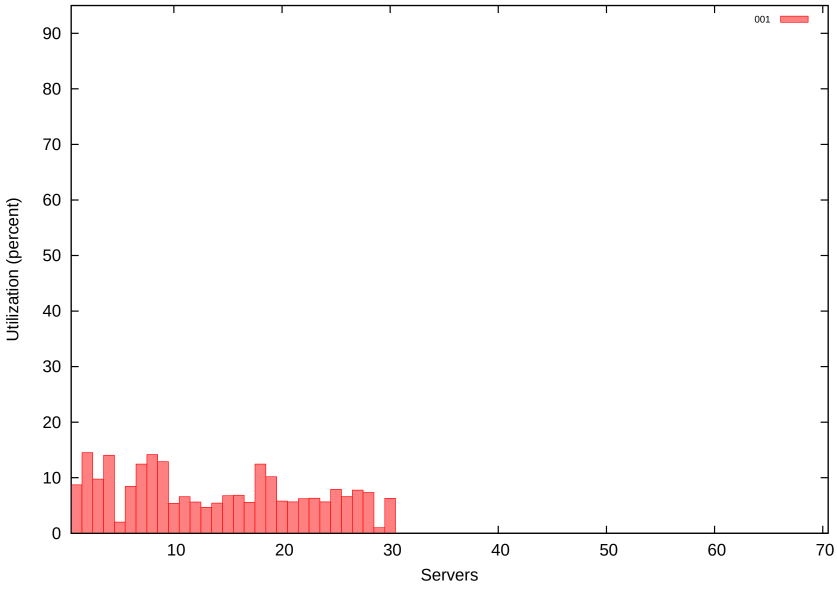


Figure 4.4: Utilization (%) of 30 surrogate servers for serving 1000k client requests (Load-Balance).

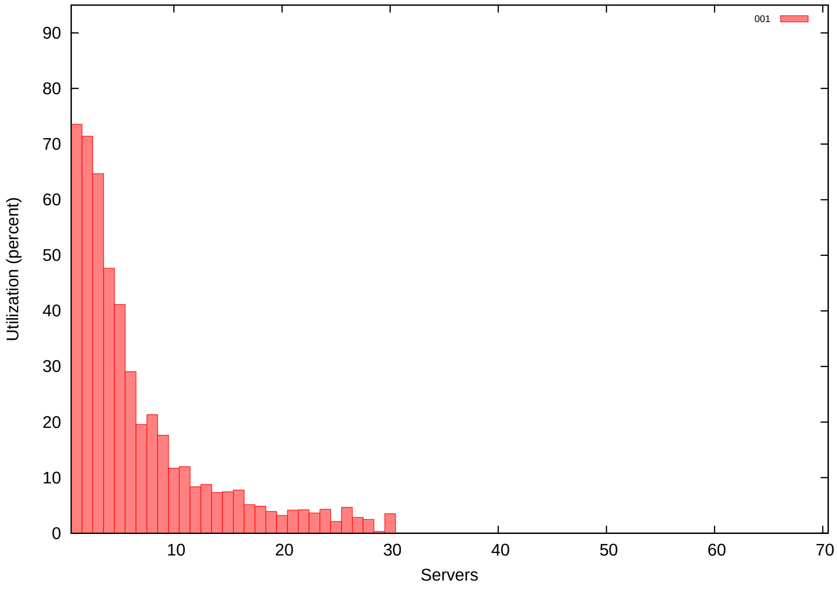


Figure 4.5: Utilization (%) of 30 surrogate servers for serving 1000k client requests (Load-Unbalance).

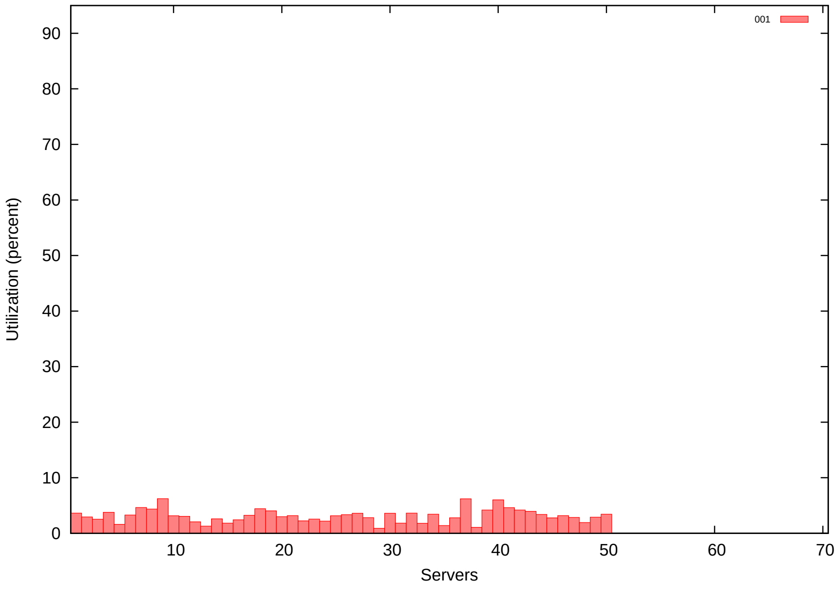


Figure 4.6: Utilization (%) of 50 surrogate servers for serving 400k client requests (Load-Balance).

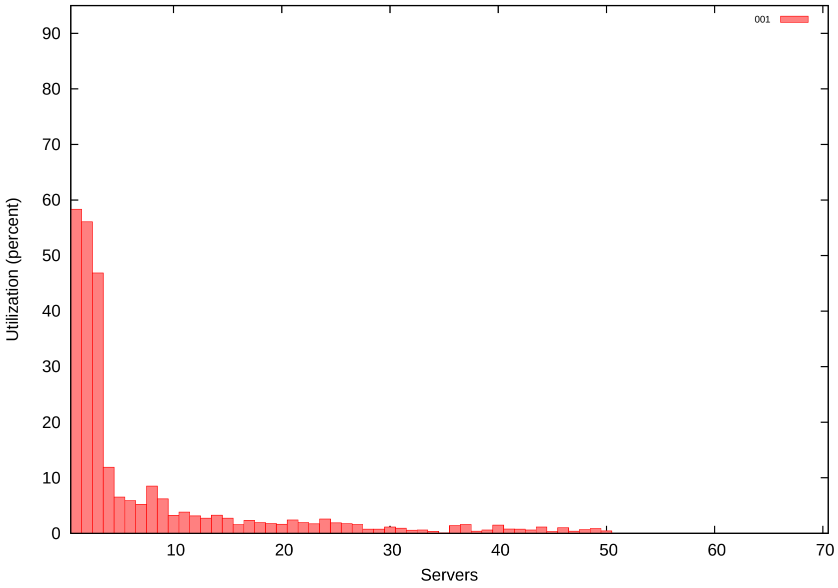


Figure 4.7: Utilization (%) of 50 surrogate servers for serving 400k client requests (Load-Unbalance).

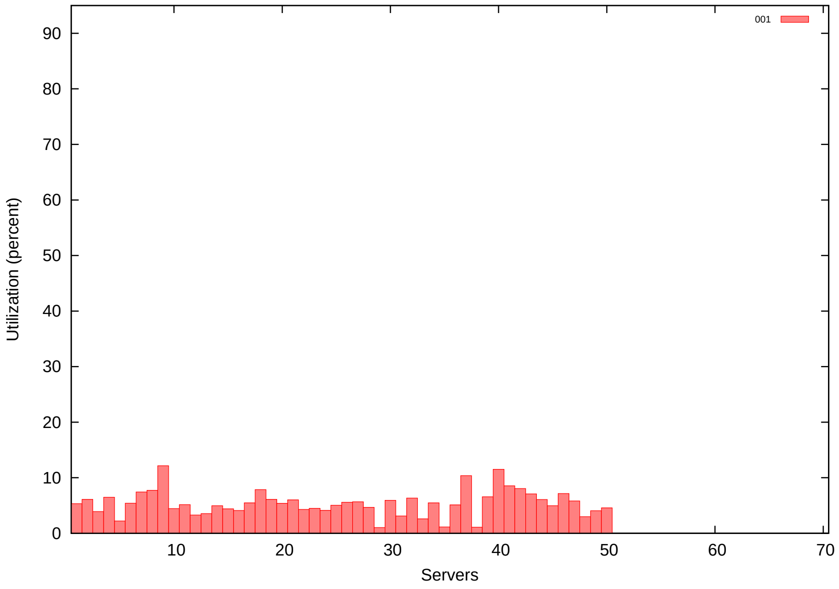


Figure 4.8: Utilization (%) of 50 surrogate servers for serving 1000k client requests (Load-Balance).

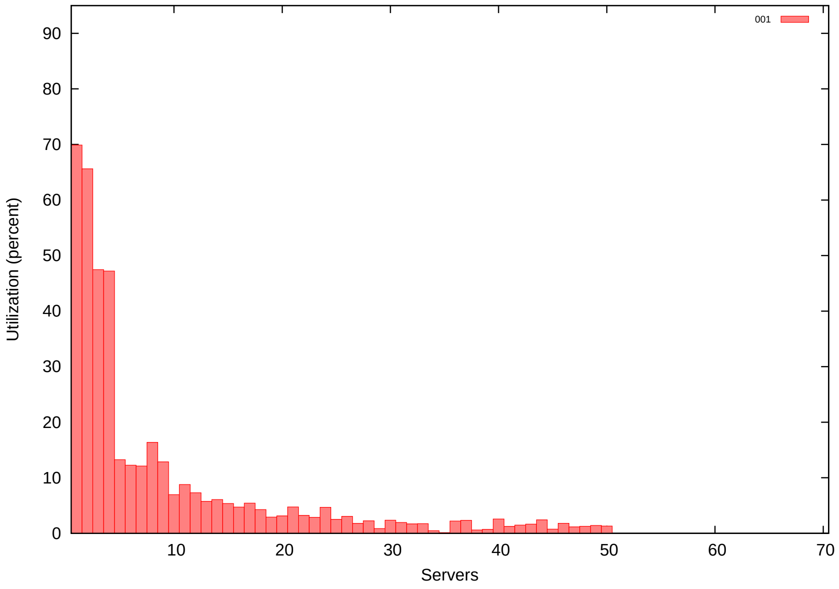


Figure 4.9: Utilization (%) of 50 surrogate servers for serving 1000k client requests (Load-Unbalance).

where P_s is the power consumed by the surrogate server s. P_{idle_s} is the minimum possible power the surrogate s can consume. In this case when a surrogate server is turned on it is supposed to consume a constant amount of power if it is idle and doesn't have any request to serve i.e. it is completely unloaded. P_{Max_s} is the maximum possible power a surrogate server s can consume, when it is fully loaded.

Between time intervals t_i and t_j the energy consumption $E_{[t_i,t_j]}$ can be calculated as:

$$E_{s_{[t_i,t_j]}} = \sum_{k=i}^{j-1} P_{s_{[t_k,t_{k+1}]}} * (t_{k+1} - t_k) \tag{4.6}$$

$$E_{s_{[t_i,t_j]}} = \sum_{k=i}^{j-1} (P_{idle_s} + \frac{Conn_{s_{[t_k,t_{k+1}]}}}{ConnMax_s}(P_{Max_s} - P_{idle_s})) * (t_{k+1} - t_k) \tag{4.7}$$

$$E_{s_{[t_i,t_j]}} = (t_j - t_i) * P_{idle_s} + \frac{1}{ConnMax_s}(P_{Max_s} - P_{idle_s}) \sum_{k=i}^{j-1} Conn_{s_{[t_k,t_{k+1}]}} * (t_{k+1} - t_k)$$

Linking the energy consumption and the utilization model proposed earlier, we obtain:

$$E_{s_{[t_i,t_j]}} = (t_j - t_i) * P_{idle_s} + (P_{Max_s} - P_{idle_s}) * U_{s_{[t_i,t_j]}} \tag{4.8}$$

Finally the total energy E consumed by a surrogate server s is shown by the following equation:

$$E_s = E_{s_{[0,T]}} \tag{4.9}$$

4.3 Energy Aware CDN Redirection Policies

It is observed that traditional CDN client request redirection policies don't take into account the energy conservation. We have derived the CDN client request redirection policies from the recently discussed CDN basic policies i.e. Load-Balance and Load-Unbalance. We have applied two popular energy conservation techniques to these basic policies, in order to gain the energy conservation and also to analyze their impact on CDN operations and services. We have applied powering-off servers and Dynamic Voltage Frequency Scaling (DVFS) techniques. The under-utilized surrogate servers are not necessary to be kept ON. It is better to turn-off the under-utilized surrogate servers, after serving the current requests, in order to gain the energy savings. The processor frequency can be adjusted to the lower scales or can be changed dynamically according to the load of the surrogate servers, to attain the reduction in energy consumption. The proposed policies are evaluated into the following parts.

4.3.1 DVFS Aware Policies

In normal CDN conditions, utilization of the surrogate servers is not according to their capacity. Mostly the surrogate server has low or medium load. In case of lower load the user requests can be served even by having lower processor frequencies. One of the energy saving techniques is to turn-off the underutilized servers, but in some conditions, it is not always useful, for example when the availability of the service is crucial. Also, the turning-off some servers may have an impact on the services provided i.e. dropping some requests, delays in request completion etc. In that case there is need to find the opportunity to conserve energy with the high availability of the service and keeping in view the user experience parameters as well.

One of the popular energy saving techniques is DVFS (Dynamic Voltage Frequency Scaling). In typical CDN conditions, a surrogate server always run at higher processor frequency. Processor frequency can be an important factor to consider to save energy consumption in a surrogate server. A surrogate server can serve the requests at lower frequencies as well but this affects the performance of the request completion process. There is a need to make a balance between energy saving techniques and quality of the service. According to DVFS technique, the frequency of processor clock is adjusted dynamically to have a corresponding reduction in the supply voltage. The processor frequency can be adjusted to more than one mode to conserve energy. There is need to consider the different requirements of the request completion process and processor frequency can be set accordingly. The processor frequency can be set into modes like minimum, medium and maximum. The load of the surrogate servers can be considered to apply DVFS in CDNs as well. While considering the case of CDN, DVFS can be applied at local or global scenarios.

To have this functionality, we have proposed local/variable and global/fixed DVFS techniques to derive DVFS aware CDN redirection policies. Global techniques takes into account all surrogate servers in the same way. All the surrogate servers are configured similarly. Local DVFS techniques concentrate on the individual surrogate server level. We have considered only three modes for frequency scaling because it provides the efficient configuration to have energy savings as shown in [Pierson 2011]. With three modes, energy saving is significant enough. Adding extreme intermediate modes doesn't lead to a significant energy saving while complicating the process.

We consider all the surrogate servers have the same specifications. All the surrogate servers are consider to have the Intel (R) Xeon (R) $E5620$ processor. We propose two major kinds of frequency scaling techniques. (1) Global or fixed frequency scaling (2) Local or variable frequency scaling.

In order to apply frequency scaling techniques, we consider the three modes of the processor frequency.

- $F_{min} = 1.6GH_z$:refers to the minimum frequency of the processor
- $F_{med} = 2GH_z$:refers to the medium frequency of the processor and
- $F_{max} = 2.4GH_z$:refers to the processor's maximum frequency

4.3.1.1 Global or Fixed Frequency Scaling.

This technique of frequency scaling is equally applied to all surrogate servers available to serve the requests. All surrogate servers work on a fixed frequency of processor clock while serving the client requests. According to this category of frequency scaling, processor frequency is set to the value which remain constant during the processing time for all types of client requests load. This policy can be more useful in the conditions where the behavior of the load is known. For example, different time of the day or night or during different seasons, the load can have a specific behavior that can allow the content provider to set the processing at a particular rate. However, we have considered them for all kind of traffic loads, in order to have energy/performance constraints, in all cases. We have exhibited the following policies in this regard.

- ***FreqMin:*** This policy shows one extreme of frequency scaling techniques. According to this policy, the processor frequency always work at minimum possible rate F_{min}. This policy is considered to have higher energy saving but it is not always the case because energy consumption is related with the time and processing for long time can augment energy consumption as well.
- ***FreqMed:*** This policy takes the average way to opt frequency scaling techniques between two extremes of F_{min} and F_{max}. This policy considers to fix the processor frequency at the average rate F_{med}.
- ***FreqMax:*** In order to have good system performance and to provide the services in a better way, the processing at the higher rates can be important. For any kind of system load, processing at higher rates accelerates the processing. Regarding the CDN services, providing higher rates of processing client requests can augment the CDN performance and user experience as well, particularly when a large number of contents are requested at the same time. So, at higher loads conditions, higher processing can be useful. Keeping in view its importance, this policy provide the higher processing rate of processor's clock. According to this policy, processor frequency of surrogate server, always function at the highest possible frequency F_{max}, when system has some load.

We have derived following policies by applying global frequency scaling to Load-Balance and Load-Unbalance policies.

- Load-Balance FreqMin
- Load-Balance FreqMed
- Load-Balance FreqMax (or Load-Balance for short)
- Load-Unbalance FreqMin

Table 4.1: Global DVFS Policies

Policy	**Function**	**Surrogate Server Processor Frequency**
Load-Balance FreqMin	Load-Balancing	$F_{min} = 1.6GH_z$
Load-Unbalance FreqMin	Load-Unbalancing	$F_{min} = 1.6GH_z$
Load-Balance FreqMed	Load-Balancing	$F_{med} = 2.0GH_z$
Load-Unbalance FreqMed	Load-Unbalancing	$F_{med} = 2.0GH_z$
Load-Balance	Load-Balancing	$F_{max} = 2.4GH_z$
Load-Unbalance	Load-Unbalancing	$F_{max} = 2.4GH_z$

- Load-Unbalance FreqMed
- Load-Unbalance FreqMax (or Load-Unbalance for short)

All these policies have the basic functioning of load-balancing and load-unbalancing as explained earlier while surrogate servers serve the client requests at the corresponding frequency rates shown in Table 4.1.

4.3.1.2 Local or Variable Frequency Scaling.

In contrast to global frequency scaling policies, this policy is applied to the individual server level. Instead of fixing frequency scaling value, this technique considers the variation in the processor frequency rate dynamically while processing the requests. Frequency of the process is based on the load of client requests. When load of a surrogate server is changed, frequency of the processor of surrogate server can be set accordingly. When a surrogate server has higher load, there is need to process requests rapidly and its processor frequency can be set to maximum speed. When a surrogate server has average load then it is not good to process the requests at the minimum frequency that can slow down the request completion process and the surrogate server can have the higher loads in case of new coming requests but processing them at maximum frequency can lead to increase in energy consumption. So in order to save the energy consumption while server having average load, the surrogate server can work at medium processor frequency. While minimum frequency of the processor can handle lower load of the surrogate server.

At minimum processor frequency, the request completion process is slow. The new coming requests can increase the number of connections and the utilization of surrogate servers is increased. So the surrogate servers are utilized in a better way. Similarly, load of the surrogate server can also be classified into different categories e.g. higher, average and lower loads.

We consider the two local frequency scaling policies, (1) FreqAdapt (2) FreqAdapt2, in order to analyze the different affects of the processor frequency in the context of energy conservation in CDNs. We took load of the surrogate server as the basic parameter to define the DVFS policies. The average load of the surrogate server is divided into three chunks as shown in Table 4.2

- *DVFS MinLoad:* Defines the range when a surrogate server is considered to have the minimum utilization for applying the DVFS energy conservation technique.
- *DVFS MedLoad:* Shows the range when a surrogate server utilization is consider to process at medium frequencies.
- *DVFS MaxLoad:* Illustrates the range of load when the surrogate server is supposed to be utilized at its maximum.

The above classification of the load of surrogate servers provides us opportunity to apply DVFS technique where the frequency of the processor is changed according to the given conditions. At the basis of load classification, the frequency of the processor is changed accordingly. Algorithm 2 presents the pseudo-code for local frequency scaling policies.

- ***FreqAdapt:*** This frequency scaling technique is based on the surrogate servers utilization. It takes the surrogate utilization as a parameter and applies the frequency scaling technique accordingly. We consider the three modes for processor frequency as shown in Table 4.2. According to *FreqAdapt* policy, if a surrogate server has an average load $0 \geq load \leq 0.5$, the surrogate works at the minimum processor frequency F_{min}. The surrogate server processes the

Algorithm 2 Pseudo-code for Local Frequency Scaling Policies

```
$L$ = Load of a surrogate server $s$
$F$ = Clock frequency of a surrogate server's processor
if ($L < DVFSMinLoad$) then
    $F = F_{min}$
else if ($L > DVFSMedLoad$) then
    $F = F_{max}$
else
    $F = F_{med}$
```

requests slowly. The processor frequency $Freq$ of the surrogate server is set to F_{med} when a surrogate server has the average load from $0.5 < load \leq 0.7$. The client requests are completed at the medium speed. When a surrogate server has the load $0.7 < load \leq 0.9$, the request completion process is faster as the processor frequency of the surrogate server is set to F_{max}. It should be noted that in any case, the load is kept below $MaxLoad$ (in our case 0.90).

- ***FreqAdapt2:*** This policy also applies the frequency scaling technique in CDN. Just like FreqAdapt, it is also based on the surrogate server load and the corresponding processor frequencies of the surrogate servers. The average load of the surrogate servers is considered to change the processor frequency dynamically. According to FreqAdapt2, the same modes and values of the processor frequency are considered i.e. F_{min}, F_{med} and F_{max}. While the load classification is changed for the corresponding frequencies as shown in Table 4.2. The minimum processor frequency F_{min} is set for the loads lies in the range $0 \geq load \leq 0.2$. The surrogate servers processor functions at F_{med} when a surrogate server has the average load from $0.2 < load \leq 0.7$ where the surrogate server processes the requests at medium frequency. The processor frequency F of the surrogate server is set to F_{max} when the server has the higher loads $0.7 < load \leq 0.9$.

FreqAdapt and FreqAdapt2 are proposed to identify the different behaviors of the frequency scaling technique. FreqAdapt2 policy goes more aggressively for F_{min} as compared to FreqAdapt policy, that is the only difference between two policies. We studied also other policies (changing thresholds) but these two were considered

Table 4.2: Local DVFS Policies

Parameter	**FreqAdapt**	**FreqAdapt2**	**Corresponding Processor Frequency**
DVFS MinLoad	$0 \leq load \leq 0.5$	$0 \leq load \leq 0.2$	$F_{min} = 1.6GH_z$
DVFS MedLoad	$0.5 < load \leq 0.7$	$0.2 < load \leq 0.7$	$F_{med} = 2GH_z$
DVFS MaxLoad	$0.7 < load \leq 0.9$	$0.7 < load \leq 0.9$	$F_{max} = 2.4GH_z$

relevant and selected to be presented in this document.

We apply FreqAdapt and FreqAdapt2 to basic proposed CDN redirection policies i.e. (1) Load-Balance (2) Load-Unbalance, to derive the following policies for energy conservation:

- Load-Balance: FreqAdapt
- Load-Balance: FreqAdapt2
- Load-Unbalance: FreqAdapt
- Load-Unbalance: FreqAdapt2

4.3.1.3 DVFS Energy Consumption Model.

As already detailed in Section 4.2.4, power consumption P of a surrogate server between time intervals t_1 and t_2 can be calculated as follow,

$$P_{s_{[t_1,t_2]}} = P_{idle_{cpu}} + \frac{Conn_{s_{[t_1,t_2]}}}{ConnMax_s}(P_{Max_{cpu}} - P_{idle_{cpu}}) \tag{4.10}$$

where P_s is the power consumed by the the surrogate server s. $P_{idle_{cpu}}$ is the minimum possible power the surrogate s can consume. It represents the power consumed by the server s when it is turned-on and it doesn't have any request to serve. In this case the load of its CPU is 0%. When CPU usage is 100%, it consumes the maximum power denoted by $P_{Max_{cpu}}$.

If the processor of a surrogate server s operates on the different frequencies then its power consumption P between the time intervals t_1 and t_2 is

$$P_{s_{[t_1,t_2]}} = P_{idle_{cpu(fs_{[t_1,t_2]})}} + \frac{Conn_{s_{[t_1,t_2]}}}{ConnMax_s}(P_{Max_{cpu(fs_{[t_1,t_2]})}} - P_{idle_{cpu(fs_{[t_1,t_2]})}}) \quad (4.11)$$

Where the $fs_{[t_1,t_2]}$ is the current frequency of the processor between time interval t_1 and t_2. We consider processor frequency constant between t_1 and t_2.

Between time intervals t_i and t_j the energy consumption $E_{[t_i,t_j]}$ can be calculated as:

$$E_{s_{[t_i,t_j]}} = \sum_{k=i}^{j-1} P_{s_{[t_k,t_{k+1}]}} * (t_{k+1} - t_k) \quad (4.12)$$

$$E_{s_{[t_i,t_j]}} = \sum_{k=i}^{j-1}(P_{idle_{cpu(fs_{[t_i,t_j]})}} + \frac{Conn_{s_{[t_k,t_{k+1}]}}}{ConnMax_s}(P_{Max_{cpu(fs_{[t_i,t_j]})}} - P_{idle_{cpu(fs_{[t_i,t_j]})}})) * (t_{k+1} - t_k) \quad (4.13)$$

4.3.2 Consolidation and DVFS Aware Policies

4.3.2.1 Surrogate Server Consolidation.

Consolidation of servers aims to minimize total number of servers or locations in order to utilize the computer server resources efficiently. In existing CDNs normal network traffic conditions, the surrogate servers are not utilized efficiently according to their capacity. Most of the time, most of the surrogate servers are underutilized. They are available to serve the user requests without considering the intensity of the network traffic. A waste of power is examined when the surrogate servers are not utilized according to the capacity and are kept ON even without any load. In some situations, same number of user requests can be fulfilled with a small number of surrogate servers, if they are utilized properly according to their capacity. By minimizing the loss of the energy consumption the price of the product or services can be minimized. When a surrogate server is underutilized, it can be considered to be turned-off in order to save the energy cost. A surrogate server with lower loads also has some requests to serve. It is important that before turning-off surrogate server, the current user requests should be fulfilled. The current requests of the servers can also be redirected towards the other surrogate servers. While turning-off surrogate servers there is a risk to lose some requests that is considerable in order to satisfy the user demands.

If the surrogate servers have the requests to serve then it is not good decision to switch them off, that degrades the services provided to the customers. This method of getting the energy savings on the cost of a higher degradation of the services is not appreciated as the customers and the service providers are interested in the quality of the services. In order to apply the switching-off technique for energy conservation, there is need of a policy in which we get a number of servers underutilized. To get under-utilization of the servers, the client requests should be redirected in a manner that some of the surrogate servers capture most of the traffic. For that purpose, load-unbalance policy provides the opportunity to get a pretty amount of the underutilized surrogate servers and then to apply the switching-off technique to get energy savings.

Load-Unbalancing behavior gives the opportunity to utilize some of the surrogate servers efficiently. The surrogate servers with less or medium load are considered to apply energy saving techniques e.g. switching-off the underutilized surrogate servers, adjusting the processor frequency of the surrogate servers according to their load.

The availability of the surrogate servers is also important. If a server is loaded to 100% of its capacity then it becomes hotter and it needs more cooling that can augment the cooling energy cost. If a surrogate server is loaded above a threshold, then it is not considered to receive the incoming user requests until it satisfies some current requests to minimize its load.

When a surrogate server has no load, it is said in the idle mode. The surrogate servers in the idle mode are considered to turn-off. If all the available surrogate servers have the loads equal to their capacities then there is a need to turn-on a server to satisfy the user requests. In that case, all the switched-off servers have the equal probability to be turned-on. So, a surrogate server is picked up randomly to be turned-on from the pool of the switched-off surrogate servers.

4.3.2.2 Consolidation and DVFS.

Different techniques are used to conserve energy in the distributed systems. Two of the popular techniques are powering-off the servers and Dynamic Voltage Frequency Scaling (DVFS). When the higher availability of the services is needed and energy conservation is also demanded then DVFS technique provides the better opportunity by providing the servers available while having energy conservation by changing the frequency of the processor dynamically. When the higher processing is required and energy conservation is also considered to minimize the service costs, powering-off the underutilized surrogate servers can be useful. In some cases, combining the both techniques of powering-off surrogate servers and changing the frequency of surrogate server processor dynamically or setting it to the lower rates is valuable.

In case of load-balance policy, the distribution of the requests is uniform. All the surrogate servers have almost the same behavior of utilization. So the DVFS technique is feasible to apply in such environment. The policy load-unbalance redirects the majority of the requests to the fewer surrogate servers and the rest of the surrogate servers are underutilized. The underutilized surrogate servers can be considered to be powered-off after serving the current requests. The load-unbalance policy also provides the opportunity to serve the requests at different frequencies. For that reason, we select the load-unbalance policy to apply both above discussed energy conservation techniques. In order to evaluate the impact of the both energy conservation techniques (powering-off and DVFS) in the CDN surrogate infrastructure, the following policies are proposed and implemented in CDNsim.

- ***Consolidation and Global DVFS Aware Policies:*** These policies are derived from the load-unbalance policy. As a group of surrogate servers captures most of the client requests, they become the bottle necks and the rest of the servers have lower utilization. For underutilized surrogate servers, the technique of the switching-off servers is applied. In order to apply this policy, we have determined two thresholds. One of the thresholds is set to identify when to consider a surrogate server to be switched-off. In this policy we consider the

Table 4.3: Global DVFS Policies

Policy	**Function**	**Surrogate Server Processor Frequency**
Load-Unbalance Power-Off FreqMin	Load-Unbalancing + Consolidation	$F_{min} = 1.6GH_z$
Load-Unbalance Power-Off FreqMed	Load-Unbalancing + Consolidation	$F_{med} = 2.0GH_z$
Load-Unbalance Power-Off FreqMax	Load-Unbalancing + Consolidation	$F_{max} = 2.4GH_z$

load for powering-off a surrogate server if the condition $0 <= load <= 5\%$ holds. The underutilized surrogate servers serve the current load and then they are considered to be switched-off. While the other threshold is fixed to reconsider, when new surrogate servers are needed to be switched-on. When the working surrogate servers get higher loads, then the powered-off surrogate servers are turned-on dynamically. All surrogate servers process the client requests on a selected constant processor frequency shown in Table 4.3. It includes the following policies,

- Load-Unbalance Power-Off FreqMin
- Load-Unbalance Power-Off FreqMed
- Load-Unbalance Power-Off FreqMax

- ***Consolidation and Local DVFS Aware Policies:*** These policies are also derived from the load-unbalance policy. These policies also use the same mechanism for surrogate server consolidation as previously discussed policies (global DVFS aware CDN redirection policies). The only difference between previous and this policy is of DVFS mechanism. In these policies, processor frequency of surrogate servers is changed dynamically according to the load of the surrogate server as shown in Table 4.2. These policies include:
  - Load-Unbalance Power-Off FreqAdapt
  - Load-Unbalance Power-Off FreqAdapt2

Figure 4.10 shows the classification of our proposed energy-aware CDN redirection policies.

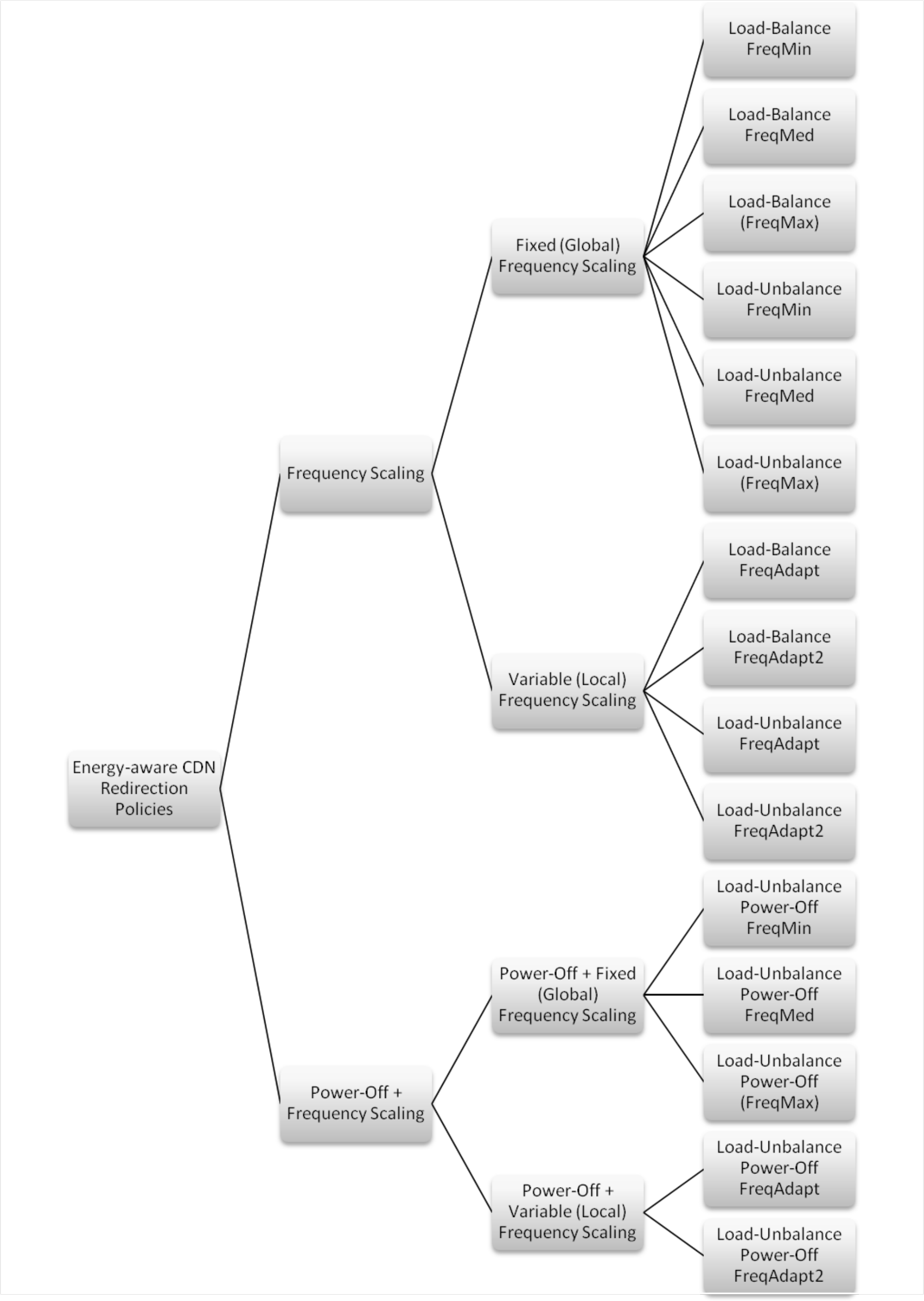

Energy-aware CDN Redirection Policies
Frequency Scaling
Power-Off + Frequency Scaling
Fixed (Global) Frequency Scaling
Variable (Local) Frequency Scaling
Power-Off + Fixed (Global) Frequency Scaling
Power-Off + Variable (Local) Frequency Scaling
Load-Balance FreqMin
Load-Balance FreqMed
Load-Balance (FreqMax)
Load-Unbalance FreqMin
Load-Unbalance FreqMed
Load-Unbalance (FreqMax)
Load-Balance FreqAdapt
Load-Balance FreqAdapt2
Load-Unbalance FreqAdapt
Load-Unbalance FreqAdapt2
Load-Unbalance Power-Off FreqMin
Load-Unbalance Power-Off FreqMed
Load-Unbalance Power-Off (FreqMax)
Load-Unbalance Power-Off FreqAdapt
Load-Unbalance Power-Off FreqAdapt2

Figure 4.10: Energy-aware CDN redirection policies.

