# Efficient Resource Energy Management in Data Centers for Cloud Computing

Project Report submitted in partial fulfilment of the requirement for the degree of

Bachelor of Technology.

in

**Computer Science Engineering**

under the Supervision of

**Dr. Amit Kumar Singh**

By

Anushtha Sharma (121217)

Shubham Kumar Singh (121314)

To

**Department of Computer Science & Engineering,**

**Jaypee University of Information Technology Waknaghat, Solan – 173234, Himachal Pradesh India**

**JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY**
(Established under the Act 14 of Legislative Assembly of Himachal Pradesh)
Waknaghat, P.O. Domehar Bani. Teh. Kandaghat, Distt. Solan- 173234(H.P)
Phone: 01792-245367, 245368,245369

Fax-01792-245362

This is to certify that project report entitled "**Efficient Resource Energy Management in Data Centers for Cloud Computing**", submitted by Anushtha Sharma and Shubham Kumar Singh in partial fulfillment for the award of degree of Bachelor of Technology in Computer Science Engineering to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree.

**Date:** 31 May, 2016

**Supervisor's Name: Dr. Amit Kumar Singh**

**JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY**
(Established under the Act 14 of Legislative Assembly of Himachal Pradesh)
Waknaghat, P.O. Domehar Bani. Teh. Kandaghat, Distt. Solan- 173234(H.P)
Phone:01792-245367, 245368, 245369 Fax-01792-245362

I hereby declare that the work reported in the B. Tech thesis entitled "**Efficient Resource Energy Management in Data Centers for Cloud Computing"** submitted by "**Ms. Anushtha Sharma" and "Mr. Shubham Kumar Singh"** at Jaypee University Of Information Technology, Waknaghat is an authentic record of our work carried out under the supervision of **Dr. Amit Kumar Singh**. This work has not been submitted partially or wholly to any other university or institution for the award of this or any other degree.

Anushtha Sharma (121217) ASharma
Shubham Kumar Singh (121314) Shubham Kumar Singh

# Acknowledgement

We would like to express our gratitude and appreciation to all those who gave us the possibility to complete this report. A special thanks to our final year project coordinator, Dr. Yashwant Singh, whose help, stimulating suggestions and encouragement, helped us to coordinate our project especially in writing this report.

We would also acknowledge the support of Luxembourg FNR in the GreenIT project and the European Research Consortium for informatics and Mathematics.

Last but not least, many thanks go to the head of the project, Dr. Amit Kumar Singh who have given her full effort in guiding the team in achieving the goal as well as her encouragement to maintain our progress in track. We would to appreciate the guidance given by other supervisor as well as the panels especially in our project presentation that has improved our presentation skills by their comment and tips.

Date: 31 - May - 2016

Name of the students:

Anushtha Sharma (121217)

Shubham Kumar Singh (121314)

# Table of Content

Title Page............................................................................. (1)

Certificate ............................................................................ (2)

Acknowledgement ................................................................... (4)

Table of Content ..................................................................... (5)

List of Figures ........................................................................ (6)

List of Tables ........................................................................ (7)

Abstract ................................................................................ (8)

1. Chapter-1 Cloud Computing: An Introduction ........................... (9)

2. Chapter-2 Literature Survey ..................................................... (18)

3. Chapter-3 System Development .................................................. (23)

4. Chapter-4 Experimental Results and Performance Analysis ....... (39)

5. Chapter-5 Conclusions and Future Directions........................... (51)

References.............................................................................. (52)

## List of figure

Fig 1.1Two-tier data center architecture ........................................................... (16)

Fig 1.2 Three-tier data center architecture ........................................................... (17)

Fig 3.1. Result Simulation in GreenCloud ........................................................... (26)

Fig 3.2. Pie chart presentation of simulation ........................................................... (27)

Fig 3.3 Example of a server with its components ........................................................... (28)

Fig 3.4. Resource allocation on a node with heterogeneous resources ........................................ (30)

Fig 3.5 DENS algorithm ........................................................... (35)

Fig 4.1 Small heterogeneous topology – relative results ........................................................... (50)

## List of Tables

Table 2.1 Paper Review on Efficient Energy Management for Cloud Computing ................ (18)

Table 2.2 Paper Review on Efficient Energy Management for Cloud Computing ................ (19)

Table 3.1 How GreenCloud is better ........................................................................ (25)

Table 4.1 Small Homogeneous Topology- Result ............................................... (49)

# ABSTRACT

Two scheduling algorithm random and HEROS (Heterogeneous efficient resource optimization scheduling) are compared to find out which one is effective in computing efficient energy. Using modified three-tier high speed architecture framework and principle, efficient resource energy management in data center for cloud computing is computed.

Cloud is grid of clusters, where grid of cluster means set of connected components collected to achieve a common goal. Cloud computing has appeared as the basic necessity for IT task. Data center accommodating cloud computing faces difficulties like high computational expenses, worldwide environment changes and high carbon footprints etc. Following problems can be resolved using green cloud techniques. This paper presents outline of new metrics able to measure performance and energy efficiency of cloud computing.

# CHAPTER – 1

# Cloud Computing: An Introduction

Clouds are a large pool of easily usable and accessible virtualized resources such as hardware, development platform and services. Cloud is an internetwork of different types of server which share resources. Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort.

Cloud computing is emerging as a model that use "**everything as a service**" and which in turn provided as cloud services. Virtualized physical resource, virtualized infrastructure, virtualized middleware platforms and business applications are being provided and consumed as service in the cloud. For example, a business solution model is either being built by using cloud service or being provided as a cloud service. The cloud service has stack of services which is arranged from top to bottom on the three layers that are, Hardware, system and application layer. Each component in this stack provides different types of service to cloud. These cloud computing services has much better than the traditional service provisions in context of reduced upfront investment, expected performance, high availability, infinite scalability and tremendous fault tolerance capability.

A cloud may be public, private, community or hybrid

**1.1 Public cloud:** Public clouds are made available to the general public by a service provider who hosts the cloud infrastructure. Generally, public cloud providers like Amazon AWS, Microsoft and Google own and operate the infrastructure and offer access over the Internet. With this model, customers have no visibility or control over where the infrastructure is located. It is important to note that all customers on public clouds share the same infrastructure pool with limited configuration, security protections and availability variances.

A public cloud is the obvious choice when:

- Your standardized workload for applications is used by lots of people, such as e-mail.
- You need to test and develop application code.
- You need incremental capacity (the ability to add compute resources for peak times).

- You're doing collaboration projects.

**1.2 Private cloud:** It is cloud infrastructure dedicated to a particular organization. Private clouds allow businesses to host applications in the cloud, while addressing concerns regarding data security and control, which is often lacking in a public cloud environment. It is not shared with other organizations, whether managed internally or by a third-party, and it can be hosted internally or externally.

There are two variations of private clouds:

1. On-Premise Private Cloud: This type of cloud is hosted within an organization's own facility. A businesses IT department would incur the capital and operational costs for the physical resources with this model. On-Premise Private Clouds are best used for applications that require complete control and configurability of the infrastructure and security.
2. Externally Hosted Private Cloud: Externally hosted private clouds are also exclusively used by one organization, but are hosted by a third party specializing in cloud infrastructure. The service provider facilitates an exclusive cloud environment with full guarantee of privacy. This format is recommended for organizations that prefer not to use a public cloud infrastructure due to the risks associated with the sharing of physical resources.

When is a Private Cloud for you?

- You need data sovereignty but want cloud efficiencies
- You want consistency across services
- You have more server capacity than your organization can use
- Your data center must become more efficient
- You want to provide private cloud services

**1.3 Hybrid cloud:** it is composition of two or more clouds (private, community or public) that remain unique entities but are bound together offering the advantages of multiple deployment models. In a hybrid cloud, you can leverage third party cloud providers in either a full or partial manner; increasing the flexibility of computing. Augmenting a traditional private cloud with the resources of a public cloud can be used to manage any unexpected surges in workload.

Hybrid cloud architecture requires both on-premise resources and off-site server based cloud infrastructure. By spreading things out over a hybrid cloud, you keep each aspect of your business in the most efficient environment possible. The downside is that you have to keep track of multiple cloud security platforms and ensure that all aspects of your business can communicate with each other.

Here are a couple of situations where a hybrid environment is best:

- Your company wants to use a SaaS application but is concerned about security.
- Your company offers services that are tailored for different vertical markets. You can use a public cloud to interact with the clients but keep their data secured within a private cloud.
- You can provide public cloud to your customers while using a private cloud for internal IT.

## 1.4 Types of cloud computing

Cloud computing is providing developers and IT departments with the ability to focus on what matters most and avoid undifferentiated work like procurement, maintenance, and capacity planning. As cloud computing has grown in popularity, several different models and deployment strategies have emerged to help meet specific needs of different users. Basically three types of cloud computing Software-as-a-service (SaaS),

Platform-as-a-Service (PaaS), Infrastructure-as-a-Service (IaaS).

**1.4.1 Software as a Service (SaaS):** Software as a Service provides you with a completed product that is run and managed by the service provider. In most cases, people referring to Software as a Service are referring to end-user applications. With a SaaS offering you do not have to think about how the service is maintained or how the underlying infrastructure is managed; you only need to think about how you will use that particular piece software. A common example of a SaaS application is web-based email where you can send and receive email without having to manage feature additions to the email product or maintaining the servers and operating systems that the email program is running on.

#### 1.4.1.1 Benefits of SaaS

- You can sign up and rapidly start using innovative business apps

- Apps and data are accessible from any connected computer
- No data is lost if your computer breaks, as data is in the cloud
- The service is able to dynamically scale to usage needs

**1.4.2 Platform as a Service (PaaS):** Platforms as a service remove the need for organizations to manage the underlying infrastructure (usually hardware and operating systems) and allow you to focus on the deployment and management of your applications. This helps you be more efficient as you don't need to worry about resource procurement, capacity planning, software maintenance, patching, or any of the other undifferentiated heavy lifting involved in running your application.

#### 1.4.2.1 Benefits of PaaS

- Develop applications and get to market faster
- Deploy new web applications to the cloud in minutes
- Reduce complexity with middleware as a service

**1.4.3 Infrastructure as a Service (IaaS):** Infrastructure as a Service, sometimes abbreviated as IaaS, contains the basic building blocks for cloud IT and typically provide access to networking features, computers (virtual or on dedicated hardware), and data storage space. Infrastructure as a Service provides you with the highest level of flexibility and management control over your IT resources and is most similar to existing IT resources that many IT departments and developers are familiar with today.

