Energy Efficient Cloud Data Center

Praful Anand

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769008, Odisha, India
May 2015

Energy Efficient Cloud Data Center

Thesis submitted in partial fulfillment of the requirements for the degree of

Master of Technology

in

Computer Science and Engineering

(Specialization: Software Engineering)

by

Praful Anand

(Roll No.- 213CS3179)

under the supervision of

Prof. A. K. Turuk

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela, Odisha, 769 008, India

May 2015

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India.

Certificate

This is to certify that the work in the thesis entitled ” ***Energy Efficient Cloud Data Center*** ” submitted by ***Praful Anand*** is a record of an original research work carried out by him under our supervision and guidance in partial fulfillment of the requirements for the award of the degree of Master of Technology with the specialization of Software Engineering in the department of Computer Science and Engineering, National Institute of Technology, Rourkela. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Prof. Ashok Ku. Turuk
Associate Professor
Department of CSE
National Institute of Technology
Rourkela-769008

Place: NIT,Rourkela-769008
Date: 26-05-2015

Acknowledgment

First of all, I would like to express my deep sense of respect and gratitude towards my supervisor Prof Ashok Kumar Turuk, who has been the guiding force behind this work. I want to thank him for introducing me to the field of Cloud Computing and giving me the opportunity to work under him. His undivided faith in this topic and ability to bring out the best of analytical and practical skills in people has been invaluable in tough periods. Without his invaluable advice and assistance it would not have been possible for me to complete this thesis. I am greatly indebted to him for his constant encouragement and invaluable advice in every aspect of my academic life. I consider it my good fortune to have got an opportunity to work with such a wonderful person.

I wish to thank all faculty members and secretarial staff of the CSE Department for their sympathetic cooperation.

During my studies at N.I.T. Rourkela, I made many friends. I would like to thank them all, for all the great moments I had with them.

When I look back at my accomplishments in life, I can see a clear trace of my family's concerns and devotion everywhere. My dearest mother, whom I owe everything I have achieved and whatever I have become; my beloved father, for always believing in me and inspiring me to dream big even at the toughest moments of my life; and my brother and sister; who were always my silent support during all the hardships of this endeavor and beyond.

Praful Anand
213CS3179

Abstract

Cloud computing has quickly arrived like a deeply accepted computing model. Still, the exploration and investigation on cloud computing is at a premature phase. Cloud computing is facing distinct issues in the field of security, power consumption, software frameworks, QoS, and standardization. The management of efficient energy is one of the most challenging research issues. The key and central services of cloud computing system are the SaaS, PaaS, and IaaS. In this thesis, the model of energy efficient cloud data center is proposed. Cloud data center is the main part of the IaaS layer of a cloud computing system. It absorbs a big part of the aggregate energy of a cloud computing system. Our goal is to supply a better explaining of the design issues of energy management of the IaaS layer in the cloud computing system. Servers and processors are the main component of the data center. Virtualization technologies that are the key features of the cloud computing environment provide the ability for migration of VMs between physical servers of the cloud data centre to improve the energy efficiency. This is called dynamic server consolidation that has direct impact on service response time. Energy efficient cloud data center reduces the overall energy consumed by the data center. This results in, reduction of cost incurred by the data center, long life of hardware components, green IT environment, and making more user friendly. Many VM placement techniques, server consolidation techniques have been proposed. They do not show optimal solution in every circumstances. They show optimum result only for a certain data set. They did not consider both VM placement and its migration simultaneously. They did not attempt to minimize the VM migrations during server consolidation. Still, forceful consolidation can result in the performance degradation and may lead the SLA negligence. So, there is a trade-off between performance and energy. A number of heuristics, protocols and architectures have explored and investigated for server consolidation using VM migration to reduce energy consumption. The primary objective is to minimize the overall energy consumption by servers without violating the SLA. Our proposed model and scheme show the better result at most of the data set. It is based on virtualization technique, VMs, their placement and their migration. Our study focuses on problems like huge amount of energy consumption by server and processor. So, here energy consumption is reduced without violating SLA and to meet certain level of QoS parameters. Server consolidation is performed with minimum number of VM migration. Here, maximum utilization of resources is tried to achieve, but utilization of resources is not compared with the existing scheme. Our scheme may show different better result for different configuration of the data center for the same data set. Problem is formulated as a knapsack problem. Proposed scheme inherits some feature from heuristics approach like BF, FF, BFD, and FFD. These are used for greedy-bin-packing problem. For simulation, input data set is taken as random value. These random values are general data set used in real scenario and by the existing scheme. From simulation, it is found that proposed model is achieving the desired objectives for a number of data set, and for another data set, some percentage loss of objectives is occurring.

Keywords: Server consolidation, Cloud data center, Cloud computing, Energy efficiency, IaaS, Resource utilization, VM migration, Green IT, Virtualization.

Contents

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

Today, Cloud computing is an emerging technology in academia and industry. It is adopted as research, course work, and providing different applications to customers as services. Cloud computing is a very novel computing technology, a distributed kind of internet based computing in which the resources like infrastructure, platform, software etc. are provided by the cloud service providers as a scalable, reliable, fault tolerant service to the customers on their request on a pay per usage basis and provides advantages like on-demand access, broad network access, rapid elasticity etc. Cloud computing is inherited from distributed computing with virtualization as additional technique. With virtualization traditional data center is converted into cloud data center. 3 key characteristics of Cloud computing are virtualization, pay-on-demand, and scalability. It is also called as utility computing due to payment on incremental and request foundation model. Companies like Amazon Web Service (AWS), Google, Salesforce.com, IBM, Microsoft and Oracle have converted their traditional data center to cloud data center. As a starter Amazon is the one who started giving cloud services in 2006 with Elastic Compute Cloud(EC2) Amazon's total revenues are 61 billion dollar followed by Microsoft and Rackspace.com. Google has started with Google compute engine, which is much faster, than AWS.

Virtualization enables infinite capacity in the cloud data center. It supports better isolation between services by encapsulating concurrent services into different VMs and manageability. Virtualized resources, dynamic deployment of VMs, on-demand resource provisioning among the hosted VMs, and so on are applying virtualization. These works result in the improvement of the performance of the virtualization, resource utilization, and consuming energy. So, Virtualization is being extensively used for server consolidation in the cloud computing.

A company or organization having large infrastructure with virtualization save energy 68-87% and results in reduction of carbon emission. It is an approach towards green environment. Now, Companies in U.S would get the saving in energy of 11.0 billion dollar and reduction in carbon emission of 83.5 million metric tonnes per year by 2022. It is comparable to over 14.6 million vehicle's carbon emission per year. Cloud data center provides services at a lower cost than traditional data center. Cloud data center has few applications , homogeneous h/w environment , standardized management tools and s/w architecture , simple workload , and minimal application updating and patching. Infrastructure layer is the most important layer of the cloud computing architecture. Energy The management of energy or to build energy model is one of the demanding issue in the cloud data center of the infrastructure layer. For building energy efficient cloud data center, we will optimize each and every component of the data center e.g. optimize processor, server, memory, hard disk, and cooling system. So, different techniques are proposed for energy efficient cloud data center. Server consolidation with VM migration, Optimal VM placement, Energy efficient resource allocation are some of them and these are discussed here. Energy efficiency is achieved by running minimum number of physical servers within data center with minimal and acceptable violation of SLA. There are many QoS for cloud computing. QoS are many for many users with various profiles and different needs. QoS like throughput, delay, jitter, loss rate, bandwidth etc is highly dependent on application and image resolution, sound quality, appropriate language etc are dependent on the end user. There is a trade-off between these QoS.

1.1 Literature Survey

In this section, a review on the study of different techniques for server consolidation using VM migration is presented. Different authors applied different strategies for server consolidation and VM placement with some variation in their objectives. They created their own model for server consolidation. They took their data from different sources like google data center, TU Berlin university, amazon data center etc. or with mathematical equation like Poisson Distribution, cumulative inverse function distribution, cumulative distribution function, t-distribution, normal distribution, Gaussian distribution etc and applied their model on these data.

Author	Work Done
Yao et al.[31]	They defined DVS model. Jobs are scheduled in finite and specified period of time on optimal processor.
Cheng and Goddard [6]	They considered input/output resources. This results in larger processing, and computing time, additional power consumption on system and processor.
Qiu et al.[23]	They suggested an algorithm for parallel loops across n number of processors. Processors's voltage and frequency were changed according to the current workload. Their model minimized efflux, changing, and progressive energy through DVS and ABB. Their model strictly considered deadline, priority and sequence defined for task operations.
Chen et al.[4][5]	They took account a framework for tasks that have specified constraints on homogeneous multi-processor energy-efficient processor and suggested a 1.13-approximation algorithm. They also suggested a 1.283-approximation efflux-awake algorithm for only tasks having only timing constraint to minimize energy consumption.
Pinheiro et al.[22]	They proposed LC technique. LC solves the problem of overloaded and underloaded servers by distributing the load among servers dynamically. In such a way underloaded server is released and switched off.
Jejurikar et al.[13]	CPU frequency is increased, CPU voltage is decreased, CPU is shutdown in case of no workload to minimize power consumption.
Zhu et al.[33]	Suggested a scheduling algorithm. Slack is shared among n number of processors of embedded system. A queue that contains ready tasks is declared globally. Tasks are selected from queue in such a way that unutilized time of one task is utilized to increase the completion time of another task by slowing down the speed of CPU.
Heath et al.[11]	Considered the heterogeneous environment within the cloud data-center.
Elnozahy et al.[8]	power supplies are more efficient relatively low. Their work investigated and explored IVS and CVS.
Younge et al.[32]	In a multi-core system, they showed that increase in the number of cores does not proportionate to the rise in energy consumption.
Rodero et al.[24]	They minimized energy consumption of the HPC system that have virtual environment in online manner.
Laszewski et al.[30]	suggested a energy-aware algorithm that schedules VMs in a cluster of server that is based on DVS.
Stoess et al.[28]	framed a new 2-leveled energy management model that takes into account of VM system based on hypervisor.
Kim and Buyya et al.[15]	They took into account QoS requirements of clients e.g. total cost, soft real time constraints of services of applications for energy aware placement of VMs in the data center that is based on DVS.
Kusic et al.[16]	Proposed a model for dynamic allocation of resources in the cloud data center.
Meisner et al.[18]	Their technique carried out the cloud data center between low and high power states frequently to conserve energy among server cluster.
Lee and Zomaya[17]	They proposed 2 heuristic based power-aware task consolidation algorithms to maximize resource utilization. They took into account energy consumption of server cluster in both the active and idle state.
Jing et al.[14]	They created a green cloud by state-of-the art technique and proposed model for energy conservation in the IaaS layer of cloud.

