WESTERN SYDNEY
UNIVERSITY

Study of challenges to make cloud computing systems energy efficient

Balkar Singh

18497494

A report submitted for

300598 Master Project 2

in partial fulfillment of the requirements for the degree of

Master of Information and Communication Technology (Advanced)

Supervisor: Yogesh Sharma

School of Computing, Engineering and Mathematics
Western Sydney University

October 2017

ABSTRACT

Cloud Computing Technology has changed the whole IT sector and world. It provides many services we use every day such as e-mail, phone-backup, and much more. Because of furious development and adoption of cloud computing the expansion of data centers has also been increased to run the cloud based applications. Due to such expansion and adoption of cloud computing paradigm, the enegy requirement to operate the underlying computing infrastructure also increases. Energy is a scarce resource which is created from fossil fuels, nuclear power, water, and other resources. As the energy consumption is increasing, service providers have to bare more expenses in terms of operational costs and users have to pay more for the expensive services. However, to operate large underlying infrastructure, more electricity also need be produced using various sources such as fossil fuels, water etc. which causes adverse effects on environment such as a rise in carbon emissions, normal temperature and greenhouse gasses. This study is based on finding different method and techniques of making cloud computing more power and energy efficient. The review the existing technologies and developments in this sector is already done in the first phase of the study. Now, in this phase the simulation of the cloud environment is done using cloud based discrete event simulator ‘Cloudsim’. Different resource allocation and provisioning algorithms are simulated on the cloud framework designed using cloudsim simulator and performance evaluation of the proposed policies is done using various metrics.

ACKNOWLEDGMENTS

This research study is done to accomplish the prerequisites for the MICT (Master of Information and Communication Technology) degree. I want to say thanks all the people who helped me during this research. I am thankful to Dr. Bahman Javadi (DAP - Post Grad ICT, Western Sydney University) and to my research supervisor Yogesh Sharma (Ph.D. candidate, Western Sydney University). Because of Dr. Bahman expert guidance and suggestions, I was able to do my research in right manner and direction. Also, the consistent advice and field knowledge provided by Yogesh helped me a lot to achieve to desired results of the study. Finally, I want to say thanks to Western Sydney university for providing library with huge number of online resources which helped me to collect the data for my research.

TABLE OF CONTENTS

.

LIST OF TABLES

LIST OF FIGURES

CHAPTER I: INTRODUCTION

This research study is based on cloud computing. So, let's first discuss what is cloud computing. It is a technology through which computing services are provided by the cloud providers at the user's demand. Large IT firms like Google, Amazon and many others are the cloud providers. These firms have huge number of computing resources which are collectively called as Cloud Infrastructure. Cloud Infrastructure inclues datacenters. A firm can have any number of data centers. In a datacenter, mainly we have two types of resoures which are hardware and software. We get the term cloud when these softwares and hardwares works collectively in an organized manner to complete the user requests. The idea of network based computing was introduced in 1960's but the term cloud computing is firstly used by Google's CEO Eric Schmidt in 2006 confrence. As the time passes this technology become more mature and advanced and it becomes one of the most used and popular technology in today's digital world.

Cloud Computing provides three types of services which are SaaS, PaaS and IaaS. SaaS stands for Software as a service for example web applicatons like photoshop, word and excel provided by Adobe and Microsoft online. Any user can rent these applications according to their need like one day, week, month or a for a year and can use them over internet without installing them on their computer. PaaS stands for Platform as a service like Google app engine. It is cloud service through which a user can develop and host a web app in google cloud. IaaS stands for Infrastructure as a service like Goolge compute engine and Web services by Amazon (Vaquero, 2008). This type of service allows a user to rent a space in the cloud as per their requiremnt. As cloud computing provides a limitless scability so a user can rent any amount of computing resources ranging from a VM(Virtual Machine) to host, host to a server or maybe several servers in a datacenter.

Nowadays, cloud computing is used by everyone ranging from big companies like Snapchat to a normal smartphone user. Because of its wide useability its growth rate is very high. Since 2009 spending on cloud is 4.5 times more as compare to spending on IT. According to a research, in 2020 public cloud computing spending will be $162 billion earlier in 2015 it was only $67 billion ('Salesforce Releases New Research ' 2016). This much rise in the cloud computing growth leads to the tremendous increase in its power consumption. Facebook datacenters power consumption is 60M watts approximately whereas power consumption of google is 300M watts approximately. According to IBM its spending on electricity is 45% of its total operational expenditure (Sharma et al. 2016). If we talk about number of datacenters 2094 federal datacenters were recorded in 2010 whereas in 1998 they were only 424. The total power consumption of datacenters all over the world is 30B watts approximately and to generate such amount of power we need at least 30 nuclear plants (Glanz 2012).

Every day number of cloud computing users are increasing and so as its power consumption so, on one side we have increased power consumption and on the other side there is a lot of wastage of power on different levels of consumption of power in the datacenters. Computer hardware in datacenters are not the only source of power consumption actually half of the datacentre power is used by the cooling hardware like fans and air conditioners. Also, the power consumed by the servers inside the datacenters is not productive all the times because the servers are not utilized all the times there are times when servers remain idle and consume power unnecessarily.

There are many harmful effects on environment due to power generation and consumption in such huge amounts and then there is power wastage also which makes this condition more worse. Global warming has already become the big problem for the world but after knowing about it also nobody started taking preventive measures to stop it. Everybody have to take some steps to solve this problem. There are many reasons behind the increase in the normal temperature and heat is one of them. Heat is dispersed in the environment when power is generated and consumed. When power consumed sometimes there is dual heat dispersion for example when severs at datacenters consume power they become hot spread heat and to cool them cooling hardware like air conditioners are used which again dispersed heat into the environment. Not only the heat dispersion IT sector is also responsible for the 2 percent of the total global CO_2 emissions and we all know that how badly CO_2 effects our environment. According to a research there is a correlation between the technology export and the emissions of CO_2 which is 83 percent (Florea, Sommer & Ahmadabadi 2016). In today's digital world of social media everybody is sharing without thinking and for suppose if someone watches 1 second video on YouTube then it will release .2 gram of CO_2 into the environment. The total number of videos can range up to 6B if we add up all social platforms videos. The total amount of carbon emission released into environment every year is unimageable because only four billion videos of one second can release eight thousand tons of CO_2 emission every day. All the researchers are working together to find new ways to decrease these CO_2 emissions and one simple way is to reduce the power consumption because less power consumption will lead to less heat and CO_2 dispersion and also then we need less power generation which will decrease the burning of the fossil fuels which disperse huge amount of CO_2 gases and heat which ultimately reduces the harmful effects on environment. Major IT corporations like google and apple have taken some steps to reduce carbon emissions and start using the green energy. Google policy is to make themselves hundred percent carbon free and to achieve that they are using green energy at present they are using thirty seven percent of green energy. Another research project of google is to make use cold water circulation in datacentre building structure to replace the cooling the hardware and completely vanish the heat emission done by such appliances.

Apple is also following the footsteps of the google they are presently using eighty seven percent of green energy in their California office.

After taking all of the information into account it can be clearly said that there is lack of methods which can reduces the power consumption of cloud computing and make it more power efficient. In the literature review section of this report different methods to reduce the power consumption are discussed which are purposed by different researchers. In the methodology part of this report, the procedure used to conduct this research are discussed including software used for the simulation of the cloud based datacentre along with the data collection and evaluation methods. Later in the results section, the results of this research study are discussed which overall depicts that we can reduce the power consumption by using various resource allocation and provisioning algorithms.

CHAPTER II: Literature Review

2.1 Power and Energy

Before talking about various methods of making cloud systems more power/energy efficient. It is important to understand that the terminology of power and energy and their relation.

Power (P): Power is the rate at which system performs work.

Energy (E): Energy is the total amount of work performed over a period of time.

$$P = \frac{W}{T} \qquad\qquad E = P * T \qquad\qquad (1)$$

W is work. Power is measured in Watts (w) and energy is measured in Watt-hour (Wh)

Due to a lot of research is done in the area of increasing performance per watt ratio but power consumption of devices is not reducing which is the main problem.

2.2 Types of Power Consumption

There are two types of power consumption by devices one is Static power consumption and other is Dynamic power consumption.

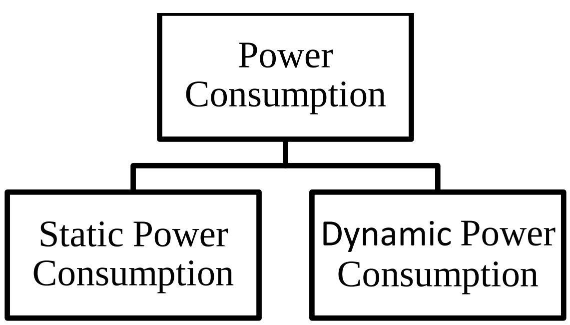

Figure 1: Types of Power Consumption (Beloglazov, Buyya, Lee, & Zomaya, 2011)

2.2.1 Static Power Consumption is closely related to low-level system design (circuits). It is determined by transistors and process technology. Power leakages are a big problem in system circuits. To improve the power efficiency re-designing of low-level system design should be done.

2.2.2 Dynamic Power Consumption has two factors one is short-circuited other is Switched capacitance. The short circuit only consumes 10-15% of power where major power consumption is done by Switched capacitance.

$$P_{dynamic} = a * c * v^2 * f \tag{2}$$

a= Switching Activity, C= physical capacitance, v=voltage, f= clock frequency

Low-Level System Design   Dynamic Voltage Frequency Scaling (DVFS)

DVFS is the process to reduce the power consumption by handling the voltage and clock frequency of the CPU. DVFS is discussed in detail in next section of this report.