#### 1.4.3.1 Benefits of IaaS

- No need to invest in your own hardware
- Infrastructure scales on demand to support dynamic workloads
- Flexible, innovative services available on demand

Each type of cloud service, and deployment method, provides you with different levels of control, flexibility, and management. Understanding the differences between Infrastructure as a Service, Platform as a Service, and Software as a Service, as well as what deployment strategies you can use, can help you decide what set of services is right for your needs.

**1.5 Internet Data Center:** Internet Data Center (IDC) is a common form to host cloud computing. An IDC usually deploys hundreds or thousands of blade servers, densely packed to maximize the space utilization. Running services in consolidated servers in IDCs provides customers an alternative to running their software or operating their computer services in-house. The major benefits of IDCs include the usage of economies of scale to amortize the cost of ownership and the cost of system maintenance over a large number of machines. With the rapid growth of IDCs in both quantity and scale, the energy consumed by IDCs, directly related to the number of hosted servers and their workload has been increased. The rated power consumptions of servers have increased by 10 times over the past ten years. This surging demand calls for the urgent need of designing and deployment of energy-efficient Internet data centers. A modern state-of-the-art data center has three main components-data storage, servers, and a local area network (LAN). The data center connects to the rest of the network through a gateway router. The power consumption data for each server was obtained by first calculating the maximum power using HP's power calculator, then following the convention that average power use for midrange/high-end servers is 66% of maximum power. In the following, I outline the functionality of this equipment as well as some of the efficiency improvements in cloud computing data centers over traditional data centers. Long-term storage of data in a data center is provided by hard disk arrays, together with associated equipment.

**1.6 Green cloud Computing:** Even though there is a great concern in the community that Cloud computing can result in higher energy usage by the datacenters, the Cloud computing has a green lining. There are several technologies and concepts employed by Cloud providers to achieve better utilization and efficiency than traditional computing. Therefore, comparatively lower carbon emission is expected in Cloud computing due to highly energy efficient infrastructure and reduction in the IT infrastructure itself by multi-tenancy. improvement in energy efficiency of Cloud providers by leveraging the economies of scale associated with large number of organizations sharing the same infrastructure. Virtualization is the process of presenting a logical grouping or subset of computing resources so that they can be accessed in ways that give benefits over the original configuration. By consolidation of underutilized servers in the form of multiple virtual machines sharing same physical server at higher utilization, companies can gain high savings in the form of space, management, and energy.

According to market research conducted by Pike Research, the wide-spread adoption of cloud computing could lead to a potential 38% reduction in worldwide data center energy expenditures by 2020. The savings would be primarily achieved by consolidating data centers and maximizing power usage efficiency (PUE), improving recycling efforts, lowering carbon and gas emissions and minimizing water usage in cooling the remaining centers. Because so much of a data center's energy expenditures support data storage, the Storage Networking Industry Association (SNIA) has promoted new technologies and architectures to help save energy. Advances in SAS drive technologies, automated data deduplication, storage virtualization and storage convergence reduce the amount of physical storage a data center requires, which helps decrease its carbon footprint and lower operating expenditures (OPEX) and capital expenditures (CAPEX). Because the color green is also associated with paper money, the label green cloud is sometimes used to describe the cost-efficiency of a cloud computing initiative.

Recently, cloud computing services have become increasingly popular due to the evolving data centers and parallel computing paradigms. The operation of large geographically distributed data centers requires considerable amount of energy that accounts for a large slice of the total operational costs for cloud data centers for up to 10% of the current data center operational expenses. High power consumption generates heat and requires an accompanying cooling system that costs per year for classical data centers which drastically decreases hardware reliability and may potentially violate the Service Level Agreement with the customers. The first power saving solutions focused on making the data center hardware components power efficient. Technologies, such as Dynamic Voltage and Frequency Scaling (DVFS), and Dynamic Power Management (DPM) were extensively studied and widely deployed but their power down and power-off methodologies, the efficiency of these techniques is at best limited.

**1.7 Dynamic Voltage Scaling (DVS):** Is a power management technique in computer architecture, where the voltage used in a component is increased or decreased, depending upon circumstances. Dynamic voltage scaling to increase voltage is known as overvolting; dynamic voltage scaling to decrease voltage is known as undervolting. Undervolting is done in order to conserve power, particularly in laptops and other mobile devices, where energy comes from a battery and thus is limited. Overvolting is done in order to increase computer performance, or in rare cases, to increase reliability.

The term "overvolting" is also used to refer to increasing static operating voltage of computer components to allow operation at higher speed (overclocking).

**1.8 Dynamic Frequency Scaling (DFS):** is a technique in computer architecture whereby the frequency of a microprocessor can be automatically adjusted "on the fly," either to conserve power or to reduce the amount of heat generated by the chip. Dynamic frequency scaling is commonly used in laptops and other mobile devices, where energy comes from a battery and thus is limited. It is also used in quiet computing settings and to decrease energy and cooling costs for lightly loaded machines. Less heat output, in turn, allows the system cooling fans to be throttled down or turned off, reducing noise levels and further decreasing power consumption. It is also used for reducing heat in insufficiently cooled systems when the temperature reaches a certain threshold, such as in poorly cooled overclocked systems.

**1.9 Data Center Architecture:** The pool of servers in today's data centers over comes 100,000 hosts with around 70% of all communications performed internally. This creates a challenge in the design of interconnected network architecture and the set of communication protocols. Given the scale of a data center, the conventional hierarchical network infrastructure often becomes a bottleneck due to the physical and cost-driven limitations of the used networking equipment. Specifically, the availability of 10 Gigabit Ethernet (GE) components and their price defined the way the data center architectures evolved. The 10 GE transceivers are still too expensive and probably offer more capacity than needed for connecting individual servers. However, their penetration level keeps increasing in the backbone networks, metro area networks, and data centers.

**1.9.1 Two-tier data center architectures:** They follow the structure depicted in Fig. 1. In this example, computing Servers (S) physically arranged into racks form the tier-one network. At the tier-two network, Layer-3 (L3) switches provide full mesh connectivity using 10 GE links. The Equal Cost Multi-Path (ECMP) routing is used as a load balancing technology to optimize data flows across multiple paths. It applies load balancing on TCP and UDP packets on a per-flow basis using express hashing techniques requiring almost no processing from a switch's CPU. Other traffic, such as ICMP, is typically not processed by ECMP and forwarded on a single predefined path. The two-tier architecture worked well for early data centers with a limited number of computing servers. Depending on the type of switches used in the access network, the two-tier data centers may support up to 5500 nodes. The number of core switches and capacity of the core links defines the maximum network bandwidth allocated per computing server.

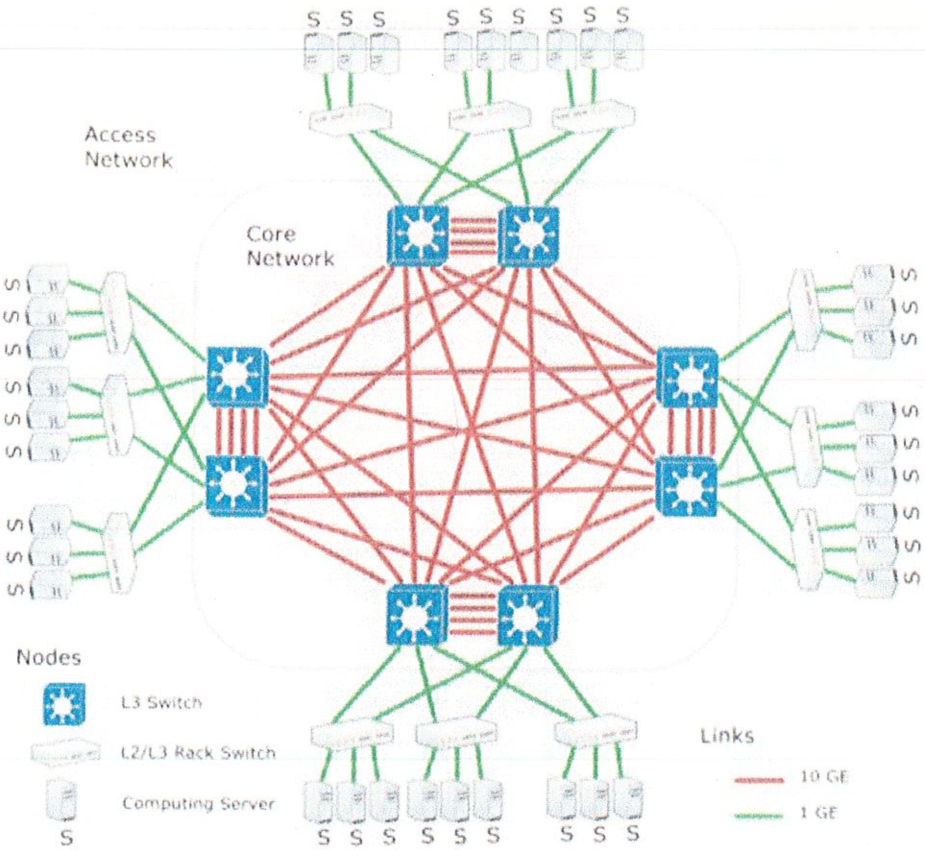


Fig 1.1 Two-tier data center architecture

**1.9.2 Three-tier data center architectures:** They are the most common nowadays. They include: (a) access, (b) aggregation, and (c) core layers as presented in Fig. 2. The availability of the aggregation layer facilitates the increase in the number of server nodes (to over 10,000 servers) while keeping inexpensive Layer-2 (L2) switches in the access network, which provides a loop-free topology. Because the maximum number of ECMP paths allowed is eight, a typical three tier architecture consists of eight core switches (only four are presented in Fig. 2). Such architecture implements an 8-way ECMP that includes 10 GE Line Aggregation Groups (LAGs), *which allow a network client to address several links and network ports with* a single MAC address. While the LAG technology is an excellent methodology to increase link capacities, its usage has several fundamental drawbacks that limit network flexibility and performance. LAGs make it difficult to plan the capacity for large flows and make it unpredictable in case of a link failure. In addition, several types of traffic patterns, such as ICMP and broadcast are usually routed through a single link only. Moreover, full mesh connectivity at the core of the network requires considerable amount of cablings. The aforementioned disadvantages have redirected the design choices for the next generation data centers to consider: (a) increasing the capacity of the core and (b) accessing parts of the network with beyond 10 GE links.