Jyothi et al.[25]	survey on different techniques for energy efficient server consolidation using VM live migration.
Ferreto et al.[9]	Heuristics approaches and LP formulation rank VMs with constant and stable capacity to restrict the VM migration.
Murtazaev and Oh[20]	Sercon algorithm which minimized the overall number of utilized PMs, and the overall number of VM migration.
Song et al.[26]	They modelled the cooperation between requests with different QoS parameters, and existing power capacity between simultaneous application services. The name of model is utility-analytic model that is based on-internet . Requests and capacities are placed in queue. Their time and period for scheduling is forecasted.
Singh and Hemalatha[21]	A VM Placement Technique known as BASIP to overcome the issue of deadlock by using a banker algorithm with Stochastic Integer Programming.
Eslam et al.[19]	VM placement technique to minimize the data transfer time consumption between VMs and thus helps in optimizing the overall application performance.
Chebiyyam et al.[3]	Motivation, Benefits, and Proposed algorithm for server consolidation.
Clark et al.[7]	Proposed cost of migration as downtime of service and the overall VM migration time. These should be reduced.
Spieksma et al.[27]	Their approach gave optimal solution for some data set and proposed an algorithm that globally optimized their problem but, showed exponential time complexity at worst case and a heuristic approach.
Toth et al.[2]	Proposed an algorithm that produced the lower bound for the solution of 2DVPP problem.
Hyser et al.[12]	VM placement and its live migration.
Gmach et al.[10]	proactive trace-based workload VM placement, migration controller scheme, policies, efficiency, and performance metrics to address the efficient management of virtualized resources.
Verma et al.[29]	Application VM placement policy in the virtualized system while considering the migration cost and power cost.
Bobroff et al.[1]	minimized the quantity of needed physical resources and the rate of SLA negligence using dynamic server consolidation and migration of VM algorithm.

1.2 Architecture of cloud computing

As shown in Fig.1.2, here the most standard and hierarchical architecture of the cloud computing is discussed. It consists of the three basic layer IaaS, PaaS, and SaaS. The bottom most layer is the Infrastructure as a Service, next is the Platform as a Service, and the upper most layer is the Software as a Service.

But in some places we find one additional layer development as a service(DaaS) layer between SaaS and PaaS. DaaS deals with mainly web based development tools that can be shared among community. It is not a new layer and is derived from traditionally delivery of development tools.

IaaS, PaaS, and SaaS layers provide infrastructure (CPU, Memory, Disk, Bandwidth etc.) as a service, platform (Python, Java, .Net etc.) as a service, software (Gmail, Youtube, Facebook etc.) as a service to customers respectively on subscription based and pay-as-you-go model via the internet. 4 general advents of Software as a Service are poly-items, individual item, curve occupancy, and poly-holders. In Fig.1.1, each layer and its sub layers with examples, its scope, and concerned fields are discussed.

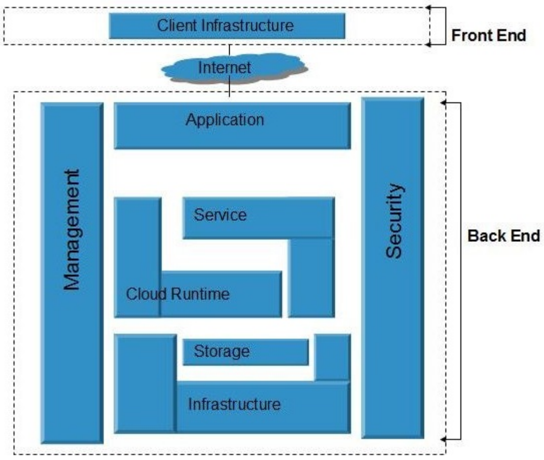


Figure 1.1: Detailed Cloud Computing Architecture

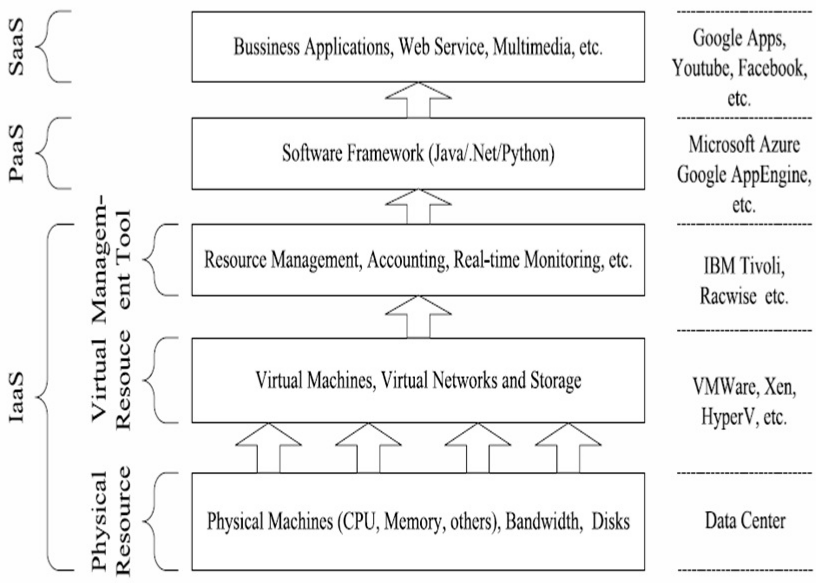

SaaS
Bussiness Applications, Web Service, Multimedia, etc.
Google Apps, Youtube, Facebook, etc.
PaaS
Software Framework (Java/.Net/Python)
Microsoft Azure Google AppEngine, etc.
IaaS
Virtual Managem-ent Tool
Resource Management, Accounting, Real-time Monitoring, etc.
IBM Tivoli, Racwise etc.
Virtual Resouce
Virtual Machines, Virtual Networks and Storage
VMWare, Xen, HyperV, etc.
Physical Resource
Physical Machines (CPU, Memory, others), Bandwidth, Disks
Data Center

Figure 1.2: Abstract Cloud Computing Architecture

1. Front End:- This part is accessed and seen by clients. Fat client, thin client, and mobile device these interact with cloud data storage via middleware like web browser. Thin client equipment runs via the internet. If internet slows down, a single point of failure is created, and the corresponding equipment is considered useless.

2. Back End:- It is a cloud of various computers, servers, data storage system. These are responsible for online storage and provide resources among multiple customers. These can be installed in public, private, community, and hybrid cloud. These should provide agility, flexibility, scalability, multi customer support, security, traffic control, and protocol known as middleware.

3. Cloud based delivery:- IaaS, PaaS, DaaS, and SaaS these are discussed above.

4. Network:- Intranet, Internet as discussed below:

The cloud network layer provides:

1. High bandwidth with low latency that allows users to access their application services and data without any interruption.

2. Agile network, that provides ICT application services and enables the accessing of resources. It can shift frequently and effectively between servers.

3. Network Security is important with multi - tenancy i.e. with multi customer environment.

1.3 Issues and Challenges

In this section, Challenges that are being faced by IT organization whose applications, services are migrated to cloud are discussed. As cloud services have been replaced current desktop services, the design, configuration, policies, equipment, plan within organizations are needed to be change, What future directions should be adopted within IT organization? and so on. These are requirements for studying this section for running business in optimum way.

1.3.1 Organizational Change

1. Cloud service providers have no custom supports to the customers. So, organization needs a central IT department that fully concentrates on providing supports of their services and products to customers.

2. Cloud service providers have not high level. Their levels are same as before migration

to the cloud. So, they need to change their working processes.

3. System administrators have control over some aspect of cloud based system. So, political rules and system administrators should not interfere on the cloud based system. Cloud service provider will directly or indirectly contact with end users.

1.3.2 Costs

1. Generally, hardware, software, and IT support costs are considered. Economic policies for application migration, procurement policies play a major role to accommodate the application of cloud computing at the enterprise level. These should collaborate with economists and business management schools and colleges to make cloud computing at the enterprise level.

2. Costs, Operational budgets, and Procurement costs need specific signatories to approve the deployment of enterprise in the cloud computing. But this leads against on-demand systems. So, enterprise may adopt cost limits and predict usage patterns, and IT usage peaks within the organization.

3. Allocation of costs to organization in isolation. This issue can be resolved by auditing, optimizing financial regulations, and developing tools and techniques.

1.3.3 Security, Legal, And Privacy

Individual manger worry about the failure of delivery of the products that have been promised. Security, legacy, and privacy issues are the major risks in the cloud computing environment. So, risk avoidance is needed. Other security issues are data loss, phishing, multi-customer algorithm, and chain of computing components in cloud computing. They need novel technique to resolve these issues.

1. Enterprises to be alert about security and regulatory issues of their applications that had shifted to the cloud. This is due to the complexity of their system, moving their some data under jurisdiction while some move to the cloud.

2. The model of legal and security issues may be developed in the near future by our cloud service providers and researchers. These model will be eventually granted and followed by our government. But, currently DOS attack is critical problem in the cloud computing. Research is ongoing to resolve the problem of the DOS attack.

3. The system that has been deployed in the cloud has less security due to the lack of control. Legal issues should be resolved before applying cloud computing technique.

Security issues when data is moving. This cab be eliminated by applying constraints. Time and cost incurred in data migration is a major concern. So, the practical issues that influence data migration should be investigated.

4. When sensitive data of a sector is moving outside the organization, certain requirements and regulations of sector should be followed and should not involved any risks.

5. When the company uses in-house and cloud based systems then there is a need to control the behavior of every component of the organization. Every user should be aware by these issues. These challenges can be solved by moving every application and service of the company on the cloud.

1.3.4 Issues in cloud computing to support IoT

Here, some are challenges that are requirement for IoT.

1. To provide dynamic resources for the system enable application flexibility.

2. Testing for the system ensures QOS parameters to support real time needs.

3. The scalability of IaaS layer enables required exponential growth of requests.

4. Reliable structure of cloud computing provides the availability of applications.

5. Privacy and security issues enhance the data protection and user privacy.

6. The efficient and effective management of energy resources mean efficient power consumption of applications.

7. The federation in the cloud environment executes the applications close to end users.

8. Interoperable and portable cloud establishes open cloud ecosystem with concerned characteristics.

1.3.5 Other Issues

Migrating to the cloud reduced infrastructure cost but increased communication cost. The flexible chain of pool made cost analysis more complicated. Re-design and re-development of the model is needed. Adding new features require more security, cost, more customization. So, viable strategy is needed to SaaS cloud provider for portability and sustainability. SLA specification should provide maximum percentage of intentions of customer and be easy to verify, evaluate, and enforced by resource allocation policy. Different layers e.g. IaaS, PaaS, and SaaS provide different SLA specifications. So, more advanced SLA specification need customer feedback for customized evaluation of SLA in future. There are a number of services and products of the organization. So, the migration of which one enables more security of its own. Hazy Cloud makes a problem for users to select a cloud. Interoperability solves this problem. Many steps of tier are involved in interoperability. First, develop the computing components and IT resources. Second, deploy many bordering methods to cloud services. Standardization helps to achieve interoperability.