As servers are generally underutilized in the data center so, low average utilization of resources leads to high power consumption. According to a research (Beloglazov et al., 2011), CPU’s are generally utilized less than 50%. The more is the lower average utilization of resources the more will be the Total cost of ownership (TCO) of Datacenter (DC). Which include cooling system, PDU, generators, power delivery facilities, UPS, etc. Augmented density and the lack of space between the server components leads to heat dissipation for which cooling is required which leads to more power consumption.

2.3 Power efficiency techniques

Both research papers (Beloglazov et al., 2011) and (Kaur & Chana, 2015) are analyzed and discussed in detail along with some other researchers. In (Beloglazov et al., 2011) power efficiency techniques are divided into four different levels which are hardware, operating system, virtualization and data centre level whereas in (Kaur & Chana, 2015) power efficiency techniques are divided into seven different types of techniques which are Virtual Machine Allocation and

Scheduling Based Techniques, Multicore Architecture Based Techniques, Consolidation based techniques, power aware management scheme based energy efficient techniques, thermal-aware based techniques, Bio-inspired computing based techniques and miscellaneous techniques.

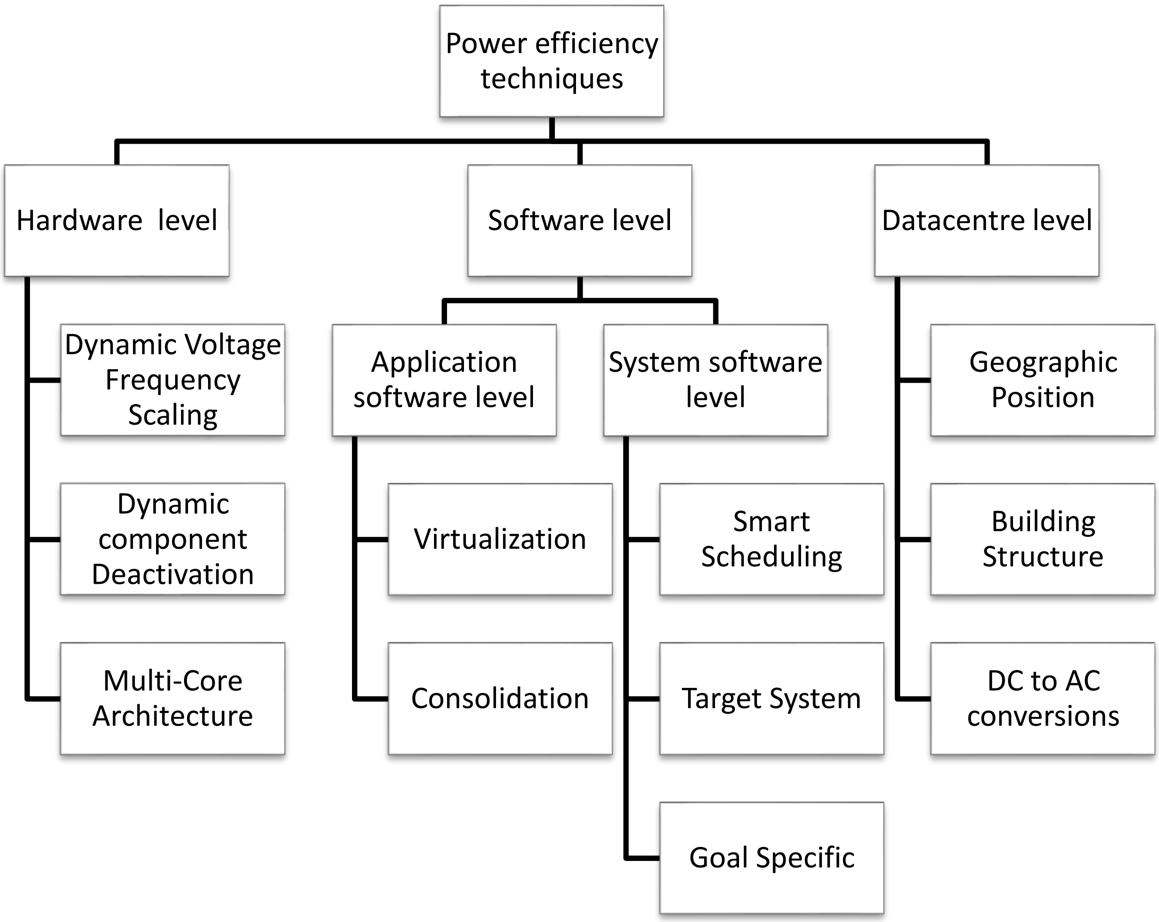

Figure 2. Power efficiency Techniques Taxonomy

2.3.1 Hardware Level

In hardware level, DCD and DVFS along with ACPI are discussed.

2.3.1.1 DCD (Dynamic component deactivation) is a technique in which a system's component is deactivated when it's idle and unnecessarily consuming power. Components must be deactivated because they don't support the dynamic performance scaling. But at run time it is hard to save power consumption through this technique because of increase power consumption in component activation also there is some performance degradation because of delays which occur due to the time is taken by component initialization. So, this technique will only be useful when the idle time is long enough which can save some power.

To solve this problem different techniques have been introduced which are divided into two categories:

- **Predictive**: The system history and close future are used in Predictive techniques. As it's a prediction so its efficiency is based on the correlation between the system history and its future. If idle time prediction is not exact means if predicted time is less than actual, then there will be fewer power savings and if the predicted time is more than the actual then there will be a performance loss.

  Predictive DCD can be

  - **Static**: In static, a time limit with no activity is decided to be used as a parameter which decides the deactivation of the component. After decided idle time is passed component is deactivated. As its static, so it can be applied to any workload type but it must be changed for every workload and to solve over and under-prediction idle time limit can be changed. The disadvantage of this technique is that as the component is activated after receiving the first request so there is some performance loss. To solve these issues:
    - Predictive Shutdown: By using the system idle history and close future predictive shutdown is done to don't miss any power saving idle window.
    - Predictive wakeup: Whereas predictive wakeup solves the problem of performance loss by activating the component before receiving the request.

- **Adaptive**: But static technique only works for known workload from system history so for unknown workload adaptive techniques are used in which component deactivation deciding parameters can be changed dynamically by using the earlier prediction results.

- **Stochastic**: Another category of techniques is stochastic in which probabilistic model of the system is created in which component activation or deactivation and requests are stochastic processes. But creating a model is not easy upon receiving the request as it takes time. In system upon receiving requests with some probability and component activation done with other probability by solving the stochastic problem.

2.3.1.2 Dynamic Voltage Frequency Scaling (DVFS) is power saving technique in which both power and CPU clock frequency are manipulated together to save power. As modern CPU's can run on various voltage levels as per its utilization. Lower the CPU utilization lower will be the power level. But by reducing the CPU clock speed or frequency the number of instructions processed by the CPU is also reduced which increases the processing time and reduces performance. Every application to perform well require clock frequency to be matched with its requirements. The order of scheduled tasks could get changed because of the slower clock frequency. DVFS should be used carefully to reduce power consumption.

DVFS can be applied in three ways.

- **Interval-based** in which system history is used to know the CPU utilization and by using this information CPU clock frequency and power is tuned for future.
- **Intertask** in which different clock speed is assigned to the different tasks in the system when the type of the workload is known.

- **Intertask** in which task processing time is divided into parts or chunks and then different clock speed is assigned to different parts of the task processing time. In this technique, the task structure information is used to find appropriate clock speed.

2.3.1.3 ACPI (Advanced configuration and power interface) is a software system which provides an interface to the power management with an additional feature of hardware discovery and configuration. No matter how good and efficient are the hardware techniques for improving power efficiency but they are hard to change as per the upgraded policies as they are implemented in hardware design so the software is the best option which can be updated easily. Before ACPI there was platform and firmware specific power management software.

- ACPI become the first platform and firmware independent software which improves the power management and configuration of hardware devices. Through ACPI, DPM is introduced in the OS which enables all the systems to dynamically manage the power.
- By using ACPI developers can improve the power management. ACPI defines the power states which are C and P states. These states can be applied while using DPM.
- P-states are the power performance states which can use any different settings of DVFS. P0 is the highest performance state and with an increase of state, performance will reduce.
- C states are CPU power states which illiterates halt, stop-clock and sleep mode. In all recent OS, ACPI is widely used to manage the system power.

2.3.2 Operating system level

In Operating system level, various researches have been discussed which are based on certain characteristics like system resources (single or multiple), Target systems (mobile or server), goal (minimize energy consumption, performance loss or meet energy budget), power saving techniques (DVFS, DCD or resource throttling) and workload (real-time or high computing applications).

2.3.2.1 Ondemand Governor (Pallipadi & Starikovskiy, 2006) is an in-kernel real-time power manager for Linux OS for fulfilling current performance requirements. Manager continuously CPU utilization, set clock frequency and supply the voltage. It keeps CPU busy up to 80% to accommodate the heavy change in workload.

Governor sample the CPU utilization in the centralized way which becomes overhead with increased number of CPU'^s^. So, to solve this problem parallel sampling independently for each CPU is purposed.