4.4 Results Discussion

It is important to note that DVFS aware CDN redirection policies apply the frequency scaling techniques but their main functioning remains the same (load-balancing and load-unbalancing). In order to evaluate our proposed policies, we considered the impact of different evaluation parameters on CDN infrastructure (number of surrogate servers) for serving different number of client requests. Simulation parameters, data set and evaluation metrics details are discussed in Chapter 3. In order to avoid the redundancy of the results having same behavior, we have shown the results of policies which show the global behavior without disturbing the generality. All the combination of policies are evaluated in the same way. In this Chapter, we choose to discuss Load-Balance FreqMax, Load-Unbalance FreqMax and Load-Unbalance Power-Off FreqMax (or shortly Load-Balance, Load-Unbalance and Load-Unbalance Power-Off). Because the other policies with different application of frequency scaling techniques show the same kind of behavior and trend in results. Some considerable difference of values which will be however discussed in Chapter 5. Load-Balance FreqMax and Load-Unbalance FreqMax policies will be used for the comparison purposes in the next chapters.

4.4.1 Load-Balance

This is one of the basic policies i.e. load-balance, as describe in section 4.1 which applies a global frequency scaling technique FreqMax. In order to evaluate this policy, different parameters are considered as described in Chapter 3.

4.4.1.1 Surrogate Server Utilization.

Figures 4.11 and 4.12 show the average utilization of the surrogate servers for different number of surrogate servers to serve different number of client requests. The x-axis represents number of client requests. Utilization of the surrogate servers has a non linear relation with the number of surrogate servers as shown in Figure 4.12. It decreases with the increase in the number of surrogate servers except in case of 30 and 40 surrogate servers where utilization is very close. The utilization curve

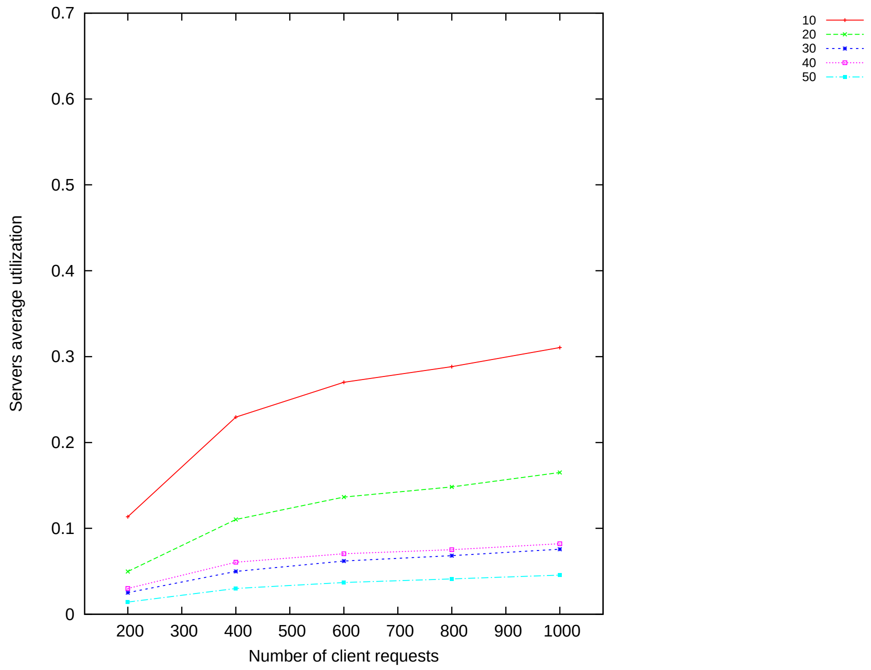


Figure 4.11: Surrogate servers Average utilization vs. number of client requests ($\times 10^3$) for different number of surrogate servers, for ***Load-Balance*** policy.

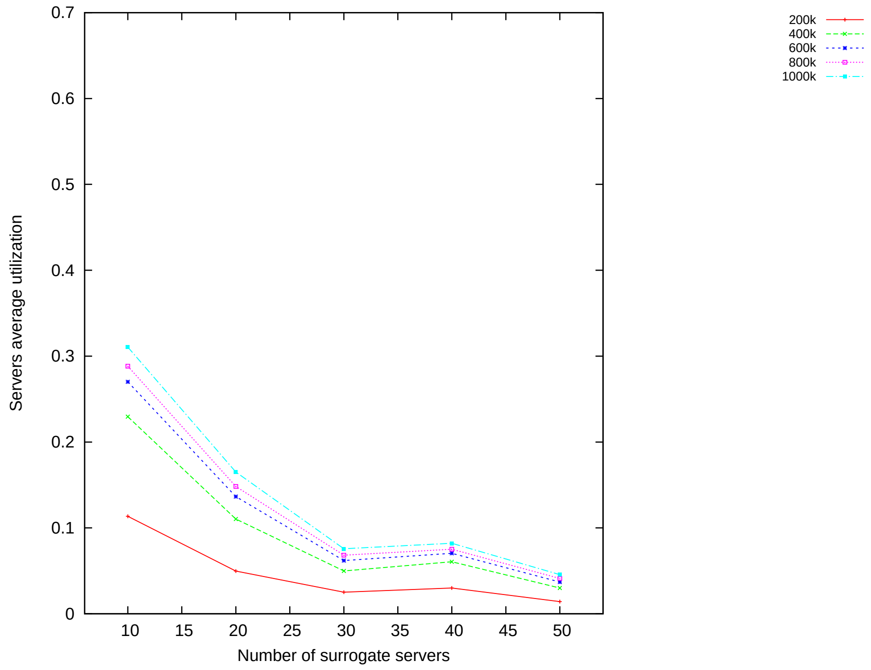


Figure 4.12: Surrogate servers average utilization vs. number of surrogate servers serving different number of client requests, for ***Load-Balance*** policy.

becomes lower as the number of servers increases from 10 to 50. In case of 10 surrogate servers, the average utilization curve shows the highest values while in case of 50 servers the utilization of the surrogate servers is the lowest. The reason for decrease in the utilization is, if we increase the number of surrogate servers for the same number of client requests then the client requests will be divided into smaller parts. In this case, smaller infrastructure shows better utilization than bigger infrastructure while balancing the client requests load. It is the number of requests and the duration of serving a request that makes the utilization of the surrogate servers (see Equation (4)). According to this policy, there is low congestion at the surrogate servers, as the traffic is divided randomly to all the surrogate servers and the traffic at hand is not too high.

The impact of the number of requests is more important here. The utilization of the surrogate servers increases with the increase in the number of client requests as shown in Figure 4.11. If we have a constant number of surrogate servers and we increase the number of requests, it takes more time to serve these requests that increases the utilization. In case of $200k$ requests, the number of requests is very low for the infrastructure which shows very low utilization as compared to the rest.

So we can conclude that in the ***Load-Balance*** policy, the average utilization of the surrogate servers increases with the increase in the network traffic i.e. number of client requests while the average utilization of the surrogate servers decreases with the increase in the number of surrogate servers.

4.4.1.2 Energy Consumption and Energy per Request.

Figures 4.13 and 4.14 present the impact of the number of surrogate servers and the number of requests to the energy consumed by the surrogate servers. Figure 4.14 shows the gradual increase in the total energy consumption by these sets of surrogate servers with increase in number of surrogate servers. There is a linear relation between the number of surrogate servers and the energy consumed by the surrogate servers. A surrogate server consumes constant energy when turned-on, the rest of the energy is proportional to its utilization (see Equation (9)). So, in case of more surrogate servers, utilization of the surrogate servers is decreased

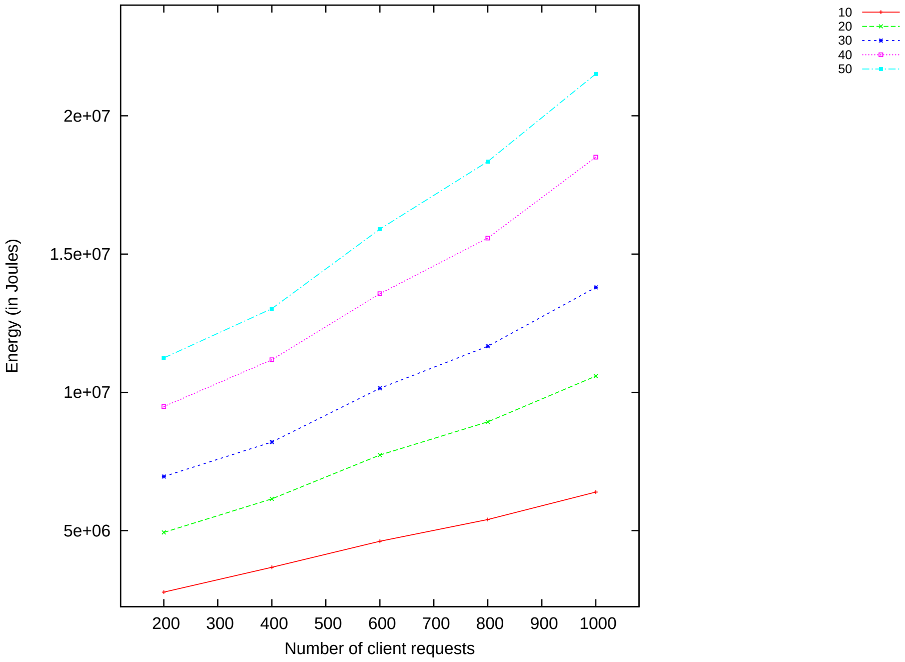


Figure 4.13: Surrogate servers energy consumption vs. number of client requests ($\times 10^3$) for different number of surrogate servers, for ***Load-Balance*** policy.

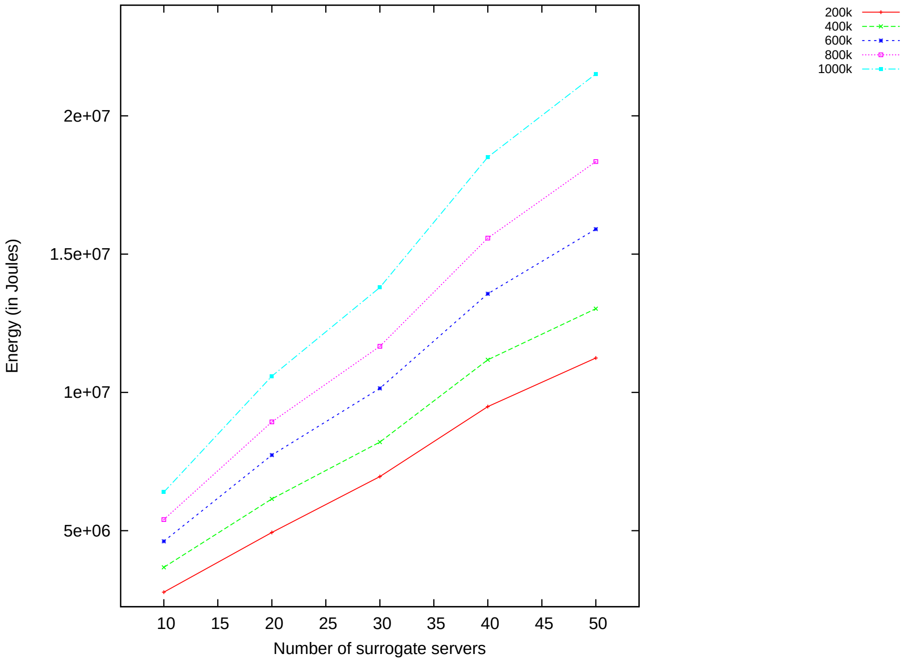


Figure 4.14: Surrogate servers energy consumption vs. number of surrogate servers for different number of client requests, for ***Load-Balance*** policy.

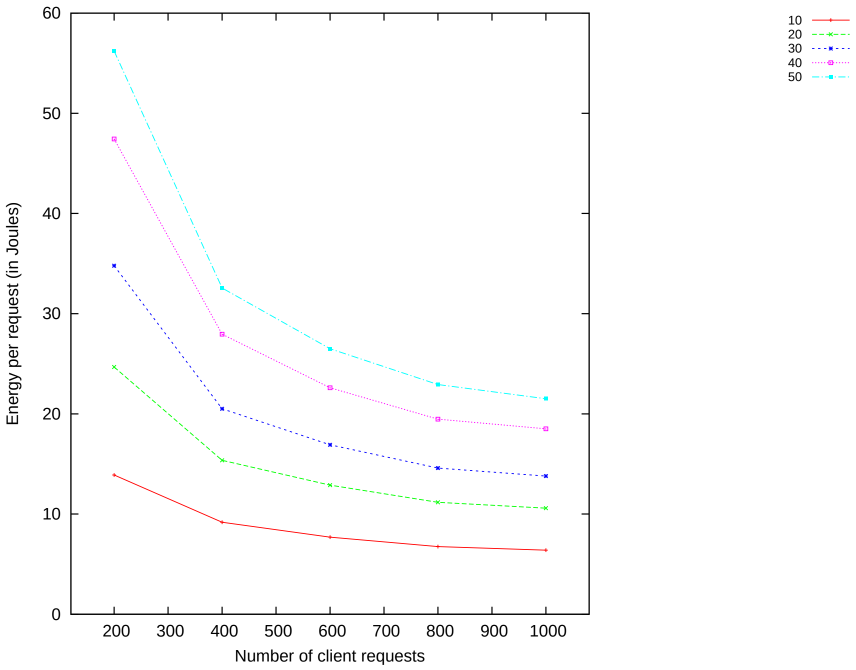


Figure 4.15: Energy per request vs. number of client requests ($\times 10^3$) for different number of surrogate servers, for ***Load-Balance*** policy.

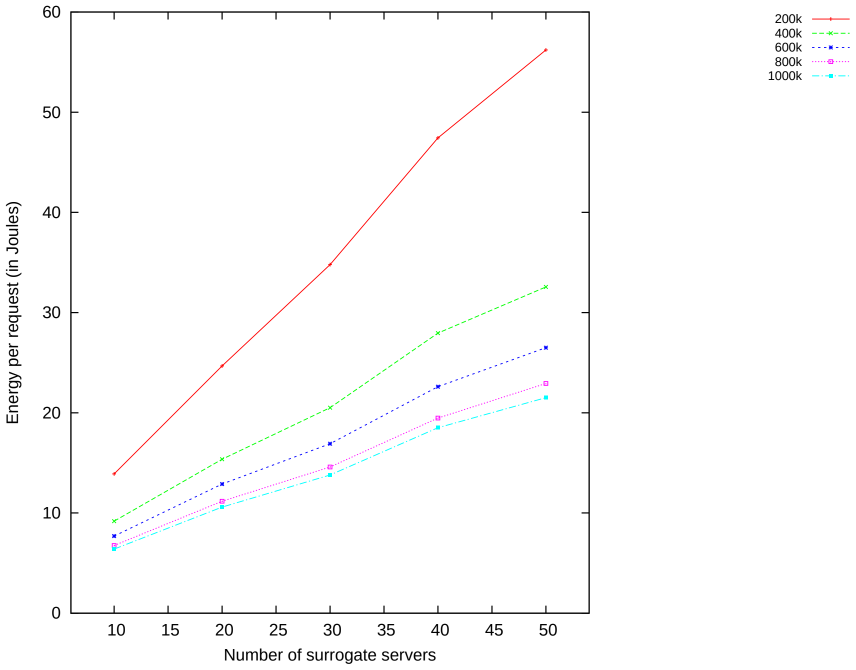


Figure 4.16: Energy per request vs. number of surrogate servers, serving different number of client requests, for ***Load-Balance*** policy.

but in case of energy consumption, it increases with the increase in the number of surrogate servers. In this case, the impact of the constant energy consumption by surrogate servers is higher than the impact of utilization of the surrogate servers, as surrogate servers have low utilization.

There is a linear relation between the number of requests and energy consumed by surrogate servers as shown in Figure 4.13. There is increase in energy consumption as the number of requests increases. Smaller number of client requests causes low power consumption in the surrogate servers. More requests increase the simulation time. As energy is directly proportional to the time consumed, so more requests ultimately result in increased energy consumption.

Figures 4.15 and 4.16 show the energy consumed per request over the different number of requests for different number of surrogate servers. Figure 4.15 shows exponential decrease in the energy consumed per request with the increase in the number of client requests. With the smaller number of client requests, the surrogate server caches are less intelligent. If the surrogate server doesn't have the demanded object in its cache, it asks to the neighboring surrogate servers. So in case of less traffic of client requests, more cooperation among surrogate servers occurs that causes increase in the energy consumption in the other surrogate servers as well. As the client requests traffic increases the caches of surrogate servers start to be more intelligent and they start to cache the popular objects that increase the probability of serving the contents by the surrogate servers receiving the requests by the clients directly. In that case the overall energy consumption by the platform of the surrogate servers is decreased. The difference of energy consumption among the number of requests decreases with the increase in number of requests. It is because of gradual increase in smartness of caches. The energy consumption per request also increases when there are more surrogate servers turned on for serving the same number of client requests traffic (Figure 4.16).

4.4.1.3 Mean Response Time.

Response time is important for client satisfaction to the service provided. Smaller response time is better for client satisfaction. Figures 4.17 shows that as the number

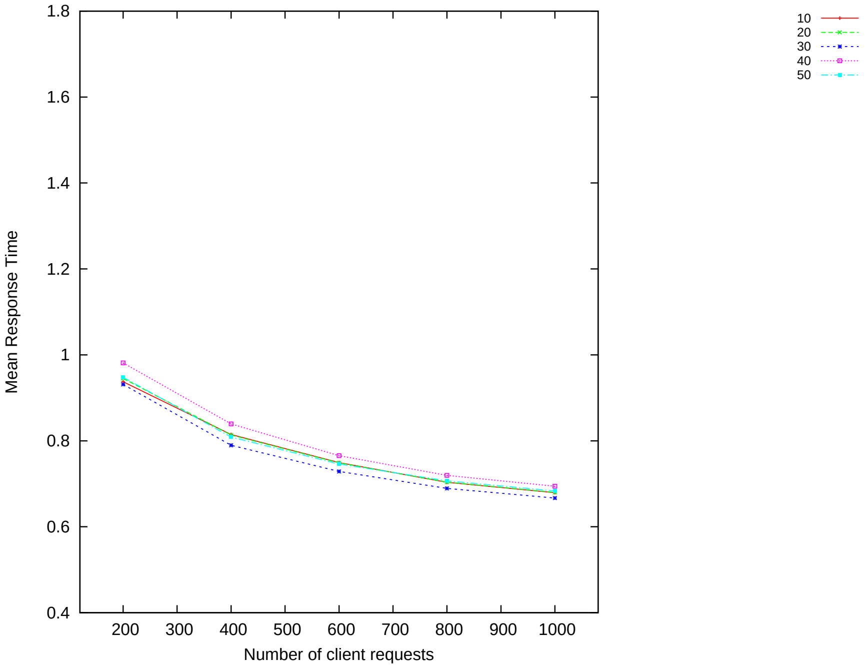


Figure 4.17: Mean response time vs. number of client requests ($\times 10^3$) for different number of surrogate servers, for ***Load-Balance*** policy.

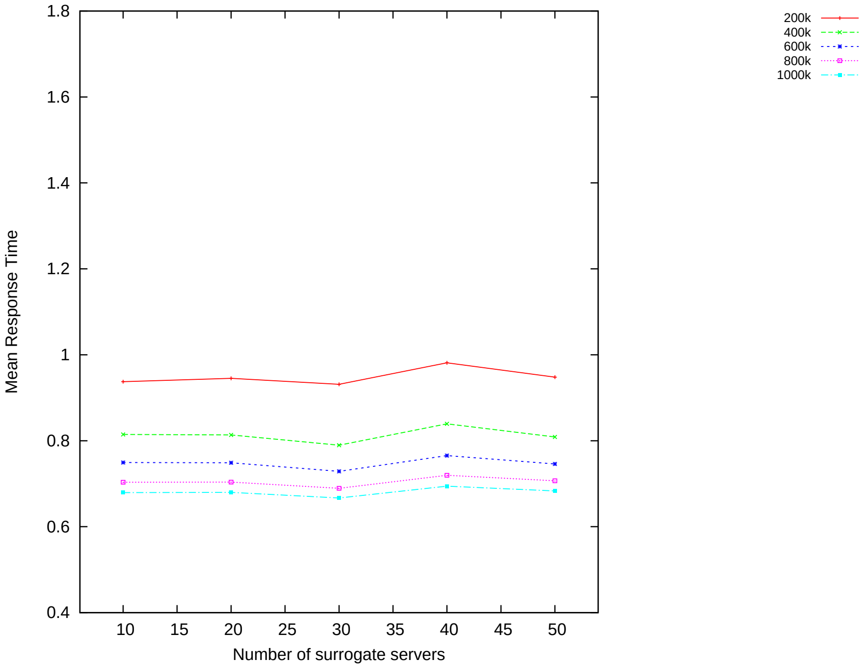


Figure 4.18: Mean response time vs. number of surrogate servers, serving different number of client requests, for ***Load-Balance*** policy.

of client requests is increased there is a gradual decrease in mean response time for any number of surrogate servers. The reason behind this decrease in response time is when we have small number of client requests, the caches of the surrogate servers are not mature and behave like dumb caches. As there are more requests sent to the surrogate servers to serve, with the increase in demand they start to become smarter since the cache replacement policy (LRU) remove older objects to keep popular ones. The contents which are demanded more frequently (popular contents), the surrogate servers caches try to keep them in their caches and delete the unpopular contents to save the space. When a client requests for the popular content, there is more probability of availability of content in the cache of the server so there are more chances that the content will be served directly by the surrogate server and thus the response time is smaller. While if a surrogate server cache is small or empty, then it doesn't differentiate among the contents whether popular or unpopular, so when a client requests for the content, if the surrogate server doesn't have the content in its cache, it asks to the neighboring surrogate servers for the contents. As the client request is not satisfied directly and the request is sent to the other servers, that takes time. The response time for the client request completion is therefore increased.

In Figure 4.18, any number of servers shows the same behavior of mean response time with the change in the number of client requests as described earlier. The impact of the number of surrogate servers is low. Since the client requests are distributed in an uniform way, which doesn't cause the problem of congestion on nodes and low bandwidth.

4.4.1.4 Hit Ratio.

Figures 4.19 and 4.20 illustrate the hit ratio in percentage. It shows the quality of infrastructure management. If a client request is sent for some specific contents, its request is directed towards the corresponding surrogate server. If target surrogate server has the contents, it sends the contents to client and release the connection. If the surrogate server doesn't have the demanded contents, it needs the cooperation of other surrogate servers. Hit ratio shows the degree of the client requests which

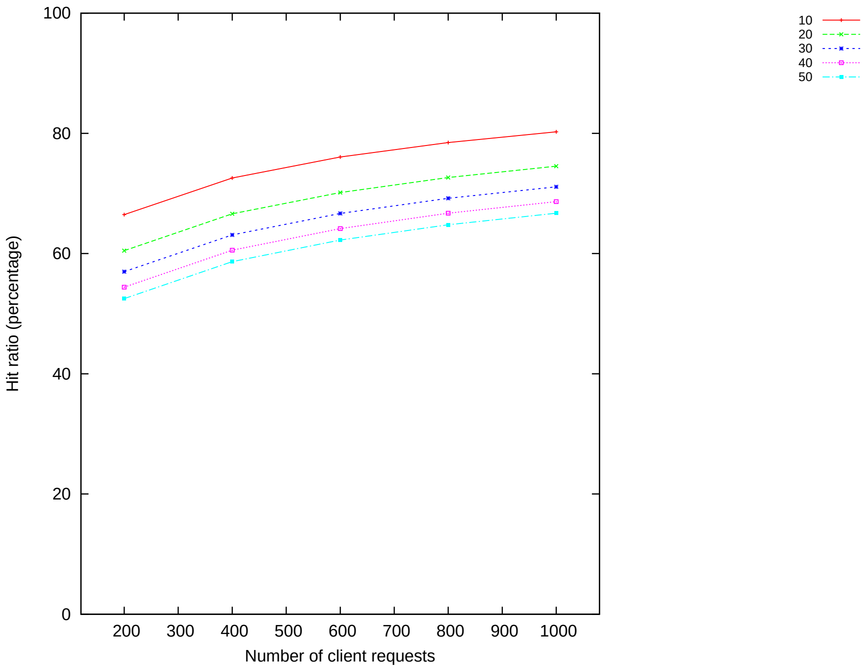


Figure 4.19: Hit ratio (%) vs. number of client requests ($\times 10^3$) for different number of surrogate servers, for ***Load-Balance*** policy.

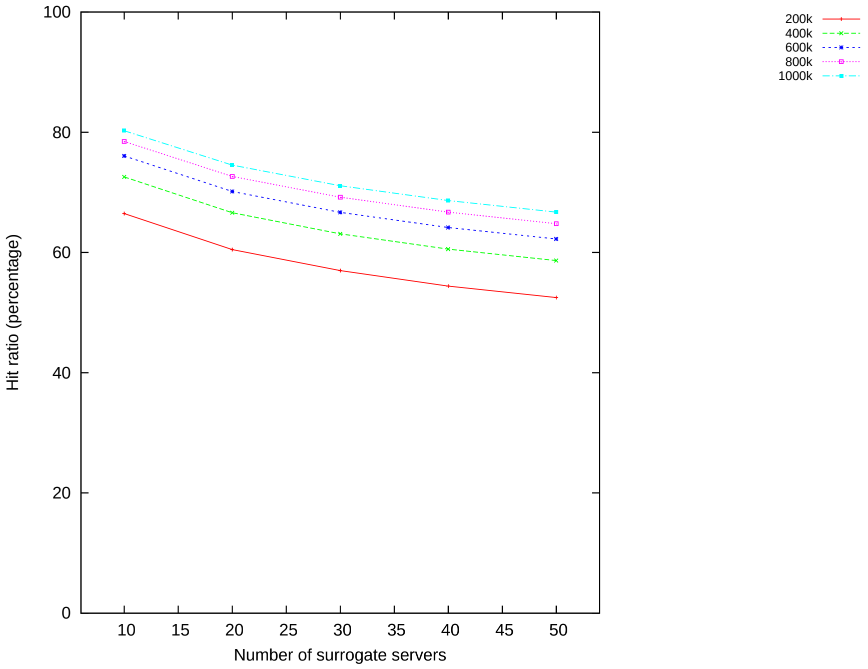


Figure 4.20: Hit ratio (%) vs. number of surrogate servers, serving different number of client requests, for ***Load-Balance*** policy.

are completed directly by the surrogate server that receives the client requests and sends the contents back to client without the cooperation of the other surrogate servers. The direct completion of requests without cooperation helps to minimize the response time. Figure 4.20 shows that hit ratio is better when the number of servers is smaller while with the increase in number of surrogate servers, hit ratio decreases. When we have small number of surrogate servers, all the client requests come to these surrogates servers, as explained earlier, with the time, caches of surrogate servers become smarter and start to cache the popular objects that increases the probability of request completion. The question is why the response time for the smaller number of surrogate servers is higher while having higher hit ratio? The response is : The value of hit ratio depends on the direct satisfaction of client request from targeted surrogate server. Hit ratio doesn't depend on the time for the completion of request. It doesn't care if a request takes lot of time to be completed. It doesn't take into account the congestion on the nodes. A request completion from the path with no congestion and delay and a request completion from the congested path with the double or triple delay have the same value for hit ratio but they definitely have different response time.

4.4.1.5 Failed Requests

Figure 4.21 shows that for all number of client requests and for all number of surrogate servers, there is no failed request (also called aborted request) and all the requested contents are delivered to the clients through the CDN process. This policy follows the modest approach and divide requests to all available surrogate servers, so there is no problem of congestion at the servers or at the network nodes that avoids the problem of denial of services. ***Load-Balance*** policy is useful when high availability of the contents is required.

4.4.2 Load-Unbalance

Load-Unbalance policy with the application of global frequency scaling technique FreqMax is evaluated here with same evaluation metrics. than the previous policy.

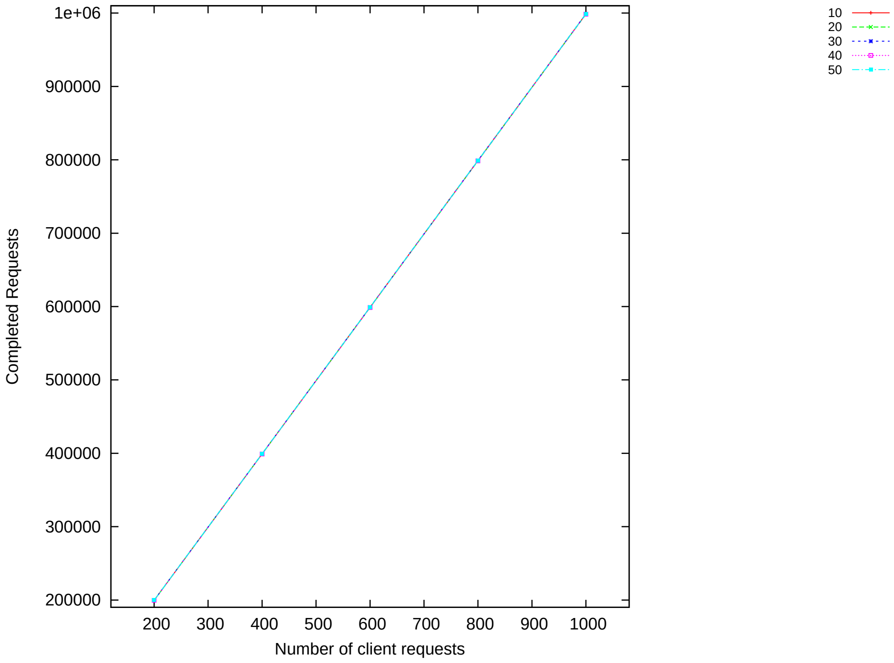


Figure 4.21: Number of completed requests vs. number of client requests (10^3) for different number of surrogate servers, for ***Load-Balance*** policy.

4.4.2.1 Surrogate Server Utilization.

Figures 4.22 and 4.23 present the average utilization of surrogate servers for different number of surrogate servers while serving different number of client requests. The behavior of surrogate servers utilization is similar to ***Load-Balance*** policy. Utilization of the surrogate servers decreases with the increase in the number of surrogate servers and it increases with the increase in the number of client requests.

4.4.2.2 Energy Consumption and Energy per Request.

Figures 4.24 and 4.25 illustrate the energy consumed in joules by the number of surrogate servers while serving different number of client requests. Energy consumption in surrogate servers shows the similar behavior like ***Load-Balance*** policy. There is increase in energy consumed with increase in surrogate servers and client requests traffic. Figures 4.26 and 4.27 also present the similar behavior like ***Load-Balance*** policy for energy consumed per request for different number of client requests in a CDN environment with different number of surrogate servers. It shows energy consumed per request has a non-linear relation with the number of client requests while it has linear relation with the CDN infrastructure of surrogate servers.

4.4.2.3 Mean Response Time.

Figures 4.28 and 4.29 present the mean response time for client requests in CDN infrastructure of different number of surrogate servers, serving different number of the client requests. Mean response time shows again the nonlinear behavior for most of the client requests traffic. The mean response time for client requests decreases with increase in the number of requests. For different number of surrogate servers, mean response time also varies with the change in client requests traffic. For smaller number of client requests, the mean response time for different number of surrogate servers is more than higher number of client requests. In case of different number of

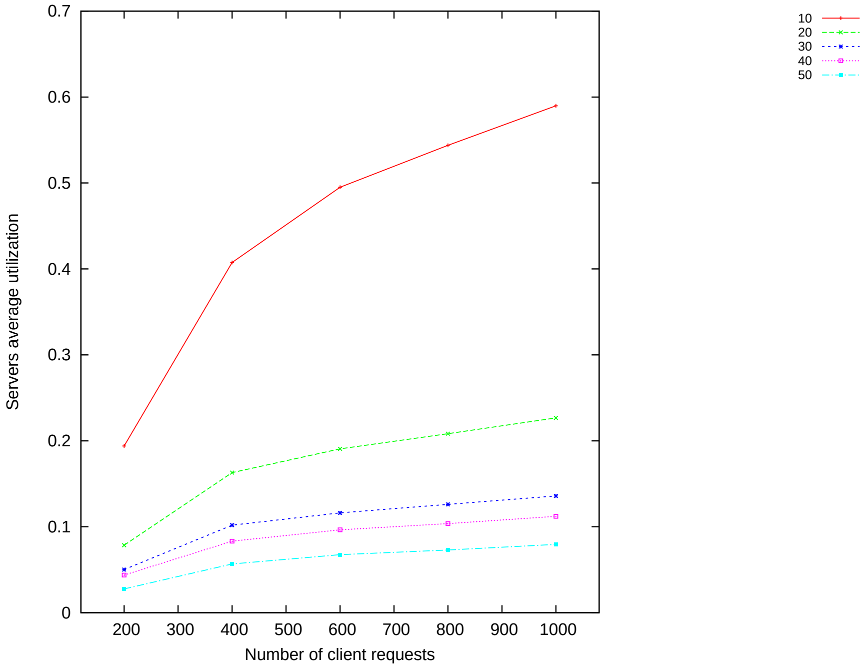


Figure 4.22: Surrogate servers Average utilization vs. number of client requests ($\times 10^3$) for ***Load-Unbalance*** policy.

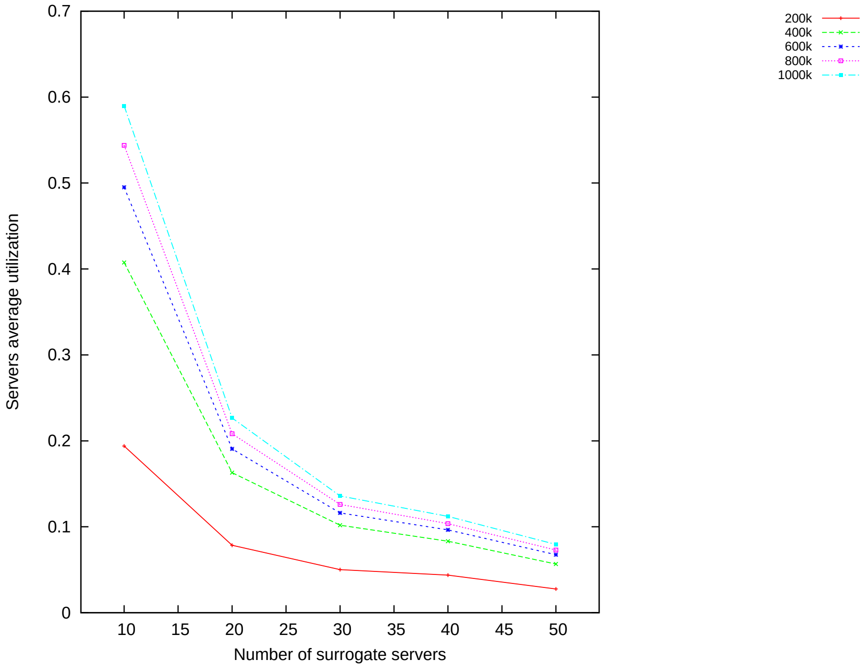


Figure 4.23: Surrogate servers Average utilization vs. number of surrogate servers, serving different number of client requests, for ***Load-Unbalance*** policy.

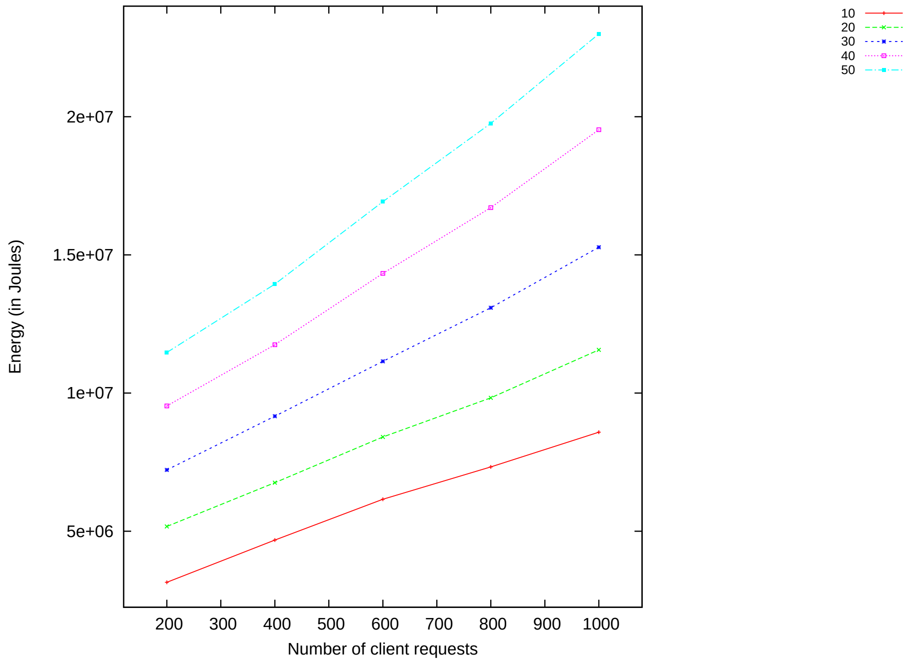


Figure 4.24: Surrogate servers energy consumption vs. number of client requests ($\times 10^3$) for ***Load-Unbalance*** policy.