1.4 Motivation

1. Information can be accessed by multiple computing devices. 33% companies adopted cloud computing. It has 33% mobility and 17% cost effective characteristics.
2. A very few companies went to down, while most of them is growing exponentially after adopting cloud computing.
3. A large number of business firms that are 82% of all companies are saving money with cloud. But, saving is not so much.
4. Current business is going towards cloud. 65% of companies have subscribed for cloud computing last one year.
5. It has reduced energy consumption that has resulted in the reduction of carbon emission and making a green environment.
6. 93% of companies that adopted cloud computing technique have been grown its one or more field of IT department.
7. 52% of them noticed their increased efficiency of data center, and utilization of resources. 47% of them observed their low operating cost.
8. 80% of them noted their IT department improvement within six months. 74% of small scale business there is no resistance to move to cloud within the organization.
9. Many countries are preparing their employees for cloud adoption. In which 97% of Brazilian companies prepared their employees.
10. Very less change occurred in the data security policies after adopting cloud computing. only 25% companies announced more anxiety after adopting cloud computing. 47% of Singapore companies have more worry while 47% of Brazilian companies have less worry about their data security policies.
11. 50% of USA government IT workers are working towards cloud related fields. 48% USA government agencies moved one or more work flow to the cloud following ”cloud-first” policy.

12. Cloud data center spends vast amount of energy in a cloud computing system.

13. Amazon's Data Center consumes 42% cost of the total budget in which 53% costs are incurred by only servers of data center.

14. More power requirement incurs more cost to set up a cloud system, reduces profit of cloud service provider, makes environment polluted, heating of the h/w component.

15. To achieve energy efficient data center there should be a good knowledge of cloud computing, data center, virtualization etc.

16. Around 30% servers are under-utilized but spend remarkable quantity of energy per year.

17. Virtualization enables infinite capacity in cloud data center. Cloud data center provides services at a lower cost than traditional data center.

19. Cloud data center has few applications, homogeneous h/w environment, standardized management tools and s/w architecture, simple workload, and minimal application updating and patching.

20. Cloud customer faces the problem of protection of their own data and application. This will largely determine whether, and upon which underlying conditions, business-related sensitive data and applications may be stored in the cloud. So, different security measures are taken for private and public cloud depending on their threats profile. In IaaS layer, There is a need for improved pass of money, base for unknown provision planning, clear metering, and the administration of self-service.

1.5 Objectives

1. Minimize energy consumption of data center without violating SLA.

2. To meet a certain limit of QoS parameters.

3. Maximum utilization of resources.

4. Maximum number of requests can be serviced.

5. Minimize the number of used PMs.

6. Minimize the number of VMs migration.

7. Minimize the VMs migration time.

8. Maximize the server utilization of cluster.

1.6 Thesis Organization

Remaining part of the thesis is constructed as follow:

- In chapter-2 processor and server models are designed to consume minimum energy without violating SLA or violating SLA with a very less percentage and meet a certain level of QoS parameters. Processor model is based on DVFS technique, DVS, and DPM technique. Server model is based on LC technique, virtualization, dynamic resource allocation, and δ-Advanced-DVS policy. VM placement with online VM migration algorithm that is based on defined processor, memory, and server model is proposed to meet the objective. Overall efficiency of the algorithm is different for different data set.

- In chapter-3 Server consolidation is performed using VM migration. We have some initial state of the cluster. Model and server consolidation algorithm are proposed to minimize number of used PMs and number of VMs migration. Nodes are almost homogeneous and VM's capacity does not change during server consolidation. Energy is minimized with very less cost incurred by VM migration.

- In chapter-4 overall thesis conclusion and future works are addressed.

CHAPTER 2

VM PLACEMENT AND ONLINE SERVER CONSOLIDATION IN THE CLOUD DATA CENTER

2.1 Introduction

Cloud data center is a major component of the IaaS layer of the cloud computing. It consumes a huge amount of energy. This incurs to more investment, more infrastructure and computing cost, less profit to cloud service provider and makes cloud data center not Eco-friendly. So, a number of research has been done and currently, researches are ongoing to minimize energy consumption by data center. Data center consists of physical servers, processors, memory, storage, and cooling system. So, optimize each and every component to reduce energy consumption without violating SLA or violating the SLA to a threshold value. It is also necessary to keep in mind about QoS parameters like throughput, bandwidth, delay, jitter, loss rate. The QoS parameters are image resolution, sound quality, appropriate language etc. for end users. There is a huge number of underutilized servers in such a data center and create a big issue for cloud service providers. When dealer's application services execute in separation to meet security issues. Then it generally results in under utilization of servers. Physical servers consume a huge amount of energy for servicing requests. In this chapter a model is proposed to minimize energy consumption by data center with maximum utilization of resources. This model adopts some properties of heuristic algorithms like FF and BF. FF and BF have objective to minimize the number of PSs, but our algorithm has objective to maximize the resource utilization as well. We optimized each component by applying existing and our proposed techniques. We proposed a algorithm for server consolidation using VM migration technique. Server consolidation means that service the requests with minimum number of used physical servers at a time. Server consolidation provides advantages like total reduction in CPU cycles , total reduction in administration and energy prices, resultant reduction in shifting, connection, hard disk, memory and repository and reduction in carbon emission from data center.

Virtualization enables server consolidation in a very efficient and effective way. One of the major application of virtualization technique is server consolidation. It also supports maximum utilization of servers. It offers better isolation, manageability, resource provision on demand basis for server consolidation. Virtualization is a technique that makes virtual version of anything e.g. OS, Physical servers, Storage, and network resources in IT sector. Resource virtualization, dynamic deployment of VM, resource allotment according to pay-as-you-go model among the assigned VMs, and so on advance to upgrade in the achievement of resource utilization and virtualization. Performance unpredictability, proof of requests arrival distribution, and the performance of application services are phenomenons in virtualization. A similar aspect of a chain of resources is supported by virtualization. A server of cloud data center can accommodate a number of computing resources encapsulated as a VM. These VMs are isolated from each other. It makes applications disjoint from the system's hardware, and provides simple administration of VMs to data center managers. Applications enclosed as VMs can be paused, terminated, copied, blocked, and provide a virtualization layer between the OS layer and system's hardware layer. The state of VM is saved and data package is stored in repository as a substitute to repair from catastrophic failure or error. Virtualization is one of the major feature of cloud computing. Virtualization supports scalability, easy manageability features of the cloud computing. Here, the hosting of VMs is dynamically exchanged without preventing their running of services. Virtualization extracts the descriptions of system's hardware and supplies virtualized resources for effective and powerful applications. The virtualization layer like a software is called a VM monitor(VMM) or hypervisor. It enables the physical resources of the server as virtual resources i.e. creates multiple VMs on the server. Each VM runs for processing of application. Different VMs may have different computing environment on the same server. VM migration that is a part of the virtualization enables the load balancing among servers of the cloud data center. VM live migration that enables shifting of VMs from one server to another server without preventing the services executing in VM is a key feature of virtualization and provides benefits like fault tolerance, maximum resource utilization, the overall balancing of current workload, and online system maintenance etc. Online VM placement is continuously optimizing due to changeable workloads supplied to applications. The main causes for huge energy consumption are huge amount of computing resources, energy inefficient hardware, and inefficient management of resources. Absolutely idle servers waste about 65% of their crest power. When computing resources consume 1 W of power, an extra 0.3-0.8 W is needed for the cooling system.

Server consolidation improves resource utilization and minimizes energy consumption. Simultaneous application services are encapsulated into different VMs. Thus, these services are isolated from one another. The variations and inconsistencies on resource requirements of the simultaneous services provide occasions to server consolidation for maximize the resource utilization and minimize the energy consumption. We can compare between dedicated and consolidated servers. The current workload is measured and its achievements are measured in terms of probable value. The workloads are consolidated in various ways. Each way results in the saving of different number of servers and different amount of power. Results of simultaneous application services that have been affected by virtualization are varying. So, there is an another challenging task to create a model that definitely realizes the potential revenue of cloud data center. Dynamically turning on/off servers, utility management, and on-demand resource management are other techniques for server consolidation in the cloud data center to save energy and progress the QoS parameters. These techniques are active and take decisions during the execution of application services. The dynamic allocation of resources and the mapping of VMs are not only techniques to explain the vast usage of virtualization. But, only these techniques are able to target on the arrangement of scope of the data center that are based on internet when many application services are to be consolidated within the cloud data center. So, in this chapter such type of problem and its solution is discussed.

We are again discussing about the relation between resources and application services. This relation is affected by virtualization. This chapter models a utility analytic model for server consolidation that is based on internet, defining the relationship between different requests of service arrival with different QoS parameters, and the potential passing across simultaneous application services in the cloud data center to solve the above issues. This model uses a queue and predicts the time for assigning the CPU or resources to the task and the completion time for task execution. Utility analytic model gives the topmost constrained of the number of severs after consolidation to grant the QoS parameters but having equal probability of loss of requests like in the dedicated servers. Simultaneously, function of power and utilization of PSs, the aerial of virtualization, and the algorithm for the dynamic allocation of resources in the cloud data center is formulated for the evaluation of server consolidation.

2.2 Research Background

This section describes the data set used for evaluation of proposed model. Another data set is used for processor model within the data center. Another thing is that proposed algorithm is inherited from heuristic algorithms NF, FF, FFD, BF, and BFD.

2.2.1 Random Data Selection

Random data set is normalized to obtain better accuracy with proposed algorithm.

1. We took ten servers with maximum load capacity and energy consumed by them as shown in Table 2.1.

Table 2.1: Each server's maximum load capacity and energy consumption in unit time

No of Server	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
Max. load capacity	10	20	30	40	50	60	70	80	90	100
Energy Consumed	50	60	70	80	90	100	110	120	130	140

2. We took another ten servers with maximum load capacity and energy consumed by them as shown in Table 2.2.

Table 2.2: Each server's maximum load capacity and energy consumption in unit time for another data set

No of Server	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
Max. load capacity	25	39	79	89	100	110	129	138	145	156
Energy Consumed	110	700	770	780	900	950	1000	1005	1010	1020

2.2.2 Heuristic Algorithm

Here, We inherited the features of Best-Fit algorithm, First-Fit algorithm from bin-packing problem. We sorted servers in increasing load capacity. At the time of migration, we traverse servers from highest load capacity to lowest and put all request to first server that service requests.