2.3.2.2 Eco System (Zeng, Ellis, Lebeck, & Vahdat, 2002) researchers split the OS power management into two categories:

- **System devices**
- **Applications**

The currentcy concept is introduced by researchers. One unit of currentcy represents the right to consume some energy for some fixed period.

- When user select battery lifetime and application properties then Eco System provides some currentcy to each application.
- When application used all the currentcy then that processes are not further serviced or scheduled.

2.3.2.3 Nemesis OS (Neugebauer & McAuley, 2001) provides an accurate and easy accounting for every consumption by individual applications. It is built to solve a battery lifetime management problem. Applications work and adapt as per the data provided by the OS. If some applications aren't able to do that then the user must shut down the low-priority tasks.

2.3.2.4 Illinois Grace Project (Vardhan et al., 2009) researchers have created a plan to save power with help of synchronized adaptation at different system layers according to variations in requests for system resources by the applications.

Three different levels of adaptation have been purposed by researchers:

- Global adaptation: keeps eye on all application which is active in the system on all the layers and on entry or exit applications into the system.
- Per application adaptation: work in isolation on the application and used when time frame adjusting system resources for application demand.
- Internal adaptation works separately on diverse resources shared by many different applications and adjusts their states.

Enhanced version of Grace works on mobile multimedia systems by using hierarchical adaptation. It also works on network bandwidth constraints and reduces network transmission power. By using Grace on both network and CPU adaptations can save 32% of energy savings.

2.3.2.5 Linux/RK (Rajkumar et al., 2000) researchers have purposed four different algorithms to implement DVFS in a real-time system. Research results showed 50% of energy savings upon the use of purposed algorithms. Algorithms are implemented with help of Linux kernel called Linux/ Resource kernel. The system chooses the best algorithm as per the system needs.

- Sys-clock (system clock) frequency assignment is reasonable for frameworks where the overhead of frequency scaling and power is too high to perform at each setting switch.
- PM-clock (Priority-Monotonic clock) frequency assignment is appropriate for frameworks with a low-frequency scaling and power overhead permitting modification of the frequency and power settings at every setting switch.
- Opt-Clock (Optimal Clock) frequency assignment utilizes a non-linear model for every application to decide an ideal frequency that limits the power utilization. This method is appropriate for offline use because of high computational complexity.
- DPM-Clock (Dynamic priority-monotonic clock) frequency assignment appropriate for frameworks where the normal execution time of an application is altogether not as much as the worst-case scenario.

2.3.2.6 Coda and Odyssey (Flinn & Satyanarayanan, 2004) are the systems created for managing mobile systems limited battery lifetime, computing power and changeability of network connection. The research results show savings of 30% of energy by using the purposed system but to use the system all necessary applications must be modified. In the context of a distributed file system application transparent adaptation is used by Coda whereas managing and initializing application adaptations is done by Odyssey. By its use, data can be transferred over the network in appropriate quality which will use limited computing power and save power. The data quality can be adjusted as per application request.

2.3.2.7 Power Nap (Meisner, Gold, & Wenisch, 2011) system is created for saving power consumption in servers by investigating the transition between the low to high power states of the CPU. The planned approach make use of short idle periods that happen because of workload changeability. Power savings are linear if the transition time less than 10ms and it outperforms the outcome from DVFS for low utilization. But the application of this method is difficult because the current transition time of servers is 300ms which way more than requested by the researcher but if requirements are met then the system can save up to 74% of energy consumption.

2.3.3 Virtualization Level

Virtualization in cloud computing is running Virtual machines on a single physical machine which reduce power consumption by reducing a number of physical machines and using machines to full utilization. As various virtual machines(VM's) can be executed on the single physical machine so the hardware is shared between those VM's.

Therefore, to keep things coordinated virtualization layer is created between VM's and hardware which is called as Virtual Machine Monitor(VMM). VMM can be used to implement power saving techniques. For example, VMM can monitor all VM's, their performance and can use various energy efficient techniques like DCD or DVFS to increase overall system performance.

Various virtualization solution providers are available in the market like VMware and KVM (Beloglazov et al., 2011). All these solutions use power management with some additional features.

2.3.3.1 Xen hypervisor is a virtualization solution which uses ACPI p-states which are explained earlier in this report. Four different types of states are used by xen which are on demand, user space, performance and powersave. According to present system need ondemad selects the best P-state. In userspace, the user can set the p-state. To achieve the highest performance, highest p-state is used by performance whereas to save power lowest p-state is used by powersave. Xen also implements the various c-states which allow saving energy like sleep mode. Along with c and p States xen also provide a function to migrate VM from one physical machine to another both online or offline which save power by shutting down the underutilized servers and transferring VM’s to a minimal number of physical machines.

2.3.3.2 VMware also implements c and p states like xen. ESX and ESXi are virtualization solutions for enterprise level. Apart from services provided by zen VMware also provides VMotion and Distributed resource scheduler services(DRS). These two services work together to save power consumption. DRS provides the service of monitoring the servers in the data center to keep them fully utilized and with a balanced load and for achieving this DRS uses VMotion service which performs VM migration between physical nodes.

2.3.3.3 KVM virtualization solution is implemented as Linux kernel module. KVM supports sleep and hibernation states.

Apart from these available virtualization solutions, various researches are also done to improve the power efficiency through virtualization. In (Rodero et al., 2012) researchers extended the functionality of cluster VM allocation which schedules similar types of tasks to the same system which saves power because similar types of processes process faster as compare to mix of different types of processes like i/o and memory-based processes. Also, send unused servers to sleep mode which also saves power. Reducing the number of servers required to complete the user's requests is called consolidation.

Another research (Kaur & Chana, 2015) works on Dynamic VM consolidation to reduce power consumption. This technique uses the VM’s historical data for regulating utilization thresholds automatically and migrating the appropriate VM’s to complete the user requests. In another research (Quan, Mezza, Sannenli, & Giafreda, 2012) researchers have created an algorithm which collects the power consumption data of all the servers of the data center. As the resulting algorithm provides the list of servers with the lowest power consumption and highest commutation performance then the VM’s with heavy data are transferred to those servers.

2.3.4 Data Center level

At data center level energy efficiency is highly required because it is the highest level in the hierarchy and there are various inefficiencies present at this level like under or over resource provisioning, unoptimized VM allocation, underutilized servers and much more. The main two technologies help in reducing power consumption are consolidation and virtualization. As consolidation reduces the number of servers needed to complete requests and virtualization helps in running many systems(VM’S) in one system also VM’s can be easily transferred to the best server selected by various techniques like for providing best performance or for saving energy.

A research (Buyya, Beloglazov, & Abawajy, 2010) is done on energy efficiently managing the data center resources. The research finds the way to provision the datacentre resources energy efficiently while meeting the quality standards specified in SLA agreements. Researches have divided the problem into various levels. Firstly user sends a request for the resource, the resource broker gets the request. Secondly, resource broker analyses the request and translate it into provisioning VM’s then forward it to many data centers. Thirdly, datacentres reply back to the broker with the price of provisioning the VM’s and then brokers choose the data centers with the lowest price. Lastly, the VM is provisioned for the user request.

Another research (Gao et al., 2014) is done on energy efficient resource management based on service level agreement. At datacentre level, Researchers have used server consolidation and DVFS to attain application level performance and power efficiency. Relocation of VM's is used as per user requirements while meeting the minimum resource requirement specified in the SLA. A programming model is created to find the best cluster in real time for example which VM should be transferred to which server and which server should be activated. The selected server cluster will provide the lowest power consumption. Through experiments, researchers have quoted that purposed system of resource management can save power up to 50%.

From a different perspective of saving energy a research (Kaur & Chana, 2015) is done. The research is focused on the scheduling the jobs in such a manner that the heat emission from servers will remain low. Reduction in power consumption of servers is important but power consumption of cooling equipment is also important. Another similar research is done in which solution is divided into two parts. One is workload monitor and another is power thermal manager(PTM). Workload manager takes account of all the incoming user requests and predicts incoming workload also activate/deactivate servers according to workload whereas PTM takes decisions about the active servers.

Power aware scheduling of tasks to reduce power consumption is also used by many researchers. A research is done to reduce the overall power consumption of servers by scheduling the jobs in power aware manner. In a cluster of power-aware systems, the processor's frequencies are altered from on-off or off-on based on the requirement. Although, Scheduler handles trad off between performance and power. Another research is done in which a purveying algorithm is used for real time access of VM's in energy aware manner. Initially, system search for the energy conscious schemes based on DVFS then apply the purveying algorithm.

After analysing the power efficiency techniques classified on basis of various researches, levels and on basis of different techniques a comparison is done by combining both the basis to have a better and clear understanding of energy efficiency techniques.

2.4 Power efficiency Techniques Taxonomy

In figure 2 power efficiency techniques are shown. In this section power efficiency techniques taxonomy will be discussed. Software level and Datacenter level will be discussed in detail as hardware level is already discussed.

2.4.1 Virtualization

Virtualization is an important feature of cloud computing which is realised with the help of Virtual machine. In a single physical machine, there can be many virtual machines. A virtual machine is a simulation of an actual machine created within a physical machine with the help of software like VMware, VirtualBOX, Windows Virtual PC, and many more. Various virtual machines shares the resources of the single physical system which helps in achieving full utilization of resources, reduction in CPU idle time and also in reducing power consumption as many machines are running only on the cost of running single physical machine. The physical machine is called the Host machine and the virtual machines are called the Guest machines. In the user aspect, every user has a single guest machine whereas many users share the same host machine. Between the virtual machines and physical system hardware, there is Virtualization layer called as VMM (Virtual Machine Monitor) which is responsible for managing the resource provisioning, keeping each VM isolate, power management, and other managerial decisions.