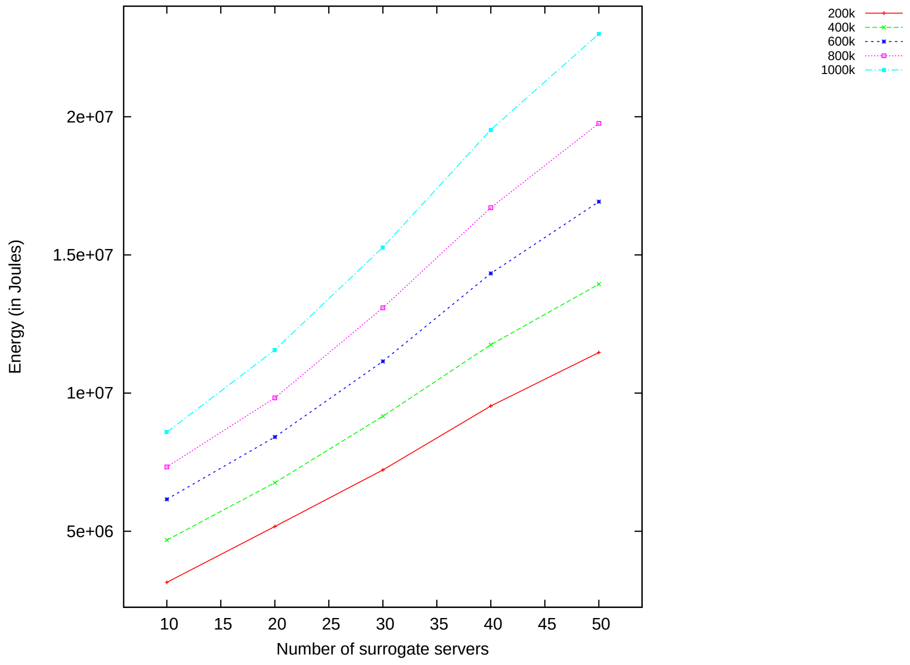


Figure 4.25: Surrogate servers energy consumption vs. number of surrogate servers, serving different number of client requests, for ***Load-Unbalance*** policy.

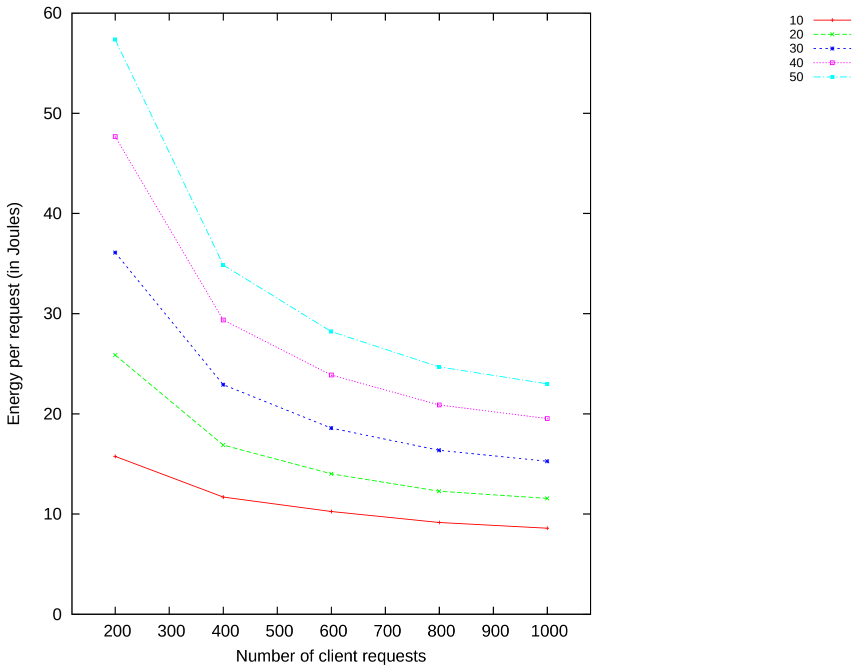


Figure 4.26: Energy per request vs. number of client requests ($\times 10^3$) for different number of surrogate servers, for ***Load-Unbalance*** policy.

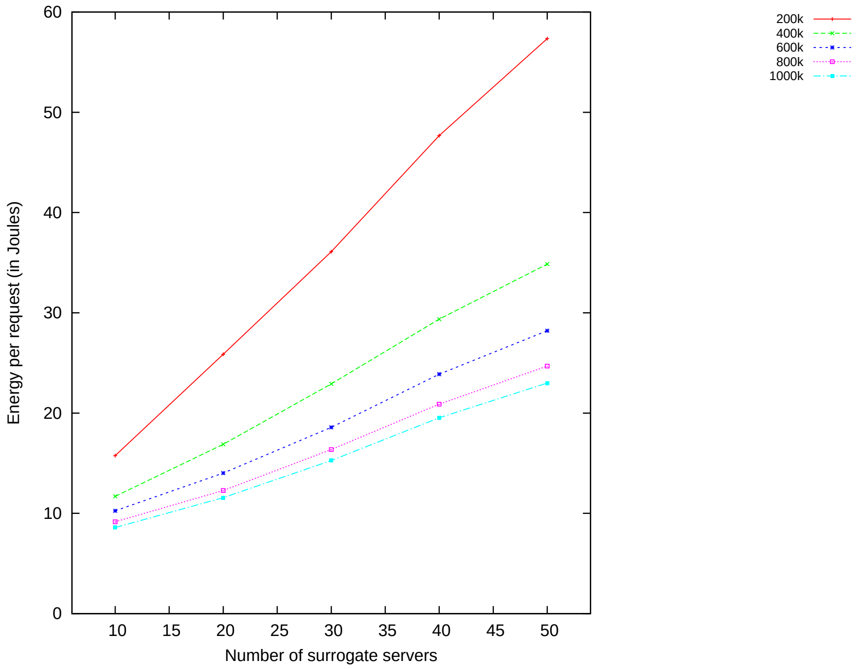


Figure 4.27: Energy per request vs. number of surrogate servers, serving different number of client requests, for ***Load-Unbalance*** policy.

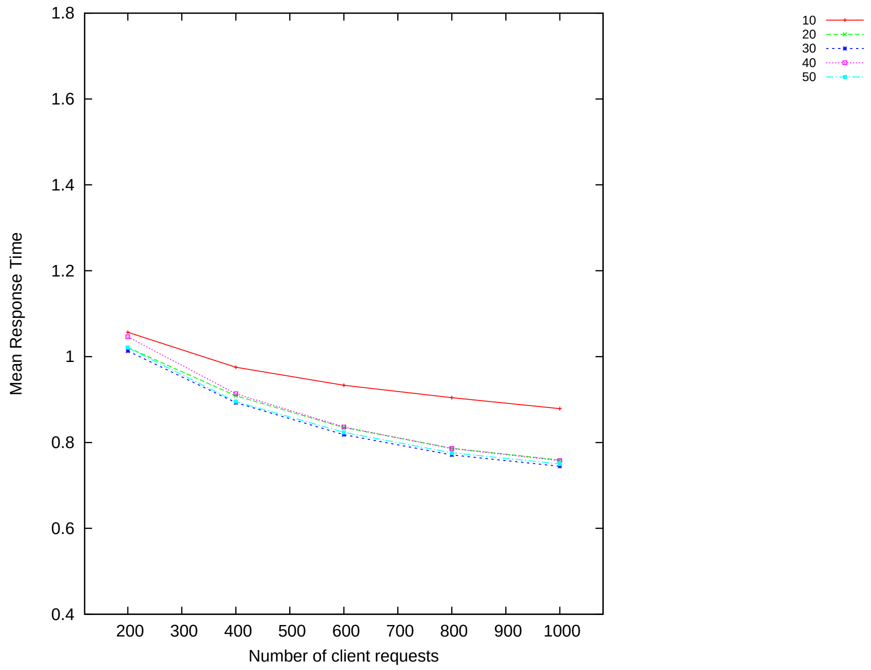


Figure 4.28: Mean response time vs. number of client requests ($\times 10^3$) for different number of surrogate servers, for ***Load-Unbalance*** policy.

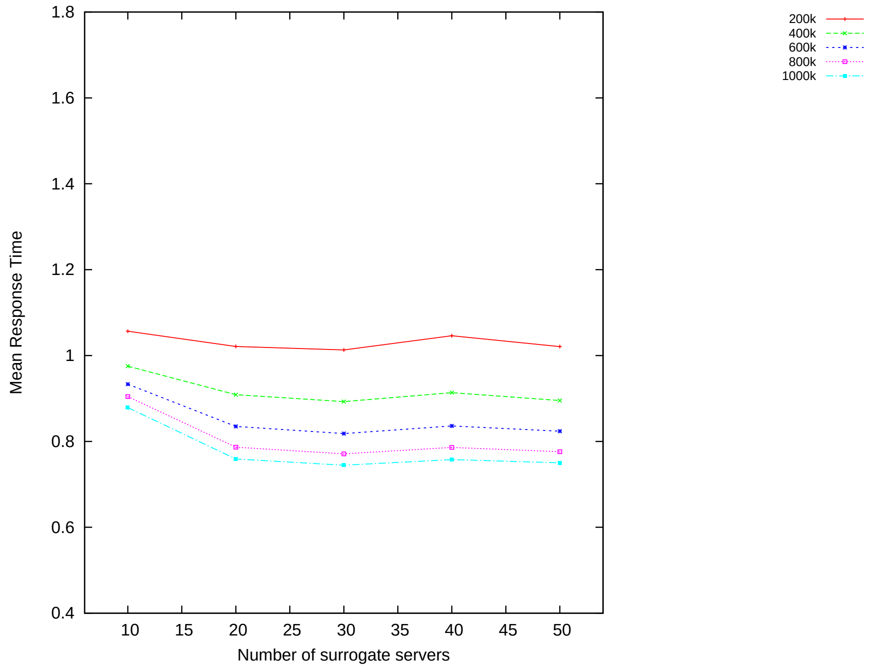


Figure 4.29: Mean response time vs. number of surrogate servers, serving different number of client requests, for ***Load-Unbalance*** policy.

surrogate servers, the mean response time for 10 servers is higher while the values for the other number of surrogate servers (i.e $20, 30, 40, 50$) are closer. In case of load-unbalance, the client requests are redirected towards a subgroup of surrogate servers. Majority of the client request traffic is handled by a small number of surrogate servers. When we have a small number of surrogate servers for a larger number of client requests, the requests are forwarded towards a very small group of surrogate servers which become the bottlenecks and congestion on the nodes occurs. The concentration of the client requests occurs towards fewer servers that causes the saturation on the servers. This behavior causes the increase in mean response time as shown in case of 10 surrogate servers.

4.4.2.4 Hit Ratio

Figure 4.30 and 4.31 show the hit ratio. It shows that in case of Load-Unbalance policy, hit ratio is better when the number of servers is smaller while with the increase in number of surrogate servers, hit ratio decreases. For the CDN infrastructure with smaller number of surrogate servers, the client requests are concentrated to a small group. The caches of the surrogates become intelligent to cache the popular contents that increases the hit ratio. While with the increase in the number of requests, the hit ratio increases because the surrogate server caches become more intelligent with the more traffic of client requests and the popular contents are cached. The availability of the popular contents increases the hit ratio.

4.4.2.5 Failed Requests

Figure 4.4.2.4 presents the failed requests which could not be completed. It shows in case of 10 surrogate servers, some requests could not be completed (i.e. 0.03% of the requests are failed to complete while serving $1000k$ requests) because of the load of the traffic on fewer surrogate servers while the majority of the client requests are satisfied. For $20, 30, 40, 50$ surrogate servers, the number of aborted or failed requests is zero.

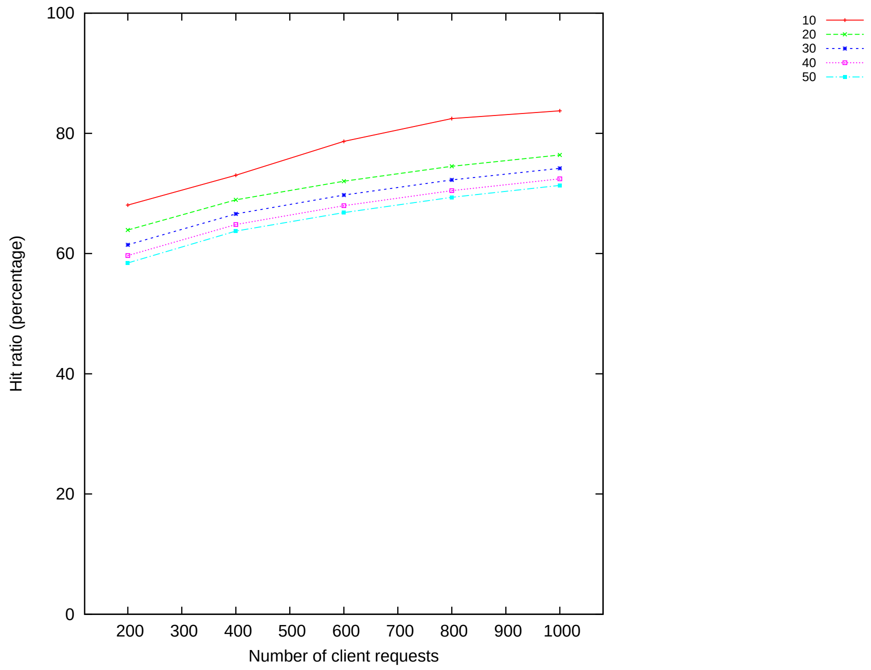


Figure 4.30: Hit ratio (%) vs. number of client requests ($\times 10^3$) for different number of surrogate servers, for ***Load-Unbalance*** policy.

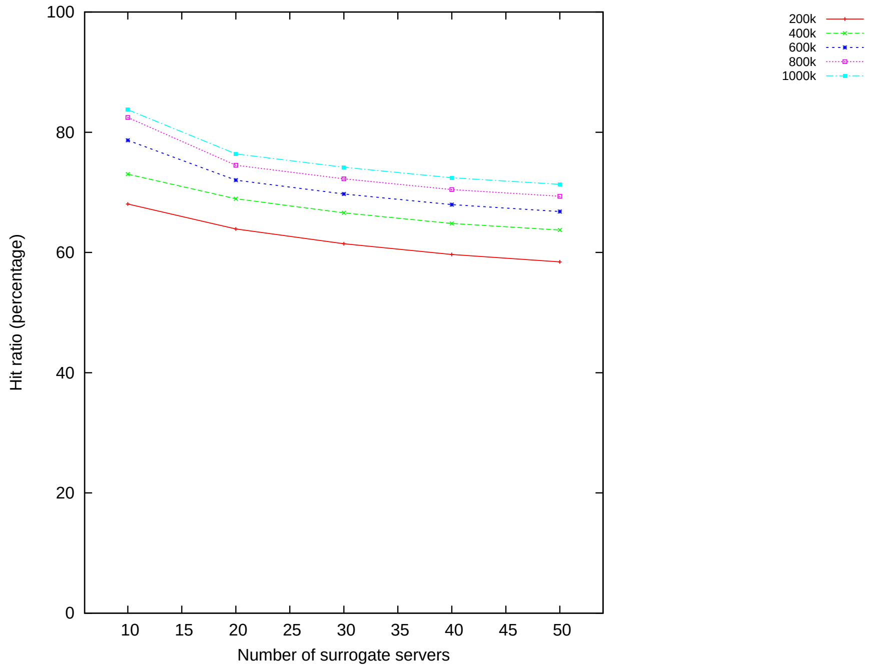


Figure 4.31: Hit ratio (%) vs. number of surrogate servers, serving different number of client requests, for ***Load-Unbalance*** policy.

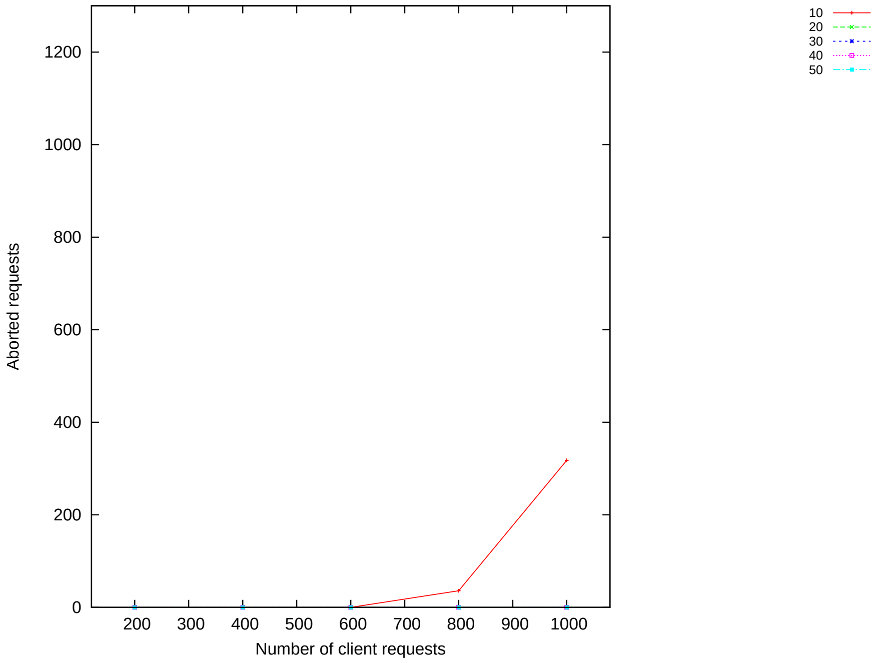


Figure 4.32: Number of failed requests vs. number of client requests ($\times 10^3$) for different number of surrogate servers, for ***Load-Unbalance*** policy.

4.4.3 Load-Unbalance Power-Off

This policy is derived from the policy load-unbalance while processing of requests at surrogate server level is done according to the energy scaling technique of FreqMax. As a group of surrogate servers captures the most of the client requests, they become the bottle necks and the rest of surrogate servers have lower utilization, switching-off servers is applied.

4.4.3.1 Surrogate Server Utilization.

Figures 4.33 and 4.34 show the average utilization of the surrogate servers, for different number of servers and client requests. It shows the surrogate servers under the application of ***Load-Unbalance Power-Off*** policy are utilized in a better way. The shape of the curve is same as the ***Load-Balance*** and ***Load-Unbalance*** policies for CDN infrastructure size and for client requests traffic amount. According to this policy the utilization of the surrogate servers is better than the ***Load-Balance*** and ***Load-Unbalance*** policies because this policy considers the technique of switching-off surrogate servers. Some under-utilized surrogate servers are turned-off, so the requests are divided to the rest of surrogate servers. If the same number of client requests are divided into less number of surrogate servers, that obviously increases the utilization.

4.4.3.2 Energy Consumption and Energy per Request.

Figures 4.35 and 4.36 present the energy consumption for different number of surrogate servers to serve different traffic of client requests. Energy consumption increases with the increase of client requests. Energy consumption also increases with the increase of surrogate servers but its impact is lower as compared to the number of client requests to the energy consumed. Not all the surrogate servers are available through out the time to serve the client requests. Turning-off surrogate servers takes place particularly when the number of available surrogate is 20 or onward. In case of more surrogate servers, the probability of shutting-down surrogate servers increases given a particular traffic. With the increase in number of surrogate servers,

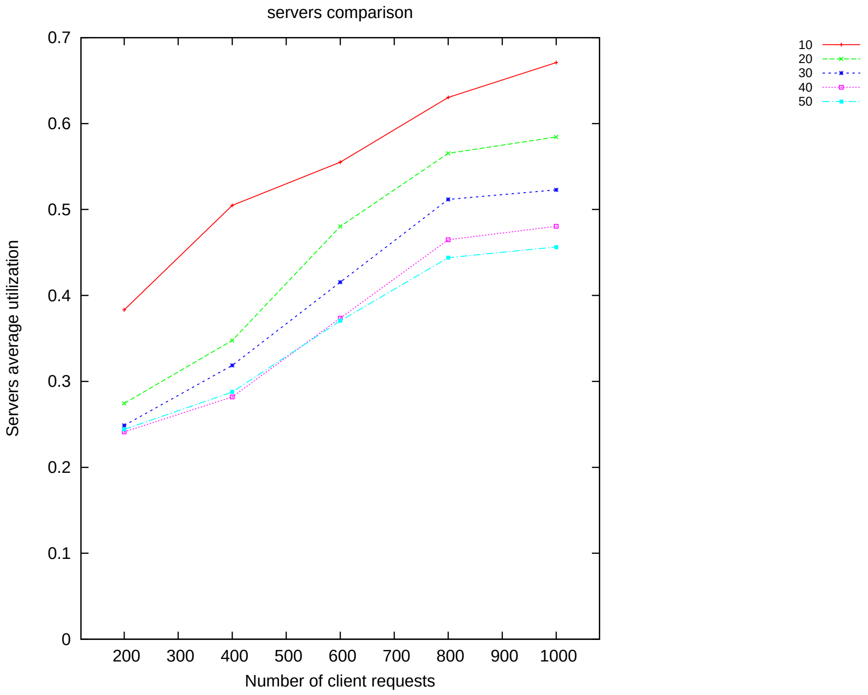


Figure 4.33: Surrogate servers average utilization vs. number of client requests ($\times 10^3$), for different number of surrogate servers, for ***Load-Unbalance Power-Off*** policy.

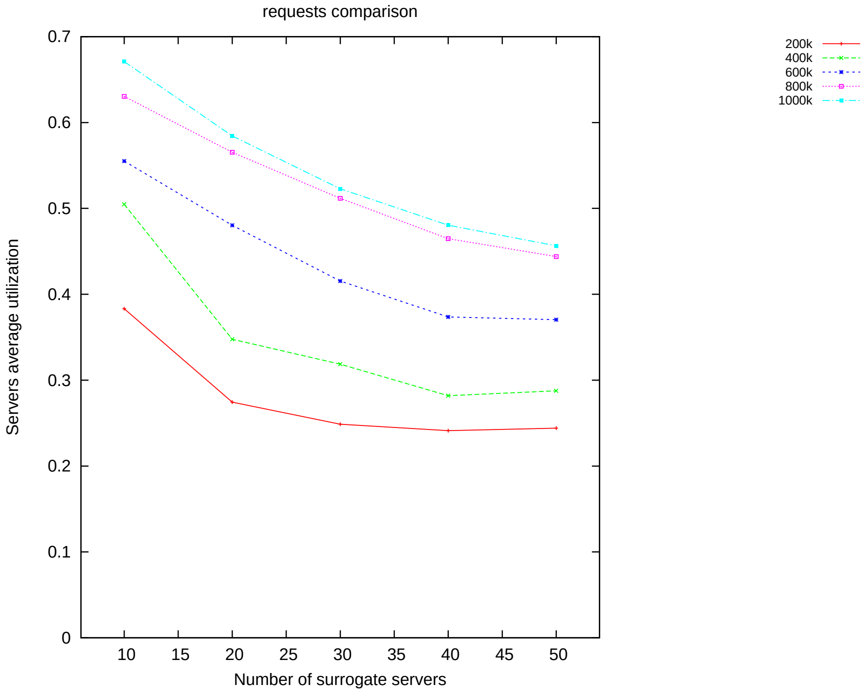


Figure 4.34: Surrogate servers Average utilization vs. number of surrogate servers, serving different number of client requests, for ***Load-Unbalance Power-Off*** policy.

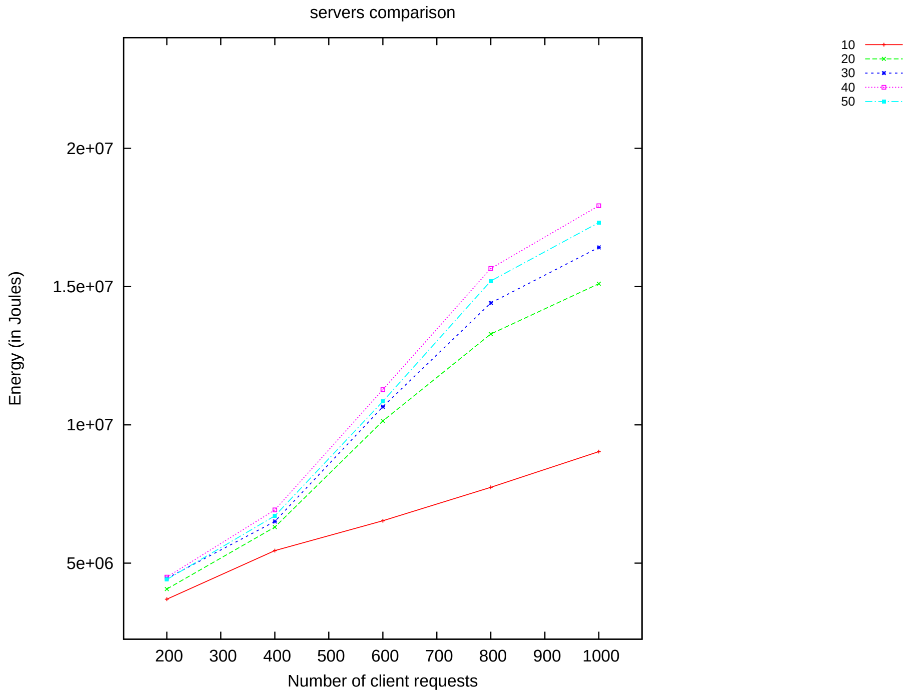


Figure 4.35: Surrogate servers energy consumption vs. number of client requests $(\times 10^3)$ for different number of surrogate servers, for ***Load-Unbalance Power-Off*** policy.

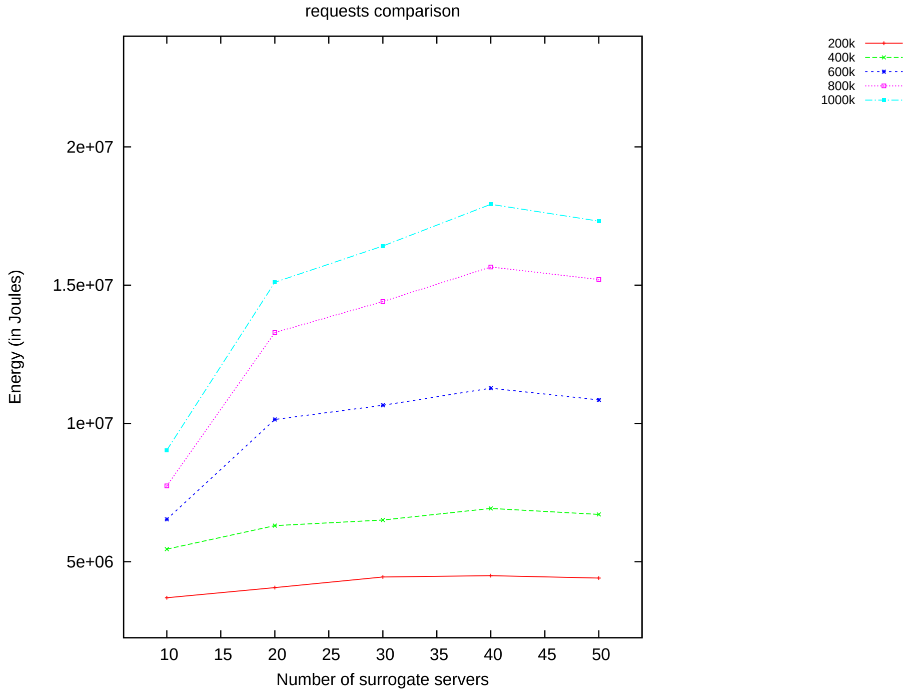


Figure 4.36: Surrogate servers energy consumption vs. number of surrogate servers, serving different number of client requests, for ***Load-Unbalance Power-Off*** policy.

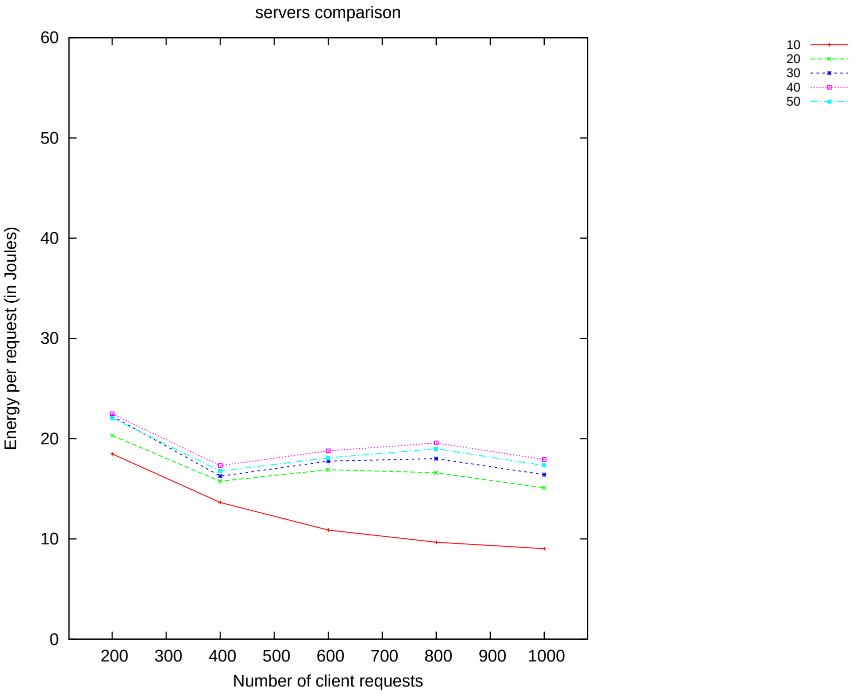


Figure 4.37: Energy per request vs. number of client requests ($\times 10^3$) for different number of surrogate servers, for ***Load-Unbalance Power-Off*** policy.

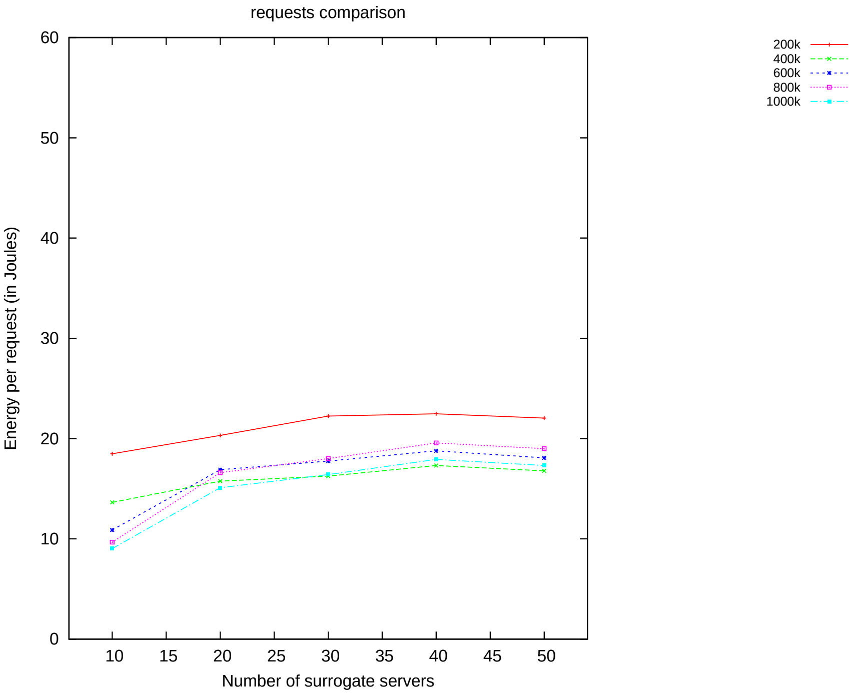


Figure 4.38: Energy per request vs. number of surrogate servers, serving different number of client requests, for ***Load-Unbalance Power-Off*** policy.

the impact of increase in the overall energy consumption is smaller as compared to the previously discussed policies where turning-off surrogate servers doesn't take place i.e. ***Load-Balance*** and ***Load-Unbalance***. This is because of decrease in the static energy consumption (power consumption caused by the surrogate servers when they are powered-on). Figures 4.37 and 4.38 show energy consumption per request. Energy per request increases with the increase in the number of surrogate servers. Energy per request for 10 surrogate servers decreases with the increase in the number of requests because of the cache smartness as discussed earlier. For 10 surrogate servers, energy per request is lesser than the rest of the cases because there are less surrogate servers to serve the requests and the caches become smarter faster, hence it increases the availability and decreases the cooperation among the servers, as a result energy per request decreases. There is more difference of energy consumed for 10 surrogate servers and 20 surrogate servers but this difference is decreased as the number of servers are increased from 20 to 50. But from 20 to 50 surrogate servers, energy consumed doesn't decrease with the increase in the client requests. It shows the impact of the turning-off surrogate servers. When the surrogate servers are turned-off, then at restart, their caches are less intelligent and takes some time to be intelligent that increases the cooperation among the surrogate servers that increases the energy consumption. But at the other hand, as it decreases the number of available servers then it minimizes the number of cooperation contacts and makes faster the smartness of the caches as well. As a result, with the increase in the number of client requests, energy per requests is increased but with the increase among the number of surrogate servers, the difference of energy consumed per request is decreased.

4.4.3.3 Mean Response Time.

Figures 4.39 and 4.40 illustrate the mean response time for different number of surrogate servers for serving different number of client requests. Mean response time is not changing a lot regarding the surrogate servers except in the case of 10 surrogate servers where less surrogate servers serve the requests and because of faster smartness of caches lower response time values are experienced. Mean response time

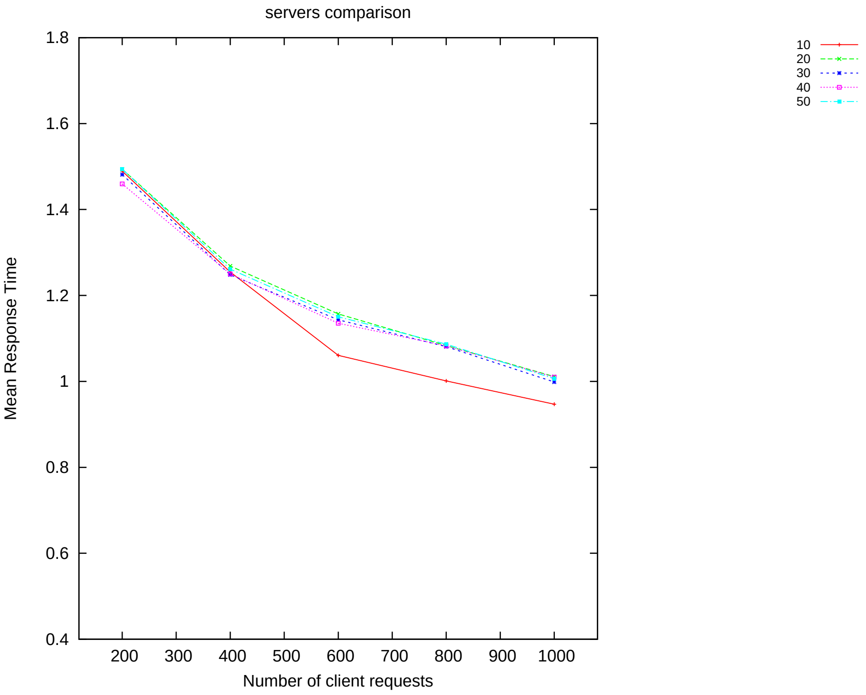


Figure 4.39: Mean response time vs. number of client requests ($\times 10^3$) for different number of surrogate servers, for ***Load-Unbalance Power-Off*** policy.