Algorithm 1: Best-Fit
Input: lVM, lPM
Output: MapVMPM

```
foreach $VM_i \in lVM$
  AssignIx⟵0
  foreach $PM_j \in lPM$
    if($VM_i$ fits in $PM_j$)
    then rCap[AssignIx]⟵Compute RemainCap($VM_i$, $PM_j$)
         AssignIx⟵AssignIx+1
    end if
  end foreach
  MinCapIx⟵Find IndexMinCap(rCap)
  MapVMPM[i]⟵Perform MappingVMPM(rCap, $VM_i$, $lPM$, MinCapIx)
end foreach
```

2.3 Proposed Model

2.3.1 Objectives

1. Minimize energy consumption of data center without violating SLA.

2. To meet a certain limit of QoS parameters.

3. Maximum utilization of resources.

4. Maximum number of requests can be serviced.

2.3.2 Assumptions

1. Each server has different maximum load capacity.

2. The server that has larger load capacity , consumes greater energy independently the current load servicing by that server.

3. We are not aware of the behavior of the incoming loads.

4. Initially server cluster is in the idle state.

2.3.3 Processor Model

1. Switching off components of the chips by DPM technique.

2. Minimum number of processors should , be run , have excellent performance , capability per W, more effective workload management , apply the virtualization , and the potential of running in higher temperature environments.

3. Processors have energy efficient task , job , thread scheduling mechanism.

4. Execute task at critical speed.

5. An optimal loop scheduling and voltage assignment algorithm minimizes both leakage and dynamic energy using DVS and ABB.
6. Multi-core and Multi-processor system.
7. Slowing down the computing speed at the low workload.
8. Processors have survivability and fault tolerant features.
9. Small and simple 1st level cache and way prediction decrease power consumption.
10. Compiler optimization i.e. any improvement at compile time improves power consumption.
11. No traffic within processor.

Mathematically, We can write one of the equation as follows:
Let one chip has n parts. Each consumes energy as follows:

$$E_1, E_2, E_3, ..., E_n$$

in unit time.

Total Energy = TE = $\sum_{i=1}^{n} E_i$

Let first k parts are powered off due to currently not using.

Total Energy after powering off k parts = MTE = $\sum_{i=1+k}^{n} E_i$

So, save in energy = SE = TE - MTE (1)

Another equation is as follows:
This equation is for finding save in energy when we use multiprocessor system. Let, k be the number of processors in the multiprocessor system.
Energy consumed for executing one task by server = Energy consumed by processor to execute one task + Energy consumed by other parts of server.

TE = E + RE

Let, We have n tasks.

Energy consumed in executing n tasks by server = n * TE

NTE = nE + nRE.

Let $1 < k < n$
Energy for processing of k tasks:

KTE = kE + RE.

Energy for processing of remaining n-k tasks:

RTE = (n-k)E + RE

Total energy consumed for processing n tasks:

$$ONTE = KTE + RTE$$

So, save in energy:

$$\begin{aligned} SE &= NTE - ONTE \\ &= NTE - KTE - RTE \\ &= nE + nRE - kE - RE - nE + kE - RE \\ SE &= (n-2)RE \qquad (2) \end{aligned}$$

2.3.4 Server Model

1. Apply LC technique(only improve overall energy consumption , under heavy load does not work properly).

2. Apply the combination DVS , CVS and cluster reconfiguration technique.

3. Adopt Virtualization and VM live migration across physical servers for energy efficiency.

4. Sever consolidation with VM live migration.

5. Dynamic allocation of resources and DVS.

6. Energy-aware allocation of VMs to provide services in the cloud data center that support DVS.

7. δ-Advanced-DVS policy to schedule VMs, to reduce energy consumption, to maximize the acceptance rate of provisioning requests in the real time.

8. Multi-core environment.

9. To drop the power of part of the larger host system that are not needed by the VMs placed on it.

10. For specific allotment, Optimal resource allotment strategies for specific in the dynamic algorithm.

11. Operation overhead is considered in decision making.

12. The explanation and answer must be scalable as the size of cloud data center continuously increases.

Table 2.3: No of processors currently using and corresponding save in energy.

No of processors using	3	4	5	6	7	8	9	10	11
save in energy (kj)	1	2	3	4	5	6	7	8	9

2.3.5 Proposed Algorithm for Server Consolidation

Algorithm 2 Server Consolidation

```
Step 1: Input: newrequest
Step 2: Assign the request to first server that can service the request. Highlight that server as maximum utilized or not.
Step 3: Input: newrequest.
Step 4: Assign the request to the first server that is not maximum utilized and can service the request. If service could not be serviced by any server. Display output as request can not be serviced.
Step 5: After assignment in Step 4 if server is maximum utilized. highlight that server. goto Step 3
Step 6: Migrate all VMs that are servicing request to the maximum capacity server that can service all request.
Step 7: Highlight that server as maximum utilized or not. goto Step 3
```

2.4 Simulation and Result

In this section, simulation environment, Result obtained from implementation of two equations and proposed algorithm, and observation from these results are described.

2.4.1 Simulation Environment

We implemented the above two equations using Matlab, ×86 architecture system, Windows 7 as OS, Intel Core2 processor, 4GB RAM, Windows network elements. Proposed algorithm is implemented in C++. We took data from obtained result and draw the result in graphical view using matlab with the system having features described above.

2.4.2 Result

Figure 2.2 depicts equation number 1. From Figure 2.2, The observation is as Table 2.4. So, As the no of parts are powered off due to currently not using save in the energy increases linearly.

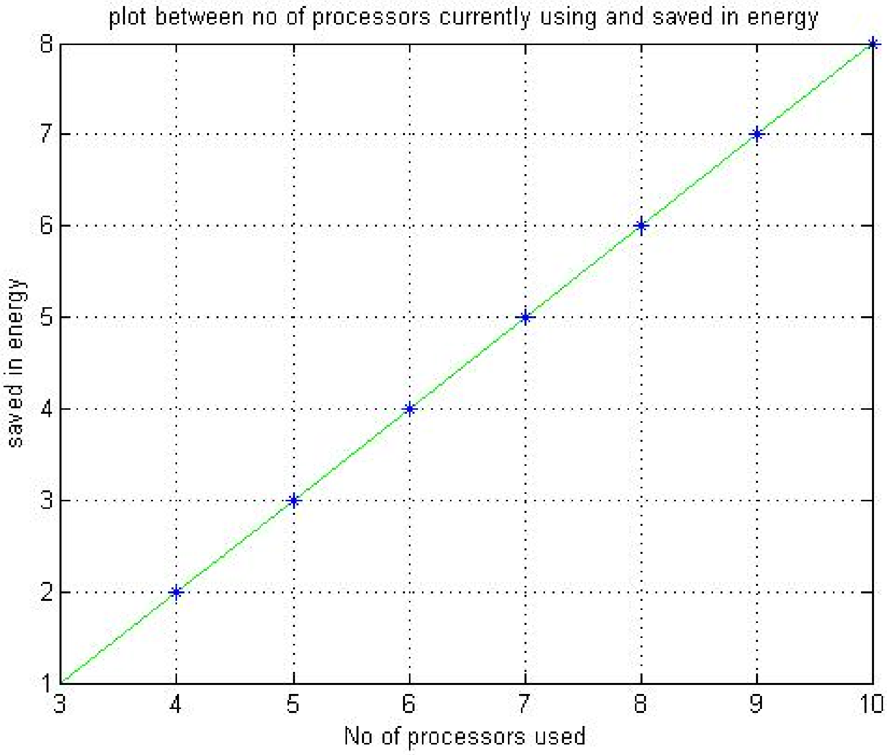


Figure 2.1: Graph between no of processors used and save in energy in different time.

Table 2.4: No of parts of a chip is powered off and corresponding save in energy.

No of parts powered off	0	2	4	6	8	10	12	14	16
save in energy (kj)	0	2	4	6	8	10	12	14	16

Figure 2.1 depicts equation number 2. From Figure 2.1, The observation is as Table 2.3. So, As the no of processors are increasing save in the energy increases linearly.

In Table 2.5 we showed the energy consumption by the data center at every new load for Table 2.1.

Table 2.5: Result for a chain of requests with configuration of data center as Table 2.1

Time	T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9
New Load	34	49	56	12	69	78	23	90	10	15
Energy Consumed	80	78	140	270	270	240	340	340	390	380

In Table 2.6 we showed the energy consumption by the data center at every new load for Table 2.2.

From Figure 2.3 and 2.4 we can see that on increasing load energy consumption by data center

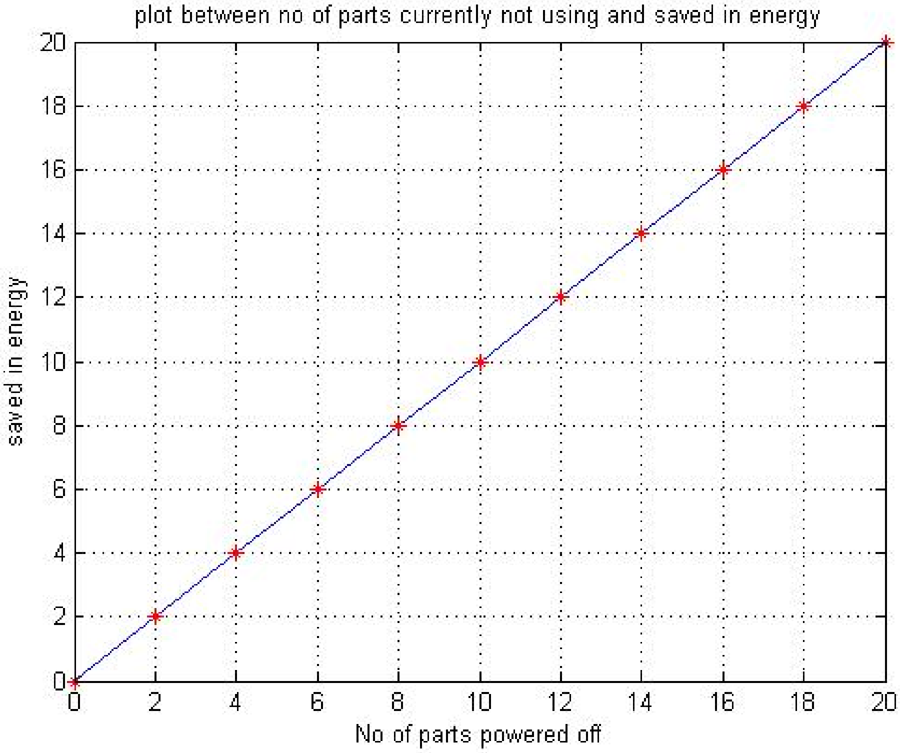


Figure 2.2: Graph between no of pats powered off and save in energy in different time.