2.4.2 Consolidation

Consolidation is a technique in which a total number of servers or resources required for fulfilling user request are reduced by making efficient use of available resources. Consolidation makes use of DVFS and DCD techniques to achieve energy efficiency. Consolidation is used at different levels of server consolidation, VM consolidation, and task consolidation. In server consolidation, consolidation is done on the server as a whole like if two servers are running and both of them are only 40% utilized or are underutilized then data of the first server will be transferred to the second server and then the first server will be turned off. Consolidation also helps in workload balancing by dividing the work into a minimum number of servers equally and prevent the servers from being underutilized or over utilized. Which saved a huge amount of energy as only those servers are running which are required and others are switched off or are in sleep mode. Workload balancing is achieved through VM consolidation in which VM migration is performed. VM migration is done to reduce the number of active physical machines. In task consolidation, a similar type of tasks is allocated to a single server to reduce the processing time because tasks which are similar in nature takes less computing time as compare to tasks of different nature. Through task consolidation process computing time is reduced which leads to reduction energy consumption.

2.4.3 Multicore Architecture

All latest modern CPU’s (central processing unit) are equipped with Multicore architecture. Multicore architecture CPU’s are those CPU’s which have more than one core, for example, Intel manufactured CPU’s core 2 duo and i3 have two cores whereas in i5, i7 there are four cores. More the cores mean more processing or computational power in the power consumption of single core CPU. Multicore architecture really helps in increasing energy efficiency in cloud computing because on multicore CPU many VM’s can run on a single machine and also with help of multicore many applications can be consolidated on the same server. As in multicore multiple, VM's runs on a single machine so there are far fewer machines required as compared to machines required in case of single core architecture which also leads to fewer infrastructure costs. Today's Multicore technology is helping Data Providers in increasing the Data center performance while keeping the power consumption and heat at minimum levels.

2.4.4 Data Centre Level

At data center level many different factors play their role in increasing energy efficiency of cloud computing but mainly they are categorized into two different categories which are internal factors and external factors. Internal factors include management of power and its distribution at data center level and data center cooling techniques. External factors include Data center geographic position and Data Center building structure.

Power management is very important at data center level. Power management is done at CPU level with different power levels which are called DVFS (dynamic voltage frequency scaling). Choosing level as per the CPU utilization more the utilization more will be the power level. This works with multicore architecture and virtualization. At a data center level, the power consumption of all servers are managed with the help of power-aware algorithms like user request scheduling which schedules the request to the appropriate server whose utilization is not full yet. The method of power distribution in the data center can also save the power loss or consumption. Generally, in data centers from main source power is distributed to each server in DC (direct current) form and then each server convert DC to AC (alternate current) because all machines run on AC but in the conversion from DC to AC there is some power loss. It became huge inefficiency if we accumulate all this individual power conversion loss. Solution to this problem to convert all the power to AC before transferring to the servers so that power loss can be minimized. Another major source of power consumption in the data center is Colling Infrastructure. Nowadays major companies are looking more into the conventional ways of managing the heat dissipation from the servers, for example, using the cold water supply in data center building to reduce the heat or cool down only the specific area of the data center where major heat is generated.

Apart from the internal factors, internal factors are also important. Recently choosing the appropriate geographic position for the data center is become very important like companies are choosing the place where temperature are low and where power is available at cheap rates. Also, the building structure of data center is very important nowadays companies are building the sustainable building structure in which there are free air flow and plantation of trees which helps in keeping the temperature low without power consumption.

2.4.5 System Software Level

Using the energy efficient measures at system software level is very important because it effects other techniques of energy efficiency like virtualization, consolidation and Multicore architecture. Different managing policies is used to increase the energy efficiency. OS policies to use the different cores of the CPU at different power levels as per the requirement to maximize the energy efficiency. Similarly different OS functions for running and managing the virtual machines. Smart scheduling is also very important. In smart scheduling, similar processes are scheduled together because the process of similar nature takes lesser time to process as compared to processes of different nature. The operating system customization is done as per the target system like for mobile system, servers or some arbitrary system because different systems have different power uses and sources. Also for the power efficiency, OS should have goal specific policies like meeting the energy budget, meeting performance goal and minimizing energy consumption.

CHAPTER III: METHODOLOGY

3.1 Research Phases

3.1.1 First Phase

- This research is done in two phases. In the first phase, all the different methods and techniques purposed by different researchers to reduce the power consumption in cloud computing is studied.

- The result of first phase is the power efficiency techniques taxonomy which is shown in Figure 1 and discussed in literature review part of this report.

3.1.2 Second Phase

- In the second phase, different resource allocation and provisioning techniques are implemented using the “CloudSim” simulator which simulates the working datacentre.

- The results of the simulator are discussed in the results section of this report.

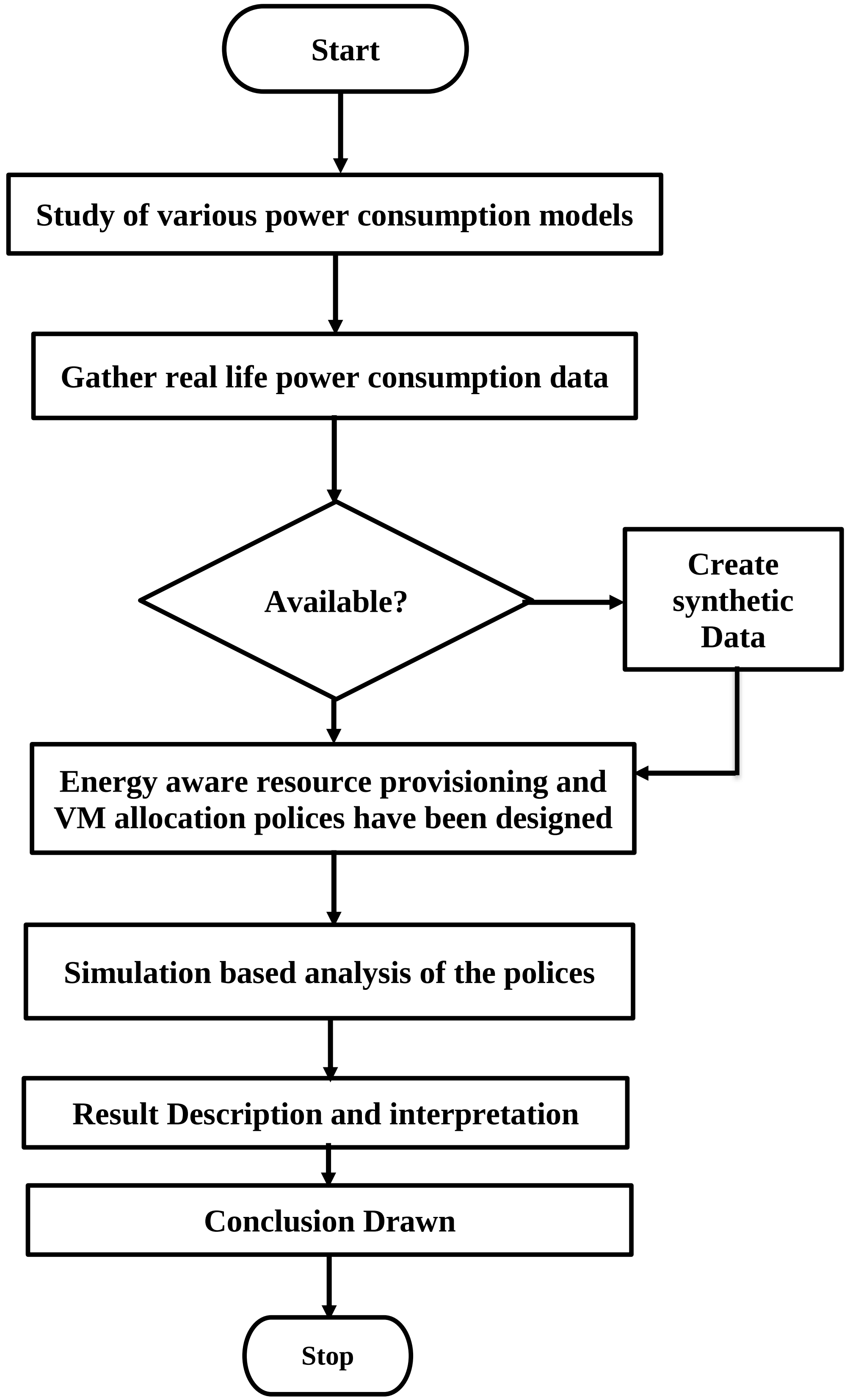

Figure 3: Methodology Flowchart

The methodology is divided into 6 different phases as shown in the figure 3.

Phase I: All the information required in third phase is gathered in this phase. Information about power consumption is collected by studying the different power consumption models.

Phase II: In stage two power consumption information is gathered. At first, a research was performed to find the real power consumption data but after search no ideal data was found. As real data was not found so synthetic data was created utilizing factors utilized as a part of the power consumption models.

Phase III: In third phase, energy aware resource provisioning and VM allocation polices have been designed. These policies are provisioning the cloud resources and to allocating the VMs on the provisioned resources in an energy efficient way.

Phase IV: In the Fourth phase simulation based analysis was done to analyse all the energy aware resource provisioning and VM allocation polices designed.

Phase V: In the fifth phase, the result is discussed which are derived from the analysis done in the fourth phase.