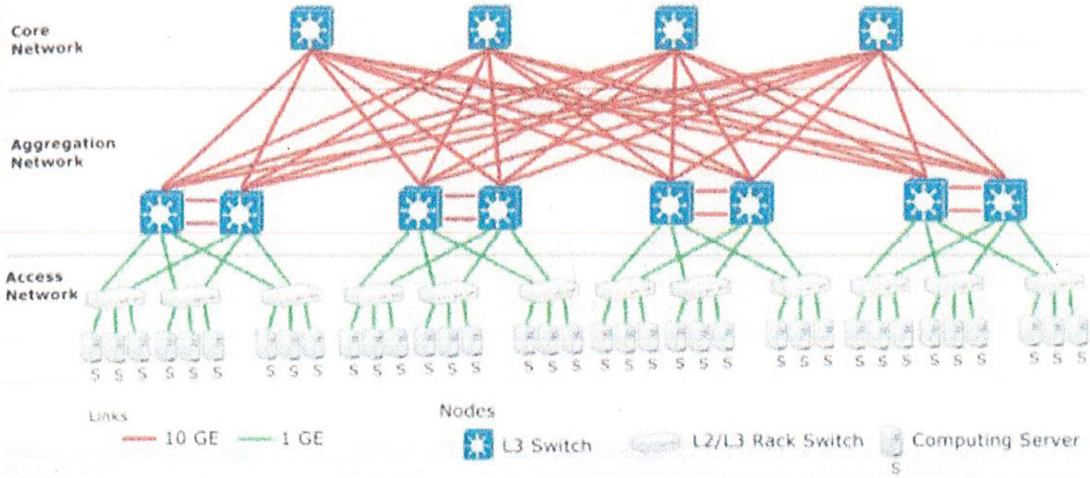


Fig 1.2 Three-tier data center architecture

# CHAPTER 2

## Literature Survey

Table 2.1 Paper Review on Efficient Energy Management for Cloud Computing

| SN. | Author, Year | Technologies Used | Performance Matrix (Maximum Power Saving) | Result |
|---|---|---|---|---|
| 1. | Rajkumar Buyya, 2010 | Virtualized Cloud Data Centers | 66% | % in comparison to a system that applies only DVFS technique |
| 2. | Zhiming Shen, 2011 | Elastic resource scaling | 8-10% | % total energy consumption |
| 3. | Beant Singh Gill, 2013 | Analysis using Green Cloud Simulator | 65% | % in comparison to a system that applies No saving technique |

Table 2.2 Paper Review on Efficient Energy Management for Cloud Computing

| SN. | Author, Year | Technologies Used | Performance Matrix (Maximum Power Saving) | Result |
|---|---|---|---|---|
| 4. | Congfeng Jiang, 2010 | Dynamic Voltage and Frequency Scaling (DVS/DFS) | 22% | energy savings offered by implementing per-core DVS/ DFS |
| 5. | Yuvapriya Ponnusamy, 2012 | DVFS, Migration and Dynamic Shutdown | 50% | save vast amounts of energy while minimally impacting performance |
| 6. | Anusuya Rajan, 2014 | Green Cloud: A Pocket-Level Simulator | 35-37% | % in comparison to a system that compared with the existing work |

Buyya et al. [1] proposed a technical paper on Application of Green Cloud Computing for Efficient Resource Energy Management in Data Center. Virtualization technology allows one to create several Virtual Machines (VMs) on a physical server and, therefore, reduces amount of hardware in use and improves the utilization of resources. The simulated data center consists of 100 heterogeneous physical nodes. Each node is modelled to have one CPU core with performance equivalent to 1000, 2000 or 3000 MIPS, 8 Gb of RAM and 1 TB of storage. One of the heuristics leads to significant reduction of the energy consumption by a Cloud data center – by 66% in comparison to a system that applies only DVFS technique but does not adapt allocation of VMs in run-time. Among the benefits of virtualization are improved fault and performance isolation between applications sharing the same resource. Some advantages associated with these technologies are location independence –VMs can be moved to a place where energy is cheaper and scaling up and down resource usage can be adjusted to current requirements.

Shen et al. [2] proposed a technical paper on Elastic resource scaling lets cloud systems meet application service level objectives (SLOs)with minimum resource provisioning costs. In this paper, we present CloudScale, a system that automates finegrained elastic resource scaling for multi-tenant cloud computing infrastructures. CloudScale employs online resource demand prediction and prediction error handling to achieve adaptive resource allocation without assuming any prior knowledge about the applications running inside the cloud. CloudScale can resolve scaling conflicts between applications using migration, and integrates dynamic CPU voltage/frequency scaling to achieve energy savings with minimal effect on application SLOs. We have implemented CloudScale on top of Xen and conducted extensive experiments using a set of CPU and memory intensive applications (RUBiS, Hadoop, IBM System S). The results show that CloudScale can achieve significantly higher SLO conformance than other alternatives withlowresource and energy cost. CloudScale isnon-intrusive and light-weight, and imposes negligible overhead (< 2% CPU in Domain 0) to the virtualized computing cluster

Gill et al [3] proposed a technical paper on Application of Green Cloud Computing for Efficient Resource Energy Management in Data Centers for two tier, three tier, and three-tier high-speed data center architectures which demonstrate the effectiveness of the simulator in utilizing different power management schema, such as voltage scaling, frequency scaling, and dynamic shutdown. On average, the data center consumption is around 432 kWh during an hour of the runtime. The processing servers share around 70% of total data center energy consumption, while the communicational links and switches account for the rest 30%. Furthermore, the consumption of switches breaks with 17% allocated for core switches, 34% for aggregation switches, and 50% for the access switches. It means that after computing servers lowering the power consumption of access switches will have the highest impact. The core and aggregation switches together account for 15% of total energy consumption.

| Parameters | No energy saving | DVFS | DNS | DVFS+DNS |
|---|---|---|---|---|
| Data Center | 503 | 96 | 37 | 35 |

| | | | | |
|---|---|---|---|---|
| Servers | 351 | 97 | 39 | 37 |
| Switches | 152 | 95 | 32 | 31 |
| Energy Cost | $441 | $435 | $163 | $157 |

The DVFS scheme alone reduces power consumption to only 96% from the nominal level. The most effective results are obtained by DNS scheme. It is equally effective for both servers and switches as the most of their energy consumed shows no dependency on the operating frequency. GreenCloud is designed to capture details of the energy consumed by data center components as well as packet-level communication patterns between them.

Jiang et al. [4] proposed a technical paper on Application of Green Cloud Computing for Efficient Resource Energy Management in Data Centers. “Dynamic Voltage and Frequency Scaling (DVS/DFS)” scheme is a key technique to save energy when energy-efficiency is an important concern. DVFS that adjusts the voltage and frequency of CPU according to current utilization. Within DVS/DFS schemes, when system workloads are under average level, considerable power consumption can be reduced through CPU voltage or frequency scaling, e.g. slowing down CPUs or switching CPUs to low-power modes. The potential system-wide energy savings offered by implementing both fine-grained and per-core DVS/DFS in a 4-core chip multi processors(CMP) system. Some disadvantages associated with individual DVS/DFS algorithms when they are ported to DCs, where virtualization technologies and emerging multi-core processors are widely deployed. In DCs with legacy hardware and software applications, it is impossible to implement global DVS/DFS schemes among all machines due to the heterogeneity.

Ponnusamy et al. [5] proposed a technical paper on Application of Green Cloud Computing for Efficient Resource Energy Management in Data Centers in which we are using the technology Dynamic Voltage and Frequency Scaling (DVFS), Migration and Dynamic Shutdown. This technology is meant to define efficient computing resource management. One technique being explored is the use of Dynamic Voltage and Frequency Scaling (DVFS) within Clusters and Supercomputers. By using DVFS one can lower the operating frequency and voltage, which results in decreased power consumption of a given computing resource considerably. Green Cloud framework maximizes performance per watt within a Cloud.we have found new ways to save vast amounts of energy while minimally impacting performance.

Rajan et al. [6] proposed a technical paper on Application of Green Cloud Computing for Efficient Resource Energy Management in Data Centers. We used the Dynamic Voltage Frequency Scaling or Dynamic Shut-down Techniques at both the component and system levels. Green Cloud is to automatically make the scheduling decision on dynamically migrating/consolidating VMs among physical servers to meet the workload requirements meanwhile saving energy and specially for performance-sensitive. It reduces the costs of operating the IT equipment. Also, cooling and increase server density enlarging the capacity of existing data center facilities, but it is not centralized and scheduled. Time(Sec)Existing Proposed Difference

| | | | | |
|---|---|---|---|---|
| Energy Consumption | 100 | 16 | 10 | 6 Joules |
| Packet Delivery Ratio | 100 | 92 | 93 | 1% |
| Packet Received | 100 | 2789 | 3204 | 415 packets |
| Throughput | 100 | 0. 2789 | 0. 3204 | 0.415 bytes |
| End-to-end Delay | 100 | 1.57 | 1.35 | 0.22 sec |

The result represents the data centers energy usage. Here, we minimize the energy usage by introducing the protocol for connectivity with the Green Cloud concept. The result represents, the energy saved up to 6 Joules when compared with the existing work.