Table 2.6: Result for a chain of requests with configuration of data center as Table 2.2

Time	T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9
New Load	20	25	111	100	80	90	40	10	120	28
Energy Consumed	110	1020	1020	1920	2930	3935	3935	3935	4935	4935

does not increase significantly and further observation is as follows:

Energy consumption, overall utilization of the server cluster at any time t depends on the current load and some set of previous load that has been processed or processing. Max. load capacity and energy consumed by each server.

$$U_i = T_u/T_t$$

T_u = time for which server is used.

T_t = time for which server could be used.

$$E_i = E_u/E_t$$

. E_u=Energy consumed by one server.

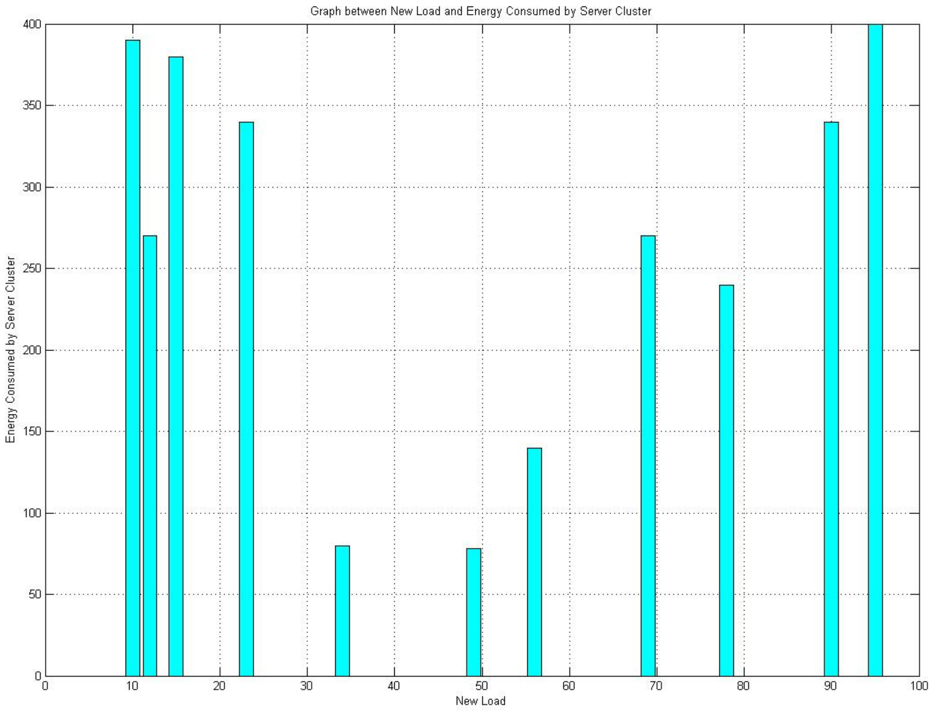


Figure 2.3: Graphical view of result of Table 2.5

E_t=Total energy consumed by all server.

$$O_i = U_i + E_i$$

2.5 Summary

We reviewed the existing energy efficient techniques for creating the cloud data center. Then we modeled, energy efficient cloud data center at the IaaS layer of cloud computing, minimized the energy consumption to a convinced and positive limit and also assisted to decrease the CO_2 release from the cloud data center with considering my objective using some existing techniques, hybrid techniques, modified techniques and proposed techniques.

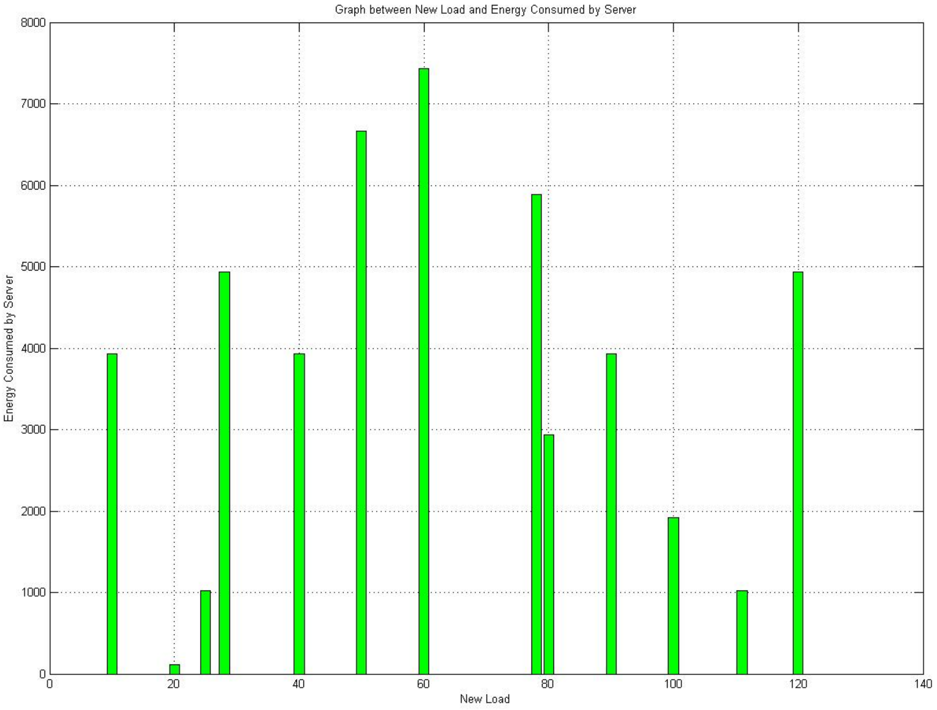


Figure 2.4: Graphical view of result of Table 2.6

CHAPTER 3

SERVER CONSOLIDATION USING VM MIGRATION

3.1 Introduction

Cloud computing has 3 important properties virtualization, pay on-demand, and scalability. Pay-on-Demand and scalability provide cloud users to change their resource demand in a very small time and pay to cloud provider accordingly. The work that we did come in IaaS type. One of the great application of virtualization is server consolidation in the cloud data center. Virtualization enables resources as a chain of resources rather than dedicated for the individual application. server can run many application enclosed as a VMs with multiple computing environments. In the result, it provides better management, on-demand resource provision, and isolation. Server consolidation is the technique by which VMs are consolidated or assigned to minimum number

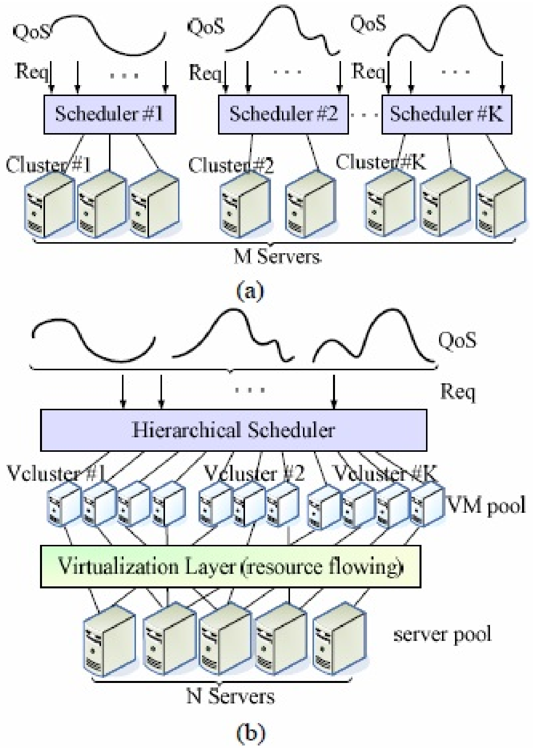


Figure 3.1: Multiple Services assigned on (a) Dedicated PSs with Multiple Schedulers (b) Consolidated PSs with One Hierarchical Scheduler

of servers within the data center. This is not a type of batch job. VMs are dynamically hosted on servers according to need and freed periodically. The main motto of Server consolidation is to minimize the power consumption by powering off idle PMs within the data center. This results in the green IT or green computing and reduces the cost in servicing of requests. In Figure 3.2 and 3.3, server consolidation using VM live migration is clearly shown. In the perspective of scalability, server consolidation increases resource utilization, reduces the problem of under utilization and over utilization of servers, reduces CPU count, this is not available in the dedicated server where a server can not be shared by multiple services. In Figure 3.1.a A scheduler assigns one service to one server, another scheduler assigns a single service to another server, and so on. This ensures isolation between services, and the wastage of resources and power among cluster of server. In Figure 3.1.b There is one virtualization layer between scheduler and cluster of servers. Services are encapsulated as VMs. These VMs are isolated to one another and assigned to servers with the help of virtualization layer. This ensures maximum resource utilization, minimum power consumption with acceptable degradation in another QoS parameters due to the variation in resource requirements with time among these VMs. This is done with some loss probability of requests. In server consolidation, minimum number of VM migration is one of the challenging issue, because a number of VM migration takes place in achieving minimum number of used nodes. VM live migration is a costly operation. It consumes some CPU cycles, memory, and network bandwidth. Some prior works did not consider a large number of VM migrations during server consolidation. In this chapter, An algorithm named NodMig is proposed. server consolidation is done with minimum number of VM migrations. We took some idea from greedy bin-packing algorithms like BF,