Phase VI: This is the last phase in this phase conclusion is drawn based on the results discussed in the earlier phase.

CHAPTER IV: RESULTS

4.1 Introduction

This section of the report is all about the results of the research work done. This section also includes all the information about the simulation environment in which the research work is done for example simulation software used, types of computers used in datacentre and datacentre characteristics. All energy aware resource provisioning and VM allocation polices designed are discussed in this section along with performance evaluation metrices. At last the simulation results of all the policies are compared and discussed with the help bar graphs.

4.2 System Model

Let's discuss about the system in which simulations were performed. As simulation of datacentre is done in the study so the type of targeted system is IaaS environment in which cloud providers given some part of infrastructure (a VM, Host or a Server) on rent to user on their demand. The simulated datacentre consists the different types of physical nodes. The configuration of each physical node contains the set of MIPSs (Millions of Instructions Per Second), available RAM (Random Access Memory), available storage and network bandwidth. The simulated environment does not have any type of information regarding the user workloads.

4.2.1 Datacenter Configuration

The datacentre simulated for this research contains the 50 hosts and the configuration of these hosts are discussed in the power model section. Along with the hosts 1,000 different cloudlets or user requests are considered for this project. The length of a cloudlet could be in the range from 70,000 to 1,00,000. Other configurations of the cloudlets include UserId which tells the id of the user who launched the particular cloudlet, Cloudlet file size gives information about the input file size of this Cloudlet before execution in byte, Cloudlet output size depicts the output file size of this Cloudlet after execution in byte. Number of pes tells about the number of Pe required to execute this job where one Pe acts as one core of the host machine.

4.2.2 Power Model

How power is consumed by different entities like Memory, CPU, Hard Disks and Cooling systems in a datacentre is already discussed in the literature review part of this report. However, the power consumption of the CPU is minimised with the help of (DVFS) Dynamic Voltage and Frequency Scaling but in this functioning also there are limited set of frequencies available to use dynamically. Also, as discussed earlier the DVFS mechanism cannot be applied to RAM memory, hard disks and network interfaces. In this research, the real power consumption data provided by spec2008benckmark for different hosts is used. For the calculation of the power consumption the maximum and minimum power consumption values are used provided by benchmark. Also, the number of cores created for host are similar to the values provided for the corresponding host in the benchmark this is done to make the simulation more realistic. Two servers with dual and quad core CPU's are used in this study. The first server selected is Ibm X3250 and its configuration are (Intel Xeon X3470, 4 cores x 2933 MHz, 8GB) and the second server selected is HP ProLiant ML110 G5 and its configuration are (Intel Xeon 3075, 2 cores×2660 MHz, 4 GB). Power consumption of these selected servers at different load level is shown in table 1. In table 1 0% shows the minimum power consumption vales of the corresponding servers and 100% depicts the maximum power consumption values.

Table 1: Power consumption of servers at different load levels in Watts

SERVER	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
IBM X3250	41.6	46.7	52.3	57.9	65.4	73	80.7	89.5	99.6	105	113
HP ML110 G5	93.7	97	101	105	110	116	121	125	129	133	135

The given below equation calculates the power consumption of *VMi* which runs on node n_j with utilization of u_i. *Pmax* and *Pmin* values are taken from the table 1and *fracj* is the fraction of *Pmin*.

$$P_j(u_i) = (frac_j \times Pmax_j) + ((1 - frac_j) \times Pmax_j \times u_i) \quad \textbf{(3)}$$

By using above calculated power consumption value, we can calculate the energy consumption because energy consumption is the amount of power consumed per unit time.

$$Evm_{ij} = P_j(u_i) \times l_i \quad \textbf{(4)}$$

E_{vmij} is the total energy consumption by the *VMi* which is executing task *li* (l= task length) on the node *nj*.

4.3 Energy aware resource provisioning and VM allocation polices

This section is dedicated to the polices which are designed with motive of reducing energy consumption. Total 9 such polices were designed which are then implement and simulated using the Cloudsim datacentre simulation tool. Different policies use different methods to achieve the required results which are discussed in detail in this section.

Table 2: Nomenclature for algo 1

Notation	Explanation
C	Set of Bag of Tasks/cloudlets
R	Set of Resources
V	Set of Virtual Machines (VMs)
t_i	i^{th} task
l_i	Length of i^{th} task
C_{sorted}	Set of Sorted Cloudlets
H_{sorted}	Set of Sorted Hosts
V_{sorted}	Set of Sorted Virtual Machines
VM_i	i^{th} Virtual Machine
U_i	Utilization of the i^{th} VM
$VM_{cores\ i}$	Cores required by the VM to execute the i^{th} task
$H_{cores\ j}$	Total number of Cores left on the j^{th} host
Pow_i	Power consumption by the i^{th} VM

Given below is the algorithm in which all of the 9 policies are included. In Algorithm 1 there are two types of inputs which are C: set of cloudlets or user requests and R: set of resources available in the datacentre. The output of algorithm 1 are set of resources which are provisioned for VM and Set of VMs which are allocated to hosts for executing the user requests or cloudlets.

Algorithm1: Energy aware resource provisioning and VM allocation

```
1.  Input: Set of Cloudlets (C), List of Resources (R)
2.  Output: Set of provisioned resources and allocated VMs
3.  IF (Policy == IHIC) Then
         Csorted ← IHIC_Cloudlets(C);
         Hsorted ← IHIC_Hosts(R);
     ELSEIF (Policy == IHDC) Then
         Csorted ← IHDC_Cloudlets(C);
         Hsorted ← IHDC_Hosts(R);
     ELSEIF (Policy == DHDC) Then
         Csorted ← DHDC_Cloudlets(C);
         Hsorted ← DHDC_Hosts(R);
     ELSEIF (Policy == DHIC) Then
         Csorted ← DHIC_Cloudlets(C);
         Hsorted ← DHIC_Hosts(R);
     ELSEIF (Policy == IC) Then
         Csorted ← IC_Cloudlets(C);
    ELSEIF (Policy == DC) Then
         Csorted ← DC_Cloudlets(C);
    ELSEIF (Policy == IH) Then
         Csorted ← IH_Hosts(R);
    ELSEIF (Policy == DH) Then
         Csorted ← DH_Hosts(R);
    ELSE
    //Default case is for RHRC policy
    Endif
4. For all ti ∈ Csorted do
         VMi ← Ci;
         V ← VMi;
         Ui ← li/lmax;
         U ← Ui;
    END FOR
5. For all Vi ∈ Vsorted do
         VMcores i ← vmi.coresRequired();
         For all hj ∈ Hsorted do
              If (Hcores i ≥ VMcores i)
              Hj ← Vi.allocate();
              Hcores j ← Hcores j - VMcores i;
              Powi ← VMi.Powermodelj();
         END FOR
    END FOR
```

The above given algorithm will select one required policy out of 9 policies and according to the selected policy both hosts and cloudlets will be sorted. After that cloudlets will be assigned to the VMs and utilization of each VM will be calculated also being sorted accordingly. After that algorithm will compare the number of cores required by the VM to execute the cloudlet with the number of cores available in the host. If the number of cores required by the VM is less than number of cores available in the host then algorithm will allocate the required number of cores to the VM. Then allocated number of cores will be deducted from the total number of cores of the host and then host core list will be updated. This process will be repeated until all the cores are allocated to the VMs and all the cloudlets are processed.

4.3.1 IHIC (Increasing Hosts Increasing Cloudlets)

As the name suggest this policy is created to sort the hosts and Cloudlets in increasing order. It sorts the hosts in such a manner that the host with lowest power consumption will be on the top of the sorted list and host with the highest power consumption will be at the last position in the list. Power consumption of host can be calculated with the help of minimum and maximum power consumption values which are shown in table 1. Now let’s talk about cloudlets. According to this policy Cloudlets are sorted in such a manner that cloudlet with the shortest length will be at the top of the list whereas cloudlet with longest length will at the last position in the list.

```
Function1.1 Increasing Hosts Increasing Cloudlets (IHIC)
Function IHIC_Hosts (R)
// Sorting the hosts in the increasing order as per power
their power consumption
1.   for all j ∈ R do
2.   Hsorted ← rj.SortHostIncreasingOrder()
3.   end for
4. return Hsorted
 End function
```

```
Function1.2 Increasing Hosts Increasing Cloudlets(IHIC)
Function IHIC_Cloudlets (C)
// Sorting the cloudlets in the increasing order as per
their length
1.   for all i ∈ C do
2.   Csorted ← ic.SortCloudletIncreasingOrder()
3.   end for
4.   return Csorted
End function
```

4.3.2 IHDC (Increasing Hosts Decreasing Cloudlets)

This policy is like previous one but in this policy cloudlets are sorted in decreasing order. This policy is created to sort the hosts in increasing and Cloudlets in decreasing order. It sorts the hosts in such a manner that the host with lowest power consumption will be on the top of the sorted list and host with the highest power consumption will be at the last position in the list. Power consumption of host can be calculated with the help of minimum and maximum power consumption values which are shown in table 1. Now let's talk about cloudlets. According to this policy Cloudlets are sorted in such a manner that cloudlet with the longest length will be at the top of the list whereas cloudlet with shortest length will at the last position in the list.

```
Function2.1 Increasing Hosts Decreasing Cloudlets (IHDC)
Function IHDC_Hosts (R)
// Sorting the hosts in the increasing order as per power
their power consumption
5.   for all j ∈ R do
6.   Hsorted ← rj.SortHostIncreasingOrder()
7.   end for
8. return Hsorted
End function
```

Function2.2 **Increasing Hosts Decreasing Cloudlets**(IHDC)

```
Function IHDC_Cloudlets (C)
// Sorting the cloudlets in the decreasing order as per their length
    for all i ∈ C do
    C_sorted ← ic.SortCloudletDecreasingOrder()
    end for
    return C_sorted
End function
```

4.3.3 DHDC (Decreasing Hosts Decreasing Cloudlets)

As the name suggest this policy is created to sort the hosts and Cloudlets in decreasing order. It sorts the hosts in such a manner that the host with highest power consumption will be on the top of the sorted list and host with the lowest power consumption will be at the last position in the list. Power consumption of host can be calculated with the help of minimum and maximum power consumption values which are shown in table 1. Now let’s talk about cloudlets. According to this policy Cloudlets are sorted in such a manner that cloudlet with the longest length will be at the top of the list whereas cloudlet with shortest length will at the last position in the list.