# CHAPTER 3

# SYSTEM DEVELOPMENT

## 3.1 GREENCLOUD SIMULATOR

GreenCloud is a sophisticated packet level simulator for energy-aware cloud computing data center with a focus on cloud communications. It offers a detalied fine grained modelling of the energy consumed by the data center IT equipment, such as computing server, network switches, and communication links. Green Cloud can be used to develop novel solutions in monitoring, resource allocation, workload scheduling as well as optimization of communication protocols and network infrastructures. It can simulate existing data centers, guide capacity extension decision as well as help to design future data center facilities. GreenCloud, released under the General Public License Agreement, is an extension of the well known NS2 network simulator. About 80 percent of GreenCloud code is implemented in C++, while the remaining 20 percent is in the form of Tool Command Language (TCL) scripts. GreenCloud has been elaborated in the context of ECO-CLOUD and GreenIT projects. GreenCloud is a well-known simulation tool which offers fine-grained simulation of modern cloud computing environment focusing on data center communications and energy efficiency. GreenCloud is based on ns-2 simulation platform. It features a detailed modelling of the energy consumed by the elements of the data center, such as computing servers, switches, and network links. It also implements a set of energy efficient metrics. GreenCloud supports traditional three-tier data center architecture as well as modern data center architectures, such as DCell, BCube, FiConn, and DPillar. The three-tier architecture, used in this study, consists of the topmost core tier, the aggregation tier that is responsible for routing, and the access tier that hold the pool of computing servers arranged into racks. An important drawback of such topology is potential oversubscription. The GreenCloud simulator was extended with functionalities necessary to model heterogeneous servers to enable the implementation of the HEROS scheduler.

### 3.1.1 Getting Started: Download and Installation

GreenCloud simulator is an extension of NS-2 network simulator. It comes as an archived source tree, or as a pre-configured VM, which works with VirtualBox and VMWare Player. The VM also includes a pre-configured Eclipse environment, so it is the easiest way to download GreenCloud and start running simulations and/or modifying the source code. If you are building GreenCloud on a non-VM machine, here are the basic instructions:

1. Download GreenCloud.

2. Unpack the downloaded software. It comes already integrated into NS-2 source code.

3. Navigate to the extracted directory.

4. Run ./install.sh to do a full installation (it should work on any Debian based system with a 3.2+ kernel, i.e. Ubuntu 12.x and higher).

5. Execute the simulation script by running ./run . 6. View the dashboard by opening show-dashboard.html.

### 3.1.2 Simulation Setup

The simulation is setup using TCL files located in ./src/scripts/ directory. The main file main.tcl determines the data center topology and simulation time. It also calls a set of the following simulation scripts:

- setup_params.tcl contains general configuration of servers, switches, tasks, monitoring and migration
- topology.tcl creates the data center network topology
- dc.tcl creates the data center servers and VMs
- user.tcl defines behaviour of cloud users
- record.tcl sets up runtime results reporting procedures
- finish.tcl calculates and reports simulation statistics. The simulation is run by executing the./run/ script.

It contains the following three simulation parameters:

- Data center load defines the number and computing requirements of incoming tasks with the respect to the data center capacity. Usually the load should be between 0 and 1. The load close to 0 represents an idle data center, while the load equal or greater than 1 would saturate data center.

• Simulation time shows the maximum time allowed for task execution, while tasks deadlines have impact on the timesharing behavior of tasks. Longer deadlines allow more tasks to be executed in parallel on a single host or a VM.

• Memory requirement defines the maximum size of the simulated memory resource that can be used in multitasking

### 3.1.3 Why GreenCloud Simulator

Table 3.1 How GreenCloud is better

| Parameter | GreenCloud | CloudSim | MDCSim |
|---|---|---|---|
| Platform | Ns2 | SimJava | CSIM |
| Language/Script | C++/OTcl | Java | C++/Java |
| Availability | Open source | Open source | Commercial |
| Simulation time | Tens of minutes | Seconds | Seconds |
| Graphical support | Limited (Network animator) | Limited (CloudAnalyst) | None |
| Application models | Computation, Data transfer, and Exec. deadline | Computation, Data transfer | Computation |
| Communication models | Full | Limited | Limited |
| Support of TCP/IP | Full | None | None |
| Physical models | Available using plug in | None | None |
| Energy models | Precise (servers + network) | None | Rough (servers only) |
| Power saving modes | DVFS, DNS, and both | None | None |

```
greencloud@greencloud:~$ cd greencloud/
greencloud@greencloud:~/greencloud$ ./run


*****************
BUILDING TOPOLOGY
*****************

Data center architecture: three-tier debug
Creating switches CORE(2) AGGREGATION (4) ACCESS(6)...
Creating 30 servers...
Creating 1 cloud user(s)...

*********************
SIMULATION PARAMETERS
*********************

Simulation time: 60.0 seconds
Data center computing capacity: 30000030 MIPS

Power management of computing servers: No
Power management of network switches: No

Progress to      0 %
Progress to     10 %
Progress to     20 %
Progress to     30 %
Progress to     40 %
Progress to     50 %
Progress to     60 %
Progress to     70 %
Progress to     80 %
Progress to     90 %
```

Fig 3.1. Result Simulation in GreenCloud

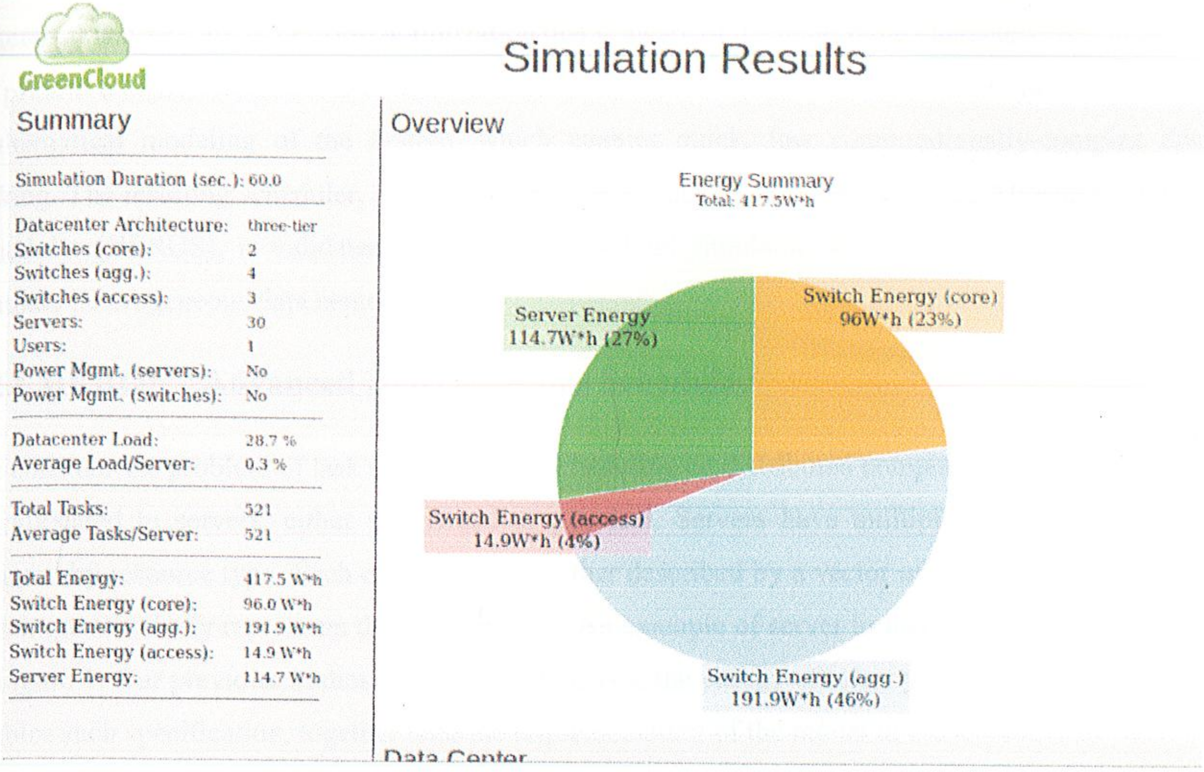


Fig 3.2. Pie chart presentation of simulation

**3.2 Algorithm:** Heterogeneity is a growing trend in distributed systems, including cloud computing. The increasing manufacturing capabilities combined with the need for high performance and high computational density result in growing diversification and specialization of the hardware. Examples of these trends include the growing utilization of general-purpose processing on graphics processing units (GPGPUs), low-power system on a chips (SoCs), multi- and many-core architectures, asymmetric cores, coprocessors, and solid-state drives. Even standardized settings, such as data centers composed of containers are facing heterogeneity. The power consumption trend of electronic hardware is one of the reasons behind the growth of the heterogeneity. Equipment specialization increases energy efficiency, which in its turn requires technologies such as Dynamic Shutdown (DNS) or dark-silicon. In essence, they aim to use the most efficient hardware (or its components) for the shortest period of time. Further developments in the field of software, notably virtualization, enable workload consolidation and exploitation of the low-power hardware states. Cloud computing, which comprises large pools of resources accessed via common resource management framework, facilitates such optimization by creating more opportunities for aggregation. Virtualization can add an additional dimension to the heterogeneity by the introduction of various hypervisors, and by containers1 which may be encapsulated in Virtual Machines (VMs). A hypervisor has impact not only on performance, but also on energy-efficiency. Modern Information Technology (IT) systems are becoming structurally complex, with elaborated software stacks. To get the most out of these systems, it is necessary to perform optimization that is aware of the underlying characteristics. In this paper, we present a highly scalable load balancer, which exploits heterogeneity in data centers and is based on a mathematical modeling of the system, which enables quick, low computationally-complex decision making. The resulting scheduler, named Heterogeneous Energy-efficient Resource Allocation Optimizing Scheduler (HEROS), is validated using the GreenCloud simulator, which recent extensions enable to simulate heterogeneous data centers.

### 3.2.1 HEROS – Advanced Heterogeneous Scheduler

We consider the problem of task (user request) scheduling on distributed computing infrastructures. Tasks are allocated to servers, either virtualized or physical. Servers have multiple components, which are grouped by resource type. Each component is further described by a vector of numbers, called capacities, which quantitatively represents their capabilities. An example of server in this representation is presented in Figure 1. Our previous studies present how to derive the energy-efficiency parameters for a model that enables such specification, together with the implementation of the model in the GreenCloud simulator.