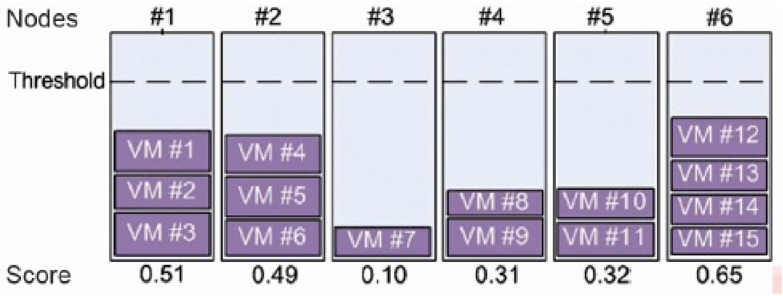


Figure 3.2: Before Server Consolidation

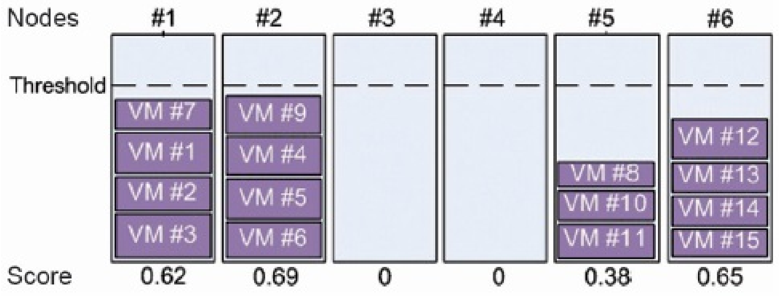


Figure 3.3: After Server Consolidation

FF, FFD, and BFD. The server that has least workload and contains at least one VM among all server is selected as a donor server. Rectangle and vector packing are two types of packing problem. Here vector packing problem is discussed with two dimensions as CPU and memory. Use of energy efficient hardware, energy minimization in network, server consolidation, energy-aware scheduling, these are another ways to reduce power consumption within the data center. There are two types of server consolidation technique offline and online. Online technique is discussed in chapter 2. Here, offline technique is discussed where we have some intermediary state of cluster of server to achieve required final state. In online technique, we are not aware the type of upcoming request or application. while in offline we are aware. Score of nodes is calculated by mathematical equation in each iteration to find donor node. On donor node, the score of each VM is calculated by mathematical expression. Two constraints are considered for VM migration between servers. Metrics are defined to calculate the efficiency of NodMig algorithm. Finally, we implemented the algorithm in C++ programming language. We took the data from obtained result and presented graphical results using MATLAB. In implementation, we took a standard data as input. We compared NodMig algorithm with FFD algorithm. Finally, Results of simulation is analyzed and its effectiveness is calculated with different parameters. Cloud data centers that are based on internet do not certainly perceive their possible earnings using virtualization for server consolidation.

Note:-Node, PS, PM, Host and Server words are used interchangeably.

3.2 Research Background

This sub-section presents the data set being used for the evaluation of NodMig. NodMig has some properties of greedy-bin-packing algorithm FFD, BFD, FF, BF. These algorithms are described here. We took standard data for the configuration of data center. We simulated with the variation of CPU threshold value and maximum allowed VM migration during server consolidation. Then, we found the optimal threshold value for CPU utilization and maximum allowed VM migration during overall server consolidation.

3.2.1 Configuration of Data Center

We took a random normalized data for current state of data center. There are 6 servers and 15 VMs as shown in Figure.3.2 or 3.3. Each server has different CPU and memory capacity.

Table 3.1: State of Cluster of Server

Servers	S_0	S_1	S_2	S_3	S_4	S_5
No of VMs	3	3	1	2	2	4
VMs	V_0, V_1, V_2	V_3, V_4, V_5	V_6	V_7, V_8	V_9, V_{10}	$V_{11}, V_{12}, V_{13}, V_{14}$

Table 3.2: VM’s Capacity

VMs	V_0	V_1	V_2	V_3	V_4	V_5	V_6	V_7
CPU Capacity	1.0	1.0	1.5	1.0	1.2	1.0	0.5	0.5
Memory Capacity	0.5	0.25	1.0	0.25	0.5	1.0	0.5	0.5

Table 3.3: VM’s Capacity

VMs	V_8	V_9	V_{10}	V_{11}	V_{12}	V_{13}	V_{14}
CPU Capacity	1.0	1.0	1.0	1.5	1.0	1.0	0.5
Memory Capacity	1.0	0.5	0.75	1.0	0.75	0.5	0.25

Algorithm 3 First-Fit-Decreasing Algorithm

Input: V, S

Output: totalMig, totalRelNe, totalUseNe.

```
PointVM[ ]⟵ VMPoint Calculation(V)
SortedVM[ ]⟵ SortVM byPointDecrg(PointVM, V)
foreach VM∈ SortedVM[ ]
    for j⟵0 to |S|-1
        Result⟵CheckForMigration(VM, j)
        if(Result)
        then Migrate VM to j
                break for loop
    end for
end foreach
```

Where

V=Set of Virtual Machine.

S=Set of Server.

$|S|$=No of Server within Data Center.

j=Index for Server.

3.3 Proposed Model

3.3.1 Problem Statement

$N_1, N_2, N_3, N_4, ..., andN_{nN}$ are nN number of physical servers within the cloud data center.

Number of VMs for N_j PM = p_j = $|N_j|, 1 \leq j \leq nN$.

Let, the value of bNS_j indicates the status of N_j PM.

$$bNS_j = \begin{cases} 1 & \text{if } N_j \text{ is an active PM.} \\ 0 & \text{otherwise.} \end{cases}$$

$$Min\{\sum_{j=1}^{nN} bNS_j\} \tag{3.1}$$

$R(N_j)$ indicates resource utilization for N_j.

$$Max\{\frac{\sum_{j=1}^{nN} R(N_j) \times bNS_j}{\sum_{j=1}^{nN} bNS_j}\} \tag{3.2}$$

Let, the value of bVM_{ksd} indicates the migration status of the kth VM.

Total number of VMs= $tnVM = \sum_{j=1}^{nN} \sum_{i=1}^{p_j} 1.$

$$bVM_{ksd} = \begin{cases} 1 & \text{if s}\neq\text{d.} \\ 0 & \text{otherwise.} \end{cases}$$

$$Min\{\sum_{k=1}^{tnVM} bVM_{ksd}\} \tag{3.3}$$

3.3.2 Assumptions

1. Multidimensional PMs and VMs.
2. Vector packing problem not Rectangle packing problem.
3. Cluster of servers is not in ideal state. VMs are already placed on servers.
4. Offine algorithm i.e. We are aware of characteristics of upcoming requests.
5. Heterogeneous cloud data center.
6. VM Live Migration.
7. The maximum resource capacity of VMs does not change during server consolidation.

3.3.3 Constraints for VM migration

The ratio of sum of CPU capacity of all VMs on individual server to the maximum CPU capacity of server must be less than or equal to the threshold value of CPU utilization. Formal representation is as follows:

$$\sum_{i=1}^{p} \frac{VC_{ij}}{C_j} \leq UTC \text{ , } \forall \text{ j} \tag{3.4}$$

where

UTC=Threshold value for utilization of CPU on jth server.

p=Number of VMs on jth server.

VC_{ij} =CPU capacity of ith VM on jth PS.

C_j =CPU capacity of jth PS.

j=jth server.

The sum of capacity of each resource of VMs on server must be less than the capacity of server for that particular resource. In this model, we are considering 2D type of VM and server. Each server consists of memory and CPU and each VM is encapsulated as memory and CPU. Formal representation is as follows:

$$\sum_{i=1}^{p} VC_{ij} < NC_j \text{ , } \forall \text{ j} \tag{3.5}$$

$$\sum_{i=1}^{p} VM_{ij} < NM_j \text{ , } \forall \text{ j} \tag{3.6}$$

If we consider server and VM that has n resources as n-dimensional vector. Formal condition is as follows:

$$\sum_{i=1}^{p} V_{ij} < N_j \text{ , } \forall \text{ j} \tag{3.7}$$

Eqn.(3.7) is the combination of eqn.(3.5) and eqn.(3.6).

where

j=jth Server.

p=No of VMs on jth server.

VC_{ij} =CPU capacity of ith VM on jth server.

VM_{ij} =Memory capacity of ith VM on jth PS.

NC_j =CPU capacity of jth PS.

NM_j =Memory capacity of jth PS.

$N_j =< rN_{1j}, rN_{2j}, rN_{3j}, ..., rN_{nj} >$.

$V_{ij} =< rV_{1ij}, rV_{2ij}, rV_{3ij}, ..., rV_{nij} >$.

If we schedule a VM V_{iS} for live migration from N_S to N_D. Then server N_S and N_D should satisfy the eqn.(3.4) and eqn.(3.7) after the VM live migration. VM V_{iS} is converted to V_{iD}.

Another thing is that NodMig algorithm supports a maximum number of VM migration through overall server consolidation. So, at each iteration total number of VM migration from the first iteration to the current iteration is checked against the maximum number of allowed VM migration. If total number of VM migration exceeds the maximum number of allowed VM migration then no more server consolidation takes place. Maximum number of allowed VM migration is directly proportional to the total number of VMs within the cloud data center. We calculate the maximum number of allowed VM migration as follows:

$$mAllowed \propto \sum_{j=1}^{nN} \sum_{i=1}^{p_j} 1 \tag{3.8}$$

where

$mAllowed$ =Maximum number of allowed VM migration through server consolidation.

p_j=Number of VMs on jth server.

$\sum_{j=1}^{nN} \sum_{i=1}^{p_j} 1 = tnVM$ =Total Number of VMs.

3.3.4 Relative Workload for Server and VM

We are calculating the relative workload of node and its assigned VMs as points. Node's points help in finding the source PS and destination PS for VM migration. VM's points ensure minimum performance degradation, minimum service downtime, and maximum utilization of server. First, point of server is calculated. Server's points are ordered in decreasing order. Least pointed node that has at least one VM is selected as a source or a donor node. On donor node, VMs are listed in decreasing order of their points. VMs are migrated from the most pointed to least pointed VM of the sorted VMs list. VM Migration is performed using eqn.(3.4) and eqn.(3.7). Mathematical Equation to calculate point for server is as follows:

$$RWLC_j = \frac{\sum_{i=1}^{p} VC_{ij}}{C_j} \text{ , } \forall \text{ j} \tag{3.9}$$

$$RWLM_j = \frac{\sum_{i=1}^{p} VM_{ij}}{M_j} \text{ , } \forall \text{ j} \tag{3.10}$$

where

M_j =Memory capacity of jth server.

$RWLC_j$ =Relative CPU workload on jth server.

$RWLM_j$ =Relative memory workload on jth server.

The point of jth server is calculated as follows:

$$pN_j = \frac{RWLC_j^2 + RWLM_j^2}{\sum_{j=1}^{nN} RWLC_j^2 + RWLM_j^2} \text{ , } \forall \text{ j} \tag{3.11}$$

where

pN_j =Point for jth node.

nN=Number of nodes.

Mathematical Equation to calculate point for VM is as follows:

$$pV_{ij} = \frac{VC_{ij}^2 + VM_{ij}^2}{\sum_{i=1}^{p} VC_{ij}^2 + VM_{ij}^2} \text{ , } \forall \text{ j} \tag{3.12}$$

where

pV_{ij} =Point for ith VM on jth server.

3.3.5 Algorithm for Server consolidation

The following algorithm is for server consolidation using VM migration. The modeling of all functions that are addressed in NodMig are modeled above in mathematical equations.