Function3.1 **Decreasing Hosts Decreasing Cloudlets** (DHDC)

```
Function DHDC_Hosts (R)
// Sorting the hosts in the decreasing order as per power their power consumption
    for all j ∈ R do
    H_sorted ← rj.SortHostDecreasingOrder()
    end for
    return H_sorted
End function
```

Function3.2 **Decreasing Hosts Decreasing Cloudlets**(DHDC)

```
Function DHDC_Cloudlets (C)
// Sorting the cloudlets in the decreasing order as per
their length
    for all i ∈ C do
    Csorted ← ic.SortCloudletDecreasingOrder()
    end for
    return Csorted
End function
```

4.3.4 DHIC (Decreasing Hosts Increasing Cloudlets)

This policy is like previous one but in this policy cloudlets are sorted in increasing order. This policy is created to sort the hosts in decreasing and Cloudlets in increasing order. It sorts the hosts in such a manner that the host with highest power consumption will be on the top of the sorted list and host with the lowest power consumption will be at the last position in the list. Power consumption of host can be calculated with the help of minimum and maximum power consumption values which are shown in table 1. Now let's talk about cloudlets. According to this policy Cloudlets are sorted in such a manner that cloudlet with the shortest length will be at the top of the list whereas cloudlet with longest length will at the last position in the list.

Function4.1 **Decreasing Hosts Increasing Cloudlets** (DHIC)

```
Function DHIC_Hosts (R)
// Sorting the hosts in the decreasing order as per power
their power consumption
    for all j ∈ R do
    Hsorted ← rj.SortHostDecreasingOrder()
    end for
    return Hsorted
End function
```

```
Function4.2 Decreasing Hosts Increasing Cloudlets(DHIC)
```

```
Function DHIC_Cloudlets (C)
// Sorting the cloudlets in the increasing order as per
their length
    for all i ∈ C do
    C_sorted ← ic.SortCloudletIncreasingOrder()
    end for
    return C_sorted
End function
```

4.3.5 IC (Increasing Cloudlets)

This policy is created to sort Cloudlets in increasing order whereas hosts are not sorted at all. According to this policy Cloudlets are sorted in such a manner that cloudlet with the shortest length will be at the top of the list whereas cloudlet with longest length will at the last position in the list.

```
Function5.1 Increasing Cloudlets(IC)
```

```
Function IC_Cloudlets (C)
// Sorting the cloudlets in the increasing order as per
their length
    for all i ∈ C do
    C_sorted ← ic.SortCloudletIncreasingOrder()
    end for
    return C_sorted
End function
```

4.3.6 DC (Decreasing Cloudlets)

This policy is like previous one but it sorts the Cloudlets in decreasing order whereas hosts are not sorted at all. According to this policy Cloudlets are sorted in such a manner that cloudlet with the longest length will be at the top of the list whereas cloudlet with shortest length will at the last position in the list.

```
Function6.2 Decreasing Cloudlets(DC)
```

```
Function DC_Cloudlets (C)
// Sorting the cloudlets in the decreasing order as per
their length
    for all i ∈ C do
    C_sorted ← ic.SortCloudletDecreasingOrder()
    end for
    return C_sorted
End function
```

4.3.7 IH (Increasing Hosts)

As the name suggest this policy is created to sort the hosts increasing order whereas cloudlets are not sorted at all. It sorts the hosts in such a manner that the host with lowest power consumption will be on the top of the sorted list and host with the highest power consumption will be at the last position in the list. Power consumption of host can be calculated with the help of minimum and maximum power consumption values which are shown in table 1.

```
Function7.1 Increasing Hosts(IH)
```

```
Function IH_Hosts (R)
// Sorting the hosts in the increasing order as per power
their power consumption
    for all j ∈ R do
    H_sorted ← rj.SortHostIncreasingOrder()
    end for
    return H_sorted
End function
```

4.3.8 DH (Decreasing Hosts)

This policy is like previous one but it sorts the Hosts in decreasing order whereas cloudlets are not sorted at all. As the name suggest this policy is created to sort the hosts decreasing order. It sorts the hosts in such a manner that the host with highest power consumption will be on the top of the sorted list and host with the lowest power consumption will be at the last position in the list. Power consumption of host can be calculated with the help of minimum and maximum power consumption values which are shown in table 1.

```
Function8.1 Decreasing Hosts(DH)
Function DH_Hosts (R)
// Sorting the hosts in the decreasing order as per power
their power consumption
    for all j ∈ R do
    H_sorted ← rj.SortHostDecreasingOrder()
    end for
    return H_sorted
End function
```

4.3.9 Default Policy

This policy will work when no other policy is selected. In this policy, no sorting is done at all. The list of both host and cloudlets are in random order.

4.4 Performance Evaluation Metrices

For the evaluation of the Energy aware resource provisioning and VM allocation polices some metrices have been created and are discussed in detail in the results and discussions section.

1. **Average Energy Consumption:** While executing the cloudlets, the average energy consumed by the provisioned resources are stored into this metrices. The given below equation shows how its calculated. The summation runs for i (cloudlet) to n where n= total number of cloudlets. AEC is calculated when Pji (power consumption of node j while running cloudlet i) is multiplied by Tli (task or cloudlet length) and divided by n.

$$\textbf{AEC} = \frac{\sum_{i=0}^{n} Pji * Tli}{n}$$

2. **Average Cost:** The average cost of running the cloudlets on the provisioned resources is stored into this metrices. The give below equation show how average cost is calculated. Summation runs for i (cloudlet) and j (host) where n= number of cloudlets and m= number of hosts. We get the AC when Cj (host cost) is multiplied by l_i (cloudlet length) divided by total number of cloudlets.

$$\textbf{AC} = \frac{\sum_{i=0}^{n} \sum_{j=0}^{m} Cj * li}{n}$$

3. **Average Make span:** It stores the average make span time of the cloudlets which is calculated by subtracting the cloudlet execution start time from the cloudlet execution finish time. The give below equation show how average make span is calculated. Summation runs for i=0 to n when n= number of cloudlets. We get AMS when Si (cloudlet execution starts time) is subtracted from Fi (cloudlet execution finish time) divided by total number of cloudlets.

$$\textbf{AMS} = \frac{\sum_{i=0}^{n} (Fi - Si)}{n}$$

Results shown and discussed are the average of 5 simulations with confidence interval of 95%. All the simulations are done using 1000 cloudlets or user tasks where each cloudlet length can range from 70000 to 100000.

4.5 Result and Discussions

In this section, all the results of various policies from simulations are compared to each other. Figure 4 shows the bar graph of average energy consumption according to different policies. Figure

5 shows the average cost incurred during simulations according to all policies. Figure 6 shows the bar graph of average make span of cloudlets according to different policies.

If we talk about figure 4 it can be clearly said that policy DHIC has the lowest energy consumption as compared to other policies whereas policy IHDC has the maximum energy consumption. Also, the results of policy IHIC is very similar to the IHDC, there is 0.01% difference. And in both the policies the hosts were sorted in the increasing order. The difference between policy IHIC and IH is also only 0.01% and in both the policies hosts were sorted in increasing order.

The results of Policies DHDC, IC, DH are very good as compare to IHIC, IH, IHDC and default policies. The average energy consumption of policies IHDC and IH is 0.26% more as compare to DHIC. In conclusion, it can be said that IHDC and IH are the worst policies whereas DHDC, DH are the good policies and DHIC is the best option regarding the average energy consumption.