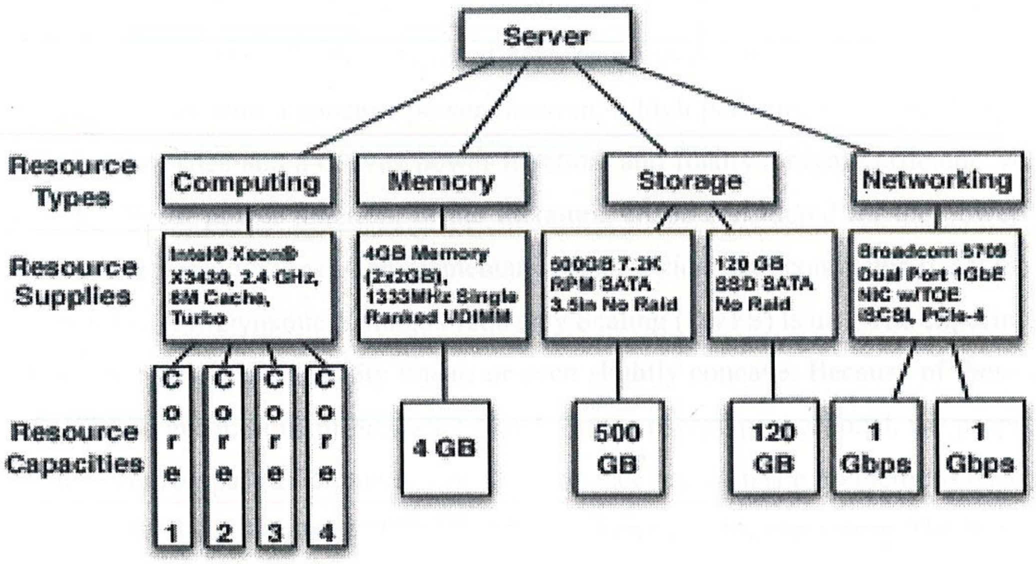


Fig 3.3 Example of a server with its components.

Tasks are indivisible units of work and they are described using the same model as servers, enabling composed allocations as presented in Figure 2. In practice, each task is described by the following mandatory parameters: input and output communications volumes, and the number of CPU instructions to be executed. These parameters impact utilization of networking and computing resources, respectively. It is possible that a task requires additional resources, e.g. storage on local drive or physical memory. Descriptions of tasks can be also fully heterogeneous, including several requirements for the same type of resource. The energy-efficient scheduling has two contradictory objectives: consumed energy and mean response time (or mean flowtime). The minimization of the energy consumption is achieved by consolidation of the load and putting idle servers to the sleep state. The energy consumption is defined as the total energy consumed by the servers. Response time is defined as the time difference between the creation of a task and the end of its output communication. The minimization of the response time is operated by a wide distribution of the workload, which minimizes the response time. The intelligent, energy-efficient scheduling combines both of these characteristics. The presented solution for the optimization problem is called Heterogeneous Energy-efficient Resource allocation Optimizing Scheduler (HEROS). The HEROS methodology is based on DENS and e-STAB, and backward compatible. It indeed relies on a similar approach for establishing server selection and communication potential functions. Similarly, to DENS, HEROS allocates tasks to the server with maximum score. The score is calculated by a decision function, which has two main components: the server selection function and the communication potential function. The novel server functions of heterogeneous hardware may vary significantly. Fig. 3 presents various power functions for three types of heterogeneous computing nodes: a commodity server, which is the least efficient with a concave power function, a high performance computing (HPC) server with the highest performance and a convex power function, and finally a highly efficient, yet low power micro server with a linear power function. In the literature, shape considered for the power functions is often dependent on the assumptions. Complementary Metal-Oxide Semiconductor (CMOS) technology suggest a convex relation if Dynamic Voltage Frequency Scaling (DVFS) is use. The experimental studies show that the power function is in reality linear, or even slightly concave. Because of these divergences and the fact that future generations of hardware may be more energy proportional, we propose a general approach, which can encompass all of these cases. Performance per Watt (PpW) metric is used to underline energy efficiency and can be directly used to select the most energy efficient server. The PpW function for server s is defined as:

$$PpW_s(l) = \frac{Perf_s(l)}{P_s(l)} \qquad (3.1)$$

where $perf_s(l)$ is the performance function (e.g. performance in MIPS at load l), and $p_s(l)$ is the power consumption function. Due to the heterogeneity, it is necessary to express l at

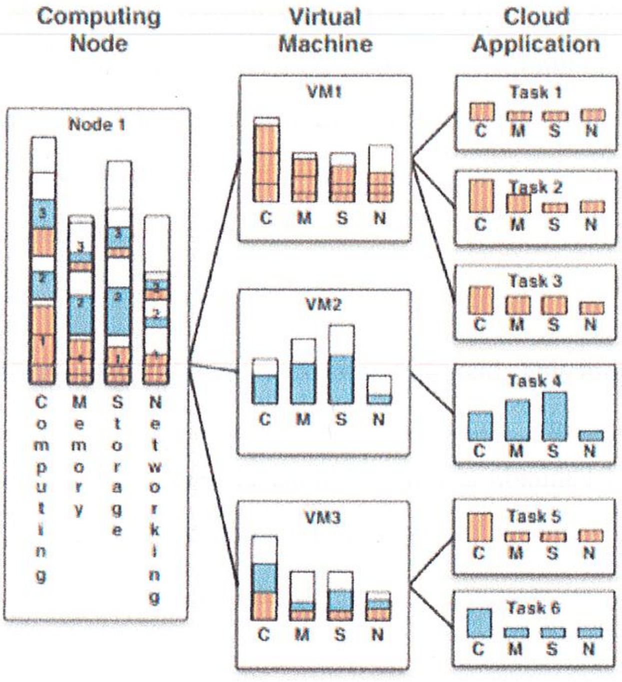


Fig 3.4. Resource allocation on a node with heterogeneous resources.

the same scale for all servers, using a standard unit relevant for a scheduled application, e.g. MIPS or a number of processed requests per second. A practical drawback of the straightforward usage of PpW is the fact, that servers become the most energy-efficient when fully loaded, which in practice can easily lead to overloading and drastic reduction of performance and energy efficiency. To prevent that, the HEROS server selection function is defined as:

$$H_s(l) = PpW_s(l).\left(1 - \gamma.\frac{1}{1+e^{\frac{-\alpha}{maxl_s}(l-\beta.maxl_s)}}\right) \qquad (3.2)$$

where $maxl_s$ is the maximum server load. The domain, or possible range of values of $l_s$, is defined as

$l_s = [0, maxl_s]$

The second term of the selection function is a sigmoid scaled to the domain $l_s$ and the range of $PpW_s(l)$. The sigmoid aim is to counter the impact of the PpW function for high values of load. The coefficient $\alpha$ determines sharpness of the descending slope, while $\beta$ is based on the maximum acceptable load of the server. In practice, these variables are set to $\alpha = 110$, $\beta = 0.9$, and $\gamma = 1.2$, to assure a smooth degradation of the selection function starting from 90% of maximum load. Fig. 4 presents server selection functions for the three servers, where thinner lines present the values of PpW functions without subtraction of the sigmoid. The communication potential Q(u) is based on the DENS communication potential, but instead of queue buffer size, it uses actual link load, and is defined as follows:

$$Q(u) = e^{-\left(\frac{2u}{U_{max}}\right)^2} \tag{3.3}$$

where u is a current link load and $U_{max}$ is the maximum link load. This function has only one component, i.e. the corresponding top-of-the-rack communication potential, which makes HEROS applicable to topologies other than three-tier. The final decision function is obtained by multiplication of the server selection function and the communication potential function:

$$F_s(l, u) = H_s(l). Q_s(u) \tag{3.4}$$

The server chosen to execute a task is the one with the highest decision function value. In case of a tie, the server is chosen randomly among the ones featuring the best value. In case of idle servers, the maximal PpW is multiplied by the communication potential, to make a balanced choice between potential energy savings and balancing workload among racks. Given that the data aggregation phase is performed on each computing node separately, the decision making shows little complexity. In this paper, the complexity of algorithm is O(n) in case of scanning a list of machines in order to find the best place. Sorting the list may further facilitate the selection procedure. IV. The novel HEROS scheduler is an extension of the stateof-the-art network- and energy-aware schedulers. HEROS is specifically designed to operate in heterogeneous systems. It bases its decisions on the aggregation of utilization and instantaneous PpW of servers with the utilization of network links. HEROS is implemented in the GreenCloud simulator, proving its effectivity in comparison with the reference scheduling approaches in homogeneous and heterogeneous systems, where it saves up to 47% of server's energy. The decision function of HEROS effectively simplifies complex description of heterogeneous servers. It also normalizes capacities and power functions of servers, making the scheduler extensible and adaptive to new settings. As a result, HEROS performs also well in homogeneous cases. Additionally, new types of servers can be dynamically added at runtime, which only requires simple calculation of their decision functions. The exact decision-making mechanism could be further elaborated. In this paper, the complexity of algorithm is O(n), in case of scanning all list of machines in order to find the best place. The future work will test weighted round robin algorithm approach, which would reduce complexity to O(1) in case scores are used to periodically update weights. More elaborated schemes may include a distributed organization, optimized to minimize network traffic while providing the required information. Future directions include performing comprehensive experimentation, with non-uniform task size and task generation patterns, and simulations of more complex, virtualized, multi-tenant environments. HEROS could be improved by extension of the set of optimized objectives, integration of other data sources, and distribution of HEROS using a multi-agent framework to enable cooperation and exchange of information by schedulers in a single data center, or even between multiple cloud computing systems. HEROS could also help in solving other, related problems, e.g. energy-efficient workflow scheduling. In this case, the combination of network-aware models with the decision function of HEROS could enable achievement of scalable and dynamic workflow allocation in cloud systems.