Algorithm 4 NodMig Algorithm
Input: mAllowed, N, cVM, cN
Output: totalMig, totalRelNe

```
totalMig⟵0
totalRelNe⟵0
while (mAllowed $\geq$ totalMig)
  for k⟵1 to $|N|$
    pointN[k]⟵compute PointN(N, cN, cVM)
  end for
  pointDecrN[ ]⟵calculate pointDecrN(pointN)
  rNI⟵releaseNodeIndex(pointDecrN)
  listVMs[ ]⟵obtainListVMs(N, pointN, pointDecrN, rNI)
  pointVMs[ ]⟵compute PointVM(listVMs)
  pointDecrVMs[ ]⟵calculate pointDecrVMs(pointVMs, listVMs)
  nMig⟵0
  foreach VM $\in$ listVMs[ ]
    for k ⟵ 1 to rNI-1
      migPos ⟵ Find MigPos(VM, cN, cVM, N, pointDecrN)
      if(migPos)
      then nMig⟵nMig+1
            scheduleVMMigration[ ]⟵create VMMigSchedule(VM, N)
            break for loop
    end for
  end foreach
  if(nMig = |listVMs[ ]|)
  then totalMig⟵totalMig+nMig
          totalRelNe⟵totalRelNe+1
          N⟵updatebyVMMig(N, scheduleVMMigration)
  else break(while loop)
end while
```

3.3.6 Number of Used Nodes, Migration Efficiency, performance degradation of VM, Utilization and Efficiency

From NodMig algorithm, we find the number of released PSs and the number of total VM migration after server consolidation. Number of used nodes is equal to the total number of physical servers minus number of released nodes. These are performance metrics for comparing results between FFD and NodMig.

$$NUN_i = TNPS_i - NRN_i \tag{3.13}$$

Where

$TNPS_i$=Total number of physical servers for ith simulation.

NRN_i=Number of released nodes for ith simulation.

NUN_i=Number of used nodes for ith simulation.

Migration Efficiency is the efficiency of algorithm in terms of number of VM migration. Less number and more number of VM migration ensure more and less efficiency of the algorithm with respect to the VM migration. Migration efficiency is equal to the ratio of difference between the total number of VMs and number of VM migration to the total number of VMs.

$$effMig_i = \frac{tnVM_i - tnVM_{migi}}{tnVM_i} \times 100 \tag{3.14}$$

where

$effMig_i$ =Migration Efficiency at ith simulation.

$tnVM_{migi}$ =Total number of VM migration after server consolidation at ith simulation.

Efficiency of algorithm is calculated in terms of overall number of released PMs and overall number of VM migration after server consolidation. Overall efficiency is equal to the ratio of number of released nodes to the number of VM migration.

$$Eff_i = \frac{NRN_i}{tnVM_{migi}} \times 100 \tag{3.15}$$

where

Eff_i =Efficiency of algorithm at ith simulation.

We are calculating degradation in the performance of VM for finding the optimal value of UTC and the maximum number of allowed VM migration. Degradation in the performance of VM is the ratio multiple of UTC and migration time for VM to the total execution time for NodMig.

$$pDegVM = \frac{T_{VMig}}{T_{tet}} \times UTC \times 100 \tag{3.16}$$

where

$pDegVM$ =Performance degradation in VM.

T_{VMig} =Time taken for VM migration.

T_{tet} =Execution time for NodMig.

Utilization of individual server and overall active server are calculated as follows:

$$R(N_j) = \frac{\sum_{i=1}^{p_j} V_{ij}}{C_j} \tag{3.17}$$

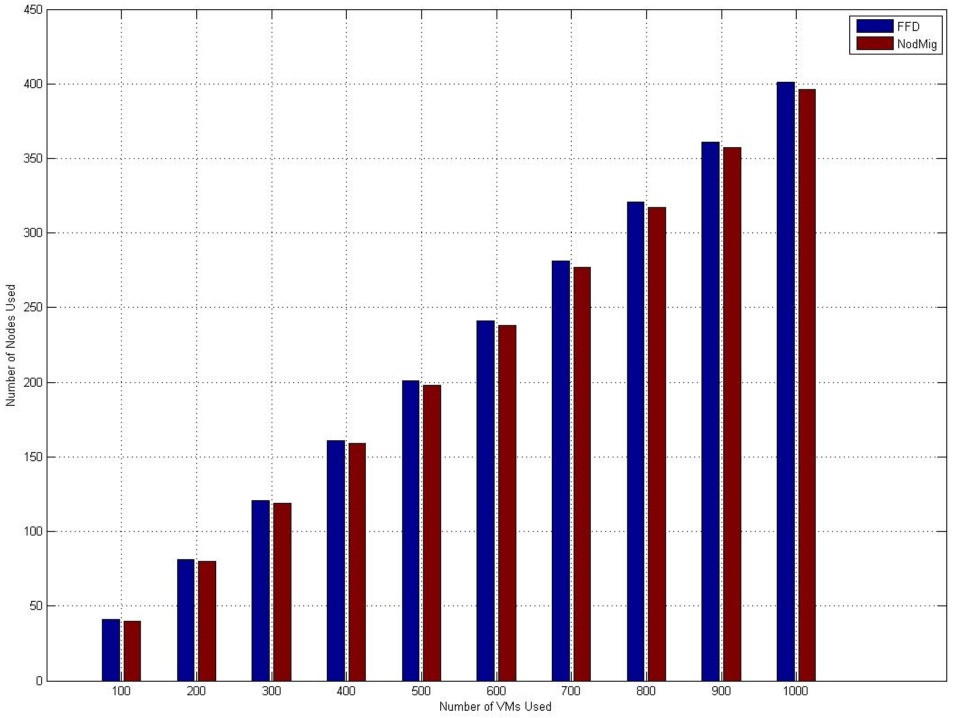
The overall utilization of the cloud data center is as follows:

$$oUTN = \frac{\sum_{j=1}^{nN} R(N_j) \times bNS_j}{\sum_{j=1}^{nN} bNS_j} \tag{3.18}$$

3.4 Simulation and Results

In this section, Simulation Environment, Result obtained from implementation of proposed algorithm, Results in graphical view, and observations from these results are described.

Figure 3.4: Graph between No of VMs and No of Used Nodes



3.4.1 Simulation Environment

We implemented NodMig and FFD in C++, with ×86 architecture system, Windows 7 as OS, Intel Core 2 Duo at 3 GHz processor, 4GB RAM, and Windows network elements. From Table 3.5, we observe that

$$mAllowed = (0.75) \times tnVM_i \tag{3.19}$$
$$UTC = 0.85$$

Result is shown in Table 3.4. We took the data from Table 3.4 and draw the result in graphical view as shown in Fig. 3.4, Fig. 3.5, Fig. 3.6, and Fig. 3.7 using MATLAB with the system having features described above.

Table 3.4: Comparison of Results between FFD and NodMig

	FFD				NodMig			
No of VMs	No of Used Nodes	No of Released Nodes	No of Migrations	Efficiency	No of Used Nodes	No of Released Nodes	No of Migrations	Efficiency
25	11	4	22	18.18	10	5	17	29.41
50	21	10	47	21.28	20	11	36	30.55
75	31	14	72	19.44	30	15	47	31.90
100	41	20	97	20.62	40	21	65	32.30
125	51	24	122	19.67	50	25	76	32.90
150	61	29	147	19.73	60	30	90	33.33
175	71	33	172	19.19	70	34	100	34.00
200	81	39	197	19.80	80	40	116	34.48
225	91	43	222	19.37	89	45	129	34.88
250	101	49	247	19.84	99	51	145	35.20
275	111	53	272	19.48	109	55	154	35.71
300	121	59	297	19.86	119	61	169	36.09
325	131	63	323	19.50	129	65	177	36.72
350	141	68	348	19.54	139	70	188	37.23
375	151	72	373	19.30	149	74	195	37.95
400	161	78	398	19.60	159	80	208	38.46
425	171	82	423	19.38	168	85	217	39.17
450	181	86	448	19.20	178	89	223	39.91
475	191	92	473	19.45	188	95	235	40.40
500	201	96	498	19.28	198	99	241	41.08
525	211	102	523	19.50	208	105	251	41.83
550	221	106	548	19.34	218	109	258	42.25
575	231	111	573	19.37	228	114	266	42.86
600	241	115	598	19.23	238	118	273	43.22
625	251	117	623	18.78	248	120	273	43.96
650	261	119	648	18.36	258	122	274	44.52
675	271	121	673	17.98	268	124	276	44.93
700	281	123	699	17.60	277	127	282	45.03
725	291	125	724	17.26	287	129	285	45.26
750	301	128	749	17.09	297	132	289	45.67
775	311	130	774	16.79	307	134	293	45.73
800	321	132	799	16.52	317	136	296	45.94
825	331	135	824	16.38	327	139	302	46.03
850	341	138	849	16.25	337	142	307	46.25
875	351	140	874	16.02	347	144	310	46.45
900	361	142	899	15.79	357	146	310	47.09
925	371	145	924	15.69	367	149	312	47.76
950	381	148	949	15.59	376	153	315	48.57
975	391	150	974	15.40	386	155	316	49.05
1000	401	152	990	15.21	396	157	318	49.37

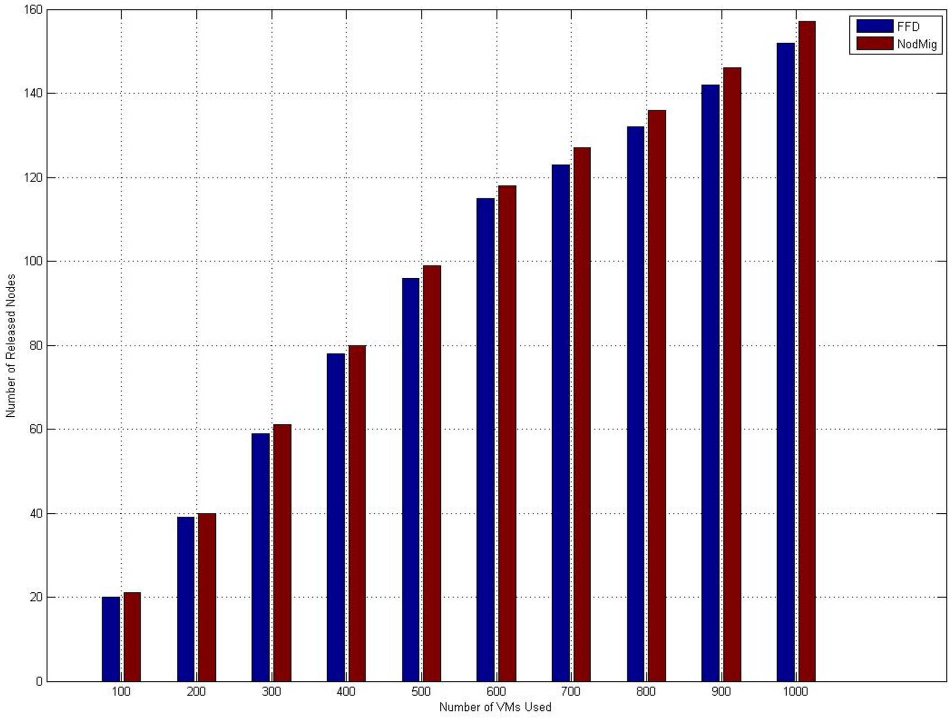
Table 3.5: Efficiency of NodMig and performance degradation of VMs at different value of UTC and mAllowed