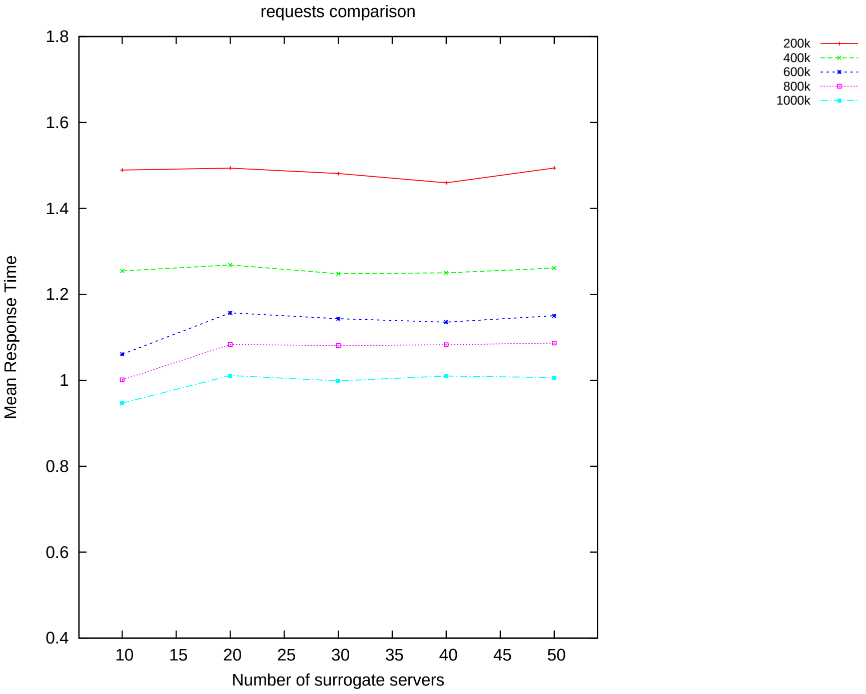


Figure 4.40: Mean response time vs. number of surrogate servers, serving different number of client requests, for ***Load-Unbalance Power-Off*** policy.

decreases with the increase in the number of client requests as discussed earlier in ***Load-Balance*** and ***Load-Unbalance*** policies.

4.4.3.4 Hit Ratio.

Figures 4.41 and 4.42 illustrate, in case of ***Load-Unbalance Power-Off***, hit ratio is better when the number of requests is very low. Figure 4.42 shows that with very small number of surrogate servers i.e. 10 surrogate servers, the hit ratio is better. In case of very small number of surrogate servers, the client requests are concentrated to a small group. The proportion of shutting down the servers is also very low. As before, the caches of the surrogates become intelligent to cache the popular contents that increases the hit ratio. For the set of 10 surrogate servers, the hit ratio have the highest value. When there is a need of restarting the powered-off servers, the restarted server starts as a new server and takes the time to be mature. If there is enough traffic after restarting a server, it serves a number of requests, it starts to be smarter. In some cases when there is very small traffic and during the CDN activity some underutilized surrogate servers are turned-off, the remaining servers fulfill the client requests and there is no need to restart the servers as in case of 200k requests. If the traffic is increased and there is need to restart the surrogate servers and after restarting there is not enough requests to serve, the surrogate caches remain dumb and it decreases the over all hit ratio as in case of 400k and 600k requests. If the surrogate server after restarting have enough requests to be mature and it restart to cache the popular contents, then it increases the overall hit ratio. As it is shown that for most of the surrogate servers 600k of requests is the worse point and after that point, with increase in number of client requests, their hit ratio is increased.

4.4.3.5 Failed Requests.

Figure 4.43 shows the failed client requests. ***Load-Unbalance Power-Off*** policy shows a very low percentage of failed requests for different number of surrogate servers with different traffic of client requests. With the traffic of 200k of client requests, there is no aborted requests and with the others, there are very few failed client requests. With 10 surrogate servers, we have the least number of aborted

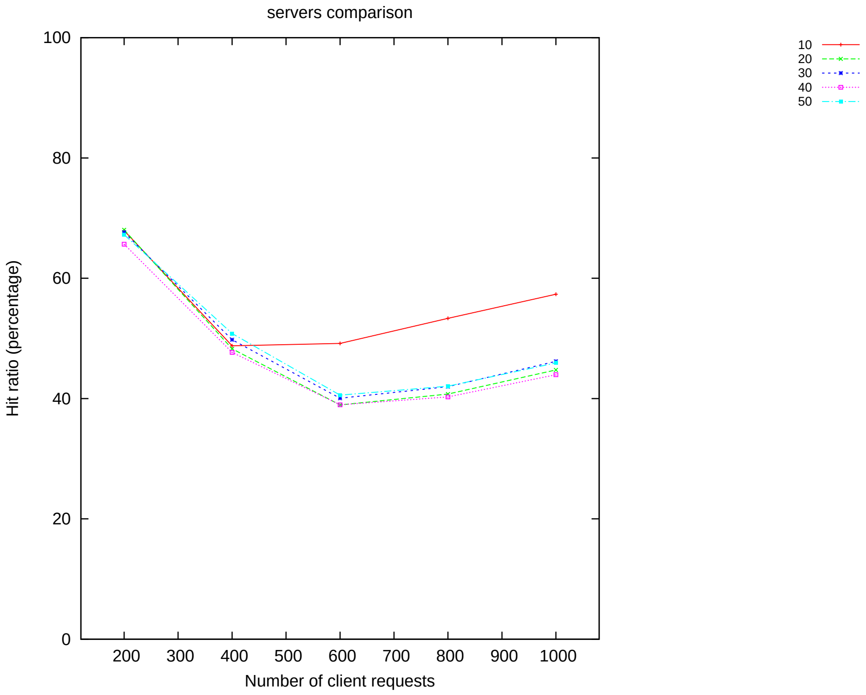


Figure 4.41: Hit ratio (%) vs. number of client requests ($\times 10^3$) for different number of surrogate servers, for ***Load-Unbalance Power-Off*** policy.

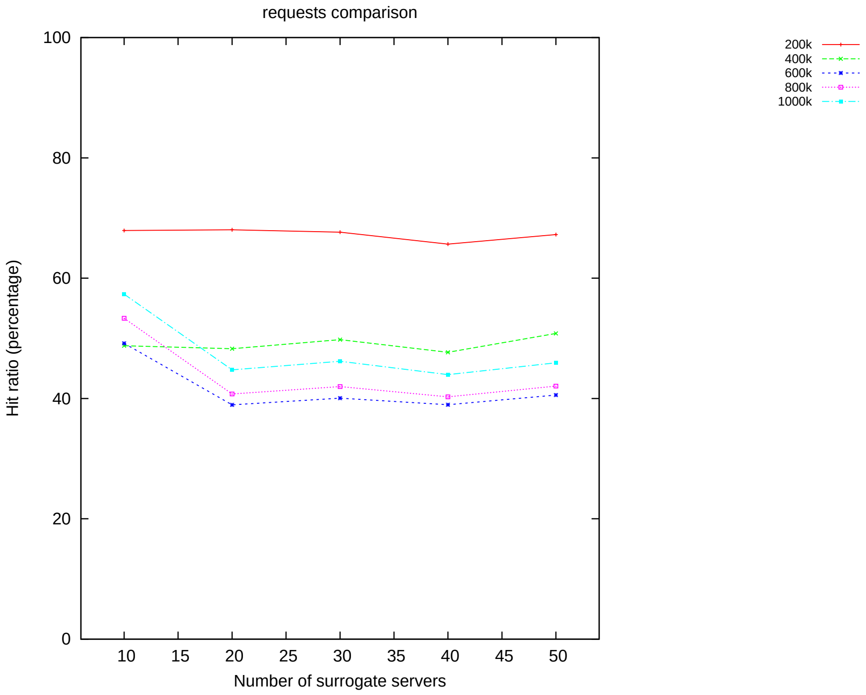


Figure 4.42: Hit ratio (%) vs. number of surrogate servers, serving different number of client requests, for ***Load-Unbalance Power-Off*** policy.

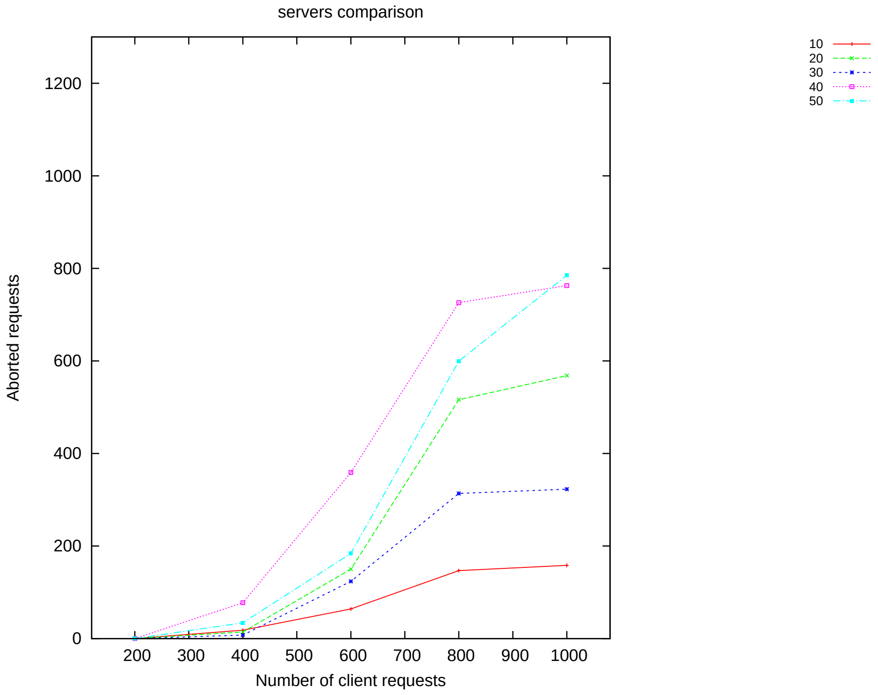
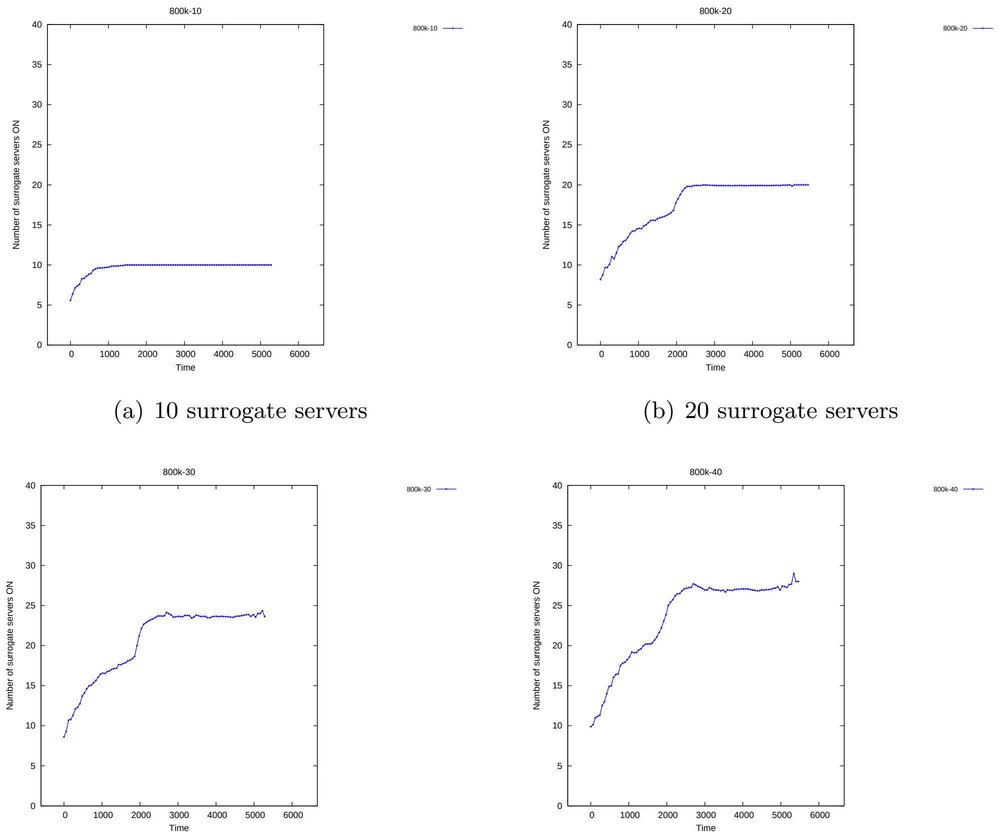


Figure 4.43: Failed number of client requests regarding number of surrogate servers vs number of client requests ($\times 10^3$), for ***Load-Unbalance Power-Off*** policy.

requests. In case of 10 surrogate servers, there is very small probability to turn-off servers that lowers the probability of the failed requests. With increase in the number of requests, the number of aborted requests is increased. For 50 surrogate servers with the 1000k requests, the maximum number of aborted requests are experienced, which is 0.07% of the total number of client requests that is very low.



(a) 10 surrogate servers

(b) 20 surrogate servers

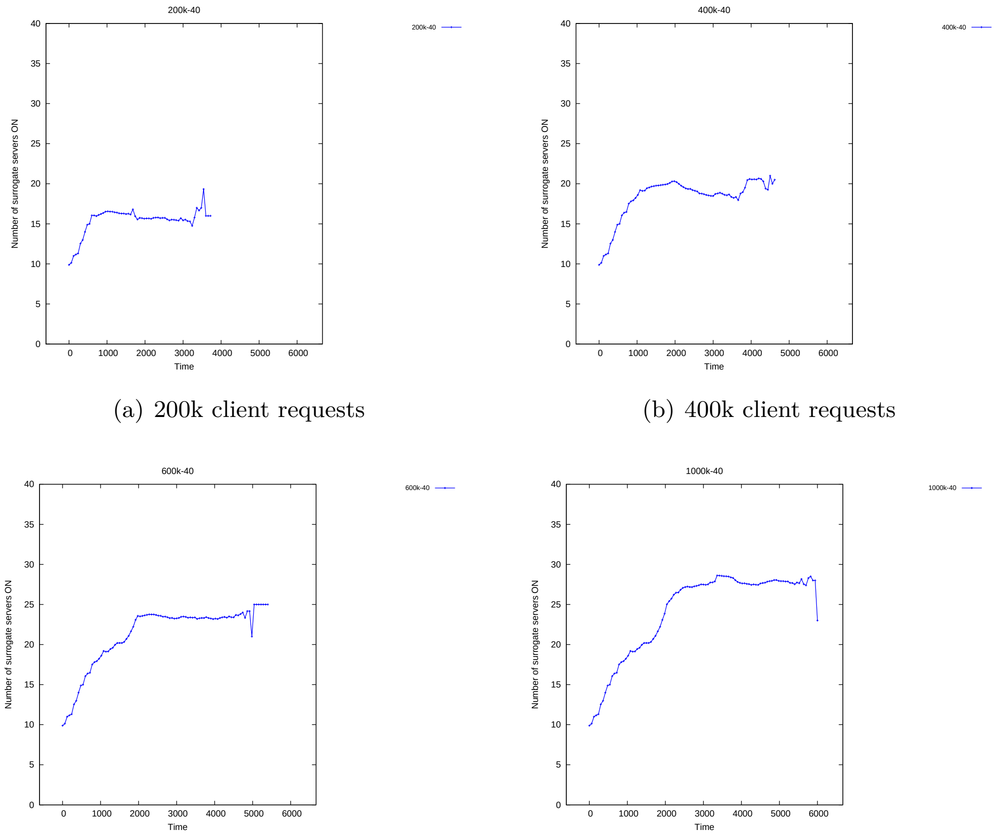
(c) 30 surrogate servers

(d) 40 surrogate servers

Figure 4.44: Number of surrogate servers powered-on over simulation time for serving 800k requests for different number of surrogate servers

4.4.3.6 Number of Surrogate Servers Turned-On.

Figure 4.44 presents surrogate servers powered-on during the simulation time for 800k client requests with different number of surrogate servers. As we use the different seeds for the experiments and each experiment has different simulation time, in order to have a global behavior, we took the average of each 100 units



(a) 200k client requests

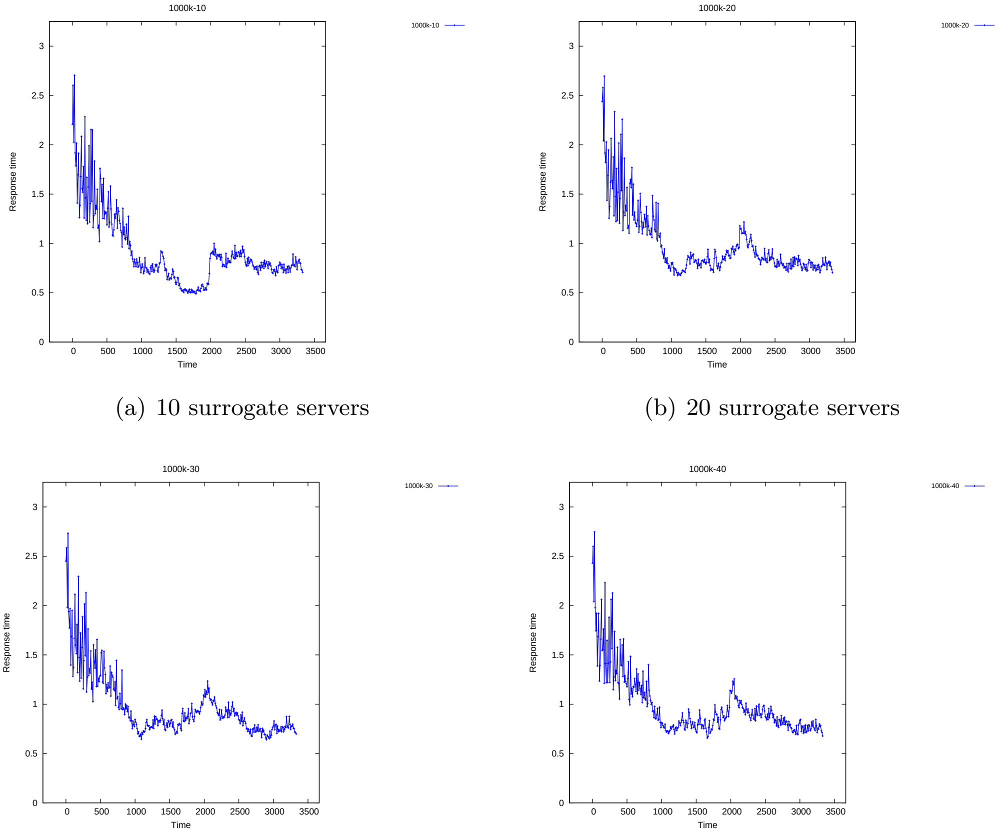
(b) 400k client requests

(c) 600k client requests

(d) 1000k client requests

Figure 4.45: Number of surrogate servers powered-on over simulation time for 40 surrogate servers for serving different number of client requests

of simulation time. In Figure 4.44, each point on x-axis is presenting the average of 100 simulation time units and corresponding average surrogate servers powered-on on y-axis during that interval. We see that, increase in the infrastructure of CDN, provides the more opportunity to power-off surrogate servers. Figure 4.45 show surrogate servers powered-on during the simulation time for infrastructure of 40 surrogate servers serving different number of client requests. With increase in client requests traffic, more surrogate servers are powered-on. We did this for all the combination of surrogate servers and client requests for all proposed policies (that use powering-of technique). All results show similar behavior.



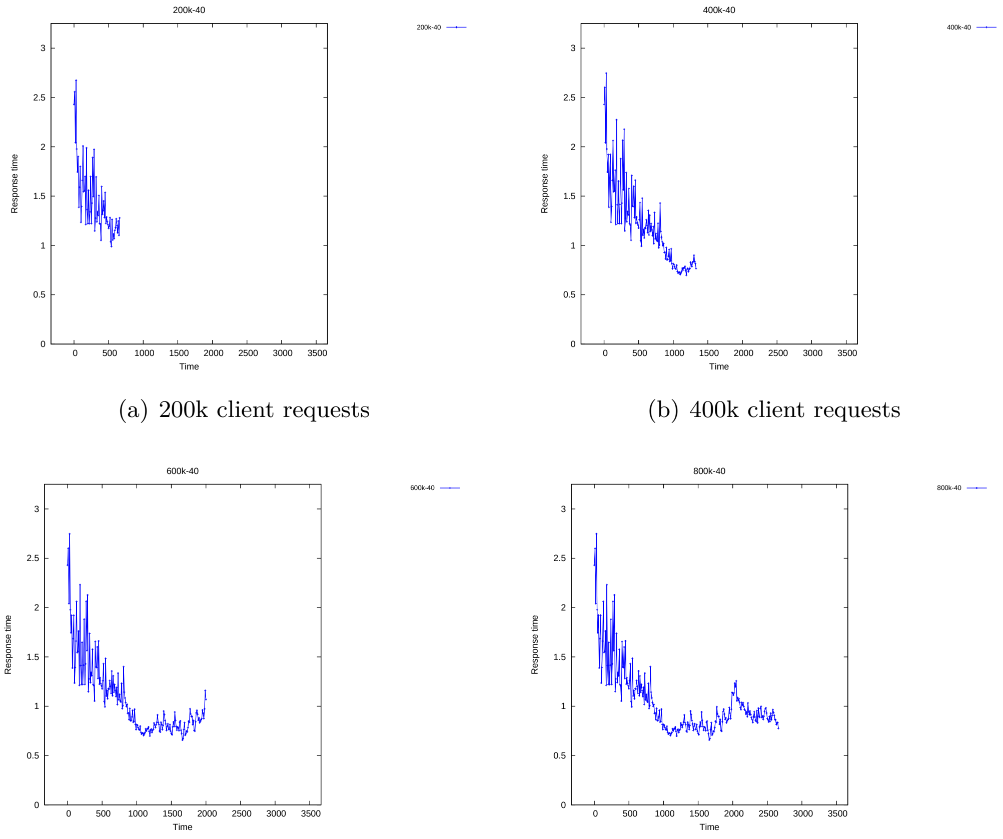
(a) 10 surrogate servers

(b) 20 surrogate servers

(c) 30 surrogate servers

(d) 40 surrogate servers

Figure 4.46: Response time for client requests over simulation time for serving 1000k requests for different number of surrogate servers



(a) 200k client requests

(b) 400k client requests

(c) 600k client requests

(d) 800k client requests

Figure 4.47: Response time for client requests over simulation time for 40 surrogate server for serving different number of client requests

4.4.3.7 Response Time.

We also calculated the response time over simulation for different number of surrogate servers serving different number of client requests. Here also we took the average of each 100 units of simulation time and the average of corresponding response time of client requests. We calculated it for all proposed policies with all the combination of surrogate servers and different number of client requests. Here we are exhibiting the case of 1000k client requests served by 10, 20, 30 and 40 surrogate servers as shown in Figure 4.46 and with 40 surrogate servers for serving 200k, 400k, 600k and 800k client requests as presented in Figure 4.47. When surrogate servers start to serve the requests, there caches are not smart and they start to be smarter with the time that causes the reason of decrease in response time and then it stays almost stable. There is low impact of infrastructure on the behavior of response time for same number of client requests. But different number of client requests causes decrease in the response time, if there is no change in the CDN infrastructure (because of the smartness of the caches as discussed earlier). Simulation time naturally increases with increase in the number of client requests.

4.5 Conclusion

The purpose of this chapter is to present the proposed CDN redirection policies. These policies will be compared in chapter 5.

CDN infrastructure behaves differently for the different client request redirection policies. It is important to identify the energy saving opportunities, proposing and developing the energy conservation techniques and policies in the CDN. Applying energy conservation techniques like DVFS and powering-off servers to the traditional CDN policies can be interesting to investigate. Keeping in view the behavior of CDN redirection policies, energy saving techniques are applied. DVFS technique is applied to both load-balance and load-unbalance policy. While having under-utilization due to the load-unbalancing, consolidation is applied to the load-unbalance. More traffic cause higher energy consumption in CDN infrastructure. But at the same time, help to mature the CDN infrastructure by making its caches smart which can have an impact on better user experience. Larger CDN infrastructure also increases the energy consumption by increasing the static as well as dynamic energy consumption. Larger infrastructure for lighter loads leads to maximize static energy consumption. It is found that powering-off surrogate servers offer considerable energy savings in the CDN infrastructure, including maximizing the utilization of the resources i.e. surrogate servers but lowering-down the user experience by increased response time and very low ratio of incomplete requests. At the other hand, smaller infrastructure for higher number of requests creates the complication in CDN services i.e. by causing failed requests, higher response time etc. It depends on the service provider and client's requirements which can lead them to select the policy.

In the next chapter, we will compare the different energy-aware policies detailed in this chapter and their impact on the different CDN evaluation metrics discussed earlier, regarding the infrastructure and the traffic of client requests.

Chapter 5

Comparison of Energy-Aware CDN Redirection Policies

Contents

Different CDN (Content Distribution Network) redirection policies cause different behavior of CDN services. It depends upon the needs of the services and the requirements of the client and service providers, to opt a CDN redirection policy. If different alternatives are proposed to provide the services, it is important to know a detailed comparison of the services provided and behavior of the adopted methods according to the different parameters. CDN redirection policies have specific constant and variable parameters. Depending on these, some policies show aggressive behavior for energy savings i.e. Load-Unbalance. However, some policies show moderate behavior i.e. Load-Balance. In order to make the choice of a policy to adopt, it is important to evaluate the policies on the same platform. In order to evaluate the policies, there are different evaluation parameters, on the basis of which the behavior of the policy is considered. Evaluation parameters considered for a request redirection policy, are discussed in Chapter 3.

Different clients may have different requirements. In some cases, clients are either quality of service oriented or cost oriented. Service providers try to deliver better services while maintaining the lower costs depending on its own priorities. Other service providers charge higher costs for better quality of services, in order to increase the profit margin. Energy consumption in a CDN has a considerable part of the overall system and services cost. This cost can be minimized or maximized depending on the approach of the service providers. Installing the energy hungry devices to provide the services in good quality causes increase in the cost of the services. Adopting green technology or optimizing the process of service providing to decrease the energy consumption results into decrease in the cost. Different policies can be adopted to provide the services. In order to make a better choice of the request redirection policy, it is important to do a detailed analysis of the policies, to see the difference in their behavior. It gives the direction to chose the right policy for the right scenario. In Chapter 4, we have proposed and discussed in detail the different request redirection policies. Here we are going to compare the policies, in order to find the different aspects of the policies to each other. We analyze the impact of load-balance and load-unbalance behavior on the different evaluation parameters:

- What are the effects on the evaluation parameters while changing the processor frequency of the surrogate servers dynamically (DVFS (Dynamic Voltage Frequency scaling))?
- How the policies behave when powering-off surrogate servers is applied?
- How the different policies react when aggressive approach of combining both energy saving techniques (DVFS + Powering-off) are taken into account?
- What is the impact of different DVFS techniques?

We have done the experiments for different number of surrogate servers i.e. 10, 20, 30, 40 and 50 to serve different number of client requests i.e. $200k$, $400k$, $600k$, $800k$ and $1000k$. Keeping in view the limitations of the time and the space, we have to chose a specific number of surrogate servers to show the behavior of the different CDN redirection policies for different number of client requests, without affecting the generality. In our case, 10 surrogate servers present the smallest CDN infrastructure. We consider turning-off surrogate servers in some of the proposed redirection policies, so the smaller number of surrogate servers are not enough to analyze the behavior of switching-off surrogate servers. Increasing the number of surrogate servers show a considerable impact of shutting-down surrogate servers but at the same time, utilization of the surrogate servers decreases with increase in the number of surrogate servers e.g. in case of 50 surrogate servers the utilization is lower. So, we selected the case of 40 surrogate servers to serve the different number of client requests, in order to show a global picture without affecting the generality.

The smaller number of requests show the behavior of the system with lower loads. As the number of requests are increased, the CDN infrastructure is loaded for more time and shows the increased utilization of the resources and have the impact on the different evaluation parameters. In order to have a global analysis, we took the $1000k$ (that is the maximum of number of client requests traffic) client requests, for different number of surrogate servers, to show the impact on the different evaluation parameters.

It is important to note that load-balance, load-unbalance and load-unbalance power-off policies refer to the policies which apply the global DVFS technique FreqMax. Here we use them for the comparison purposes to evaluate the different energy-aware policies. In this chapter, we first present curves and explain global behaviors, while actual systematic numbers will be presented at the end of this chapter.

5.1 Surrogate Server Utilization

Figures 5.1 and 5.2 exhibit the surrogate servers average utilization for different CDN redirection policies. Here, first we consider the case of 40 surrogate servers with different number of client requests and then the case of $1000k$ requests for different number of surrogate servers is presented. For all the policies, utilization of the surrogate servers increases with the increase in the number of requests and

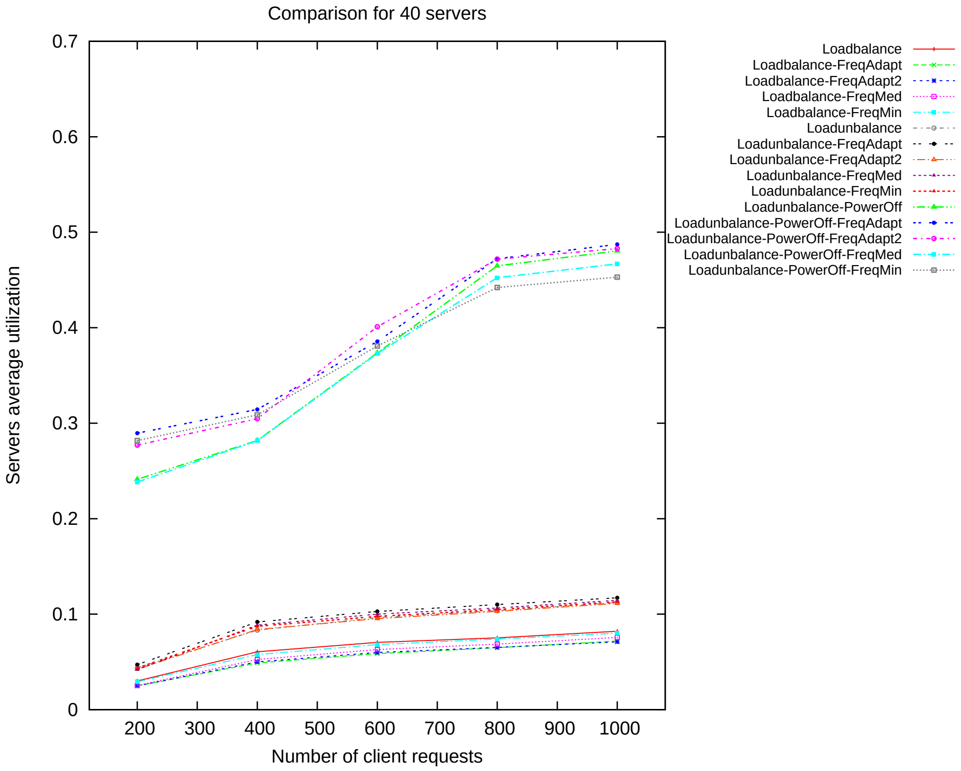


Figure 5.1: Surrogate servers average utilization vs. number of requests (10^3) for different policies.

it decreases with the increase in the number of surrogate servers. Load-Unbalance policies show better utilization than the load-balance policies. In case of load-balance, the load of the surrogate servers is divided uniformly among all the available surrogate servers. Surrogate servers share the load of the traffic and there is no congestion on the surrogate servers that causes the rapid serving of the client requests. So the duration of the connection is smaller: This minimizes the utilization of the surrogate servers. While, in case of load-unbalance policies, there is the concentration of the client requests traffic on some surrogate servers which slow down the request processing. This increases the connection duration and utilization of the surrogate servers is increased. Policies considering powering-off surrogate servers (only Load-Unbalance policies consider the powering-off servers) show better utilization as compared to the policies which don't take into account turning-off

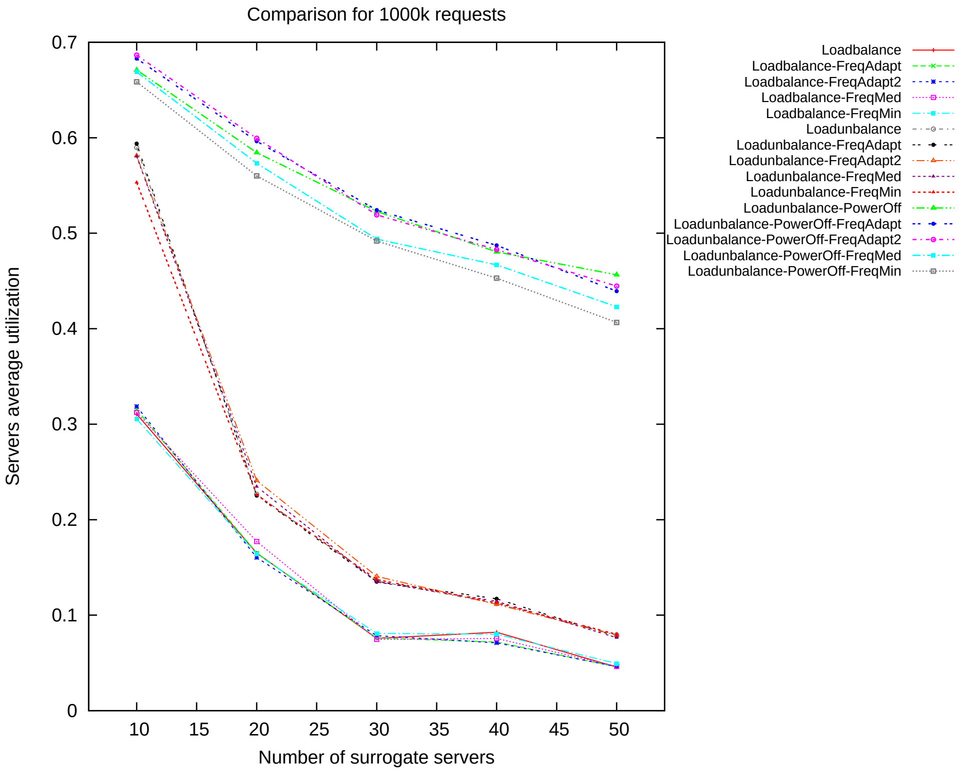


Figure 5.2: Surrogate servers average utilization vs. number of surrogate servers for different policies.

surrogates servers either considering the DVFS or not (from less than 10% to more than 30% in the case of 40 servers). Policies with powering-off surrogate servers, shut-down some underutilized surrogate servers depending on the traffic of client requests. So the number of available servers is minimized and the utilization of the available surrogate servers is maximized. Figures 5.1 and 5.2 show that the impact of DVFS on utilization of surrogate servers is not as noticeable as compared to the impact of turning-off surrogate servers. In case of DVFS, the processor frequency is changed according to the load of the surrogate servers which doesn't affect much the overall utilization of the surrogate servers.