### 3.2.2 DENS Algorithm

The DENS methodology minimizes the total energy consumption of a data center by selecting the best-fit computing resources for job execution based on the load level and communication potential of data center components. The communicational potential is defined as the amount of end to-end bandwidth provided to individual servers or group of servers by the data center architecture. Contrary to traditional scheduling solutions that model data centers as a homogeneous pool of computing servers, the DENS methodology develops a hierarchical model consistent with the state of the art data center topologies. For a three-tier data center, we define DENS metric $M$ as a weighted combination of server-level $f_s$ , rack-level $f_r$ , and module-level $f_m$ functions:

$$M = \alpha . f_s + \beta . f_r + \gamma . f_m \qquad (3.5)$$

where $\alpha$, $\beta$, and $\gamma$ are weighted coefficients that define the impact of the corresponding components (servers, racks, and/or modules) on the metric behavior. Higher $\alpha$ values favor

the selection of highly loaded servers in lightly racks. Higher $\beta$ values will prioritize computationally loaded racks with low network traffic activity. Higher $\gamma$ values favor selection of loaded modules. The $\gamma$ parameter is an important design variable for job consolidation in data centers. Taking into account that $\alpha$ +$\beta$ +$\gamma$ must equal unity, the values of $\alpha$= 0.7, $\beta$= 0.2, and $\gamma$= 0.1 are selected experimentally to provide a good balance in the evaluated three-tier data center topology. The factor related to the choice of computing servers combines the server load $L_s(l)$ and its communication potential $Q_r(q)$ that corresponds to the fair share of the uplink resources on the ToR switch. This relationship is given as:

$$f_s(l,q) = L_s(l).\frac{Q_r(q)^{\varphi}}{\delta_r} \qquad (3.6)$$

where $L_s(l)$ is a factor depending on the load of the individual servers $l$, $Q_r(q)$ defines the load at the rack uplink by analyzing the congestion level in the switch's outgoing queue $q$, $\delta_r$ is a bandwidth over provisioning factor at the rack switch, and $\varphi$ is a coefficient defining the proportion between $L_s(l)$ and $Q_r(q)$ in the metric. Given that both $L_s(l)$ and $Q_r(q)$must be within the range [0, 1] higher $\varphi$ values will decrease the importance of the traffic-related component $Q_r(q)$. Similar to the case of computing servers, which was encapsulated in eq (6), the factors affecting racks and modules can be formulated as:

$$f_r(l,q) = L_r(l).\frac{Q_m(q)^{\varphi}}{\delta_m} = \frac{Q_m(q)^{\varphi}}{\delta_m}.\frac{1}{n}\sum_{i=1}^{n} L_s(l) \tag{3.7}$$

$$f_m(l) = L_m(l) =.\frac{1}{k}\sum_{i=0}^{k} L_r(l) \tag{3.8}$$

where $L_r(l)$ is a rack load obtained as a normalized sum of all individual server loads in the rack, $L_m(l)$ is a module load obtained as a normalized sum of all of the rack loads in this module, $n$ and $k$ are the number of servers in a rack and the number of racks in a module respectively, $Q_m(q)$ is proportional to the traffic load at the module ingress switches, and $\delta_m$ stands for the bandwidth over provisioning factor at the module switches. It should be noted that the module level factor $f_m$ includes only a load-related component $l$. This is due to the fact that all the modules are connected to the same core switches and share the same bandwidth using ECMP multi-path balancing technology. The fact that an idle server consumes energy that is almost two-thirds of its peak consumption, suggests that an energy-efficient scheduler must consolidate data center jobs on a minimum possible set of computing servers. On the other hand, keeping servers constantly running at peak loads may decrease hardware reliability and consequently affect the job execution deadlines. To address the aforementioned issues, we define the DENS load factor as a sum of two sigmoid functions:

$$L_s(l) = \frac{1}{1+e^{-10\left(l-\frac{1}{2}\right)}} - \frac{1}{1+e^{\frac{-10}{\varepsilon}\left(l-\left(1-\frac{\varepsilon}{2}\right)\right)}} \tag{3.9}$$



The first component in (3.9) defines the shape of the main sigmoid, while the second component servers as a penalizing function aimed at the convergence towards the maximum server load value. The parameter ε defines the size and the incline of this falling slope. The server load $l$ is within the range [0, 1]. For the tasks having deterministic computing load, $l$ the server load can be computed as the sum of computing loads of all of the running tasks. Alternatively,

for the tasks with predefined completion deadline, the server load $l$ can be expressed as the minimum amount of computational resource required from the server to complete all the tasks right-in-time. Being assigned into racks, the servers share the ToR switch uplink channels for their communication demands. However, defining a portion of this bandwidth used by a given server or a flow at the gigabit speeds during runtime is a computationally expensive task. To circumvent the aforementioned undesirable characteristic, both (3.7) and (3.8) include a component, which is dependent on the occupancy level of the outgoing queue $Q(q)$ at the switch and scales with the bandwidth over provisioning factor $\delta$. Instead of relying on the absolute size of the queue, the occupancy level $q$ is scaled with the total size of the queue $Q_{max}$ within the range [0, 1]. The range corresponds to none and full buffer occupancy. By relying on buffer occupancy, the DENS metric reacts to the growing congestion in racks or modules rather than transmission rate variations. To satisfy the aforementioned behavior, $Q(q)$ is defined using inverse Weibull cumulative distribution function:

$$Q(q) = e^{-\left(\frac{2q}{Q_{max}}\right)^2} \quad (3.10)$$

The obtained function favors empty queues and penalizes fully loaded queues. Being scaled with the bandwidth over provisioning factor $\delta$ in (3.7) and (3.8) it favors the symmetry in the combined uplink and downlink bandwidth capacities for switches when congestion level is low. However, as congestion grows and buffers overflow, the bandwidth mismatch becomes irrelevant and immeasurable. The (3.10) is inspired by the Random Early Detection (RED) and Backward Congestion Notification (BCN) technologies. The obtained bell-shaped function favors selection of servers with the load level above average located in racks with the minimum or no congestion. The following algorithm is used to compute the DENS metric during runtime:

**DENS Algorithm**

**Initialization**

**set** weighted coefficient $\alpha = 0.7$, $\beta = 0.2$, $\gamma = 0.1$

**set** proportional coefficient $\varphi = 2$

**get** server load $l$

**get** queue size at access and aggregate switches $q$

**Server selection**

**FOR** all servers **DO**

**compute** server load $L_s(l)$, rack load $L_r(l)$, and module load $L_m(l)$

**compute** communications potentials of rack $Q_r(q)$ and module $Q_m(q)$

**compute** metric factors related to servers $f_s(l,q)$, racks $f_r(l,q)$, and modules $f_m(l)$

**compute** DENS metric as a weighted sum of $f_s(l,q)$, $f_r(l,q)$, and $f_m(l)$

**ENDFOR**

Select server with highest DENS metric

Fig 3.5 DENS algorithm

### 3.2.3 e-STAB Algorithm

The energy-efficient scheduler for cloud computing applications with traffic load balancing (e-STAB) is designed to optimize energy consumption of cloud computing data centers. The e-STAB scheduler treats communicational demands of the jobs equally important to that of the computing requirements. e-STAB is a scheduler aiming to: **(a)** balance communication flows produced by the jobs and **(b)** consolidate jobs on a minimum amount of the computing servers. As network traffic can be highly dynamic and often difficult to predict, the e-STAB scheduler analyses both the load on the network links and the occupancy of outgoing queues at the network switches. e-STAB allocates jobs favoring network resources that offer the most of the available bandwidth and penalizes resources for which the load approaches the available transmission capacity when the traffic queues growing in size. Queuing analysis aids in preventing a buildup of network congestion. Such techniques are already implemented in several transport-layer protocols that estimate buffer occupancy of the network switches and can react before congestion related losses occur. The e-STAB scheduling policy can be defined with the following two steps executed for every incoming cloud computing data center workload:

- **Step 1:** Select a group of servers S connected to the data center network with the highest available bandwidth, provided that at least one of the servers in S can accommodate the computational demands of the scheduled job. The available bandwidth is defined as an unused capacity of the link or a set of links connecting the group of servers S to the rest of the data center network.

- **Step 2:** Within the selected group of servers S, select a computing server with the smallest available computing capacity, but sufficient to satisfy the computational demands of the scheduled task.

In fat tree data centers, the servers are arranged into racks, forming a set of racks R. Subsequently, racks form the set of modules M. Therefore, to select the group of servers with the largest available bandwidth in Step 1, e-STAB must first find a module such that

$$Am(m_i) = \max_{\forall m \in M}(Am(m)) \qquad (3.11)$$

where Am is the available bandwidth of a module mi computed on a per-server basis. For a module $m_i \in$ M the available bandwidth can be computed as

$$Am_i = \frac{Cm_i - \lambda m_i}{Sm_i} \qquad (3.12)$$

where $Cm_i$ is the transmission capacity of a module i, calculated as a sum of maximum transmission speeds of all links connecting a module i to the network of core switches, is a currently effective transmission rate of the traffic, and $Sm_i$ is a number of servers hosted in the module. (12) provides an instantaneous measure of the available bandwidth. However, as most of the transmissions are bursty, which either use full network link capacity for a short time or leave the link unutilized. Therefore, the available capacity must be calculated as an average over the time interval T:

$$Am_i(t) = \frac{1}{T}\int_t^{t+T}\left(\frac{Cm_i-\lambda m_i(t)}{Sm_i}\right) dt = \frac{1}{Sm_i}\left(Cm_i - \frac{1}{T}\int_t^{t+T} \lambda m_i(t)dt\right) \quad (3.13)$$

Similar to the case of modules, for identifying a rack with the most of the available bandwidth, e-STAB must find a rack such that

$$Ar(r_j) = \max_{\forall r \in R}(Ar(r)) \quad (3.14)$$

where $Ar$ is the available bandwidth of a rack $r_j$ computed on a per-server basis. For a module $r_j \epsilon R$ the available bandwidth can be computed as

$$Ar_j(t) = \frac{\frac{1}{T}\int_t^{t+T}(Cr_j-\lambda r_j(t))dt}{Sr_j} = \frac{1}{Sr_j}\left(Cr_j - \frac{1}{T}\int_t^{t+T} \lambda r_j(t)dt\right) \quad (3.15)$$

where $Cr_j$ is the transmission capacity of a rack $j$, calculated as a sum of maximum transmission speeds of all links connecting a rack $j$ to the network of aggregation switches, $\lambda r_j$ is a currently effective transmission rate of the traffic, and $Sr_j$ is a number of servers hosted in the rack. One of the main goals of the e-STAB scheduler is to achieve load balanced network traffic and prevent network congestion. A helpful measure is the available bandwidth per compute node within the data center. However, such a measure does not capture the system dynamics, such as sudden increase in the transmission rate of the cloud applications. To have a more precise measure of the network congestion, e-STAB scales the measures of the available bandwidth $Am_i(t)$ and $Ar_j(t)$ with the component related to the size of the bottleneck queue

$$Q(t) = 1 - \frac{1}{T}\int_{t}^{t+T} e^{-\left(\frac{\rho.(q(t)-1)}{Q_{max}}\right)^{\varphi}} dt \qquad (3.16)$$

Where *q(t)* is an instantaneous occupancy of the queue measured either in bytes or packets at the time *t*, $Q_{max}$ is the maximum allowed size of the queue, while parameters K and L control the shape of the function and are explained in the following paragraph. Equation (16) is an integral version of the Weibull cumulative distribution It aims to favor the empty queues or queues with a minimum occupancy and penalize highly loaded queues that are on the threshold of buffer overflow (or on the threshold of losing packets).