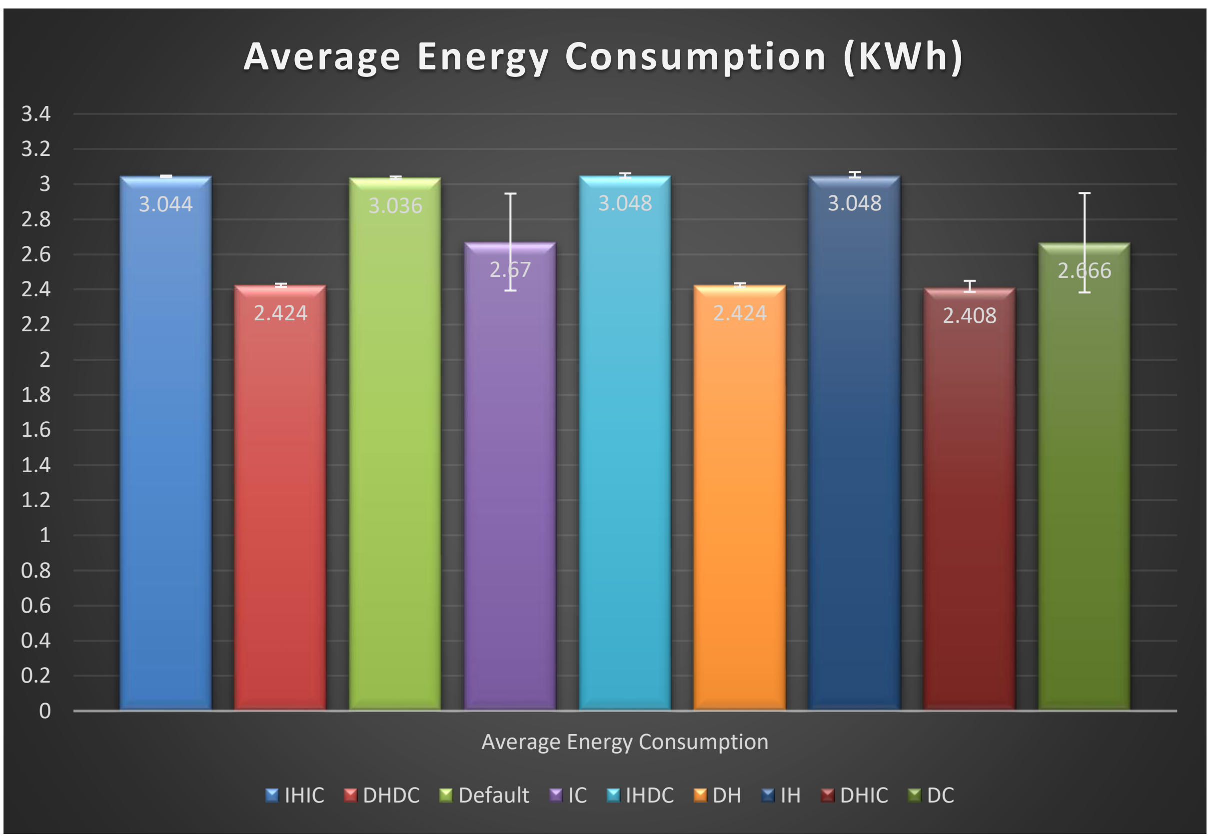

Figure 4: Average Energy Consumption

Figure 5 shows the average cost of running the 1000 user requests under different policies. The average cost result trend of different policies is similar to the result trend of the average energy consumption. The result of policies DHDC, IC, DH and DHIC are much better as compare to other policies. Policies IHIC and IHDC are 0.26% more costlier as compare to policy DHIC. The results of policies IHIC and IHDC are almost similar in both the policies the hosts were sorted in the increasing order. The result of policies DHDC and DH are exact same to each other and in both the policies the hosts are sorted in decreasing order.

And the result of these policies is very similar to the DHIC policy there is only 0.001% difference and in this policy also the hosts were sorted in the decreasing order as per their power consumption. The result of policies IHIC, IHDC, IH and default is not good in respect to average cost. There policies are much more costlier as compare to DHDC, DH and DHIC policies.

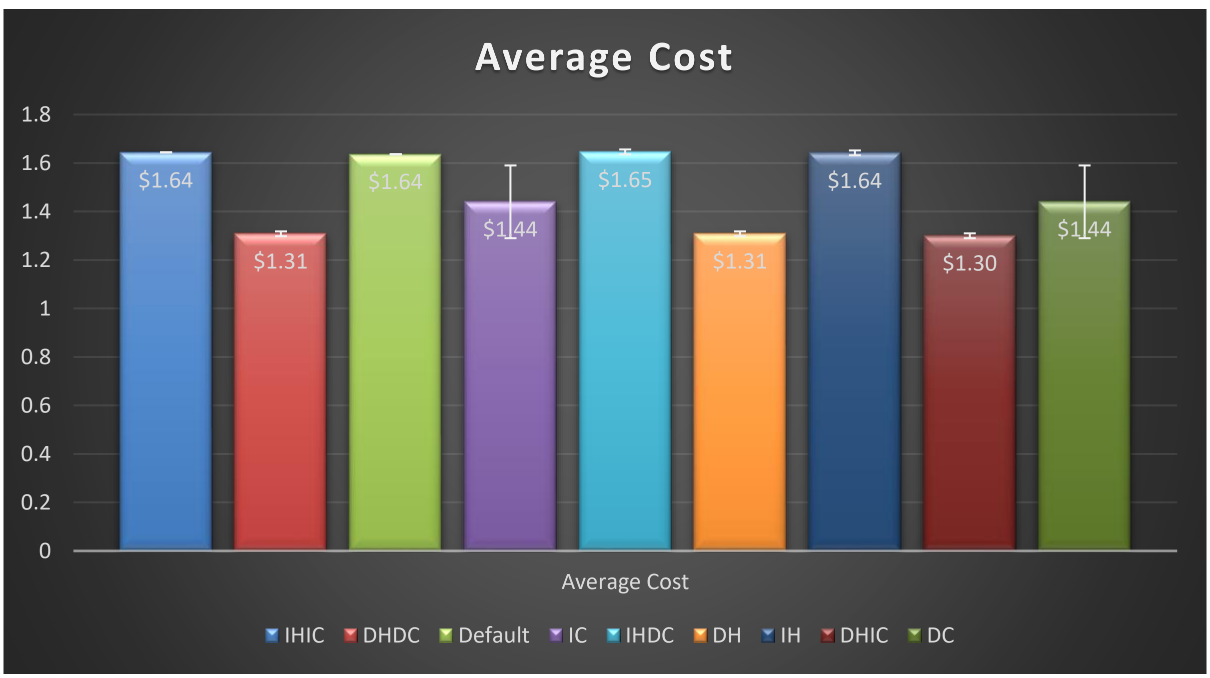

Figure 5: Average Cost

The difference between default, DHDC and IHDC, DH is of 0.25% . On one hand, if compare the results of IC and DC they are exact same which means that sorting of cloudlets doesn’t matter because in IC policy cloudlets were sorted in increasing order where as in DC cloudlets were sorted in decreasing order. But on the other hand, if we compare the results of DH and IH we can see that there is a huge difference between them unlikely the result of IC and DC which means the sorting of hosts matters a lot. The actual difference between DH and IH is 0.25% which means if your hosts are sorted in the decreasing order then you can save 0.25% on average costs which is a huge amount on a large scale.

In conclusion, from the perspective of implementation it can be said that the policies in which hosts are sorted in the decreasing order are much more cheaper as compare to other policies specially with those policies in which hosts are sorted in the increasing order.

Figure 6 shows the average make span. Make span of a cloudlet is the total time a cloudlet waited in the queue for its execution. The result tread of the average make span is not similar with the result tread of the average cost and energy consumption. Here it can be seen clearly that default policy has the minimum make span time but this can be ignored because in default there is no pre-processing or not sorting of the hosts and cloudlets. The hosts and cloudlets are selected in the random order under the default policy. Otherwise the result of the policy DHIC is good as it was in average cost and average energy consumption. The result of policies IHIC and IHDC are similar to each other as they were in other bar graphs discussed earlier. The results of policies DHDC and DH are also similar but now the results are good as they were in other bar graphs. Now the results of IHIC, IHDC, DHDC and DH are in the same range.

If we ignore the default policy the results of IC, DC and DHIC are good from the perspective of make span because these policies have the minimum make span time as compare other policies.

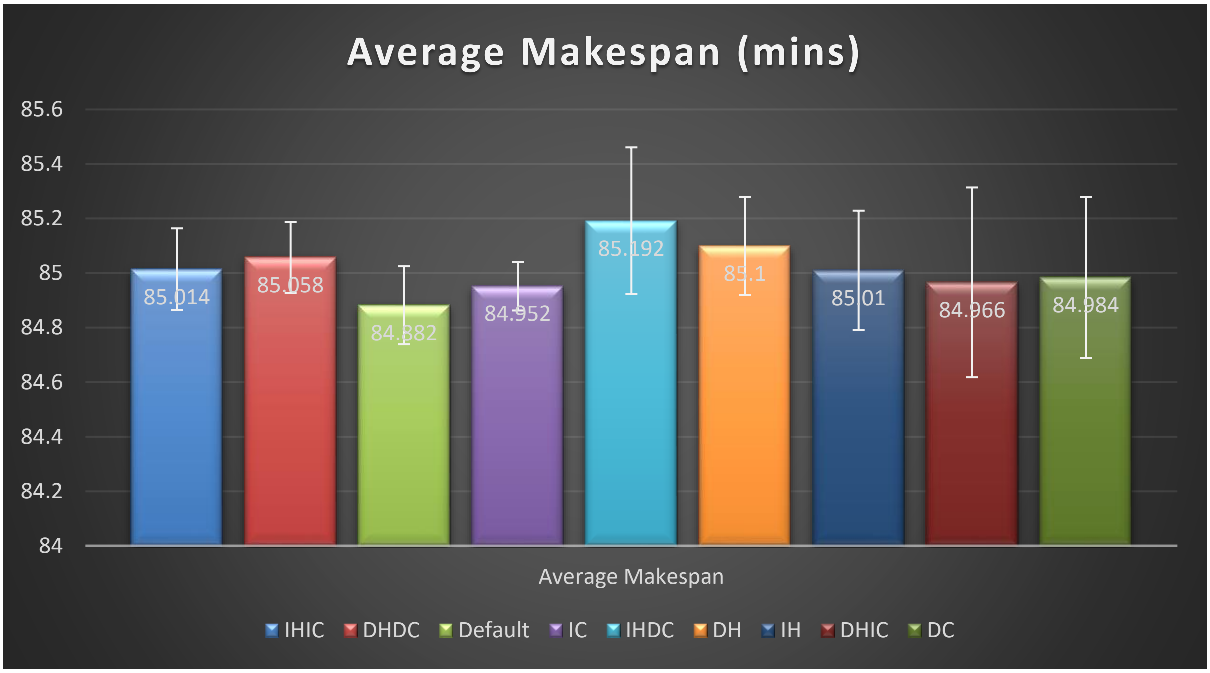

Figure 6: Average Make span

The result of the IC and DC policies are similar like they were in other evaluation metrices. This shows that order of cloudlets doesn't effect the make span time of cloudlets because in IC policy cloudlets were sorted in the increasing order whereas in DC policy they were sorted in the decreasing order. The result trend of DH and IH is the same as it was in other evaluation metrices. Which again proves the theory that order of hosts matter and it effects the results drastically.