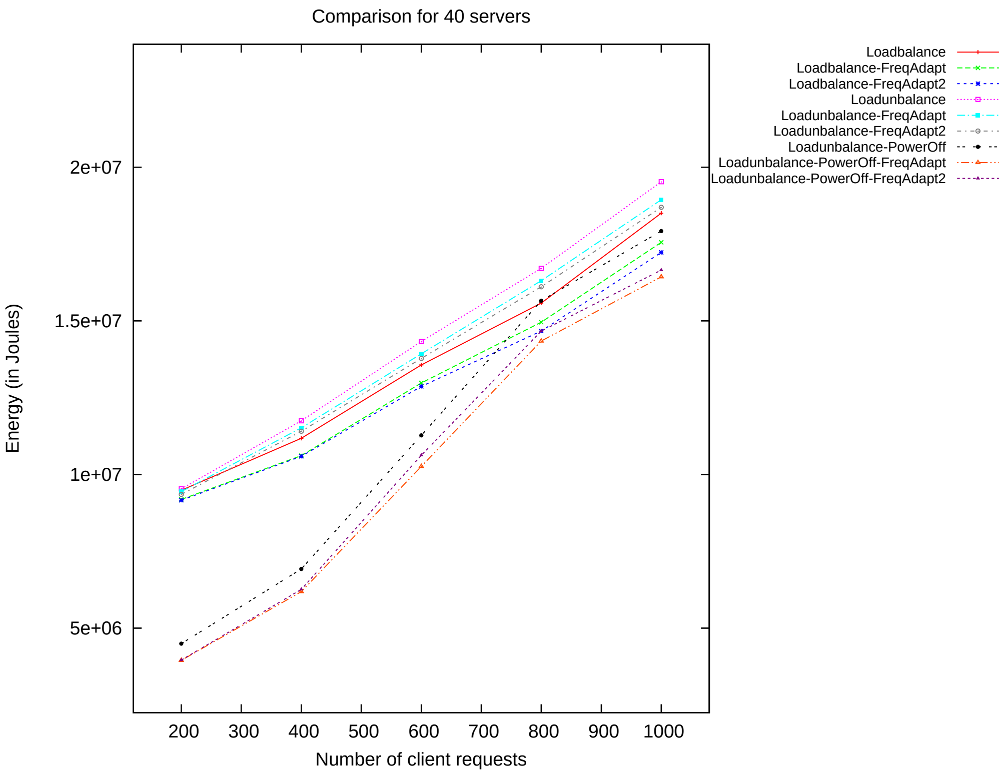


Figure 5.3: Energy consumption vs. number of requests (10^3) for different policies.

5.2 Energy Consumption

In order to simplify the presentation of energy consumption, we show results in different graphs. Putting all policies on the same graph, creates complications to understand. Figures 5.3, 5.4, 5.5 and 5.6 show energy consumption for the CDN infrastructure of 40 surrogate servers serving different number of client requests and the case of $1000k$ client requests for different number of surrogate servers respectively, comparing different CDN redirection policies. Figures 5.3 and 5.4 show the comparison for different policies describing energy consumption except the policies with the application of FreqMin and FreqMed. Where as, Figures 5.5 and 5.6 present policies with the application of FreqMin and FreqMed with the comparison of other policies. Unsurprisingly, it is noticed that policies without power-off cause more energy consumption than the policies which consider powering-off surrogate

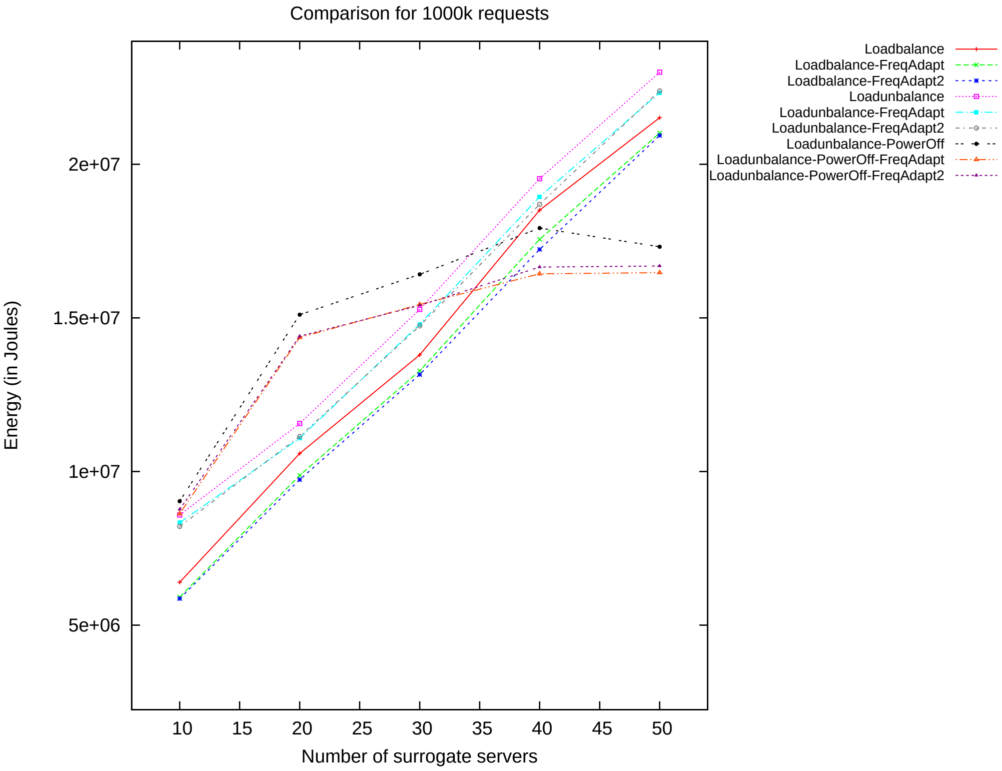


Figure 5.4: Energy consumption vs. number of surrogate servers for different policies.

servers, when the load is lower (upto 50% less for 200k and 40 servers, see Figure 5.3). Also, load-unbalance without powering-off surrogate servers causes more energy consumption in CDN infrastructure of surrogate servers than load-balance (see Figure 5.4, for more than 40 servers). As we discussed previously, average utilization of the surrogate servers is higher in case of load-unbalance, that results in more energy consumption.

Energy consumption increases with the increase in the number of client requests for all CDN redirection policies. Increase in the CDN infrastructure also results in increase in energy consumption, for all policies except policies which consider powering-off surrogate servers. DVFS also has a considerable impact on the energy savings (up to 30% see Figure 5.5, 1000k requests for instance) but it has a little impact on energy savings as compared to turning-off surrogate servers especially

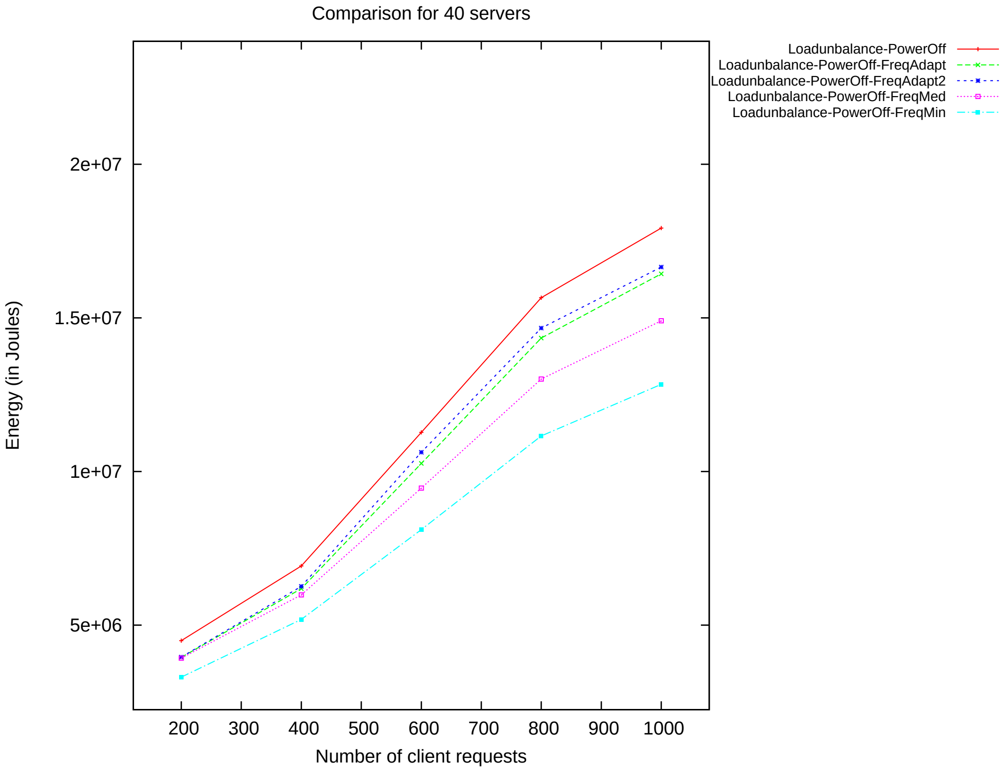


Figure 5.5: Energy consumption vs. number of requests (10^3) for different policies.

when the load is low. DVFS impact is more when surrogate server have higher loads i.e. in case of powering-off policies. Powering-off surrogate servers minimizes the active as well as the idle energy consumption in surrogate servers while DVFS only takes into account the active energy savings. However, powering-off surrogate servers for smaller infrastructure results in more energy consumption. When the infrastructure is small and the client requests traffic is large, there is a need to keep enough surrogate servers available to serve the requests. So there is less opportunity to turn-off surrogate servers. When there is low load, some servers are considered to be turned-off. When the traffic increases, the powered-off surrogate servers are needed to be turned-on. Their caches are not intelligent enough to store the popular contents, that increases the cooperation among surrogate servers which causes the increase in energy consumption. With larger number of surrogate servers, there is opportunity to turn-off more surrogate servers and the impact of

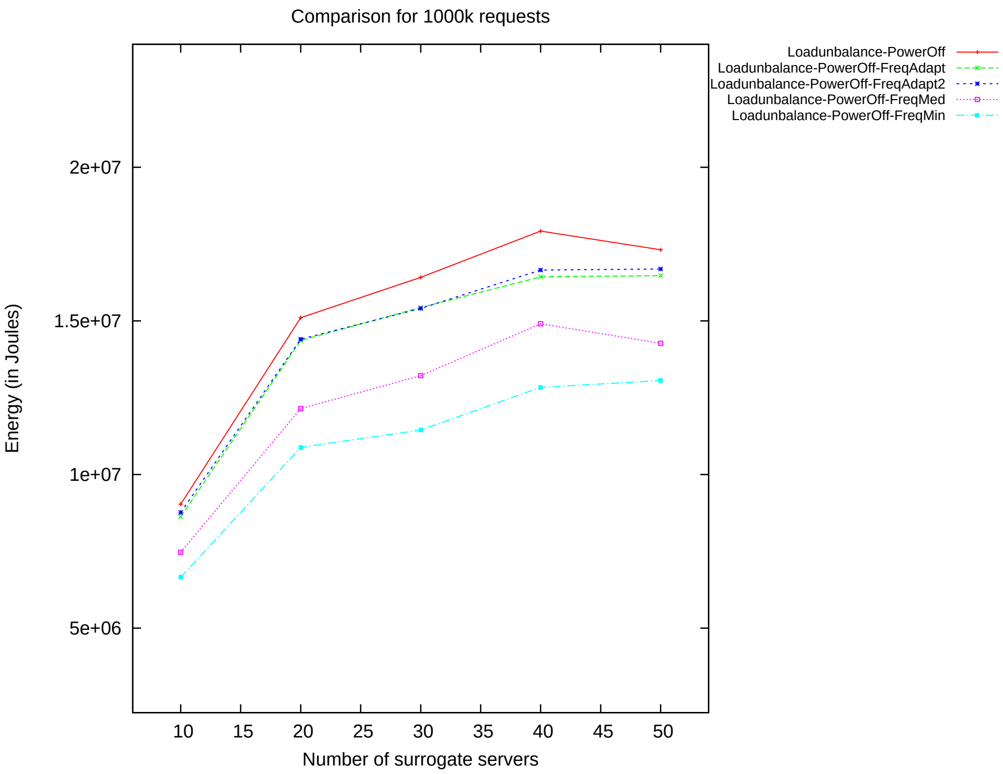


Figure 5.6: Energy consumption vs. number of surrogate servers for different policies.

gaining energy savings by switching-off surrogate servers is larger than the impact of energy increased resulted by the cooperation among surrogate servers.

Aggressive behavior of gaining energy savings by DVFS and powering-off servers (Figure 5.6), show the maximum energy savings because of having active and passive energy savings at the same time . The impact of energy savings between the local DVFS policies themselves (i.e. FreqAdapt and FreqAdapt2 policies) is very small or in some cases even negligible. When the surrogate servers are processing the requests, on lower frequencies, it spends less energy, hence the policies with FreqMin and FreqMed are showing better results.

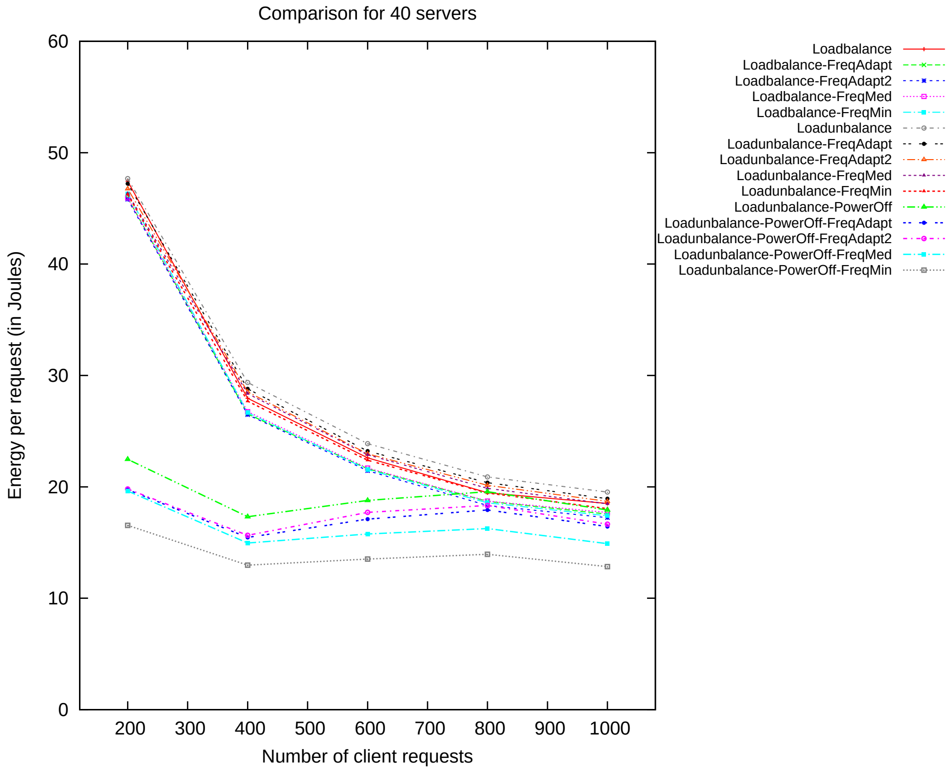


Figure 5.7: Energy per request vs. number of requests (10^3) for different policies.

5.3 Energy per Request

Figures 5.7 and 5.8 analyze the comparison of different CDN redirection policies on the basis of energy consumed per request. On Figure 5.7, there is decrease in the energy consumed per request for the policies with the increase in the traffic of client requests but in case of power-off, it remains almost constant. Initially, caches are dumb in case of lower traffic which causes cooperation which increases the energy per requests but as requests are increased energy consumption is minimized. But in case of power-off, it remains stable because of turning-off/turning-on again, it starts as a new one with dumb caches.

In case of surrogate servers (Figure 5.8), the energy per request (EpR) increases gradually for policies without powering-off but it becomes almost stable from 20 surrogate servers onward for powering-off policies. In case of 10 surrogate servers

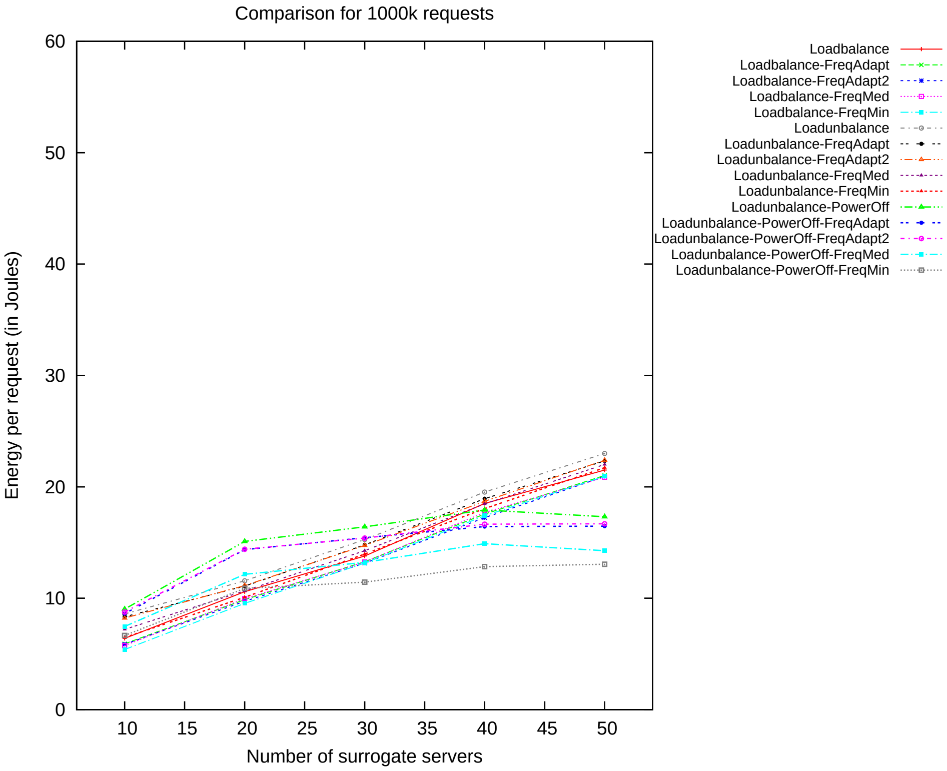


Figure 5.8: Energy per request vs. number of surrogate servers for different policies.

where not much powering-off happens, the powering-off policy behaves like the other policies. But as the number of surrogate servers increases, it remains almost stable. This stability is because of the turning-off mechanism that takes place when there are higher number of surrogate servers for serving the requests. Policies which consider the power-off and DVFS at the same time perform the best in case of energy per request because of minimizing the active and the idle energy minimization at the same time. Among the other policies, load-balance policies with DVFS perform better because of lower utilization and processing the requests on lower rates of frequency. DVFS decreases the energy consumption in the policies without power-off. Load-unbalance policy without power-off shows the worst case and consumes the highest energy per request because of higher utilization (more than twice the best case).

5.4 Mean Response Time

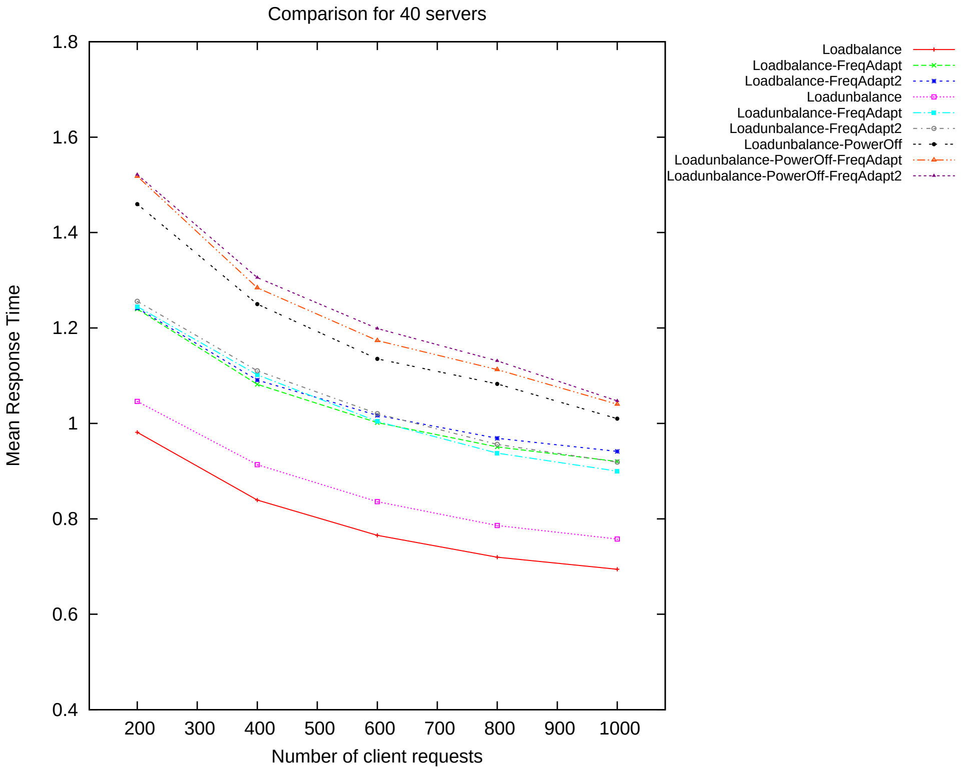


Figure 5.9: Mean response time vs. number of requests (10^3) for different policies.

Figures presenting mean response time are presented into two different divisions to simplify the understandings. Where, Figures 5.9 and 5.10 show the comparison for different policies describing energy consumption except the policies with the application of FreqMin and FreqMed. Whereas, Figures 5.11 and 5.12 present policies with the application of FreqMin and FreqMed. Load-Balance causes lower response time than load-unbalance. In case of load-balance, there is not much congestion on the surrogate servers and the network nodes as well, as the requests are distributed uniformly that causes the rapid completion of the client requests. In case of load-unbalance, the concentration of the requests towards a set of surrogate servers causes the delay in request completion.

Similarly, DVFS also has an impact on response time. Slow processing of the

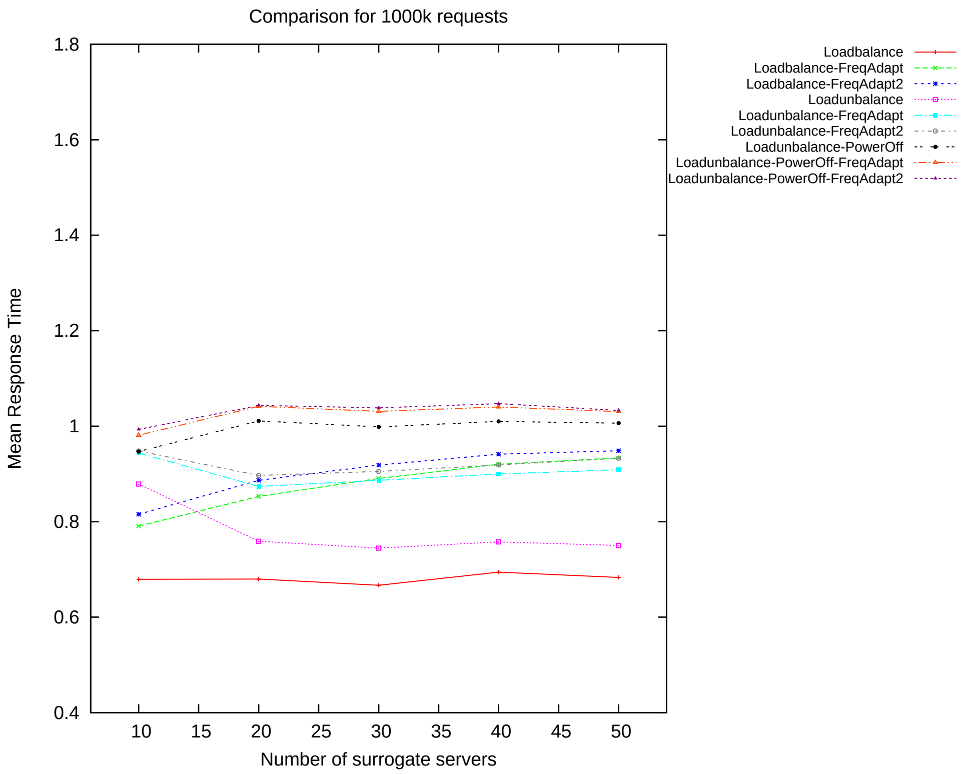


Figure 5.10: Mean response time vs. number of surrogate servers for different policies.

requests results into higher response time. But it is noticed that this difference in response time is not as high as we see in case of policies which consider turning-off surrogate servers. When some surrogate servers are turned-off, client requests are forwarded towards the other available surrogate servers which causes delay and the congestion on the other surrogate servers ,increasing the response time. Policies where surrogate servers are switched-off cause cooperation among the surrogate servers to satisfy the client requests which also increases the response time. In all policies, the increase in the number of requests causes decrease in mean response time which is because of the smartness of the caches with traffic increase and increase in direct satisfaction of requests.

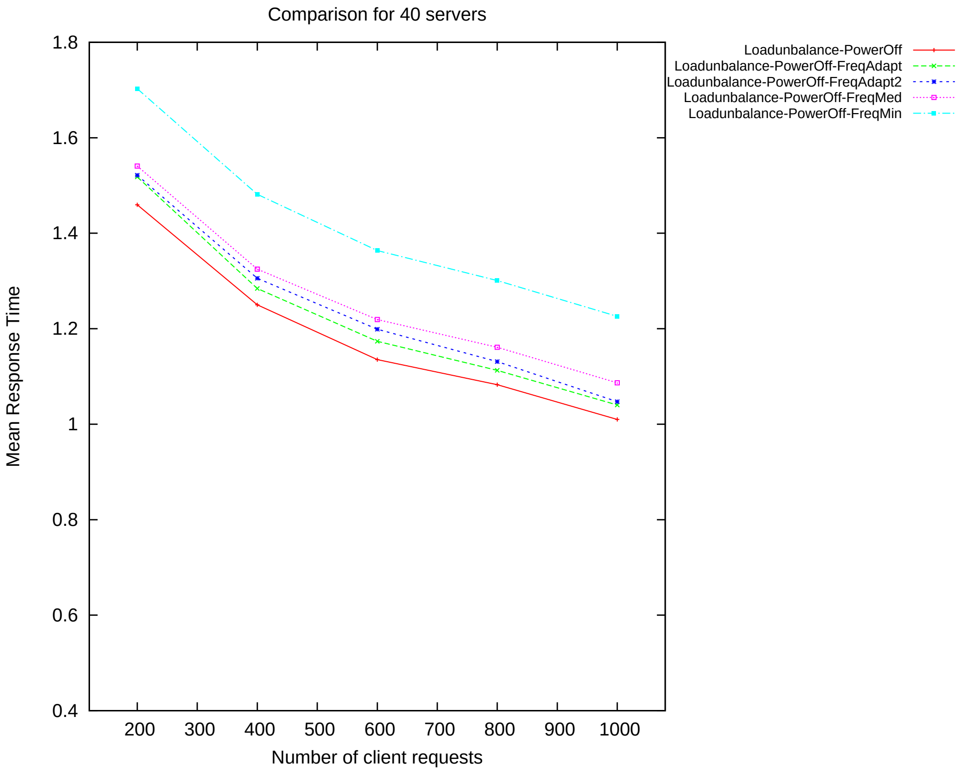


Figure 5.11: Mean response time vs. number of requests (10^3) for different policies.

5.5 Hit Ratio

Figures 5.13 and 5.14 show the hit ratio of the different CDN redirection policies for 40 surrogate servers serving different number of client requests and for different number of surrogate servers serving 1000 requests, respectively. Load-Unbalance policies present the better hit ratio as compared to Load-Balance policies. In case of load-unbalance, the concentration of client requests towards a subgroup of surrogate servers makes their caches smarter which cause increased hit ratio. In case of power-off surrogate servers, hit ratio is lowest. Processing speed of the surrogate servers doesn't much affect the hit ratio. The impact of hit ratio between *FreqAdapt* and *FreqAdapt*2 is very low, particularly in policies which don't consider powering-off surrogate servers. Increase in the number of client requests increases the probability of the hit because of the increase in the smartness of the surrogate servers caches. In

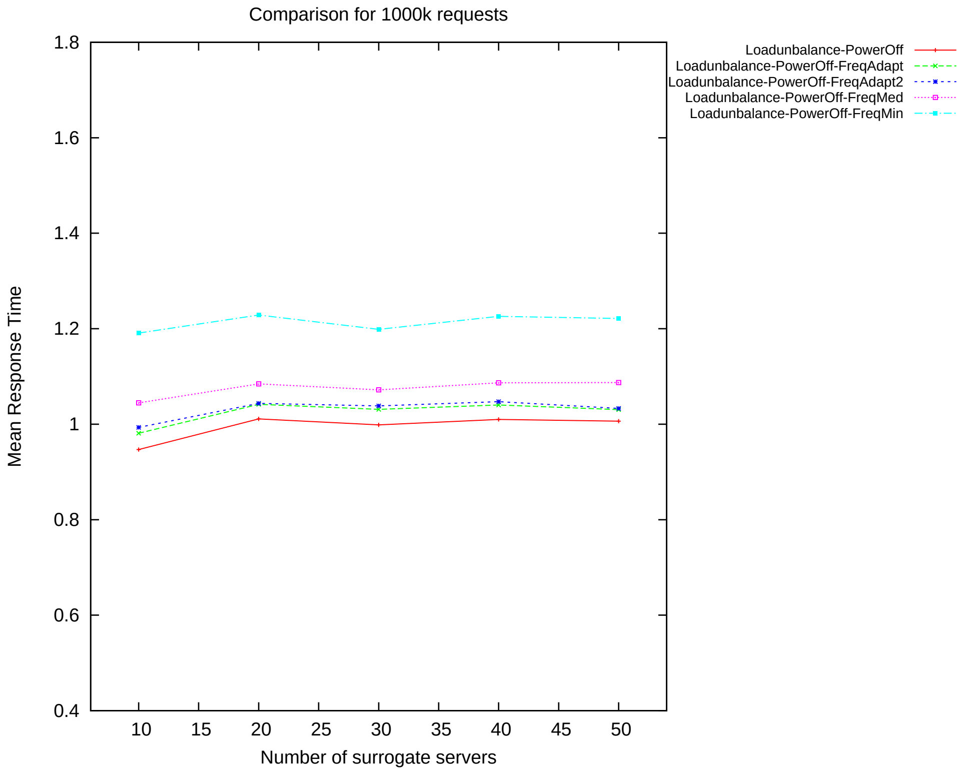


Figure 5.12: Mean response time vs. CDN redirection policies.

case of powering-off, the hit ratio is higher when there are less number of requests. When there is less traffic, more surrogate servers are powered-off and there are less surrogate servers to serve the requests. Powered-on surrogate servers are enough to serve the requests and there is not much turning-on the powered-off servers which causes the decrease in the hit ratio. So, the same number of requests are served by less surrogate servers as compared to the policies which don't consider the power-off. This smaller number of surrogate servers cause the intelligent serving of the requests which increases the hit ratio. As the number of client requests increases, there is need to turn-on some powered-off surrogate servers. When a surrogate server is turned-on after powering-off, its cache is dumb and it takes some time to be smarter. With the increase in the number of client requests it starts to become smarter. Because of that we see a sudden decrease in the hit ratio when the client requests are increased from $200k$ to $400k$ and then a an increase occurs when the

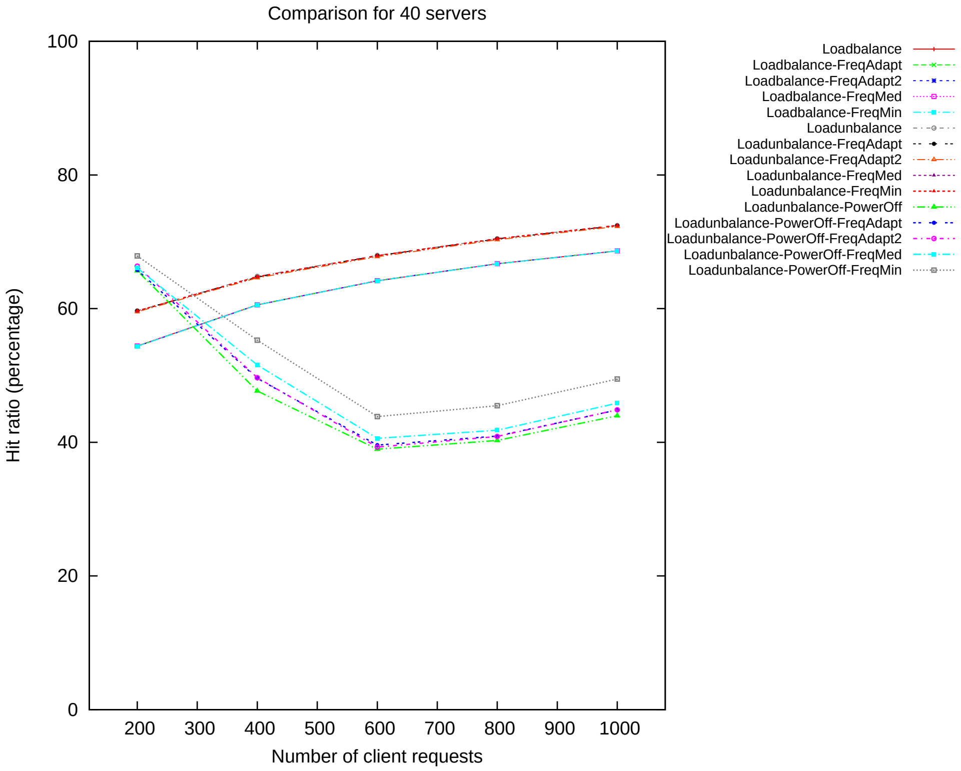


Figure 5.13: Hit ratio vs. number of requests (10^3) for different policies.

client requests increase from $600k$, as shown in Figure 5.13. On Figure 5.14, the increase in the surrogate servers number decreases the hit ratio. If there are more surrogate servers for the same number of requests, it takes more time for their caches to become smarter, as a result, hit ratio is decreased.

In case of powering-off surrogate servers, there is not much impact of the number of surrogate servers on the hit ratio. The reason is when the number of surrogate servers is increased, powering-off more surrogate servers occurs, if the traffic remains the same then there are almost the same number of surrogate servers turned-on to serve the requests.

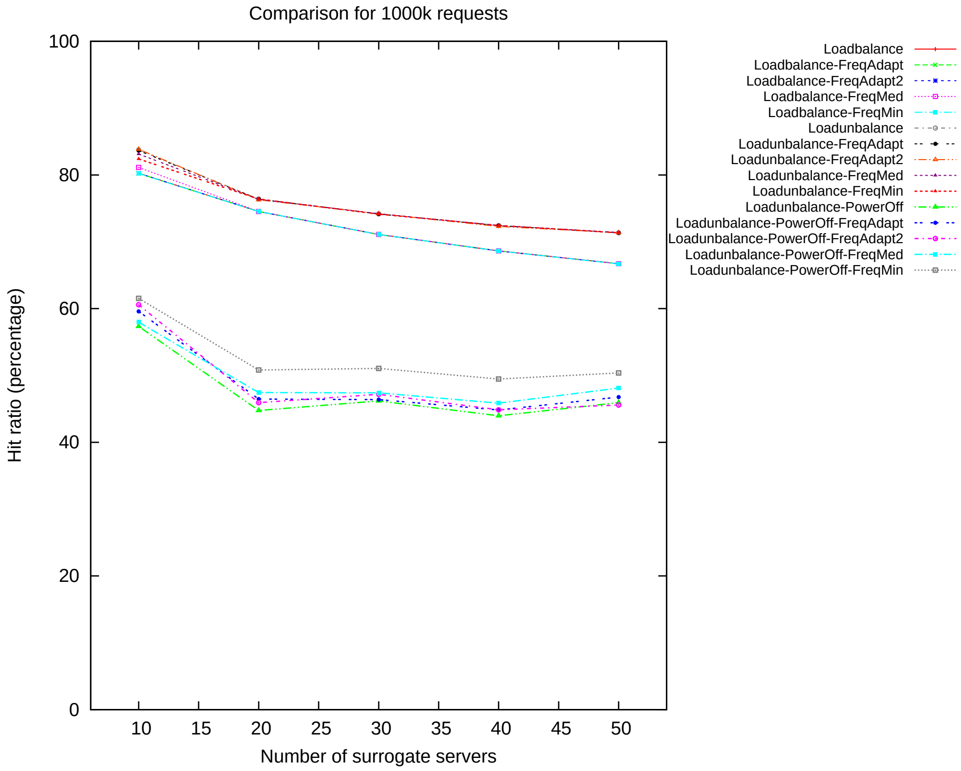


Figure 5.14: Hit ratio vs. number of surrogate servers for different policies.