# CHAPTER 4

# Experimental Results and Performance Analysis

## 4.1 Performance and Energy Efficiency Metrics for Communication Systems of Cloud Computing Data Centers

Energy efficiency is becoming a very important issue in Cloud Computing environments as more and more Internet services are deployed in data centers This paper presents an in depth review of energy efficiency in cloud computing wherein we have focused on different metrics related to energy efficiency in cloud computing to analyse power performance of cloud computing.

Cloud providers rely on large and power-consuming data centers. With the increasing deployment of many data centers and computer servers around the globe, the energy cost on running the computing, communication and cooling together with the amount of $CO_2$ emissions have increased dramatically. Energy consumption has always been a major concern in the design and cost of datacenters. The wide adoption of virtualization and cloud computing has added another layer of complexity to enabling an energy efficient use of computing power in large-scale settings.

There are two main alternatives for making data center consume less energy: (a) shutting it down or (b) scaling down its performance. The former method, commonly referred to as Dynamic Power Management (DPM) results in most of the savings as the average workload often stays below 30% in cloud computing systems. The latter corresponds to the Dynamic Voltage and Frequency Scaling (DVFS) technology that can adjust the hardware performance and power consumption to match the corresponding characteristics of the workload.

A large number of cloud servers consume massive energy and produce huge pollution. The Smart2020 analysis shows that cloud-based computing data center and the telecommunication network will generate emission about 7% and 5% each year in 2002 and 2020, respectively. From the energy efficiency perspective, a cloud computing data center can be defined as a pool of computing and communication resources organized in the way to transform the received power into computing or data transfer work to satisfy user demands. In fact, the utilization ratio of data center resource is only 30%.

Cloud Computing being a model for utility computing becoming attractive and its usage is increasing as it promises to reduce the maintenance and management costs in comparison with in-house infrastructure. However, Cloud providers also rely on large and power consuming data centers designed to support the elasticity and scalability required by their customers. Energy-efficiency remains a serious problem for providers as increasing demand, and high performance expectations and it is a challenging issue in the Cloud model due to the dynamicity of its components and flexibility of services supported by virtualization. Generally, energy efficiency and CO2 reduction within the cloud infrastructure implies that:

i. Execution of applications requires less energy, and

ii. Energy consumed during the execution comes from renewable sources or low CO2 emitting sources.

## 4.2 METRICS

**4.2.1 Thermal Design Power (TDP)**: It is the measurement of maximum amount of power required by cooling of computer system to dissipate. It is the maximum amount of power which a computer chip can take when running a real application.

**4.2.2 Power usage Effectiveness (PUE):** It is used for comparison of energy used by computing application and infrastructure equipment and the energy wasted in overhead. The PUE can be described as the ratio of overall electricity consumed by the facility of a data center to the overall electricity consumed by IT equipment's (network peripherals, servers, storage, routers, etc.)

$$PUE = \frac{Total\ Facility\ Power}{IT\ Equipment\ Power} \quad (4.1)$$

**4.2.3 Data Center Infrastructure Efficiency (DCiE):** It is the reciprocal of PUE. PUE and DCiE are most commonly used metrics that were designed for the comparison of efficiency of datacenters. It is defined as:

$$DCiE = \frac{1}{PUE}$$

$$= \frac{IT\ Equipment\ Power}{Total\ Facility\ Power} \quad (4.2)$$

IT Equipment Power can be described as the power that data center has taken for the management of IT equipments, processing of IT equipments and storing the data in disk drives or routing the data within the datacenter. Total Facility Power is IT equipment power plus power needed by uninterrupted power supply (UPS), generators (needed to provide power in case of power failure), Batteries, cooling system components such as chillers, CRACs, DX air handler units, pumps, and cooling towers.

**4.2.4 Performance per Watt:** It quantifies the energy efficacy of individual computer architecture or computer hardware. It is the processing rate that can be remitted by a processor for each watt of power absorbed by it. This must be high. It measures the rate of computation that can be delivered by a computer for every watt of power consumed by it. Normally it is measured in FLOPS (floating point operations per second) and MIPS (million instructions per second).

**4.2.5 Compute Power Efficiency(CPE):** It is a measure of the computing efficiency of a data center. As each watt consumed by server or cluster did not draw fruitful work all the time, some facility consumed power even in idle state and some consumes power for computing. Although 100% of facility capacity will never be used, but still we want maximum output from the electrical power which data center has taken. CPE is defined as

$$CPE = \frac{IT\ Equipment\ Utilization}{PUE} \qquad (4.3)$$

**4.2.6 Green Energy coefficient (GEC)**: It is a measure of green energy (energy that comes from renewable sources) that is used by the facility of a datacenter. For evaluating the environment friendly nature of a data center this metric is used. It is selected as a PUE metric by green grid organization in November 2012. Energy consumed is measured in kWh. It is defined as-

$$GEC = \frac{Energy\ Consumed}{Total\ Energy\ Consumed} \qquad (4.4)$$

**4.2.7 Energy Reuse Factor (ERF):** It is a measure of reusable energy (energy that is reused outside of a datacenter) that is used by datacenter. For making cloud, environment friendly data center should use renewable energy such as electricity generated by wind power, hydro power etc. ERF is selected as a PUE metric by green grid organization in November 2012. It is defined as

$$ERF = \frac{Reused\ Energy\ Used}{Total\ Energy\ used} \quad (4.5)$$

**4.2.8 Carbon usage Effectiveness(CUE):** It is a measure of carbon dioxide emission in environment by the data center. It is selected as a PUE metric by green grid organization in November 2012. It is defined as organization in November 2012. It is defined as-

$$CUE = \frac{E_{CO_2}}{E_{IT}} \quad (4.6)$$

$E_{CO_2}$ = Total carbon dioxide emission from total energy absorbed by the facility of a data center.

$E_{IT}$ = Total energy consumed by IT equipments. $E_{CO_2}$ includes all green house gases (GHGs) such as $CO_2$ and methane ($CH_4$) that are emitted in atmosphere. This value is taken for whole year analysis.

**4.2.9 Water Usage Effectiveness (WUE):** It is a measure of required water by a data center annually. Water is needed - a) For cooling the facility of a data center. b) For humidification. c) For apparatus associated power generating d) For production of energy. It is defined as-

$$WUE = \frac{Water\ used\ annualy}{E_{IT}} \quad (4.7)$$

**4.2.10 Data Center Productivity (DCP)**: It is a measure of amount of fruitful work yielded by data center. It is defined as-

$$DCP = \frac{Useful\ work\ done}{T_{resource}} \quad (4.8)$$

Where $T_{resource}$ = total resource taken to produce this useful work It is similar to DCeP but it considers both hardware and software resources where DCeP considers only software resources.

**4.2.11 Data Center Energy Productivity (DCeP):** It is a measure of amount of fruitful work yielded by datacenter with respect to energy consumed to yield this work. It is defined as-

$$DCeP = \frac{Useful\ work\ done}{T_{energy}} \quad (4.9)$$

Where $T_{energy}$ = energy taken to produce this useful work

**4.2.12 Space, Wattage and Performance (SWaP):** It is a Sun Microsystems metric for datacenters. It is developed for computing the energy and space requirement of a datacenter.

$$SWaP = \frac{Performance}{Space * Power} \quad (4.10)$$

The rack unit height of the system gives the space that a server occupies. The power expenditure of a server can be captured in wattage from actual benchmark runs data collected on a site or from the planning guide of vendor site. As cloud computing is becoming more and more popular, the power requirement of cloud computing is also increasing so now, it is the responsibility of designers and inventors of hardware components to look in this area by inventing Energy-Efficient Processors, Energy-Efficient servers, Energy Efficient data centers, using renewable energy sources for fulfilling power needs.

## Results of Simulation

Assuming the value of average load in server is 30%. Total number of server used in simulation is 144(1,1,144). Architecture framework is modified three- tier high speed architecture. Bandwidth on a link modified three-tier high speed architecture

Bandwidth between core & aggregation, aggregation & access and switch & host is 10 GB, 1 GB, 1GB respectively. Total computing capacity of data center is 1024102400 MIPS.

We are using Random Scheduling and HEROS Scheduling. Random scheduling is selecting a machine from a random distribution, which is uniform by default. Vendors provide same functionality with different interface at different cost. Due to variation in cost, administrator of data center use different component to construct. So, this is the reason heterogeneity is trending now a day. The power consumption trend of electronic hardware is one of the reasons behind the growth of the heterogeneity.

**RANDOM Scheduling**

Random

****************

BUILDING TOPOLOGY

****************

Data center architecture: Random

Creating switches CORE (1) AGGREGATION (2) ACCESS (64) ...

Creating 64 servers...

Loading resource specifications configuration files...

Selected DC scheduler: Random

VM static configuration...

********************

Creating cloud users

********************

Data center total computing capacity: 256025600 MIPS

Creating 1 cloud user(s)...

********************

SIMULATION PARAMETERS

********************

Simulation time: 65.5 seconds

Power management of computing servers: DVFS DNS

Power management of network switches: DVFS

Progress to 0 %

Progress to 10 %

Progress to 20 %

Progress to 30 %

Progress to 40 %

Progress to 50 %

Progress to 60 %

Progress to 70 %

Progress to 80 %

Progress to 90 %

.

.

*****************

SIMULATION REPORTS

*****************

Total tasks submitted: 14718

DC load: 26.8%

Energy consumed by servers: 137.7 W*h

Energy consumed by switches: Core(51.4 W*h) Aggregation(102.8 W*h) Access(171.0 W*h) - 325.1 W*h

Total energy consumed (servers + switches): 462.9 W*h

Average tasks per server: 0

Average CPU load per server: 0.3

Average Memory load per server: 0.0

Average Storage load per server: 0.0

Tasks failed on servers: 0

Tasks failed (rejected) on DC scheduler level: 0

Average load per vm: 0.3

Tasks failed on vms: 0

Cloud User: 0 : all tasks finished sucesfully.