In conclusion, from the perspective of implementation it can be said that policies IC and DC are good as they have less make span time as compare to other policies but result of policy DHIC is also in the same range of IC and DC results and as this policy is the best policy in other performance evaluation metrices so it can also be used.

CHAPTER VI: CONCLUSION

As the research was done in two phases so the conclusion of the first phase was the input for the second phase. In first phase for conclusion it can said that cloud computing has become the one of the most popular and used technology in today's digital world. Because of pay per use service of cloud computing everyone form big companies to normal smartphone user can use this service. But as the users of cloud computing are increasing the power consumed by cloud based datacenters is also increasing. According to a research to total power consumption of all the cloud based datacenters around the world has reached to 30 billion watts (Glanz, 2012). As more power consumption leads to generation of more power and mainly power is generated by burning the fossil fuels like coal and during this process lot of greenhouse gasses has been released into the environment which negatively impact our environment. IT sector alone is responsible for the 2 percent of the carbon emissions. To solve these environmental issue big firms like apple, google and Microsoft are taking big initiatives to reduce their carbon emission. According to a research at present Apple and Google are using 87 and 37 percent of power from the green sources respectively (Florea, Sommer, & Ahmadabadi, 2016).

During first phase, after study of different cloud computing energy efficiency techniques purposed by various researchers it was found that some researchers are focusing on the software techniques while others are focusing on hardware techniques so, to have clear understanding and to cover all the energy efficiency techniques a new taxonomy has been created in which energy efficiency techniques are categorized into 3 different categories which are hardware level, software level and Datacenter level. Hardware level comprises of, different methods which helps in increment the energy effectiveness by controlling the hardware in various ways, for example, running the CPU at various power levels and deactivating unused parts. software level procedures are additionally arranged into application software level and system software level. System software level incorporates the strategies which utilize different techniques of operating system to diminish power consumption while application software level incorporates methods like virtualization and consolidation. At Datacenter level, power consumption can be diminished by taking choices identified with the geological position of datacenter, its building structure, and datacenter interior functions.

In second phase of this research 9 different Energy aware resource provisioning and VM allocation polices have been designed and implemented in the Cloudsim simulator. All the results from the simulations are discussed in the results section. The main idea in designing the policies is to sort both host and user requests or cloudlets in different ways. Overall the results of the policies which have their hosts sorted in decreasing order are much better as compare to the policies which have their hosts sorted in increasing order.

If we have to select a best policy among all the 9 policies then policy DHIC can be selected because the result of DHIC is minimum in both average energy consumption and average cost performance evaluation metrices. The difference between DHIC and polices IHIC, IHDC, IH, default is 0.26% which means these policies consume 0.26% more average energy as compare to DHIC. Whereas when hosts are not sorted at all and only user requests are sorted in different ways (policies IC and DC) then average energy consumption remain the same. If we compare average energy consumption result of policies DHDC, DH with DHIC they are very similar with only 0.006% difference and in all of these 3 policies hosts were sorted in decreasing order as per their power consumption.

The result trend of the average cost is very similar to the result trend of the average energy consumption. Here also policy DHIC outperforms all the other policies with minimum average cost of $1.30. The difference between DHIC and polices IHIC, IHDC, IH, default is approx. 0.26% which means these policies are 0.26% more costlier to implement as compare to DHIC. Whereas when hosts are not sorted at all and only user requests are sorted in different ways (policies IC and DC) then average energy consumption remain the same which is $1.44. If we compare average cost result of policies DHDC, DH with DHIC they are very similar with only 0.006% difference and in all of these 3 policies hosts were sorted in decreasing order as per their power consumption. Also, the effect of sorting the hosts in decreasing order can also be easily seen in figure 5 because the difference between DH and IH is 0.25% which means implementing IH policy will 0.25% more costlier as compare to DH policy.

The result trend of average make span is different form the result trend of the average cost and average energy consumption. However, the make span of the default policy is minimum it can be ignored as the there is no pre-processing or sorting in this policy. Otherwise the result of DHIC policy is good as it was in average cost and energy consumption metrices.

In conclusion of the research study it can be said that we have to reduce the power consumption of cloud based datacenters because of negative environment impacts of increased power consumption, energy wastage at different levels and decrease in cloud providers profits. To reduce power consumption there are many ways found through the literature but we choose to design the energy aware resource provisioning and VM allocation polices so that datacenter resources can be used effectively and efficiently. We designed total nine such policies and implemented on them cloudsim simulator and found that policy DHIC is the best because it outperforms all the other policies.

REFERENCES

Beloglazov, A., Buyya, R., Lee, Y. C., & Zomaya, A. (2011). A taxonomy and survey of energy-efficient data centers and cloud computing systems. *Advances in computers, 82*(2), 47-111.

Buyya, R., Beloglazov, A., & Abawajy, J. H. (2010). Energy-Efficient Management of Data Center Resources for Cloud Computing: A Vision, Architectural Elements, and Open Challenges. *CoRR, abs/1006.0308*.

Flinn, J., & Satyanarayanan, M. (2004). Managing battery lifetime with energy-aware adaptation. *ACM Transactions on Computer Systems (TOCS), 22*(2), 137-179.

Florea, I. C., Sommer, M., & Ahmadabadi, E. R. (2016). GREEN IS THE NEW COLOR OF ICT - A RESEARCH ON THE POTENTIAL OF ICT TO HELP FIGHT AGAINST CLIMATE CHANGE. *Calitatea, 17*(S1), 429-435.

Gao, Y., Guan, H., Qi, Z., Song, T., Huan, F., & Liu, L. (2014). Service level agreement based energy-efficient resource management in cloud data centers. *Computers & Electrical Engineering, 40*(5), 1621-1633. doi: https://doi.org/10.1016/j.compeleceng.2013.11.001

Glanz, J. (2012). Power, pollution and the internet. *The New York Times, 22*.

Kaur, T., & Chana, I. (2015). Energy efficiency techniques in cloud computing: a survey and taxonomy. *ACM Computing Surveys (CSUR), 48*(2), 22.

Meisner, D., Gold, B. T., & Wenisch, T. F. (2011). The powernap server architecture. *ACM Transactions on Computer Systems (TOCS), 29*(1), 3.

Neugebauer, R., & McAuley, D. (2001). *Energy is just another resource: Energy accounting and energy pricing in the Nemesis OS.* Paper presented at the Hot Topics in Operating Systems, 2001. Proceedings of the Eighth Workshop on.

Pallipadi, V., & Starikovskiy, A. (2006). *The ondemand governor*. Paper presented at the Proceedings of the Linux Symposium.

Quan, D. M., Mezza, F., Sannenli, D., & Giafreda, R. (2012). T-Alloc: A practical energy efficient resource allocation algorithm for traditional data centers. *Future Generation Computer Systems, 28*(5), 791-800. doi: https://doi.org/10.1016/j.future.2011.04.020

Rajkumar, R., Abeni, L., De Niz, D., Ghosh, S., Miyoshi, A., & Saewong, S. (2000). *Recent developments with linux/rk.* Paper presented at the Proceedings of the Real Time Linux Workshop.

Rodero, I., Viswanathan, H., Lee, E. K., Gamell, M., Pompili, D., & Parashar, M. (2012). Energy-Efficient Thermal-Aware Autonomic Management of Virtualized HPC Cloud Infrastructure. *Journal of Grid Computing, 10*(3), 447-473. doi: 10.1007/s10723-012-9219-2

Vardhan, V., Yuan, W., Harris, A. F., Adve, S. V., Kravets, R., Nahrstedt, K., . . . Jones, D. (2009). GRACE-2: integrating fine-grained application adaptation with global adaptation for saving energy. *international Journal of embedded Systems, 4*(2), 152-169.

Zeng, H., Ellis, C. S., Lebeck, A. R., & Vahdat, A. (2002). *ECOSystem: Managing energy as a first class operating system resource.* Paper presented at the ACM SIGPLAN Notices.

Appendix A: Simulation data used in the research study

Policies	Average Energy Consumption	Average Cost	Average Makespan
IHIC	3.044	$1.64	85.014
DHDC	2.424	$1.31	85.058
Default	3.036	$1.64	84.882
IC	2.67	$1.44	84.952
IHDC	3.048	$1.65	85.192
DH	2.424	$1.31	85.1
IH	3.048	$1.64	85.01
DHIC	2.408	$1.30	84.966
DC	2.666	$1.44	84.984