5.6 Failed Requests

When powering-off surrogate servers occurs, it causes some failed requests due to the unavailability of the services as shown in Figure 5.15. While the rest of the policies don't have any failed request.

5.7 Conclusion

CDN redirection policies have an impact on the different evaluation parameters. It is important to compare different policies for different evaluation parameters, to have a global idea of relation among the policies and to make a choice of the policy according to the requirements of clients and CDN service provider. Policies without energy conservation techniques i.e. load-balance and load-unbalance with FreqMax show better user experience with higher availability of services but with

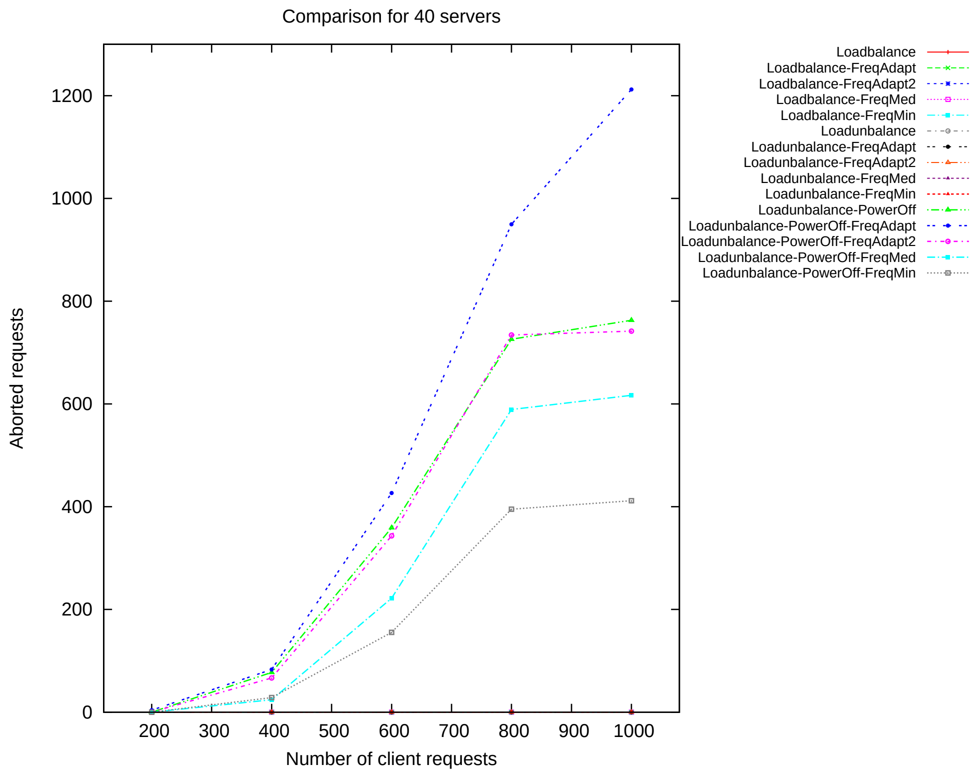


Figure 5.15: Failed requests vs. number of requests (10^3) for different policies.

the cost of higher energy consumption. Load-Balance behavior presents less energy consumption as compared to the load-unbalance behavior if the powering-off technique is not applied. It also exhibits better user experience by providing lower response time for client requests. Load-Unbalance shows higher energy savings in case of powering-off surrogate servers. Policies which apply only DVFS technique show moderate behavior of providing considerable energy conservation with higher availability of services and a low degradation of user experience. DVFS policies provide more energy gain during higher loads. Policies with servers consolidation provide higher energy conservation while having more impact on the user experience, in case of lower number of requests and having large CDN infrastructure. Finally, the policies with aggressive approach for energy conservation which merge the DVFS and powering-off servers show larger energy savings with higher impact on the user experience. Impact on different evaluation parameters between local and global DVFS policies is more considerable while combining them with server consolidation where global DVFS policies provide far better energy savings than local DVFS policies but at the cost of higher response time. Impact on the different evaluation parameters between the *FreqAdapt* and *FreqAdapt*2 techniques is very small. To summarize, Tables 5.1 and 5.2 present a detailed impact of energy conservation techniques i.e. DVFS and power-off, on the different evaluation parameters for load-balance and load-unbalance policies. In the next chapter, we will discuss the impact of intensity of load on the CDN infrastructure.

Table 5.1: Impact DVFS and Power-Off: Load-Balance and Load-Unbalance (case of 40 surrogate servers serving 1000k requests), compared with the case of Load-Balance(FreqMax) E/R = Energy per Request; MRT = Mean Response Time; HR = Hit Ratio; FR = Failed Requests; LB = Load-Balance; LUB = Load-Unbalance

Policy	Utilization	Energy	E/R	MRT	HR	FR
LB FreqAdapt	−13%	−5%	−5.13%	+24.56%	0%	0%
LB FreqAdapt2	−13.8%	−7%	−6.91%	+26.25%	0%	0%
LB FreqMed	−7.9%	−4%	−4.32%	+13.25%	0%	0%
LB FreqMin	−3.14%	−6%	−5.94%	+28.07%	0%	0%
LUB FreqMax	+26.18%	+5.25%	+5.25%	+8.36%	+5.23%	0%
LUB FreqAdapt	+29,45%	+2,31%	+2,31%	+22,86%	+5,22%	0%
LUB FreqAdapt2	+25,87%	+1,02%	+1,02%	+24,45%	+5,07%	0%
LUB FreqMed	+28,00%	−0,16%	−0,16%	+19,56%	+5,23%	0%
LUB FreqMin	+26,73%	−2,49%	−2,49%	+31,87%	+5,27%	0%
LUB Poff FreqMax	+82,80%	−3.138%	−3.135%	+31.27%	−35.94%	0.0762%
LUB Poff FreqAdapt	+83,04%	−11,19%	−11,19%	+33,25%	−34,66%	0.1212%
LUB Poff FreqAdapt2	+82,88%	−9,99%	−9,99%	+33,71%	−34,63%	0.0741%
LUB Poff FreqMed	+82,30%	−19,45%	−19,45%	+36,11%	−33,18%	0.0616%
LUB Poff FreqMin	+81,75%	−30,61%	−30,61%	+43,36%	−27,93%	0.0411%

Table 5.2: Impact DVFS and Power-Off : Load-Unbalance (case of 40 surrogate servers serving 1000k requests), compared with the case of Load-Unbalance (Freq-Max)

Policy	Utilization	Energy	E/R	MRT	HR	FR
LUB FreqAdapt	+4.44%	−3%	−3.02%	+15.79%	−0.01%	0%
LUB FreqAdapt2	−0.4%	−4%	−4.30%	+16.63%	−0.17%	0%
LUB FreqMed	+2.44%	−5%	−5.43%	+12.28%	0%	0%
LUB FreqMin	+0.70%	−7.6%	−7.63%	+25.71%	+0.04%	0%
LUB Poff FreqMax	+76.70%	−8.22%	−8.24%	+25.04%	−39.67%	0.0762%
LUB Poff FreqAdapt	+77.03%	−15.85%	−15.87%	+27.21%	−38.08%	0.1212%
LUB Poff FreqAdapt2	+76.81%	−14.72%	−14.75%	+27.69%	−38.05%	0.0741%
LUB Poff FreqMed	+76%	−23.68%	−23.71%	+30.29%	−36.69%	0.0616%
LUB Poff FreqMin	+75.29%	−34.26%	−34.25%	+38.20%	−31.70%	0.0411%

Chapter 6

Comparison of Energy Aware CDN Redirection Policies (Evaluating Impact of Client Request Frequency)

Contents

In the previous chapter, we have discussed a detailed comparison of different CDN (Content Distribution Network) redirection policies regarding number of requests and size of the CDN infrastructure. CDN can have different intensity of traffic during different periods of time. Can the behavior of a CDN change with the change in intensity of traffic? In order to know the answer to this question, we have evaluated the different CDN redirection policies (proposed earlier) with different frequencies of client requests. We have consider the case of 40 surrogate servers which gives us opportunity to analyze the powering-off impact and also the other important evaluation parameters. We took the traffic of 1000k requests which provide enough to load the CDN and to evaluate its behavior in more convincing way. In order to have different frequencies, we changed the mean inter-arrival time of requests. For that purpose, we consider the traffic with 6 different frequencies having 0.01, 0.005, 0.0033, 0.0025, 0.002 and 0.00125 as the mean inter-arrival of requests. In order to avoid presentation complications, we will present only FreqAdapt policy among local DVFS (Dynamic Voltage Frequency Scaling) policies (i.e. FreqAdapt and FreqAdapt2).

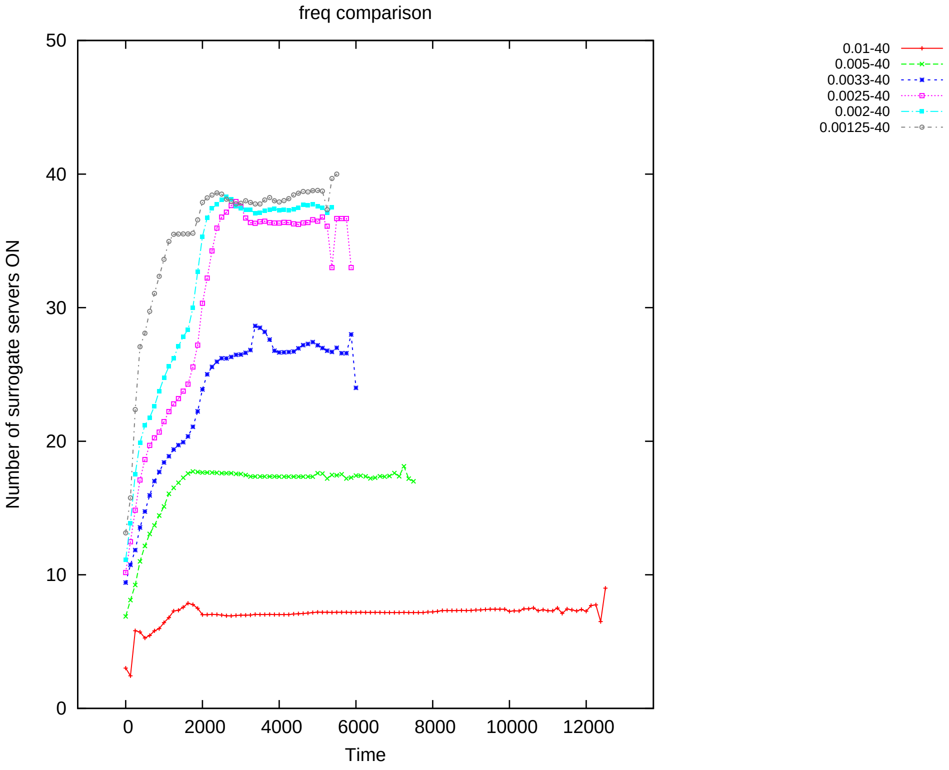


Figure 6.1: Surrogate servers ON vs. simulation time for different client request frequencies for load-unbalance power-off policy.

6.1 Number of Surrogate Servers Turned-On

Figure 6.1 shows the number of surrogate servers ON (Turned-On) over simulation time for different client request frequencies for 40 surrogate servers serving 1000k requests. The number of surrogate servers ON (i.e. surrogate servers available to serve the requests) is increased with increase in the client request frequencies. When the frequency of client requests is low, it causes lower loads in surrogates servers which allows the policy to power-off more surrogate servers. But as the frequency of client requests is augmented, less powering-off surrogate servers happens because of the higher loads and after some level the difference in number of surrogate servers ON is lower for different client request frequencies: The CDN infrastructure reaches its limit in processing requests. Figure 6.1 shows when mean inter-arrival time among requests is low, the time to process all requests is also increased.

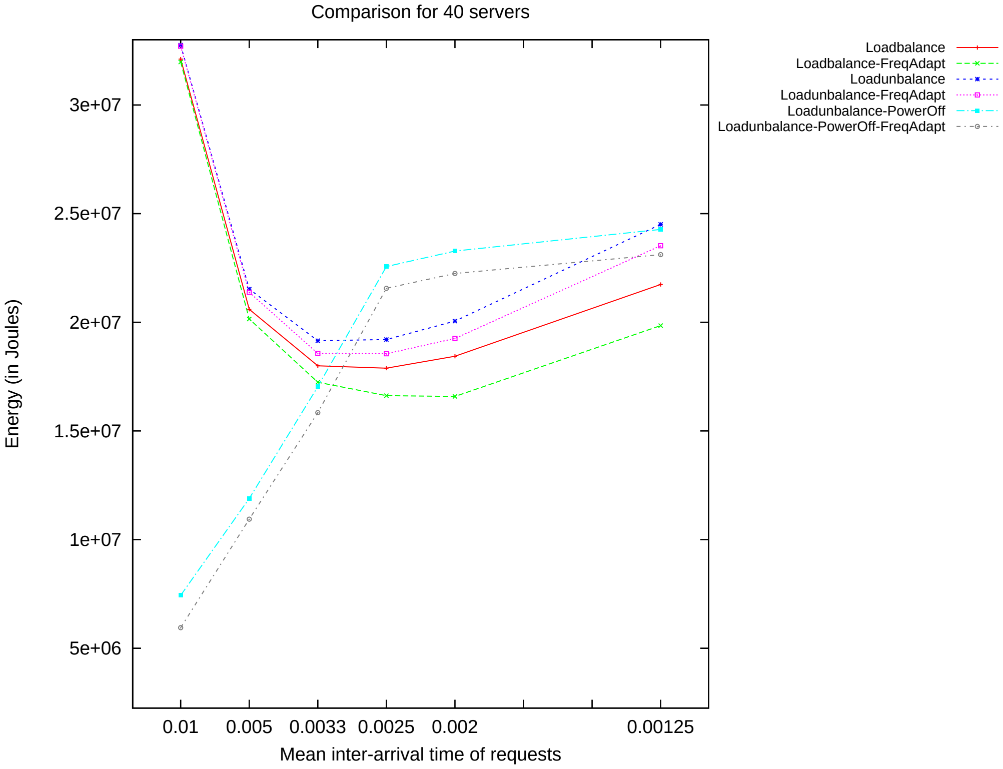


Figure 6.2: Energy consumption vs. frequency of client requests for different policies (except FreqMin and FreqMed).

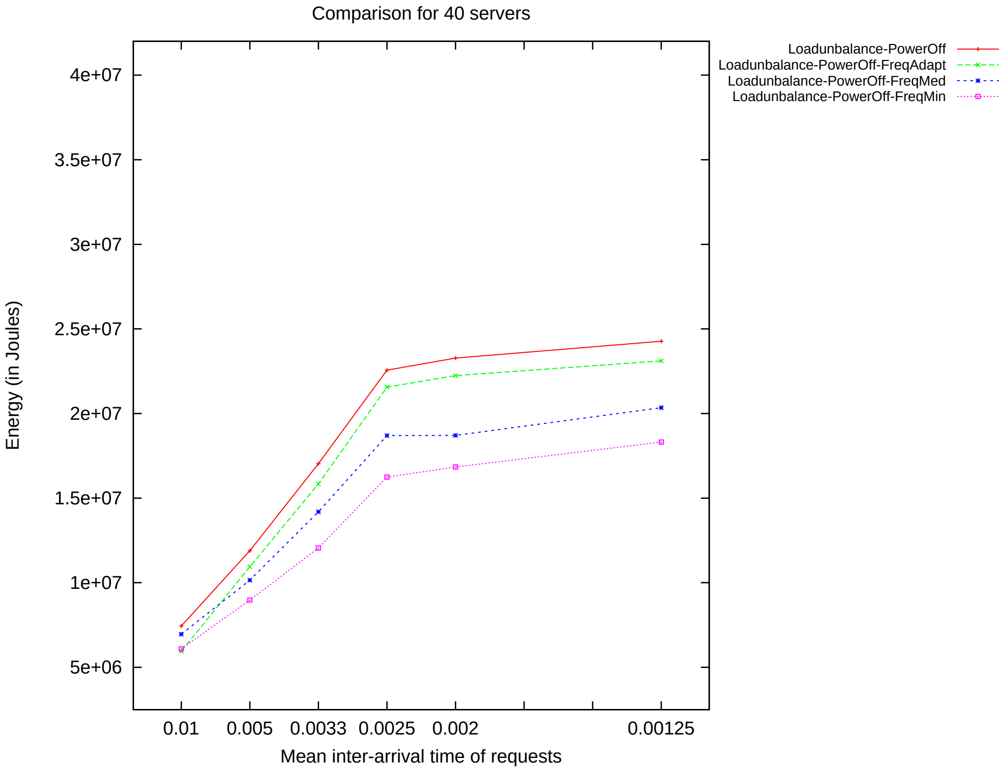


Figure 6.3: Energy consumption vs. frequency of client requests for different policies.

6.2 Energy Consumption

In order to simply the presentation, we have shown energy consumption of surrogate servers, in different figures, for comparing different policies. Figure 6.2 exhibit energy consumption for 40 surrogate servers serving 1000 client requests having different frequencies for different policies except FreqMin and FreqMed. While, Figure 6.3 presents comparison of energy consumption in the same context as previous for FreqMin and FreqMed with different policies, in the case of power-off.

Figure 6.2 shows that different policies behave differently as the frequency of the client requests is increased. We observe that load-balance and load-unbalance policies behave in the same manner while powering-off policies behave differently. We can see a peak when frequency of requests is low in case of load-balance and load-unbalance policies. This peak is due to the following reason: When client requests frequency is low, it takes more time for the system to serve the requests. It is important to remind that $E = P * T$, where E denotes energy, P represents power and T stands for time. Though the surrogate servers are less loaded, they remained powered-on for more time which causes an augmentation in the overall energy consumption. When client requests frequency is increased, overall simulation time is decreased which decreases energy consumption though dynamic energy is augmented due to higher loads. Simulation time for different client requests frequencies can be seen in Figure 6.1. Comparing load-balance and load-unbalance (without power-off), during low request frequencies, both policies have same behavior in energy consumption. But as the frequency of client requests increases, load-balance provide more energy savings than load-unbalance policies due to its uniform distribution.

In case of powering-off policies, during low loads, more surrogate servers are powered-off hence energy consumption is reduced a lot. The impact of powering-off surrogate servers on energy consumption is higher despite the impact of higher simulation time. This makes powering-off policies more energy efficient than the other policies, in low load conditions. Powering-off policies show more energy consumption at higher frequencies. This is due to the less powering-off, higher loads, and augmentation in the cooperation among the surrogate servers due to the dumbness of caches of restarted surrogate servers. Though powered-off policies show higher energy consumption in higher frequency conditions but maintains a constant value during this period as compared to the policies without powering-off mechanism.

With low request frequency, the DVFS impact on energy consumption is low. For the comparison of DVFS policies, FreqMin provides higher energy gains following FreqMed and FreqAdapt as compared to FreqMax policies.

6.3 Energy per Request

Figure 6.4 present energy consumed per request for different CDN redirection policies, for 40 surrogate servers serving 1000k traffic of different frequencies. Energy

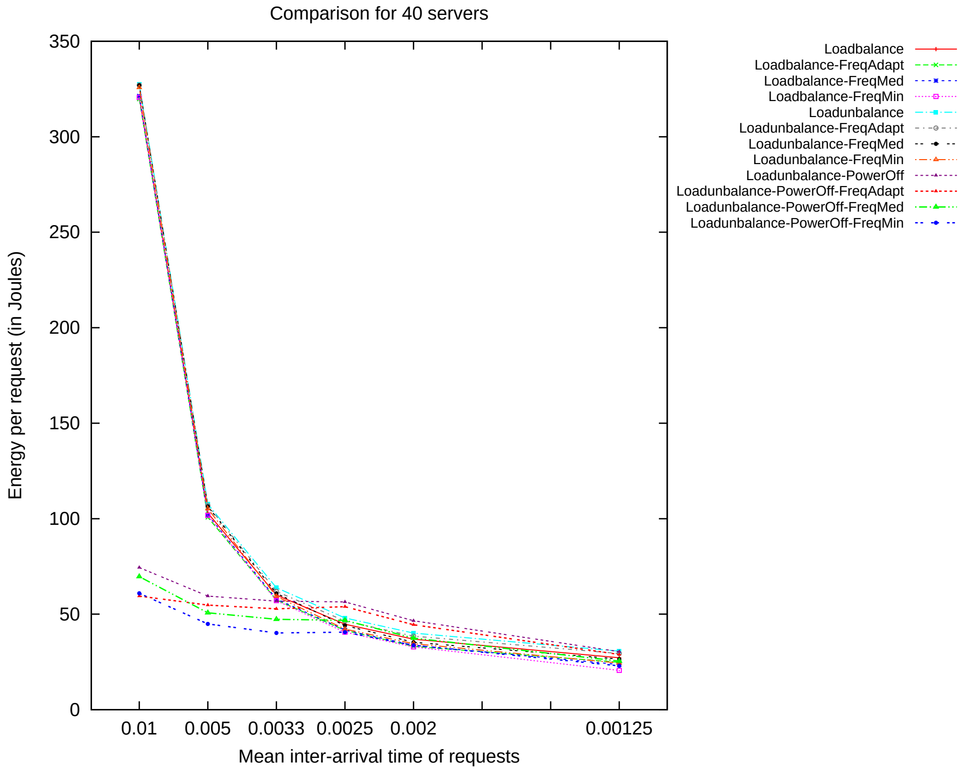


Figure 6.4: Energy per request vs. mean inter-arrival time of requests for different policies.

per request shows higher values for low loads due to higher execution time of requests when considering policies without server consolidation. With the increase in frequency of client requests, energy per request is decreased in a smoother way when the system is well loaded. Policies considering surrogate servers consolidation have more stable curve for energy per request due to the powering-off mechanism.

6.4 Mean Response Time

Figures 6.5 and 6.6 present the mean response time for 40 surrogate servers serving $1000k$ client requests with different frequencies. For low request frequencies, mean response is low. There are less requests which cause lower congestion at node and surrogate server level, that causes lower response time. Moreover, hit ratio

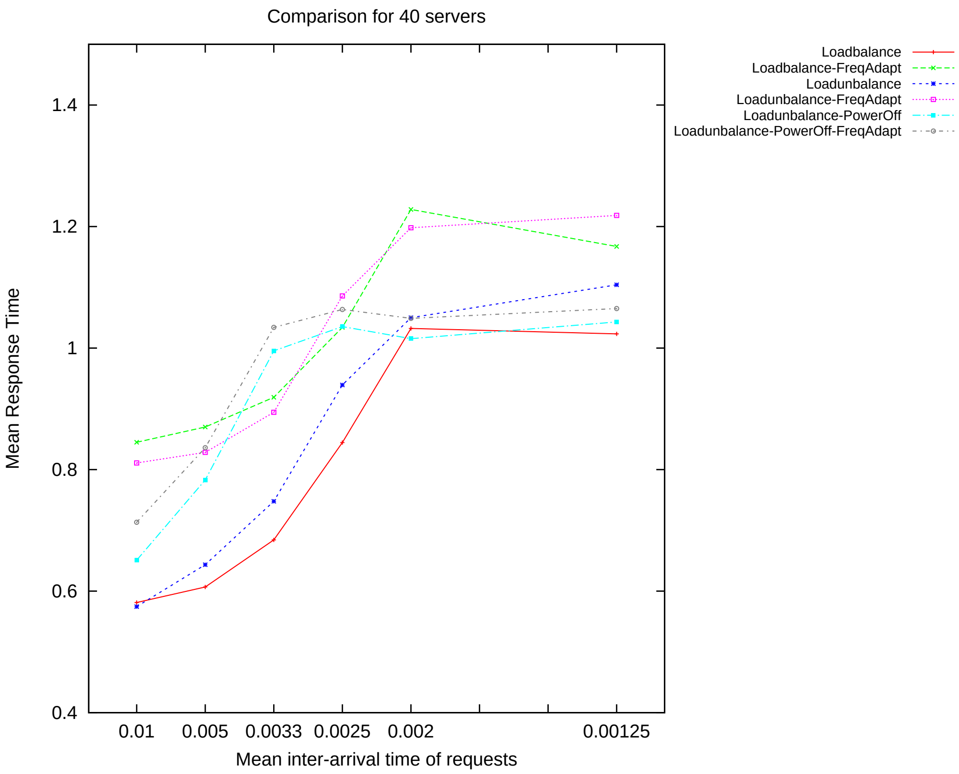


Figure 6.5: Mean response time vs. mean inter-arrival time of requests for different policies (except FreqMin and FreqMed).

is also better during lower request frequencies that can play the role to a lower response time. Mean response time increases with the increase in frequency of client requests and then it shows stable values depending on the policies pattern. In case of powering-off, it increases in beginning and then it gets stable earlier than the policy without power-off. This stability point shows when the CDN system gets an optimum point regarding response time of requests (in our case, 0.0033 is this optimum). Here the system attains the maximum response time when the system can be well loaded.

DVFS policies result in higher response time especially with higher loads. In case of FreqMax policies, they perform better in response time due to their rapid processing of requests. FreqMin, FreqMed and FReqAdapt have respectively higher response time, due to their processing speed of requests. The difference of mean re-

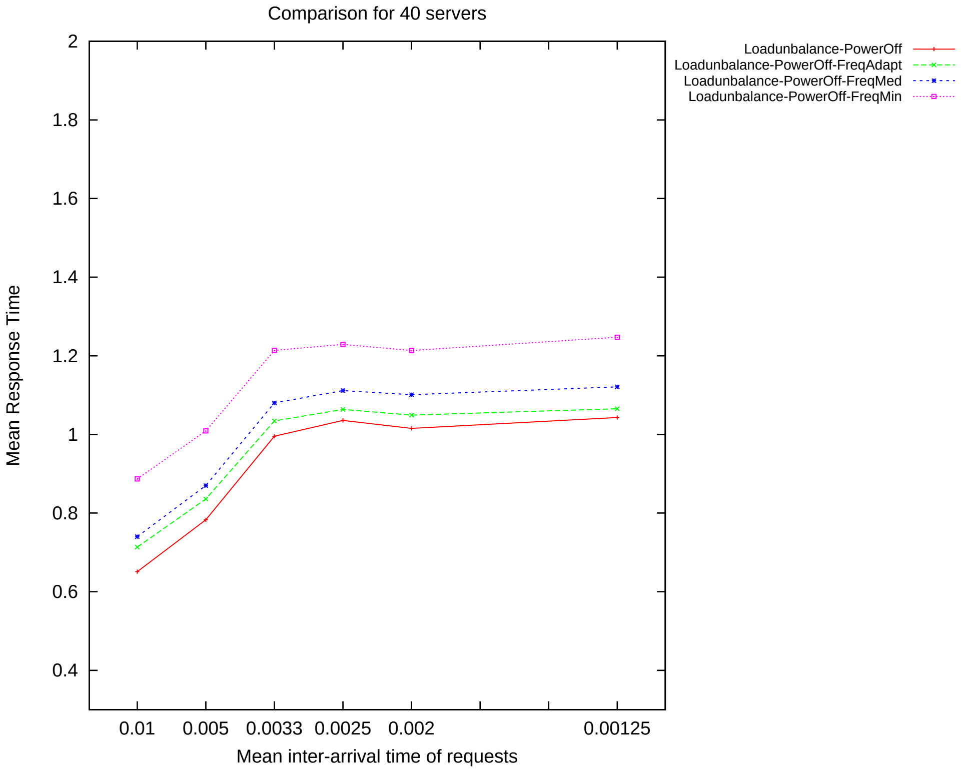


Figure 6.6: Mean response time vs. mean inter-arrival time of requests for powering-off policies.

sponse time among DVFS policies is more noticeable in case of powering-off policies as shown in Figure 6.6.

6.5 Hit Ratio

Figures 6.7 presents the hit ratio of the different CDN redirection policies for 40 surrogate servers serving $1000k$ client requests. We can see for load-balance policies and load-unbalance, the values remain almost stable and there is a decrease with higher client request frequencies. It shows that during the moderate frequencies of client requests, the impact of frequency on the direct completion of requests from the target surrogates server is very low. But when the frequency of requests is higher, it affects the hit ratio. Higher frequencies of requests cause congestion during

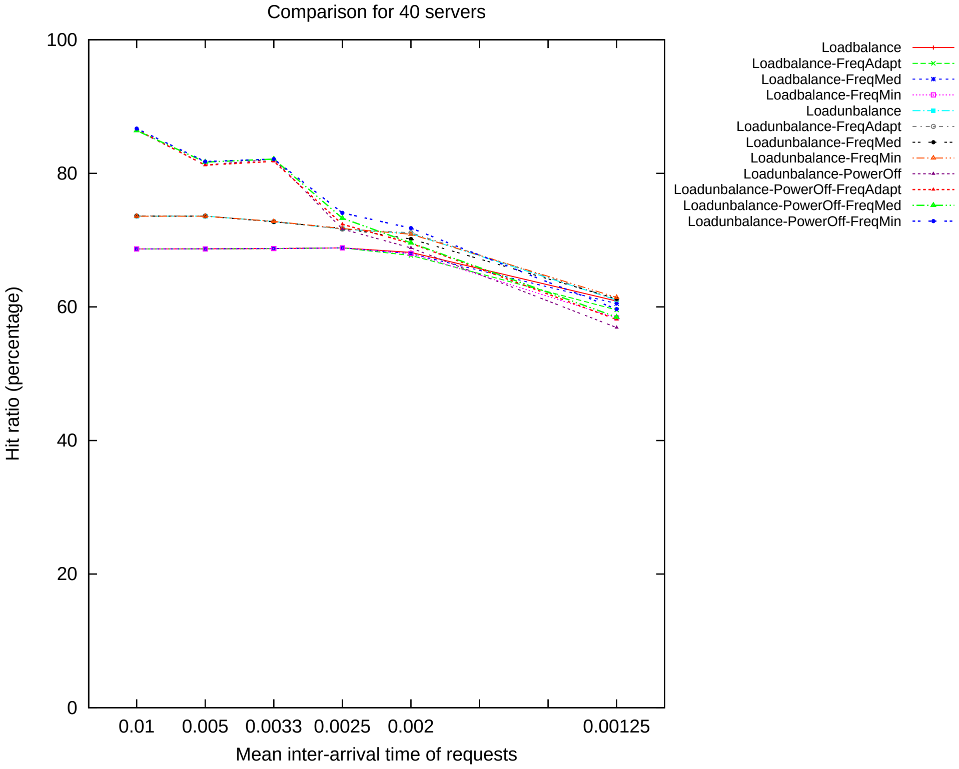


Figure 6.7: Hit ratio vs. mean inter-arrival time of requests for different policies.

some time periods which causes the retries and some failed requests as well. This decreases the overall system hit ratio. Policies which consider powering-off surrogate servers show higher hit ratio during low request frequencies and decreases during high request frequencies. Low request frequencies cause lower loads in surrogate servers which permit more powering-off surrogate servers. So, less surrogate servers are available for more time to serve the requests and their caches get smarter. After start, there is not much switching-on of the powered-off surrogate servers and for most of the simulation time, number of powered-on surrogate serves remain stable, as shown in Figure 6.1.

In case of higher frequencies, less surrogate servers are powered-off and more servers are available to serve the requests which minimizes the hit ratio. Also, the number of powered-on surrogate servers fluctuates: Newer powered-on surrogate servers behaves like dumb which decreases the hit ratio. Also, powering-off surrogate servers at higher frequencies causes more aborted requests which disturbs the overall system's hit ratio.

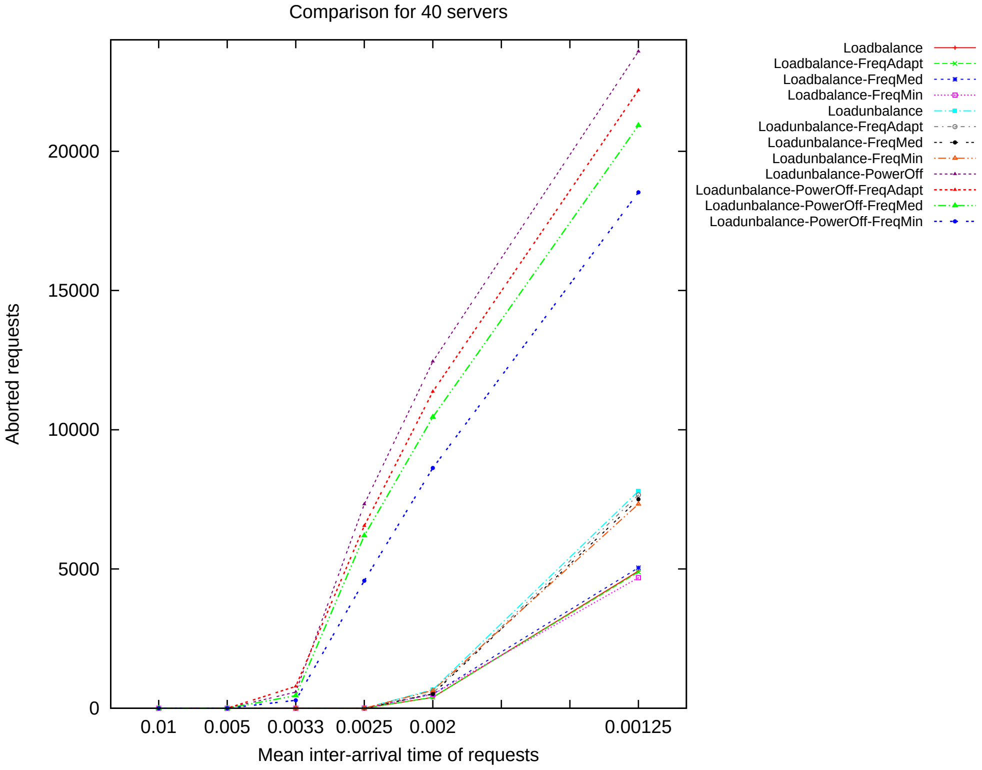


Figure 6.8: Failed requests vs. mean inter-arrival time of requests for different policies.

6.6 Failed Requests

Figure 6.8 shows that probability of the failed requests increases with the increase in the frequency of the client requests. The curve doesn't exhibit any failed requests when the frequency of requests is very low. With higher request frequency, we can see three divisions of the curves on the graph. These divisions show the failed requests for three basic policies i.e. load-balance, load-unbalance and load-unbalance power-off. Load-balance policies have minimum failed requests because of uniform distribution, which doesn't cause much congestion of requests on some surrogate servers. But when the frequency of requests is augmented, it causes some failed requests. The second best on the curve is load-unbalance. The concentration of the traffic on some surrogate servers causes failed requests when frequency of the requests is augmented that causes congestion on that group of surrogate servers which handle most of the client requests.

In case of failed requests, the policies which apply turning-off surrogate servers have the worst performance as compared to the previously discussed policies. It shows maximum number of failed requests which increases with the increase in the frequency of client requests. The reason behind it is the mechanism of powering-off surrogate servers which causes the denial of services due to unavailability of the services and the concentration of the client requests to a group of surrogate servers. Higher processing of requests also causes rapidity in the CDN cooperation process which creates the saturation and congestion in the neighborhood that causes augmentation in failed requests (in a kind of avalanche scenario).

6.7 Conclusion

This chapter concludes how the different requests frequencies have an impact on the different CDN redirection policies. It is shown that low frequencies of requests cause higher energy consumption due to their long duration without server consolidation. But in contrast, policies considering server consolidation offer higher energy savings during lower frequencies of client requests, also providing higher hit ratio with lower response time. During high request frequencies, consolidation causes higher energy consumption but more stable than the rest of the policies. Similarly, the use of DVFS also provides energy savings but at a smaller scale than consolidation with higher response time. It is seen that using DVFS and consolidation together provides considerable energy savings with a good impact of DVFS as well.