**HEROS Scheduling**

HEROS

****************

BUILDING TOPOLOGY

****************

Data center architecture: HEROS

Creating switches CORE(1) AGGREGATION (2) ACCESS(64)...

Creating 64 servers...

Loading resource specifications configuration files...

Selected DC scheduler: HEROS

VM static configuration...

********************

Creating cloud users

********************

Data center total computing capacity: 256025600 MIPS

Creating 1 cloud user(s)...

********************

SIMULATION PARAMETERS

********************

Simulation time: 65.5 seconds

Power management of computing servers: DVFS DNS

Power management of network switches: DVFS

Progress to 0 %

Progress to 10 %

Progress to 20 %

Progress to 30 %

Progress to 40 %

Progress to 50 %

Progress to 60 %

Progress to 70 %

Progress to 80 %

Progress to 90 %

*****************

SIMULATION REPORTS

*****************

Total tasks submitted: 14718

DC load: 26.8%

Energy consumed by servers: 65.7 W*h

Energy consumed by switches: Core(51.4 W*h) Aggregation(102.8 W*h) Access(171.0 W*h) - 325.1 W*h

Total energy consumed (servers + switches): 390.9 W*h

Average tasks per server: 0

Average CPU load per server: 0.3

Average Memory load per server: 0.0

Average Storage load per server: 0.0

Tasks failed on servers: 0

Tasks failed (rejected) on DC scheduler level: 0

Average load per vm: 0.3

Tasks failed on vms: 0

Cloud User: 0 : all tasks finished successfully.

Table 4.1 Small Homogeneous Topology- Result

| Scheduler | Total Energy [kWh] | Server Energy [kWh] | Tasks Failures # | Un-finished Tasks # |
|---|---|---|---|---|
| Random | 1036.7 | 308.8 | 0 | 0 |
| HEROS | 875.5 | 147.6 | 0 | 0 |

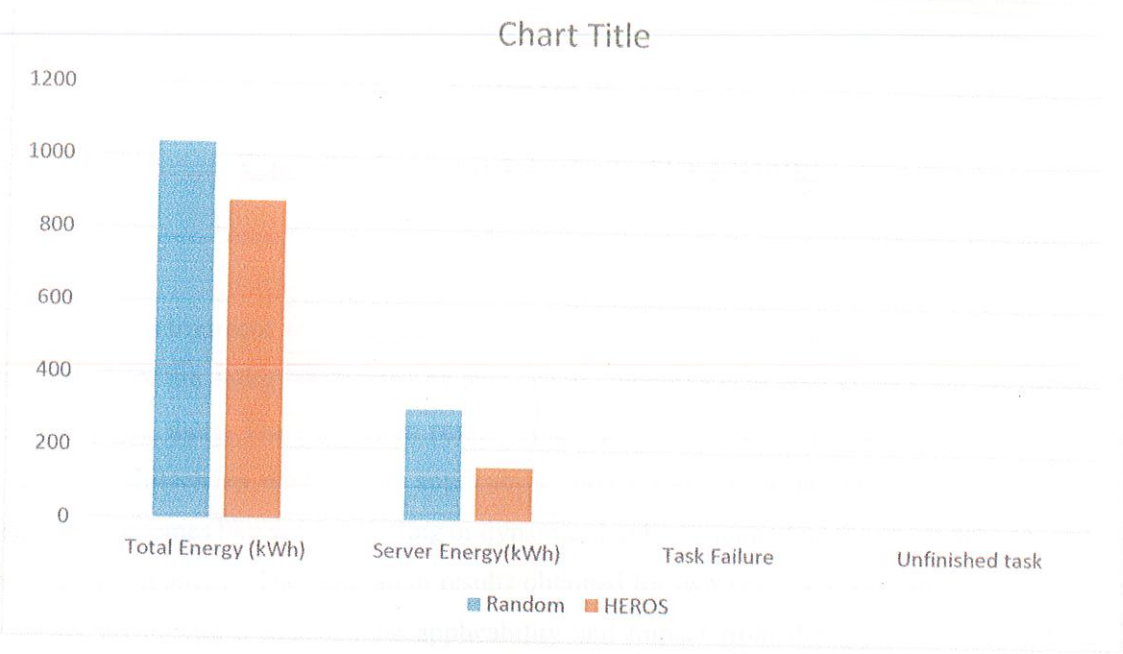


Fig 4.1 Small heterogeneous topology – relative results

For Random scheduling total energy consumed (in kWh) is 1036.7, energy consumed by server (in kWh) is 308.8 For HEROS scheduling total energy consumed is 875.5, energy consumed by server is 147.6. Result for Small Homogeneous Topology is presented in table 4. The efficiency of HEROS is better than that of Random. It is consuming 52.20 % less in server and overall 15.54% less in total. HEROS combines the best feature of DENS and e-STAB. It operates on rack-level so it can tolerate any network topology. Load balancing in HEROS is same as that of e-STAB and to reduce energy consumption in server DENS's feature is used. Decision taken in HEROS scheduling is done by collecting information and characterized by less computational complexity.

Following are the result came from simulation for the proposed metrics for HEROS scheduling algorithm.

$$Effective\ Computational\ Power = \frac{Power\ used\ in\ Server\ Computation}{Total\ Power}$$

$$= \frac{147.6}{875.5}$$

$$= 0.1686$$

$$Effective\ Networking\ Power = \frac{Power\ used\ in\ Networking\ Device}{Total\ Power}$$

$$= \frac{727.9}{875.5}$$

$$= 0.8314$$

# Chapter 5

## Conclusions and Future Directions

A simulation environment for energy-aware cloud computing data centers. GreenCloud is designed to capture details of the energy consumed by data center components as well as packet-level communication patterns between them. The simulation results obtained for two-tier, three-tier, and three-tier high-speed data center architectures demonstrate applicability and impact from the application of different power management schemes like voltage scaling or dynamic shutdown applied on the computing as well as on the networking components. The simulation results obtained for two-tier, three-tier, and three-tier high-speed data center architectures demonstrate applicability and impact from the application of different power management schemes like voltage scaling or dynamic shutdown applied on the computing as well as on the networking component. we discussed modified three-tier architecture for Random and HEROS scheduling and its advantage. Also, discussed advantages of HEROS scheduling over Random. Main aim was to pursue energy efficient cloud computing which focuses on reduction of Green House gases and to maximize the utilization of resources. Various terminologies are used and defines to calculate the efficiency of the data center in terms of power usage and energy consumed.

Future directions include the implementation of different efficient scheduling algorithm in different architecture frame work environment to make efficient energy resource management for cloud computing in data center. Focus on the simulator extension adding storage area network techniques and further refinement of energy models used in the simulated components.

# References

1. Claudio Fiandrino, Dzmitry Kliazovich, Pascal Bouvry and Albert Y. Zomaya, "Performance Metrics for Data Center Communication Systems", 2015 IEEE 8th International Conference on Cloud Computing.

2. Dzmitry Kliazovich, Pascal Bouvry and Samee Ullah Khan, "DENS: Data Center Energy-Efficient Network-Aware Scheduling", Italy: IEEE, July 2014.

3. Dzmitry Kliazovich, Sisay T. Arzo, Fabrizio Granelli, Pascal Bouvry and Samee Ullah Khan, "e-STAB: Energy-Efficient Scheduling for Cloud Computing Applications with Traffic Load Balancing", 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber,Physical and Social Computing.

4. Aimin Sang and San-qi Li, "A predictability analysis of network traffic," Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), vol. 1, pp.342-351, 2000.

5. Mateusz Guzek, Dzmitry Kliazovich, Pascal Bouvry, "HEROS: Energy-Efficient Load Balancing for Heterogeneous Data Centers", Italy: IEEE, July 2014.

6. A. Jain, M. Mishra, S. K. Peddoju, and N. Jain, "Energy efficient computing- Green cloud computing," 2013 Int. Conf. Energy Effic. Technol.Sustain.,

7. Ms Jayshri Damodar Pagare, Dr. Nitin A Koli, "Understanding of Energy efficiency in cloud

computing", International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Volume 3, Issue 6, November-December 2014.

8. Claudio Fiandrino, Dzmitry Kliazovich, Pascal Bouvry, Member and Albert Y. Zomaya, "Performance and Energy Efficiency Metrics for Communication Systems of Cloud Computing Data Centers", IEEE TRANSACTIONS ON CLOUD COMPUTING, 2015.

9. Greencloud - The green cloud simulator, available at http://greencloud.gforge.uni.lu/.

10. D. Kliazovich, P. Bouvry, and S. U. Khan, "GreenCloud: a packet-level simulator of energy-aware cloud computing data centers," Journal of Supercomputing, special issue on Green Networks, pp. 1 - 21, 2010

11. Anton Beloglazov and Rajkumar Buyya , "Cloud Computing and Distributed Systems (CLOUDS) Laboratory" in 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing.

12. Zhiming Shen, Sethuraman Subbiah and Xiaohui Gu, "Elastic Resource Scaling for Multi-Tenant Cloud Systems", in 2011Department of Computer Science North Carolina State University.

13. Beant Singh Gill, Satinderjit kaur Gill, and Prince Jain, "Analysis of Energy Aware Data Center using Green Cloud Simulator in Cloud Computing", in – Nov 2013 International Journal of Computer Trends and Technology (IJCTT) – volume 5 number 3.

14. Congfeng Jiang, Yinghui Zhao and Jian Wan, "Dynamic Voltage/Frequency Scaling for Power Reduction in Data Centers", Applied Mechanics and Materials Vols. 20-23 (2010) pp 1148-1156

15. Yuvapriya Ponnusamy and S Sasikumar, "Application of Green Cloud Computing for Efficient Resource Energy Management in Data Centers", in 2012 International Journal of Computer Science and Information Technologies, Vol. 3 .

16. Anusuya and Krishnapriya, "Green Cloud: A Pocket-Level Simulator with On-Demand Protocol for Energy-Aware Cloud Data Centers",in feb 2014 International Journal of Science and Research (IJSR) ISSN (Online): 2319-7064