Chapter 7

Conclusion and Perspectives

Contents

Internet plays an important role to connect the world. Large scale distributed systems are one of the major sources to provide the internet services. Content Distribution Network (CDN) is one of the popular large scale distributed systems which helps to serve Internet contents to the widely dispersed Internet users. Internet users are increasing rapidly with the passage of time. This is causing a fast increase in the Internet infrastructure at a very large scale that is increasing energy requirements of such systems. Increase in energy consumption results into many associated problems like increase in system and services cost, increase in the world wide atmosphere temperature etc. In order to solve this problem, it is crucial to propose, develop and to execute the techniques, strategies and technologies which provide energy-aware solutions and efficient utilization of resources while not ignoring the user requirements as well.

7.1 Conclusion

In this thesis, we have tried to solve some parts of above discussed big issue. We have concentrated on energy reduction in a CDN. We emphasize particularly on energy management in surrogate servers. Moreover, we have explored energy reduction affiliated issues like its effects on user experience, resource utilization and on quality of infrastructure management.

- ***Modeling:*** We started from modeling energy consumption in surrogate servers. To achieve this milestone, we needed a metric which can permit us to calculate energy consumption in surrogate servers. We found that surrogate server's utilization can lead us to explore energy consumption calculations in surrogate servers. We proposed linear model to compute utilization in surrogate servers. By using this model, we have derived linear energy consumption model in surrogate servers.

- ***CDN Redirection Policies:*** CDN redirection policies perform forwarding of client requests to the appropriate surrogate servers depending upon policy principles. Different CDN redirection policies result in different behavior in CDN operations. In a CDN, load-balancing has been an interesting policy to redirect client requests in an uniform way to perform CDN operations smoothly. Playing with the load of client requests and their redirection, other policies can also be derived like load-unbalancing. Hence, we proposed load-balance and load-unbalance policies. We targeted CDN redirection policies for energy conservation in a CDN. Therefore, we have considered both policies to find some energy reduction opportunities in different scenarios of client requests redirection in a CDN.

- ***DVFS (Dynamic Voltage Frequency Scaling):*** In normal CDN conditions, the peak processing capability of the underlying CPU (Central Processing Unit) is required while having higher loads, that presents a small percentage of its overall operational time. Also, there are some time periods where the load behavior remains almost at same level. Considerable power savings can be achieved by switching the CPU to a lower operating frequency setting when less processing power is needed. By considering this concept, we exhibited energy reduction technique DVFS for surrogate servers. In order to inject this techniques in CDN operations, we applied DVFS to load-balance and load-unbalance policies. To apply this technique at overall system and at machine level, we exhibited global (FreqMin, FreqMed and FreqMax) and local DVFS (FreqAdapt and FreqAdapt2) techniques, respectively.

- ***Consolidation:*** Power consumption can be directly attributed to the number of devices that are active at a given point in time. Substantial power savings can be obtained by turning-off devices which are not in use. Our mechanism begins with the premise that, if one considers the load of requests being carried by a CDN, the capacity and energy requirements of the surrogate servers which are available to receive, process and then to serve the requests and the user experience requirements of the client requests as well, then one may rationally chose a set of surrogate servers so as to satisfy the required user experience at some extent at a minimum possible energy cost. As load-unbalancing policy diverts the requests to a set of surrogate servers, we applied server consolidation technique by considering load-unbalancing policy.

- ***Evaluation Scenarios:*** CDN infrastructure size (number of surrogate servers), size of traffic (number of client requests) and intensity of client requests (client request's frequency) play an important role in CDN operations. With the variation in these parameters, CDN behavior is changed. Therefore, we considered these CDN scenarios, to simulate our proposed policies and mechanisms for evaluating the validity of our work in different CDN operational situations.

  There are different types of CDN clients and owners which can be divided into different groups, based on their requirements of cost and quality e.g. cost-oriented, cost and quality oriented and quality-oriented. Depending on this variety of CDN clients and owners requirements, we evaluated our all proposed policies in all previously discussed different CDN scenarios.

- ***Evaluation Metrics:*** In order to evaluate any techniques or mechanisms, it is important to know: how the devices are utilized, what is the energy consumption and how effectively it is consumed, at what extent the user experience is affected and how the quality of infrastructure is exhibited. For this purpose, we took surrogate server utilization, energy consumption, energy per request, mean response time, failed requests and hit ratio as the evaluation metrics, to evaluate our proposed work.

- ***Simulator:*** To evaluate our proposed work, we needed a simulator capable of providing us a complete CDN simulation environment. CDNsim provides a discrete event simulation environment basing on a solid simulation library of OMNET++. CDNsim provides a complete CDN simulation environment but it didn't offer energy-aware mechanisms. CDNsim provides various features, particularly, adding new parameters and client requests redirection policies. Hence, we have proposed and added energy-aware parameters, evaluation metrics and energy-oriented redirection policies in CDNsim. So, we transformed CDNsim into Green CDNsim.

- ***Results:*** CDN redirection policies play an important role in CDN operations. This aspect was already studied particularly by Stamos et al. [Stamos 2009] and their results prove the validity of our work in this aspect. Additionally, we take into account energy aspect which was not done before.

  Our results show that:

  - When the number of client requests in a CDN is increased, globally, more energy is consumed in surrogate servers.
  - Increase in the frequency of requests also causes increase in surrogate server load but the overall duration of execution for all requests is minimized because of which overall energy consumption at lower frequency is higher due to higher time.
  - Similarly, the size of the CDN infrastructure (number of surrogate servers) also causes increase in the energy consumption due to the increased number of turned-on servers which causes increase in the static energy consumption.
  - Load-balance policies performs better in user experience and also less energy is consumed as compared to the load-unbalance policies if powering-off is not considered.
  - By turning-off surrogate servers, we gain energy savings during lower loads but it degrades the user experience i.e. increased response time,

higher failed requests.

- DVFS also provides energy savings. Impact of energy gains is higher with higher loads. One of the advantages of using DVFS is lesser impact on user experience as compared to server consolidation technique.

7.2 Perspectives

Following steps are considered to continue our research in future:

- ***Cache policy and cache size:*** Cache of a surrogate server stores the contents. How the contents and which contents are stored for how long etc issues depend on the cache policy of a surrogate server. The size of a surrogate server cache also plays a role for the CDN operations. Different policies and different size of a surrogate server cache may play a role at its utilization and ultimately at its energy consumption. One of the near future objectives is to test the CDN with different cache policies and with different cache size and to notice its impact on energy savings as well as on user experience and on quality of infrastructure management.

- ***Network level energy savings in a CDN:*** Overall power consumption of a geographically distributed CDN may also consider the energy cost for the transportation of the contents among different sites which depends on several factors of the underlying network infrastructure and a considerable knowledge is required because of its complexity. Our research focus, in this thesis, doesn't include this aspect and can be considered for future works. Different energy saving techniques can be considered to have the knowledge of overall energy savings in a CDN e.g. sleeping or turning-off network links can be obtained using load-leveraging by diverting the traffic to fewer links which also helps to switch-off the routers connected at these paths as well, changing the voltage of network links according to the load of client requests etc.

- ***Geographical variation of energy cost:*** Energy costs in different areas in the world are different. A CDN surrogate server which is installed in United

States of America has a different energy cost than a surrogate server which is installed in France. The price of electricity is changed according to its production method and other factors. Electricity produced by atomic sources is less costly than that of off-shore sources. One of our future directions is to do the analysis of these geographic factors affecting the energy savings and its variation in prices for a CDN, including CO2 ecological costs as well.

- ***Requirements based CDN redirection policies:*** CDN clients may have different types of priorities e.g. quality based, cost based, quality and cost based etc. Similarly the surrogate servers spread in different geographic locations can also be classified regarding cost and user experience. Based on these specifications of clients and services, we can classify the user requirements and services. This classification can permit us to propose and to develop policies which can offer the redirection of the client requests according to its requirements class to the corresponding set of surrogate servers.

Part II

Summary :French version

Chapter 1

Version Française

Contents

1.1 Résumé

Les infrastructures Internet et l'installation d'appareils très gourmands en énergie (en raison de l'explosion du nombre d'internautes et de la concurrence entre les services efficaces offerts par Internet) se développent de manière exponentielle. Cela entraîne une augmentation importante de la consommation d'énergie. La gestion de

l'énergie dans les systèmes de distribution de contenus à grande échelle joue un rôle déterminant dans la diminution de l'empreinte énergétique globale de l'industrie des TIC (Technologies de l'information et de la communication). Elle permet également de diminuer les coûts énergétiques d'un produit ou d'un service. Les CDN (Content Delivery Networks) sont parmi les systèmes de distribution à grande échelle les plus populaires, dans lesquels les requêtes des clients sont transférées vers des serveurs et traitées par des serveurs proxy ou le serveur d'origine, selon la disponibilité des contenus et la politique de redirection des CDN.

Par conséquent, notre objectif principal est de proposer et de développer des mécanismes basés sur la simulation afin de concevoir des politiques de redirection des CDN. Ces politiques prendront la décision dynamique de réduire la consommation d'énergie des CDN. Enfin, nous analyserons son impact sur l'expérience utilisateur.

Nous commencerons par une modélisation de l'utilisation des serveurs proxy et un modèle de consommation d'énergie des serveurs proxy basé sur leur utilisation. Nous ciblerons les politiques de redirection des CDN en proposant et en développant des politiques d'équilibre et de déséquilibre des charges (en utilisant la loi de Zipf) pour rediriger les requêtes des clients vers les serveurs. Nous avons pris en compte deux techniques de réduction de la consommation d'énergie : le DVFS (Dynamic Voltage Frequency Scaling) et la consolidation de serveurs. Nous avons appliqué ces techniques de réduction de la consommation d'énergie au contexte d'un CDN (au niveau d'un serveur proxy), mais aussi aux politiques d'équilibre et de déséquilibre des charges afin d'économiser l'énergie.

Afin d'évaluer les politiques et les mécanismes que nous proposons, nous avons mis l'accent sur la manière de rendre l'utilisation des ressources des CDN plus efficace, mais nous nous sommes également intéressés à leur coût en énergie, à leur impact sur l'expérience utilisateur et sur la qualité de la gestion des infrastructures. Dans ce but, nous avons défini comme métriques d'évaluation l'utilisation des serveurs proxy, la consommation d'énergie, l'énergie utilisée par requête, le temps de réponse moyen, le hit ratio et le taux d'échec des requêtes. Afin d'analyser la réduction de la consommation d'énergie et son impact sur l'expérience utilisateur, nous considérons la consommation d'énergie, le temps de réponse moyen et le taux

d'échec des requêtes comme les paramètres les plus importants.

Nous avons transformé un simulateur d'événements discrets CDNsim en Green CDNsim, et évalué notre travail selon différents scénarios de CDN en modifiant : les infrastructures proxy des CDN (nombre de serveurs proxy), le trafic (nombre de requêtes clients) et l'intensité du trafic (fréquence des requêtes client) en prenant d'abord en compte les métriques d'évaluation mentionnées précédemment.

Nous sommes les premiers à proposer un DVFS et la combinaison d'un DVFS avec la consolidation d'un environnement de simulation de CDN en prenant en compte les politiques d'équilibre et de déséquilibre des charges. Nous avons conclu que les techniques d'économie d'énergie permettent de réduire considérablement la consommation d'énergie mais dégradent l'expérience utilisateur. Nous avons montré que la technique de consolidation des serveurs est plus efficace dans la réduction d'énergie lorsque les serveurs proxy ne sont pas beaucoup chargés. Dans le même temps, il apparaît que l'impact du DVFS sur l'économie d'énergie est plus important lorsque les serveurs proxy sont bien chargés. La combinaison des deux (DVFS et consolidation des serveurs) permet de consommer moins d'énergie mais dégrade davantage l'expérience utilisateur que lorsque ces deux techniques sont utilisées séparément.

Mots clés : Content Distribution Networks, Économie d'énergie, Expérience utilisateurs

1.2 Introduction

Depuis le commencement de l'évolution technologique, la vie de l'homme est basée sur l'utilisation des machines. Parfois l'on ressent que nos vies sont totalement dictées par les machines qui sont autour de nous. Un scénario simple de notre vie quotidienne commence par le bip sonore de notre réveil, un moyen de transport mécanisé pour aller au travail, un café (à la machine) pour démarrer la journée etc. Chacune de ces machines consomme de l'énergie et cette énergie produit du CO_2 (dioxyde de carbone) et des déchets nucléaires.

Durant les deux dernières décennies, nous avons commencé à ressentir les effets de ces émissions de CO_2 et de cette consommation d'énergie. Du fait de notre vaste exploitation des machines, le prix de l'énergie devient de plus en plus cher. Pour gérer cette utilisation exponentielle des machines, et donc la forte consommation d'énergie qui en découle, différent axes de recherche ont été lancés, comme la concentration et la réduction d'énergie consommée. L'objectif final de ce champ de recherche intensif est de fournir une nouvelle forme de technologie moins énergivore mais tout aussi efficace.

1.2.1 Domaine d'intérêt

Le nombre d'utilisateurs d'Internet, et donc les infrastructures l'utilisant, augmente significativement chaque année. Ce changement entraine une tendance vers un imposant système réparti géographiquement [Qureshi 2009]. Ces systèmes peuvent compter une immense quantité de serveurs et de nombreux data centers. Des millions de watts d'énergie sont nécessaires pour faire fonctionner un grand data center [Katz 2009]. Parmi les types de réseau les plus répandus, on trouve le Content Distribution Network (CDN) [Pallis 2006]. Un CDN est un réseau superposé responsable de la gestion de toutes les données des contenus des utilisateurs du Web. Un CDN consiste en un ensemble de serveurs proxy disséminés sur le Web. Ces serveurs contiennent des copies (réplicas) du contenu appartenant au serveur d'origine (conformément aux capacités de stockage spécifique de chacun). L'idée principale étant de rendre le contenu des réplicas plus proche des utilisateurs, induisant un temps de réponse plus faible et une disponibilité des contenus plus grande puisque de nombreux réplicas sont distribués.

Notre but est de définir une stratégie majeure de redirection moins énergivore qui améliorerait les travaux précédents [Stamos 2009]. Cela inclut les économies d'énergie et s'appuie sur l'utilisation de modèles de serveurs proxy. Nous nous sommes concentrés sur la réduction de la consommation d'énergie des CDN au niveau des serveurs proxy en travaillant sur les politiques de redirection des CDN. Notre travail n'inclut pas l'analyse de l'économie d'énergie au niveau des réseaux, cela fera l'objet d'études futures. Par conséquent cette recherche vise à explorer, proposer et développer des technologies moins énergivores dans les CDN et à analyser leur impact sur l'expérience utilisateur.

1.2.2 Objectifs

Les objectifs de notre recherche incluent :

- L'identification des indicateurs adéquats de CDN pour définir la consommation d'énergie dans les CDN.
- L'identification et l'analyse de la consommation d'énergie dans les CDN.
- La modélisation de la consommation d'énergie dans les CDN.
- L'étude des opportunités d'économiser l'énergie dans les systèmes de CDN.
- L'identification et le développement de techniques liées à l'économie d'énergie appliquées au développement des environnements des CDN.

1.2.3 Contributions

Les principaux progrès obtenus lors de cette recherche sont les suivants :

- La création de modèles de consommation d'énergie à partir de l'utilisation des serveurs proxy dans les CDN.
- La proposition et le développement de politiques de redirection des CDN et des réductions de la consommation d'énergie. Pour cela nous appliquons des techniques d'économie d'énergie (Consolidation et DVFS) aux politiques de redirection traditionnelles des CDN.
- Le développement d'un simulateur pour intégrer les questions liées à l'économie d'énergie et à ses techniques dans le but d'évaluer les concepts proposés.
- L'étudie de l'impact de la taille de l'infrastructure du CDN (nombre de serveurs), la taille du trafic (nombre de requêtes) et la fréquence du trafic.
- (À partir de ces techniques) la proposition de techniques d'économie d'énergie pouvant être évaluées lors de travaux futurs.

1.3 Simulation des Content Delivery Networks (CDNs)

Afin d'évaluer le fonctionnement des CDN selon différentes configurations, il est essentiel d'avoir un banc d'essai qui fournit un environnement de simulation analytique pour le CDN, car les applications du CDN en temps réel sont difficiles à obtenir dans un but de recherche. Nous avons également besoin d'un ensemble de traces des utilisateurs sur le Web (qui accèdent aux contenus d'un serveur Web via un CDN), et surtout de la topologie des contenus d'un serveur Web. Cela nous aidera à identifier les communautés de pages Web. Cet environnement comprend :

- Un modèle de système simulant l'infrastructure du CDN
- Un générateur de topologie de réseaux
- Un générateur de sites Web
- Un générateur de flux de requêtes clients

Pour cela, l'environnement de simulation CDNsim convient parfaitement [Stamos 2010].

1.3.1 CDNsim

CDNsim simule une infrastructure principale de CDN et est implémentée en langage de programmation C++. Il est basé sur la bibliothèque OMNeT++, qui fournit un environnement de simulation d'événements discrets, et prend en compte les spécificités de l'infrastructure d'Internet. Il est solide et évolutif, il fournit une large gamme de politiques de CDN. Il a également été conçu pour fournir une large gamme de services de CDN à des fins de recherche.

Le CDNsim traite dynamiquement tous les enjeux des réseaux de CDN comme la sélection des serveurs proxy, la propagation, la mise en attente, l'engorgement et les retards de traitement. Il permet de développer dans le détail des protocoles TCP (Transmission Control Protocol) / IP (Internet protocol), de développer la commutation de paquets, de retransmettre les paquets en cas d'échec, de rafraîchir des pages, etc. L'environnement de simulation CDNsim prend en compte différents paramètres d'entrée pour effectuer la simulation, par exemple la topologie de réseau, la vitesse du lien, le site Web, les flux de requêtes, la taille du cache, le nombre de serveurs, le nombre de clients, les politiques du CDNsim, etc.

Nous avons pris en compte différentes métriques d'évaluation côté serveur et côté client afin d'évaluer le comportement du CDN : le temps de réponse moyen, le hit ratio, le byte hit ratio, l'utilité du CDN, les requêtes effectuées et échouées, etc.

1.3.2 Du CDNsim au Green CDNsim

Le CDNsim original ne prend pas en compte l'aspect énergétique de la simulation CDN. Nous avons donc transformé le CDNsim en Green CDNsim. Pour cela, nous avons ajouté de nouveaux paramètres d'entrée ($MinLoad$, $MaxLoad$, $FreqMin$, $FreqMed$, $FreqMax$, $DVFSMinLoad$, $DVFSMedLoad$, $DVFSMaxLoad$, $cdnON$, $redirectionPolicy$, $cooperationON$, $ZipfUnbalance$) qui nous permettent de proposer de nouvelles politiques de redirection des CDN (voir Figure 1.1) dans le CDNsim. Nous avons également pris en compte de nouvelles métriques d'évaluation (Utilisation, Énergie, Énergie par requête, Temps de réponse, Nombre de serveurs proxy allumés).

Dans le tableau 1.1, vous trouverez les différents paramètres utilisés dans l'évaluation de cette thèse. Une étude exhaustive aurait été nécessaire pour tester toutes les combinaisons des différentes valeurs, surtout les paramètres qui peuvent influencer le comportement des serveurs proxy (taille du cache, popularité, vitesse du lien, temps moyen d'arrivée des requêtes, nombre de serveurs proxy, nombre de requêtes clients). Cependant, nous pensons que les paramètres d'évaluation proposés (certains ayant été validé sur des travaux précédents sur le CDNsim), sont assez nombreux pour évaluer la pertinence de nos approches visant à mesurer et réduire la consommation d'énergie des CDN.

1.4 Politiques pour l'économie d'énergie dans les Content Delivery Networks (CDNs)

Cette section a pour objectif d'identifier le sujet de recherche suivant et de présenter les moyens d'étudier les solutions possibles. Comment un CDN peut-il rediriger les

Table 1.1: Récapitulatif des paramètres de simulation

Paramètre	Ensemble d'expériences 1	Ensemble d'expériences 2
Taille du site Web	$1GB$	
Nombre d'objets de site Web	50000	
Paramètre z pour taille de site	1	
Taille vs corrélation de popularité	0	
Nombre de requêtes	2×10^5, 4×10^5, 6×10^5, 8×10^5, 10^6	10^6
Intervalle moyen entre les requêtes	0.0033	0.01, 0.005, 0.0033, 0.0025, 0.002, 0.00125
Répartition des intervalles	*exponentielle*	
Flux des requêtes z	1	
Vitesse de lien	$16Mbps$	
Type de topologie réseau	AS	
Nombre de routeurs dans le cœur de réseau	3037	
Nombre de serveurs proxy	$10, 20, 30, 40, 50$	40
Processeur	Intel (R) Xeon (R) $E5620$	
Fréquence minimum du processeur (FreqMin)	$1.6GHz$	
Fréquence moyenne du processeur (FreqMed)	$2.0GHz$	
Fréquence maximum du processeur (FreqMax)	$2.4GHz$	
Nombre de connexions entrantes par serveur proxy	500	
Nombre de connexions sortantes par serveur proxy	500	
Charge minimum du serveur proxy (MinLoad)	0, 0.05	
Charge maximum du serveur proxy (MaxLoad)	0.9	
Nombre de groupes de clients	100	
Nombre de fournisseurs de contenus (serveur d'origine)	1	
Nombre maximum de connexions entrantes par serveur d'origine	3500	
Pourcentage de la taille du cache par rapport à la taille du site web	40%	
Politique de remplacement du cache	LRU	
Valeur z des paramètres de déséquilibre des charges ($ZifUnbalance$) z value	0, 1	
Nombre de grains (seeds)	10	20

requêtes de contenus des clients vers ses serveurs proxy de manière à minimiser la consommation d'énergie, tout en maintenant une qualité d'expérience acceptable ? Afin de répondre à cette question, nous avons mis en place différentes étapes : des définitions théoriques aux réalisations concrètes. La répartition géographique des serveurs offre souvent différentes opportunités d'optimiser la consommation d'énergie et les coûts en répartissant intelligemment la charge de travail. Pour développer ces politiques, minimiser la consommation d'énergie ne suffit pas, il faut également prendre en compte la performance et la disponibilité des services. L'une des approches les plus populaires consiste à rediriger le trafic vers moins de périphériques et d'éteindre ceux qui ne sont pas utilisés, ou bien de les passer en mode veille (appelé également consolidation des serveurs).

Ce concept se base sur le fait que le trafic du réseau n'est pas toujours le même. Dans des conditions réseau normales, les périphériques de réseau ne sont pas utilisés à leur pleine capacité. Il faut encore introduire des mécanismes intelligents permettant d'utiliser les périphériques de réseau selon leur capacité et ainsi d'économiser efficacement de l'énergie. De même, dans des conditions de réseau normales, les serveurs fonctionnent sur des fréquences processeur élevées pour fournir aux clients les contenus demandés. Le trafic des requêtes clients est fluctuant. Si les processeurs des serveurs travaillent en basse fréquence, la consommation d'énergie sera plus faible. L'objectif est de se servir de cette idée et de régler les fréquences du processeur en fonction des conditions du trafic des clients. Cela permet d'économiser de l'énergie. Dans ce but, nous avons pris en compte deux techniques basiques de gestion de l'énergie : la consolidation des serveurs et le DVFS. Nous avons appliqué ces techniques aux politiques de redirection des requêtes du CDN.

1.4.1 Consommation d'énergie des serveurs des Content Delivery Networks

Chaque serveur proxy consomme une quantité d'énergie constante dès qu'il est allumé. Le reste de l'énergie consommée peut être considérée comme proportionnelle à l'utilisation. Dans ce contexte, nous partons du principe que la consommation d'énergie est proportionnelle au ratio des connexions actives par rapport au nombre de connexions simultanées que le serveur proxy peut supporter. Ce nombre de connexions actives explique les tâches effectuées côté serveur pour retrouver les données (gestion de l'index), l'IO disque (Input Output) pour aller chercher les données, la connexion réseau et la politique de gestion du cache. On peut utiliser un modèle élargi pour déduire la consommation d'énergie en se basant sur la charge actuelle de l'IO, des réseaux et du CPU (Central Processing Unit). Ce modèle peut également être envisagé pour de futures études. Cependant, cette hypothèse suffit pour comparer la consommation d'énergie de différents nombres de serveurs et les requêtes de trafic. Bien entendu, notre objectif principal n'est pas d'évaluer la consommation énergétique de manière précise, mais bien d'avoir une métrique permettant de comparer plusieurs scénarios.

1.4.2 Politiques

Partons du scénario d'un CDN et proposons deux politiques de base pour rediriger les requêtes du client vers les serveurs proxy du CDN. Nous utilisons la loi de Zipf afin de définir ces politiques. Nous prenons en compte la loi de Zipf avec le paramètre de déséquilibre des charges (zipfUnbalance parameter) $z \in \{0, .., 1\}$. Pour la valeur 0, nous avons une répartition uniforme et pour la valeur 1, nous avons une répartition exponentielle où seul un petit pourcentage réunit la plus grande partie de la répartition. L'algorithme de redirection client fonctionne comme suit :

- Classement des serveurs proxy en fonction de leur utilisation actuelle
- Classement du paramètre z
- Choix d'un serveur proxy de manière aléatoire en fonction de la probabilité définie par la loi de Zipf, avec une valeur de paramètre z.

En fonction de cet algorithme, on déduit l'équilibre et le déséquilibre des charges. L'équilibre des charges suit une répartition uniforme tandis que le déséquilibre des charges effectue une répartition exponentielle des requêtes. On observe que les politiques traditionnelles de redirection des requêtes clients du CDN ne prennent pas

Table 1.2: Les Politiques Locales DVFS

FreqAdapt	**FreqAdapt2**	**Fréquence du Processeur Correspondant**
$0 \leq charge \leq 0.5$	$0 \leq charge \leq 0.2$	$F_{min} = 1.6GH_z$
$0.5 < charge \leq 0.7$	$0.2 < charge \leq 0.7$	$F_{med} = 2GH_z$
$0.7 < charge \leq 0.9$	$0.7 < charge \leq 0.9$	$F_{max} = 2.4GH_z$

en compte l'économie d'énergie. Les serveurs proxy très peu utilisés ne doivent pas nécessairement rester allumés. Il vaut mieux arrêter ces serveurs proxy après qu'ils aient effectué les requêtes afin d'économiser de l'énergie (Consolidation des serveurs). La fréquence du processeur peut être diminuée ou modifiée automatiquement en fonction de la charge des serveurs proxy (DVFS (Dynamic Voltage Frequency Scaling)). Cela permettra de réduire la consommation d'énergie.

Nous avons donc déduit les politiques de redirection des requêtes clients du CDN à partir des politiques de base de CDN vues dans cette étude, c.-à-d. l'équilibre et le déséquilibre des charges. Nous avons appliqué deux techniques d'économie d'énergie populaires (consolidation des serveurs et de DVFS) à ces politiques de base afin d'améliorer l'économie d'énergie et d'analyser leur impact sur les services et les opérations du CDN. Nous avons proposé deux types de politiques DVFS : Les politiques DVFS globaux (tous les serveurs proxy fonctionnent sur la même fréquence fixe de processeur) et les politiques DVFS locaux (la frèquennce de processeur de chque serveur proxy change en fonction de sa charge, par exemple les politiques $FreqAdapt$ et $FreqAdapt2$ indiqués dans le tableau 1.2). Nous appliquons l'approche conjointe de la consolidation des serveurs et du DVFS à la politique du déséquilibre des charges. Puis nous appliquons l'approche DVFS à la politique d'équilibre des charges et déduisons différentes politiques de redirection des CDN intelligents sur le plan énergétique, comme indiqué dans la Figure 1.1. Les politiques proposées sont évaluées dans la section suivante.

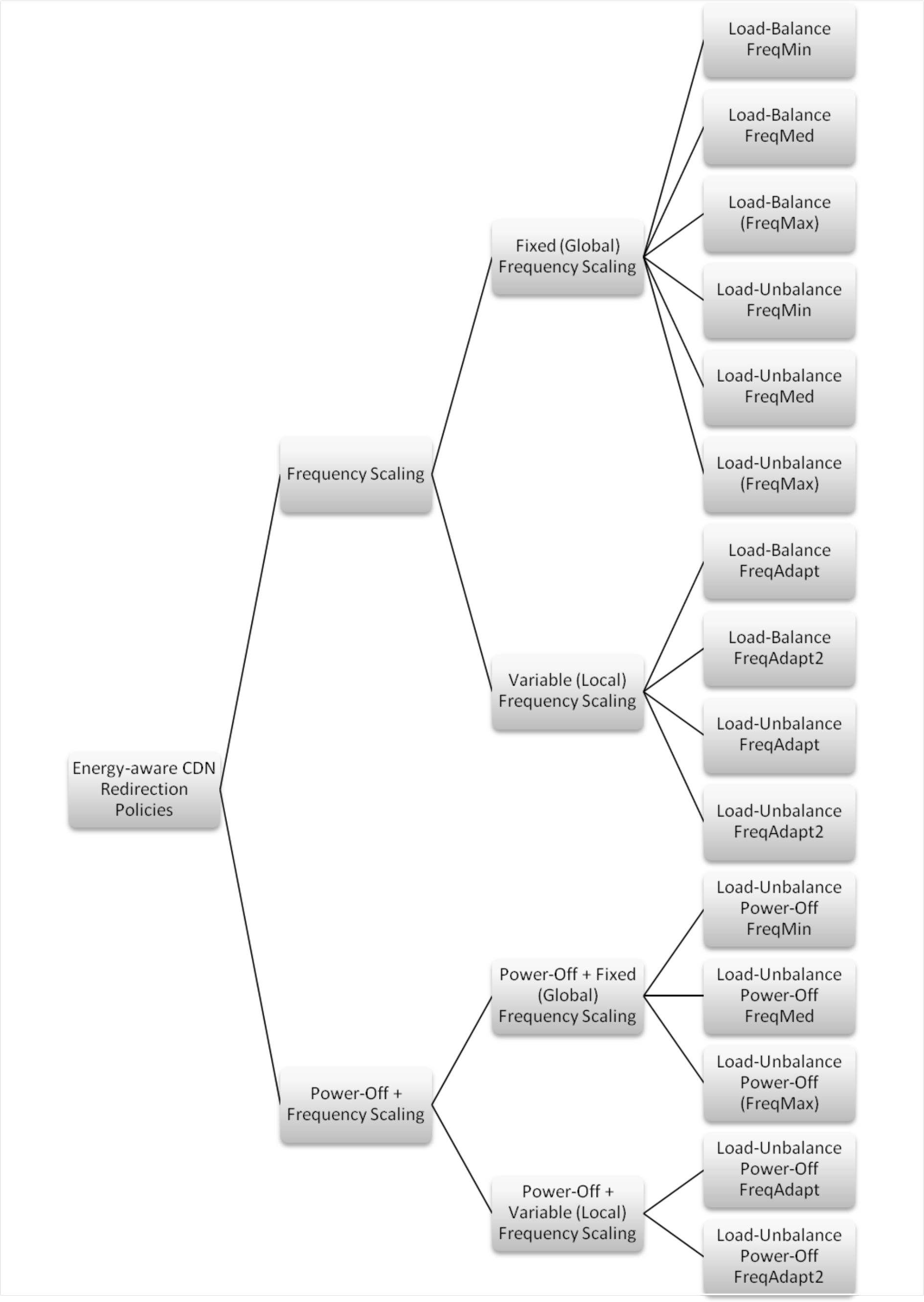

Energy-aware CDN Redirection Policies
Frequency Scaling
Power-Off + Frequency Scaling
Fixed (Global) Frequency Scaling
Variable (Local) Frequency Scaling
Power-Off + Fixed (Global) Frequency Scaling
Power-Off + Variable (Local) Frequency Scaling
Load-Balance FreqMin
Load-Balance FreqMed
Load-Balance (FreqMax)
Load-Unbalance FreqMin
Load-Unbalance FreqMed
Load-Unbalance (FreqMax)
Load-Balance FreqAdapt
Load-Balance FreqAdapt2
Load-Unbalance FreqAdapt
Load-Unbalance FreqAdapt2
Load-Unbalance Power-Off FreqMin
Load-Unbalance Power-Off FreqMed
Load-Unbalance Power-Off (FreqMax)
Load-Unbalance Power-Off FreqAdapt
Load-Unbalance Power-Off FreqAdapt2

Figure 1.1: Politiques de redirection des CDN intelligents sur le plan énergétique.

1.5 Résultats

Afin d'évaluer les politiques proposées, nous avons considéré les impacts des différents paramètres d'évaluation des infrastructures des CDN (nombre des servers proxy) dans le nombre des requêtes (taille du trafic). Nous avons également évalué l'impact des paramètres de simulation de la fréquence des requêtes clients, l'ensemble des données et les métriques d'évaluation vus dans la section1.3.

Nos résultats nous ont montré que les infrastructures des CDN ne se comportaient pas de la même manière selon les différentes politiques de redirection des requêtes clients. En tenant compte du comportement des politiques de redirection des CDN, Nous appliquons les techniques d'économie d'énergie. La technique DVFS est appliquée aux deux politiques d'équilibre et de déséquilibre des charges. Du fait de la sous-utilisation des serveurs en raison du déséquilibre des charges, nous appliquons la technique de consolidation. L'augmentation du trafic entraîne une plus grande consommation d'énergie dans les infrastructures des CDN. Mais en même temps, cela aide à faire mûrir les infrastructures des CDN en rendant les caches plus intelligents dans le but d'optimiser l'impact sur l'expérience utilisateurs.

Une augmentation de la taille de l'infrastructure de CDN entraîne également une augmentation de la consommation d'énergie du fait de l'augmentation de la consommation d'énergie statique et dynamique. Une plus grande infrastructure entraîne une plus grande consommation d'énergie statique lorsqu'elle possède moins de charge.

Nous avons montré que la consolidation des serveurs offre une économie d'énergie considérable dans les infrastructures des CDN et maximise l'utilisation des serveurs proxy. Néanmoins cela dégrade l'expérience utilisateur car cela augmente le temps de réponse. D'autre part, les petites infrastructures ayant un nombre de requêtes élevé complexifient les services des CDN, car ils créent des échecs de requêtes et augmentent le temps de réponse, etc. Le choix de la politique dépend de la demande du client et du fournisseur de service.

Les politiques de redirection des CDN ont un impact sur l'évaluation des différents paramètres. Il est primordial de comparer les différentes politiques des différents paramètres d'évaluation, d'avoir une idée globale des relations parmi les politiques et d'en choisir une en fonction des demandes des clients et des fournisseurs de service des CDN. Les politiques dépourvues de techniques d'économie d'énergie (équilibre et déséquilibre des charges) avec FreqMax donnent une meilleure expérience utilisateurs avec une plus grande disponibilité des services, mais également un coût énergétique plus élevé. En appliquant la technique de consolidation, la politique d'équilibre des charges montre des meilleurs résultats concernant l'économie d'énergie. Cela nous montre aussi une meilleure expérience utilisateurs en fournissant un temps de réponse aux requêtes plus faible. Le déséquilibre des charges nous montre une plus grande économie de l'énergie en cas de consolidation. En appliquant seulement les politiques DVFS, nous obtenons une économie d'énergie modérée avec une plus grande disponibilité des services et une faible dégradation de l'expérience utilisateurs. Les politiques DVFS fournissent une plus grande économie d'énergie pendant les chargements plus élevés. Lorsque l'on a de grandes infrastructures de CDN et moins de charges, les politiques de consolidation des serveurs montrent un plus grand gain d'énergie mais ont plus d'impact sur l'expérience utilisateurs.

Enfin, les politiques avec une approche agressive envers l'économie de l'énergie (combinant DVFS et consolidation) montrent une grande économie d'énergie avec un impact plus important sur l'expérience utilisateurs. L'impact des différents paramètres d'évaluation entre les politiques DVFS locaux et globaux sont plus importants lorsqu'ils sont combinés avec la consolidation des serveurs, dans lequel les politiques DVFS globaux assurent une meilleure économie d'énergie que les DVFS locaux (mais ont un temps de réponse plus important). L'impact sur les différents paramètres d'évaluation entre les techniques $FreqAdapt$ et $FreqAdapt2$ est vraiment insignifiant. Pour résumer, les Tables 1.3 and 1.4 présentent en détail l'impact des techniques d'économie d'énergie (DVFS et consolidation), sur les différents paramètres d'évaluation des politiques d'équilibre et de déséquilibre des charges. Nous avons enfin étudié l'impact de l'intensité de la charge sur les infrastructures des CDN.

Nous avons montré qu'une fréquence des requêtes faible entraîne une hausse de

Table 1.3: Impact de DVFS et de l'arrêt des serveurs proxy : équilibre et déséquilibre des charges : (cas de 40 serveurs proxy répondant aux 1000k requêtes utilisateurs) comparé au cas de l'équilibre des charges(LB FreqMax) E/R = Energie par requête; MRT = Temps de réponse moyen; HR = Hit Ratio; FR = Requêtes échouées; LB = Equilibre des charges; LUB = Déséquilibre des charges

Politique	Utilisation	Energie	E/R	MRT	HR	FR
LB FreqAdapt	−13%	−5%	−5.13%	+24.56%	0%	0%
LB FreqAdapt2	−13.8%	−7%	−6.91%	+26.25%	0%	0%
LB FreqMed	−7.9%	−4%	−4.32%	+13.25%	0%	0%
LB FreqMin	−3.14%	−6%	−5.94%	+28.07%	0%	0%
LUB FreqMax	+26.18%	+5.25%	+5.25%	+8.36%	+5.23%	0%
LUB FreqAdapt	+29, 45%	+2, 31%	+2, 31%	+22, 86%	+5, 22%	0%
LUB FreqAdapt2	+25, 87%	+1, 02%	+1, 02%	+24, 45%	+5, 07%	0%
LUB FreqMed	+28, 00%	−0, 16%	−0, 16%	+19, 56%	+5, 23%	0%
LUB FreqMin	+26, 73%	−2, 49%	−2, 49%	+31, 87%	+5, 27%	0%
LUB Poff FreqMax	+82, 80%	−3.138%	−3.135%	+31.27%	−35.94%	0.0762%
LUB Poff FreqAdapt	+83, 04%	−11, 19%	−11, 19%	+33, 25%	−34, 66%	0.1212%
LUB Poff FreqAdapt2	+82, 88%	−9, 99%	−9, 99%	+33, 71%	−34, 63%	0.0741%
LUB Poff FreqMed	+82, 30%	−19, 45%	−19, 45%	+36, 11%	−33, 18%	0.0616%
LUB Poff FreqMin	+81, 75%	−30, 61%	−30, 61%	+43, 36%	−27, 93%	0.0411%

Table 1.4: Impact de DVFS et de l'arrêt des serveurs proxy : déséquilibre des charges (cas de 40 serveurs proxy répondant à 1000k requêtes utilisateurs), comparé au cas de déséquilibre des charges (LUB FreqMax)

Politique	Utilisation	Energie	E/R	MRT	HR	FR
LUB FreqAdapt	+4.44%	−3%	−3.02%	+15.79%	−0.01%	0%
LUB FreqAdapt2	−0.4%	−4%	−4.30%	+16.63%	−0.17%	0%
LUB FreqMed	+2.44%	−5%	−5.43%	+12.28%	0%	0%
LUB FreqMin	+0.70%	−7.6%	−7.63%	+25.71%	+0.04%	0%
LUB Poff FreqMax	+76.70%	−8.22%	−8.24%	+25.04%	−39.67%	0.0762%
LUB Poff FreqAdapt	+77.03%	−15.85%	−15.87%	+27.21%	−38.08%	0.1212%
LUB Poff FreqAdapt2	+76.81%	−14.72%	−14.75%	+27.69%	−38.05%	0.0741%
LUB Poff FreqMed	+76%	−23.68%	−23.71%	+30.29%	−36.69%	0.0616%
LUB Poff FreqMin	+75.29%	−34.26%	−34.25%	+38.20%	−31.70%	0.0411%

la consommation d'énergie car ces requêtes sont plus longues (si l'on n'utilise pas la technique de consolidation des serveurs). Mais au contraire, les politiques utilisant la technique de consolidation augmentent l'économie d'énergie durant les fréquences des requêtes clients plus faibles. Cela entraîne aussi une augmentation du hit ratio et donc un plus long temps de réponse. Durant les fréquences élevées de requêtes, la consolidation entraîne une augmentation de la consommation d'énergie, mais une plus grande stabilité par rapport aux autres politiques. De la même façon, l'utilisation des DVFS entraîne également une économie d'énergie mais à moins grande échelle par rapport à la technique de consolidation, mais avec des temps de réponse plus longs. Nous avons constaté que l'utilisation conjointe des techniques de DVFS et de la consolidation est plus efficace pour l'économie d'énergie, avec un bon impact sur les DVFS également.

1.6 Conclusion et Perspectives

Dans cette thèse, nous nous sommes concentrés sur la réduction de la consommation d'énergie dans le CDN. Nous avons mis l'accent en particulier sur la gestion d'énergie dans les serveurs proxy. De plus, nous avons étudié les enjeux liés à la réduction d'énergie, comme ses effets sur l'expérience utilisateur, l'utilisation des ressources et la qualité de la gestion de l'infrastructure. Nous avons commencé par créer un modèle de la consommation d'énergie dans les serveurs proxy. Nous avons proposé des politiques d'équilibre et de déséquilibre des charges. Nous les avons ciblées afin de déduire des politiques de redirection des CDN intelligents sur le plan énergétique ayant pour but d'économiser l'énergie dans un CDN. Nous avons pris en compte des techniques intelligentes sur le plan énergétique, telles que le DVFS et la consolidation des serveurs, pour les intégrer dans les politiques d'équilibre et de déséquilibre des charges et ce, dans le but d'économiser l'énergie du CDN. Nous avons proposé et ajouté des paramètres de gestion de la consommation d'énergie, des métriques d'évaluation et des politiques de redirection prenant en compte la consommation d'énergie dans le CDNsim pour le transformer en Green CDNsim. Nous avons évalué les méthodes proposées dans différentes infrastructures de CDN (nombre de serveurs proxy) avec plusieurs tailles et fréquences de trafic client. Nous avons pris comme métriques d'évaluation l'utilisation des serveurs proxy, la consommation d'énergie, l'énergie utilisée par requête, le temps de réponse moyen, les requêtes échouées et le hit ratio. Nos résultats montrent que :

- Lorsque le nombre de requêtes clients sur un CDN augmente, la consommation d'énergie augmente globalement sur les serveurs proxy.
- L'augmentation de la fréquence des requêtes entraîne également une augmentation de la charge des serveurs proxy. Néanmoins, la durée globale d'exécution de toutes les requêtes diminue : la consommation d'énergie à une fréquence basse est supérieure car le temps de traitement est plus long.
- De même, la taille de l'infrastructure du CDN (nombre de serveurs proxy) augmente également la consommation d'énergie car le nombre plus élevé de serveurs allumés augmente la consommation d'énergie statique.
- Les politiques d'équilibre des charges permettent une meilleure expérience utilisateur et diminuent la consommation d'énergie par rapport aux politiques de déséquilibre des charges, si on ne prend pas en compte la technique de consolidation des serveurs.
- En arrêtant les serveurs proxy, on économise de l'énergie pendant les charges plus basses, mais cela dégrade l'expérience utilisateur en augmentant le temps de réponse et le nombre de requêtes échouées.
- Le DVFS permet également d'économiser de l'énergie. L'impact de ces économies d'énergie est plus important sur les charges plus importantes. L'utilisation de la technique du DVFS offre un avantage important : il diminue l'impact sur l'expérience utilisateur, comparé à la technique de la consolidation des serveurs.

Nous développerons les thèmes suivants dans de futurs travaux :

- Test du CDN avec différentes politiques de cache et différentes tailles de cache, et étude de leur impact sur l'économie d'énergie ainsi que sur l'expérience utilisateur et la qualité de la gestion de l'infrastructure.

- Prise en compte des différentes techniques d'économie d'énergie au niveau des réseaux afin de mieux connaître l'économie globale d'énergie dans un CDN.
- Analyse des variations géographiques de l'énergie et des dépenses énergétiques.
- Proposition de politiques de redirection du CDN en fonction des exigences du client.

Bibliography

[Anand 2006] H. Anand, C. Reardon, R. Subramaniyan and A.D. George. *Ethernet Adaptive Link Rate (ALR): Analysis of a MAC Handshake Protocol.* In Local Computer Networks, Proceedings 2006 31st IEEE Conference on, pages 533–534, 2006. (Cited on page 22.)

[Beloglazov 2011] Anton Beloglazov, Rajkumar Buyya, Young C. Lee, Albert Zomaya and Others. *A taxonomy and survey of energy-efficient data centers and cloud computing systems.* Advances in Computers, vol. 82, no. 2, pages 47–111, 2011. (Cited on page 18.)

[Benedict 2012] Shajulin Benedict. *Review: Energy-aware Performance Analysis Methodologies for HPC architectures-An Exploratory Study.* J. Netw. Comput. Appl., vol. 35, no. 6, pages 1709–1719, November 2012. (Cited on page 16.)

[Berl 2011] Andreas Berl and Hermann de Meer. *An energy consumption model for virtualized office environments.* Future Generation Computer Systems, vol. 27, no. 8, pages 1047 – 1055, 2011. (Cited on page 17.)

[Berral 2010] Josep Ll. Berral, Íñigo Goiri, Ramón Nou, Ferran Julià, Jordi Guitart, Ricard Gavaldà and Jordi Torres. *Towards Energy-aware Scheduling in Data Centers Using Machine Learning.* In Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking, e-Energy ’10, pages 215–224, New York, NY, USA, 2010. ACM. (Cited on pages 25 and 36.)

[Blackburn 2009] J. Blackburn and K. Christensen. *A Simulation Study of a New Green BitTorrent.* In Communications Workshops, 2009. ICC Workshops 2009. IEEE International Conference on, pages 1–6, 2009. (Cited on page 11.)

[Buttazzo 2002] GiorgioC. Buttazzo. *Scalable Applications for Energy-Aware Processors.* In Alberto Sangiovanni-Vincentelli and Joseph Sifakis, editeurs, Embedded Software, volume 2491 of *Lecture Notes in Computer Science*, pages 153–165. Springer Berlin Heidelberg, 2002. (Cited on page 22.)

[Carrera 2003] Enrique V. Carrera, Eduardo Pinheiro and Ricardo Bianchini. *Conserving Disk Energy in Network Servers.* In Proceedings of the 17th Annual International Conference on Supercomputing, ICS '03, pages 86–97, New York, NY, USA, 2003. ACM. (Cited on page 21.)

[Chang 2003] Fay Chang, KeithI. Farkas and Parthasarathy Ranganathan. *Energy-Driven Statistical Sampling: Detecting Software Hotspots.* In Babak Falsafi and T.N. Vijaykumar, editeurs, Power-Aware Computer Systems, volume 2325 of *Lecture Notes in Computer Science*, pages 110–129. Springer Berlin Heidelberg, 2003. (Cited on page 18.)

[Chase 2001] Jeffrey S. Chase, Darrell C. Anderson, Prachi N. Thakar, Amin M. Vahdat and Ronald P. Doyle. *Managing Energy and Server Resources in Hosting Centers.* SIGOPS Oper. Syst. Rev., vol. 35, no. 5, pages 103–116, October 2001. (Cited on pages 24 and 36.)

[Chen] H. Chen and S. Weisong. Power Measuring and Profiling:State-of-the-Art. (Cited on page 16.)

[Chen 2008] Gong Chen, Wenbo He, Jie Liu, Suman Nath, Leonidas Rigas, Lin Xiao and Feng Zhao. *Energy-Aware Server Provisioning and Load Dispatching for Connection-Intensive Internet Services.* In NSDI, volume 8, pages 337–350, 2008. (Cited on pages 25 and 36.)

[Chiaraviglio 2010] Luca Chiaraviglio and Ibrahim Matta. *GreenCoop: cooperative green routing with energy-efficient servers.* In Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking, e-Energy '10, pages 191–194, New York, NY, USA, 2010. ACM. (Cited on pages 7, 11 and 31.)

[Christensen 2004] Kenneth J. Christensen, Chamara Gunaratne, Bruce Nordman and Alan D. George. *The Next Frontier for Communications Networks: Power Management.* Comput. Commun., vol. 27, no. 18, pages 1758–1770, December 2004. (Cited on page 22.)

[Christensen 2009] Ken Christensen. *Green networks: Opportunities and challenges.* In LCN, page 13. IEEE, 2009. (Cited on page 6.)

[Comellas 2010] Josep Oriol Fitó Comellas, Inigo Goiri Presa and Jordi Guitart Fernández. *SLA-driven Elastic Cloud Hosting Provider.* In Proceedings of the 2010 18th Euromicro Conference on Parallel, Distributed and Network-based Processing, PDP '10, pages 111–118, Washington, DC, USA, 2010. IEEE Computer Society. (Cited on pages 25 and 36.)

[Contreras 2005] Gilberto Contreras and Margaret Martonosi. *Power Prediction for Intel XScale&#174; Processors Using Performance Monitoring Unit Events.* In Proceedings of the 2005 International Symposium on Low Power Electronics and Design, ISLPED '05, pages 221–226, New York, NY, USA, 2005. ACM. (Cited on page 18.)

[Da Costa 2009] G. Da Costa, J.-P. Gelas, Y. Georgiou, L. Lefevre, A.-C. Orgerie, J. Pierson, O. Richard and K. Sharma. *The GREEN-NET framework: Energy efficiency in large scale distributed systems.* In Parallel Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on, pages 1–8, 2009. (Cited on pages 7 and 18.)

[David 2011] Howard David, Chris Fallin, Eugene Gorbatov, Ulf R. Hanebutte and Onur Mutlu. *Memory Power Management via Dynamic Voltage/Frequency Scaling.* In Proceedings of the 8th ACM International Conference on Autonomic Computing, ICAC '11, pages 31–40, New York, NY, USA, 2011. ACM. (Cited on page 21.)

[Don Domingo 2010] Jack Reed Don Domingo Rüdiger Landmann. *Red Hat Enterprise Linux 6.5 Power Management Guide.* Rapport technique, 2010. (Cited on pages 20, 23 and 36.)

[Dovrolis 1999] Constantinos Dovrolis, Dimitrios Stiliadis and Parameswaran Ramanathan. *Proportional Differentiated Services: Delay Differentiation and Packet Scheduling.* SIGCOMM Comput. Commun. Rev., vol. 29, no. 4, pages 109–120, August 1999. (Cited on pages 25 and 36.)

[Economou 2006] Dimitris Economou, Suzanne Rivoire and Christos Kozyrakis. *Full-system power analysis and modeling for server environments.* In In Workshop on Modeling Benchmarking and Simulation (MOBS, 2006. (Cited on page 17.)

[Elnozahy 2003] E. N. Elnozahy, Michael Kistler and Ramakrishnan Rajamony. *Energy-efficient Server Clusters.* In Proceedings of the 2Nd International Conference on Power-aware Computer Systems, PACS'02, pages 179–197, Berlin, Heidelberg, 2003. Springer-Verlag. (Cited on pages 27 and 36.)

[Fan 2007] Xiaobo Fan, Wolf-Dietrich Weber and Luiz Andre Barroso. *Power provisioning for a warehouse-sized computer.* In ACM SIGARCH Computer Architecture News, volume 35, pages 13–23. ACM, 2007. (Cited on page 17.)

[Feldmann 2010] A. Feldmann, A. Gladisch, M. Kind, C. Lange, G. Smaragdakis and F. Westphal. *Energy trade-offs among content delivery architectures.* In Telecommunications Internet and Media Techno Economics (CTTE), 2010 9th Conference on, pages 1–6, 2010. (Cited on pages 11 and 32.)

[Flautner 2002] Krisztián Flautner, Steve Reinhardt and Trevor Mudge. *Automatic Performance Setting for Dynamic Voltage Scaling.* Wirel. Netw., vol. 8, no. 5, pages 507–520, September 2002. (Cited on page 19.)

[Flinn 2000] Jason Flinn and M. Satyanarayanan. *Energy-aware Adaptation for Mobile Applications.* SIGOPS Oper. Syst. Rev., vol. 34, no. 2, pages 13–14, April 2000. (Cited on page 19.)

[Gao 2012] Peter Xiang Gao, Andrew R. Curtis, Bernard Wong and Srinivasan Keshav. *It's Not Easy Being Green.* SIGCOMM Comput. Commun. Rev., vol. 42, no. 4, pages 211–222, August 2012. (Cited on page 34.)

[Goiri 2010] Inigo Goiri, Ferran Julia, Ramon Nou, Josep Ll. Berral, Jordi Guitart and Jordi Torres. *Energy-Aware Scheduling in Virtualized Datacenters.* In Proceedings of the 2010 IEEE International Conference on Cluster Computing, CLUSTER '10, pages 58–67, Washington, DC, USA, 2010. IEEE Computer Society. (Cited on pages 25 and 36.)

[Govil 1995] Kinshuk Govil, Edwin Chan and Hal Wasserman. *Comparing Algorithm for Dynamic Speed-setting of a Low-power CPU.* In Proceedings of the 1st Annual International Conference on Mobile Computing and Networking, MobiCom '95, pages 13–25, New York, NY, USA, 1995. ACM. (Cited on page 19.)

[Grunwald 2000] Dirk Grunwald, Charles B. Morrey III, Philip Levis, Michael Neufeld and Keith I. Farkas. *Policies for Dynamic Clock Scheduling.* In Proceedings of the 4th Conference on Symposium on Operating System Design & Implementation - Volume 4, OSDI'00, pages 6–6, Berkeley, CA, USA, 2000. USENIX Association. (Cited on pages 19, 20 and 36.)

[Guerout 2013] Tom Guerout, Georges Da Costa, Thierry Monteil, Rodrigo Neves Calheiros, Rajkumar Buyya and Mihai Alexandru. *Energy-aware simulation with DVFS.* Simulation Modelling Practice and Theory , Energy efficiency in Grids and Clouds, vol. 39, pages 76–91, 2013. (Cited on pages 21 and 36.)

[Gunaratne 2005] Chamara Gunaratne, Ken Christensen and Bruce Nordman. *Managing Energy Consumption Costs in Desktop PCs and LAN Switches with Proxying, Split TCP Connections, and Scaling of Link Speed.* Int. J. Netw. Manag., vol. 15, no. 5, pages 297–310, September 2005. (Cited on page 22.)

[Gunaratne 2006] C. Gunaratne, K. Christensen and S.W. Suen. *NGL02-2: Ethernet Adaptive Link Rate (ALR): Analysis of a Buffer Threshold Policy.* In Global Telecommunications Conference, 2006. GLOBECOM '06. IEEE, pages 1–6, 2006. (Cited on page 22.)

[Gupta 2003] Maruti Gupta and Suresh Singh. *Greening of the internet.* In Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications, SIGCOMM '03, pages 19–26, New York, NY, USA, 2003. ACM. (Cited on page 22.)

[Gurumurthi 2003] S. Gurumurthi, Anand Sivasubramaniam, M. Kandemir and H. Franke. *DRPM: dynamic speed control for power management in server class disks.* In Computer Architecture, 2003. Proceedings. 30th Annual International Symposium on, pages 169–179, June 2003. (Cited on page 21.)

[Gyarmati 2010] László Gyarmati and Tuan Anh Trinh. *How can architecture help to reduce energy consumption in data center networking?* In Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking, e-Energy '10, pages 183–186, New York, NY, USA, 2010. ACM. (Cited on page 7.)

[Heath 2005] Taliver Heath, Bruno Diniz, Enrique V. Carrera, Wagner Meira Jr. and Ricardo Bianchini. *Energy Conservation in Heterogeneous Server Clusters.* In Proceedings of the Tenth ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, PPoPP '05, pages 186–195, New York, NY, USA, 2005. ACM. (Cited on page 17.)

[HEEB 2000] Jay HEEB. *Next generation intel StrongARM technology overview.* Proc. IEICE Cool Chips III, April 2000, 2000. (Cited on pages 20 and 36.)

[Hewlett-Packard 1999] Microsoft Phoenix Toshiba Hewlett-Packard Intel. *Advanced Configuration and Power Interface*, 1999. (Cited on page 20.)

[Hlavacs 2009] H. Hlavacs, G. Da Costa and J. Pierson. *Energy Consumption of Residential and Professional Switches.* In Computational Science and Engineering, 2009. CSE '09. International Conference on, volume 1, pages 240–246, 2009. (Cited on page 7.)

[Intel Corporation 2000] Intel Corporation. *Mobile Pentium III Processor in BGA2 and Micro-PGA2 Packages.* Rapport technique, 2000. (Cited on page 20.)

[Jacobson 2009] Van Jacobson, Diana K. Smetters, James D. Thornton, Michael F. Plass, Nicholas H. Briggs and Rebecca L. Braynard. *Networking Named Content.* In Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies, CoNEXT '09, pages 1–12, New York, NY, USA, 2009. ACM. (Cited on page 29.)

[Jacobson 2012] Van Jacobson, Diana K. Smetters, James D. Thornton, Michael Plass, Nick Briggs and Rebecca Braynard. *Networking Named Content.* Commun. ACM, vol. 55, no. 1, pages 117–124, January 2012. (Cited on pages ix and 29.)

[Kamitsos 2010] Ioannis Kamitsos, Lachlan Andrew, Hongseok Kim and Mung Chiang. *Optimal Sleep Patterns for Serving Delay-tolerant Jobs.* In Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking, e-Energy '10, pages 31–40, New York, NY, USA, 2010. ACM. (Cited on pages 26 and 36.)

[Katz 2009] R. H. Katz. *Tech Titans Building Boom.* IEEE Spectr., vol. 46, no. 2, pages 40–54, February 2009. (Cited on pages 7 and 168.)

[Laoutaris 2005] Nikolaos Laoutaris, Vassilios Zissimopoulos and Ioannis Stavrakakis. *On the optimization of storage capacity allocation for content distribution.* Comput. Netw., vol. 47, no. 3, pages 409–428, February 2005. (Cited on page 41.)

[Lee 2010] Uichin Lee, Ivica Rimac and Volker Hilt. *Greening the internet with content-centric networking.* In Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking, e-Energy '10, pages 179–182, New York, NY, USA, 2010. ACM. (Cited on pages 11 and 29.)

[live stats 2014] Internet live stats. *Internet users*, 2014. (Cited on pages ix and 8.)

[Lorch 2001] Jacob R. Lorch and Alan Jay Smith. *Improving Dynamic Voltage Scaling Algorithms with PACE.* SIGMETRICS Perform. Eval. Rev., vol. 29, no. 1, pages 50–61, June 2001. (Cited on page 19.)

[Lu 2013] Tan Lu, Minghua Chen and L.L.H. Andrew. *Simple and Effective Dynamic Provisioning for Power-Proportional Data Centers.* Parallel and Distributed Systems, IEEE Transactions on, vol. 24, no. 6, pages 1161–1171, June 2013. (Cited on page 26.)

[Mandal 2011] U. Mandal, C. Lange, A. Gladisch, P. Chowdhury and B. Mukherjee. *Energy-efficient content distribution over telecom network infrastructure.* In Transparent Optical Networks (ICTON), 2011 13th International Conference on, pages 1–4, June 2011. (Cited on page 34.)

[Mathew 2012] V. Mathew, R.K. Sitaraman and P. Shenoy. *Energy-aware load balancing in content delivery networks.* In INFOCOM, 2012 Proceedings IEEE, pages 954–962, March 2012. (Cited on pages 32 and 36.)

[Mathew 2013] Vimal Mathew, Ramesh K. Sitaraman and Prashant J. Shenoy. *Energy-efficient content delivery networks using cluster shutdown.* In IGCC, pages 1–10, 2013. (Cited on pages 32 and 36.)

[Mochocki 2006] B. C. Mochocki, X. S. Hu and Gang Quan. *A Unified Approach to Variable Voltage Scheduling for Nonideal DVS Processors.* Trans. Comp.-Aided Des. Integ. Cir. Sys., vol. 23, no. 9, pages 1370–1377, November 2006. (Cited on page 19.)

[Mortazavi 2006] B. Mortazavi and G. Kesidis. *Model and simulation study of a peer-to-peer game with a reputation- based incentive mechanism.* In Information Theory and Applications Workshop, 2006, 2006. (Cited on page 48.)

[Nedevschi 2008] Sergiu Nedevschi, Sylvia Ratnasamy and Jitendra Padhye. *Hot Data Centers vs. Cool Peers.* In Proceedings of the 2008 Conference on Power Aware Computing and Systems, HotPower'08, pages 8–8, Berkeley, CA, USA, 2008. USENIX Association. (Cited on page 25.)

[Orgerie 2013] Anne-Cécile Orgerie, Marcos Dias de Assunção and Laurent Lefèvre. *A survey on techniques for improving the energy efficiency of large-scale* *distributed systems.* ACM Comput. Surv., vol. 46, no. 4, page 47, 2013. (Cited on pages 7 and 18.)

[Padmanabhan 2000] Venkata N. Padmanabhan and Lili Qiu. *The content and access dynamics of a busy Web site: findings and implications.* SIGCOMM Comput. Commun. Rev., vol. 30, no. 4, pages 111–123, August 2000. (Cited on page 39.)

[Pallis 2006] George Pallis and Athena Vakali. *Insight and perspectives for content delivery networks.* Commun. ACM, vol. 49, no. 1, pages 101–106, January 2006. (Cited on pages 7 and 168.)

[Pering 1998] Trevor Pering, Tom Burd and Robert Brodersen. *The Simulation and Evaluation of Dynamic Voltage Scaling Algorithms.* In Proceedings of the 1998 International Symposium on Low Power Electronics and Design, ISLPED '98, pages 76–81, New York, NY, USA, 1998. ACM. (Cited on pages 20 and 36.)

[Perino 2011] Diego Perino and Matteo Varvello. *A Reality Check for Content Centric Networking.* In Proceedings of the ACM SIGCOMM Workshop on Information-centric Networking, ICN '11, pages 44–49, New York, NY, USA, 2011. ACM. (Cited on page 30.)

[Pettis 2004] Nathaniel Pettis, Le Cai and Yung-Hsiang Lu. *Dynamic Power Management for Streaming Data.* In Proceedings of the 2004 International Symposium on Low Power Electronics and Design, ISLPED '04, pages 62–65, New York, NY, USA, 2004. ACM. (Cited on page 19.)

[Pierson 2011] Jean-Marc Pierson and Henri Casanova. *On the Utility of DVFS for Power-Aware Job Placement in Clusters.* In Euro-Par (1), pages 255–266, 2011. (Cited on page 78.)

[Pierson 2013] Jean-Marc Pierson, Georges Da Costa and Lars Dittmann, editeurs. Energy efficiency in large scale distributed systems - cost ic0804 european conference, ee-lsds 2013, vienna, austria, april 22-24, 2013, revised selected papers, volume 8046 of *Lecture Notes in Computer Science*. Springer, 2013. (Cited on page 7.)

[Pinheiro 2001] Eduardo Pinheiro, Ricardo Bianchini, Enrique V. Carrera and Taliver Heath. *Load Balancing and Unbalancing for Power and Performance in Cluster-Based Systems*, 2001. (Cited on pages 26, 27 and 36.)

[Pinheiro 2004] Eduardo Pinheiro and Ricardo Bianchini. *Energy Conservation Techniques for Disk Array-based Servers*. In Proceedings of the 18th Annual International Conference on Supercomputing, ICS '04, pages 68–78, New York, NY, USA, 2004. ACM. (Cited on page 21.)

[Pouwelse 2001] Johan Pouwelse, Koen Langendoen and Henk Sips. *Dynamic Voltage Scaling on a Low-power Microprocessor*. In Proceedings of the 7th Annual International Conference on Mobile Computing and Networking, MobiCom '01, pages 251–259, New York, NY, USA, 2001. ACM. (Cited on page 19.)

[Qureshi 2009] Asfandyar Qureshi, Rick Weber, Hari Balakrishnan, John Guttag and Bruce Maggs. *Cutting the electric bill for internet-scale systems*. SIGCOMM Comput. Commun. Rev., vol. 39, no. 4, pages 123–134, August 2009. (Cited on pages ix, 7, 9 and 168.)

[rep 2014] *akamai's state of the internet*. Rapport technique, sep 2014. (Cited on page 39.)

[Rivoire 2008] Suzanne Rivoire, Parthasarathy Ranganathan and Christos Kozyrakis. *A Comparison of High-level Full-system Power Models*. In Proceedings of the 2008 Conference on Power Aware Computing and Systems, HotPower'08, pages 3–3, Berkeley, CA, USA, 2008. USENIX Association. (Cited on page 17.)

[Seo 2009] Chiyoung Seo, George Edwards, Daniel Popescu, Sam Malek and Nenad Medvidovic. *A Framework for Estimating the Energy Consumption Induced by a Distributed System's Architectural Style*. In Proceedings of the 8th

International Workshop on Specification and Verification of Component-based Systems, SAVCBS '09, pages 27–34, New York, NY, USA, 2009. ACM. (Cited on page 17.)

[Sharma 2003] Vivek Sharma, Arun Thomas, Tarek Abdelzaher, Kevin Skadron and Zhijian Lu. *Power-aware QoS Management in Web Servers.* In Proceedings of the 24th IEEE International Real-Time Systems Symposium, RTSS '03, pages 63–, Washington, DC, USA, 2003. IEEE Computer Society. (Cited on pages 24 and 36.)

[Sharma 2005] Ratnesh K. Sharma, Cullen E. Bash, Chandrakant D. Patel, Richard J. Friedrich and Jeffrey S. Chase. *Balance of Power: Dynamic Thermal Management for Internet Data Centers.* IEEE Internet Computing, vol. 9, no. 1, pages 42–49, 2005. (Cited on pages 25 and 36.)

[Shekar 2010] Venkateswaran Shekar and Baback Izadi. *Energy Aware Scheduling for DAG Structured Applications on Heterogeneous and DVS Enabled Processors.* In Proceedings of the International Conference on Green Computing, GREENCOMP '10, pages 495–502, Washington, DC, USA, 2010. IEEE Computer Society. (Cited on pages 22 and 36.)

[Shuja 2012] Junaid Shuja, SajjadA. Madani, Kashif Bilal, Khizar Hayat, SameeU. Khan and Shahzad Sarwar. *Energy-efficient data centers.* Computing, vol. 94, no. 12, pages 973–994, 2012. (Cited on page 18.)

[Sitaraman 2014] Ramesh K Sitaraman, Mangesh Kasbekar, Woody Lichtenstein and Manish Jain. *Overlay Networks: An Akamai Perspective.* In Advanced Content Delivery, Streaming, and Cloud Services. John Wiley & Sons, 2014. (Cited on pages 7 and 9.)

[Stamos 2006] K. Stamos, G. Pallis, C. Thomos and A. Vakali. *A similarity based approach for integrated Web caching and content replication in CDNs.* In Database Engineering and Applications Symposium, 2006. IDEAS '06. 10th International, pages 239–242, 2006. (Cited on page 41.)

[Stamos 2009] Konstantinos Stamos, George Pallis, Athena Vakali and Marios D. Dikaiakos. *Evaluating the utility of content delivery networks.* In Proceedings of the 4th edition of the UPGRADE-CN workshop on Use of P2P, GRID and agents for the development of content networks, UPGRADE-CN ’09, pages 11–20, New York, NY, USA, 2009. ACM. (Cited on pages 11, 42, 47, 160 and 168.)

[Stamos 2010] Konstantinos Stamos, George Pallis, Athena Vakali, Dimitrios Katsaros, Antonis Sidiropoulos and Yannis Manolopoulos. *CDNsim: A simulation tool for content distribution networks.* ACM Trans. Model. Comput. Simul., vol. 20, no. 2, pages 10:1–10:40, May 2010. (Cited on pages 38 and 170.)

[The Climate Group 2008] The Climate Group. *SMART 2020: Enabling the low carbon economy in the information age.* Rapport technique, 2008. (Cited on pages 4 and 6.)

[Trobec 2013] R. Trobec, M. Depolli, K. Skala and T. Lipic. *Energy efficiency in large-scale distributed computing systems.* In Information Communication Technology Electronics Microelectronics (MIPRO), 2013 36th International Convention on, pages 253–257, May 2013. (Cited on page 16.)

[ul Islam 2011] Saif ul Islam, Konstantinos Stamos, Jean-Marc Pierson and Athena Vakali. *Utilization-aware redirection policy in CDN: a case for energy conservation.* In Proceedings of the First international conference on Information and communication on technology for the fight against global warming, ICT-GLOW’11, pages 180–187, Berlin, Heidelberg, 2011. Springer-Verlag. (Cited on pages 35 and 36.)

[ul Islam 2012] Saif ul Islam and Jean-Marc Pierson. *Evaluating Energy Consumption in CDN Servers.* In Proceedings of the Second International Conference on ICT As Key Technology Against Global Warming, ICT-GLOW’12, pages 64–78, Berlin, Heidelberg, 2012. Springer-Verlag. (Cited on pages 35 and 36.)

[Urgaonkar 2008] Bhuvan Urgaonkar, Prashant Shenoy, Abhishek Chandra, Pawan Goyal and Timothy Wood. *Agile Dynamic Provisioning of Multi-tier Internet Applications.* ACM Trans. Auton. Adapt. Syst., vol. 3, no. 1, pages 1:1–1:39, March 2008. (Cited on pages 25 and 36.)

[Valancius 2009] Vytautas Valancius, Nikolaos Laoutaris, Laurent Massoulié, Christophe Diot and Pablo Rodriguez. *Greening the Internet with Nano Data Centers.* In Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies, CoNEXT '09, pages 37–48, New York, NY, USA, 2009. ACM. (Cited on pages ix, 30 and 31.)

[Vasić 2010] Nedeljko Vasić and Dejan Kostić. *Energy-aware traffic engineering.* In Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking, e-Energy '10, pages 169–178, New York, NY, USA, 2010. ACM. (Cited on page 7.)

[Venkatachalam 2005a] Vasanth Venkatachalam and Michael Franz. *Power Reduction Techniques for Microprocessor Systems.* ACM Comput. Surv., vol. 37, no. 3, pages 195–237, September 2005. (Cited on pages 19 and 22.)

[Venkatachalam 2005b] Vasanth Venkatachalam and Michael Franz. *Power Reduction Techniques for Microprocessor Systems.* ACM Comput. Surv., vol. 37, no. 3, pages 195–237, September 2005. (Cited on page 19.)

[Vereecken 2010] Willem Vereecken, Lien Deboosere, Pieter Simoens, Brecht Vermeulen, Didier Colle, Chris Develder, Mario Pickavet, Bart Dhoedt and Piet Demeester. *Power efficiency of thin clients.* European Transactions on Telecommunications, vol. 21, no. 6, pages 479–490, 2010. (Cited on page 17.)

[Weiser 1994] Mark Weiser, Brent Welch, Alan Demers and Scott Shenker. *Scheduling for Reduced CPU Energy.* In Proceedings of the 1st USENIX Conference on Operating Systems Design and Implementation, OSDI '94, Berkeley, CA, USA, 1994. USENIX Association. (Cited on pages 19 and 36.)

[Xu 2010] Ning Xu, Jin Yang, Mike Needham, Dragan Boscovic and Faramak Vakil. *Toward the Green Video CDN.* In Proceedings of the 2010 IEEE/ACM Int'L Conference on Green Computing and Communications & Int'L Conference on Cyber, Physical and Social Computing, GREENCOM-CPSCOM '10, pages 430–435, Washington, DC, USA, 2010. IEEE Computer Society. (Cited on pages 33 and 36.)

[Yao 1995] F. Yao, A. Demers and S. Shenker. *A scheduling model for reduced CPU energy.* In Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on, pages 374–382, Oct 1995. (Cited on pages 20 and 36.)

[Zhu 2004] D. Zhu, R. Melhem and D. Mosse. *The effects of energy management on reliability in real-time embedded systems.* In Computer Aided Design, 2004. ICCAD-2004. IEEE/ACM International Conference on, pages 35–40, Nov 2004. (Cited on page 22.)