UTC	fmAllow	NUN	NRN	TVMG	MT	TET	Eiff	pDegVM
0.6	0.25	6	0	2	6	55	0	6.5
0.6	0.50	6	0	2	6	55	0	6.5
0.6	0.67	5	1	4	12	61	25	11.8
0.6	0.75	5	1	7	21	70	14.3	18
0.6	0.90	5	1	10	30	79	10	22.8
0.65	0.25	6	0	2	6	55	0	7.1
0.65	0.50	6	0	2	6	55	0	7.1
0.65	0.67	5	1	5	15	67	20	14.55
0.65	0.75	5	1	5	15	67	20	14.55
0.65	0.90	4	2	11	33	83	18.18	25.84
0.7	0.25	6	0	3	9	57	0	11.05
0.7	0.50	5	1	4	12	61	25	13.77
0.7	0.67	4	2	6	18	68	33.33	18.53
0.7	0.75	4	2	6	18	68	33.33	18.53
0.7	0.90	4	2	10	30	79	20	26.58
0.75	0.25	6	0	3	9	57	0	11.84
0.75	0.50	5	1	4	12	61	25	14.75
0.75	0.67	4	2	7	21	70	28.57	22.5
0.75	0.75	4	2	7	21	70	28.57	22.5
0.75	0.90	4	2	11	33	83	18.18	29.82
0.8	0.25	5	1	3	9	57	33.33	12.63
0.8	0.50	4	2	7	21	70	28.57	24
0.8	0.67	4	2	7	21	70	28.57	24
0.8	0.75	4	2	7	21	70	28.57	24
0.8	0.90	4	2	11	33	83	18.18	31.8
0.85	0.25	5	1	4	12	61	25	16.72
0.85	0.50	4	2	5	15	67	40	19.03
0.85	0.67	4	2	5	15	67	40	19.03
0.85	0.75	3	3	6	18	68	50	22.5
0.85	0.90	3	3	11	33	83	27.27	33.8
0.9	0.25	5	1	4	12	61	25	17.7
0.9	0.50	4	2	5	15	67	40	20.15
0.9	0.67	4	2	5	15	67	40	20.15
0.9	0.75	3	3	6	18	68	50	23.82
0.9	0.90	3	3	12	36	88	25	36.82
0.95	0.25	5	1	4	12	61	25	18.68
0.95	0.50	4	2	5	15	67	40	21.27
0.95	0.67	3	3	5	15	67	40	21.27
0.95	0.75	3	3	6	18	68	50	25.14
0.95	0.90	3	3	12	36	88	25	38.86

3.4.2 Results

Results of NodMig and FFD algorithms are shown in Table 3.4. in terms of No of Used Nodes, No of Released Nodes, No of Migrations, and Efficiency of both algorithms with different number

Figure 3.5: Graph between No of VMs and No of Released Nodes



of VMs at the interval of 25 VMs. Initially, we have some initial state of cluster. After server consolidation, we generally find the different state of cluster from the initial state with the output as defined in the objectives. State of the cluster can be viewed as shown in Table 3.4. after the server consolidation by FFD and NodMig algorithms.

Number of migrations is equal to the total number of VMs mapping from a source PS to a different destination PS during server consolidation. Finally, the efficiency of FFD and NodMig algorithms are calculated using equation (3.15).

Fig. 3.4 shows the comparison graph between different number of nodes used by FFD and NodMig algorithms at different number of VMs. As the number of VMs increases i.e. data center becomes more bigger or no of simultaneous requests increases, the rate of increase in the number of used PSs by NodMig algorithm decreases, while FFD shows the same rate of increase in the number of used PSs. Simulation shows that at every increase in 200 VMs, the rate of increase in no of used nodes decreases. But there is a limit in the decrease of rate of increase in the no of used PSs. This will be done in the future work.

Fig. 3.5 shows the comparison graph between different number of released nodes resulted by FFD and NodMig algorithms at different number of VMs. At any number of VMs, the number of released node by FFD(NRN_{ffd}) is less than or equal to the number of released node by NodMig(NRN_{nmg}).

Figure 3.6: Graph between No of VMs and No of Migrations

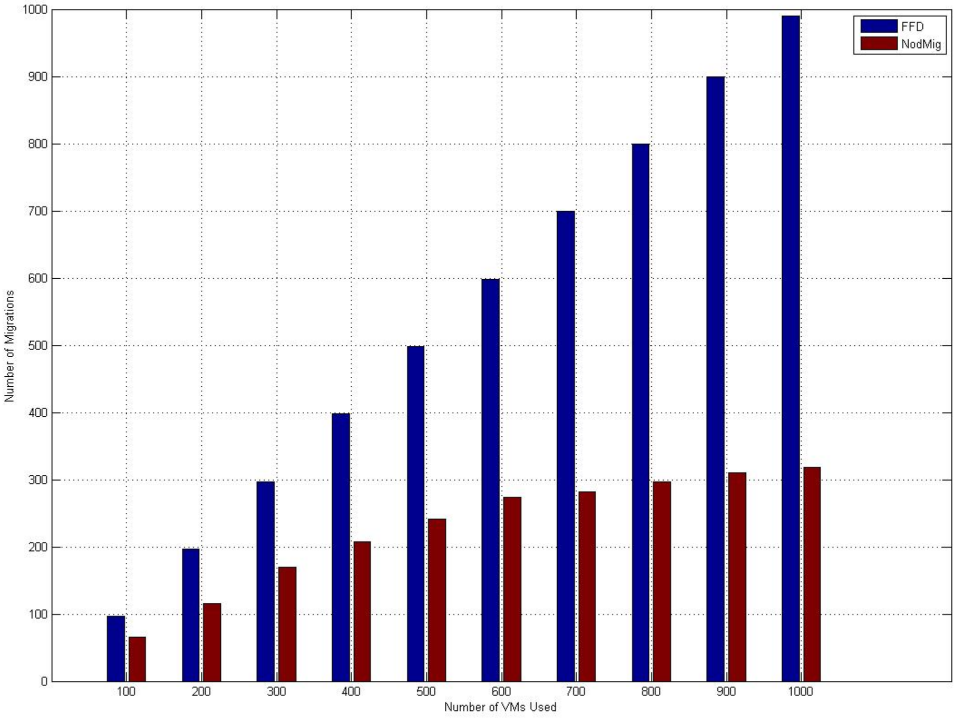
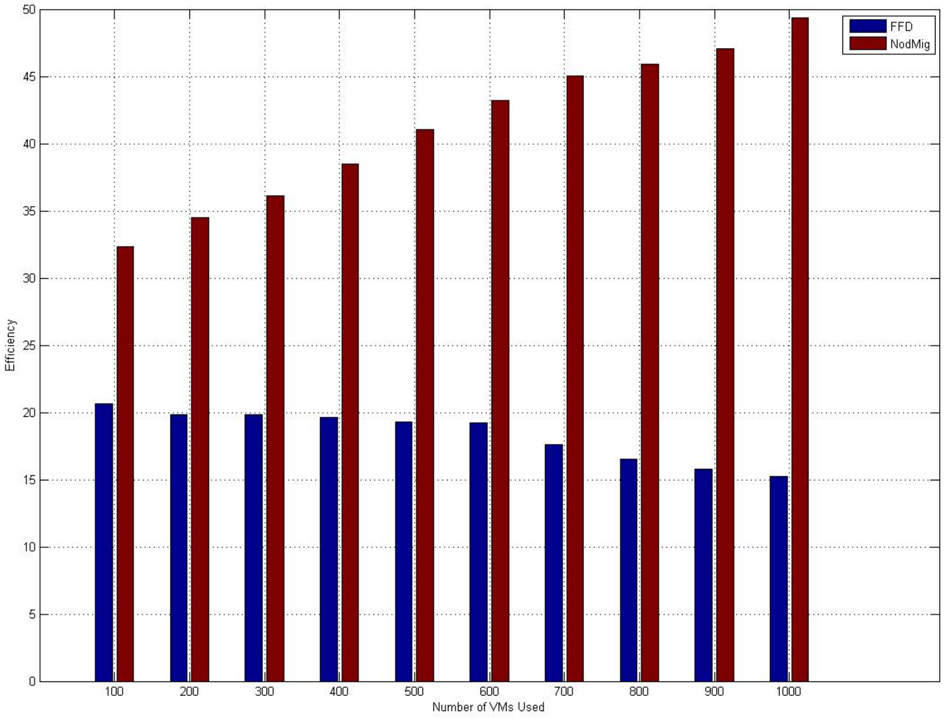


Figure 3.7: Graph between No of VMs and Efficiency



$$NRN_{ffd} \leq NRN_{nmg} \tag{3.20}$$

For NodMig, As the number of VMs increases or decreases the rate of releasing node may increase or decrease. It depends on the initial state of the cluster. Because of FFD is a heuristic approach, the same case arises for FFD. But, generally rate of releasing node decreases as the data center becomes more bigger.

Fig. 3.6 shows the comparison graph between different number of VM migrations resulted by FFD and NodMig at different number of VMs. Here, we can observe the big difference between the efficiency of FFD and NodMig in terms of VM migrations. Here, simulation shows that a very less number of increase in the number of VMs as for 25,40,50,60, a small percentage in the rate of migration efficiency of NodMig increases or decreases. But, for larger amount of increase in the number of VMs, the rate of migration efficiency generally increases. In FFD, the rate of migration efficiency either decreases or remains same for either large increase or small increase in the number of VMs. This is the main advantage of NodMig over FFD.

Fig. 3.7 shows the comparison graph between overall efficiency of FFD and NodMig at different number of VMs. The overall efficiency of FFD and NodMig is calculated by Eqn. 3.5. Simulation shows that the efficiency of NodMig always increases and the efficiency of FFD always decreases as the number of VMs increases. There is a threshold value for the increase in the efficiency of NodMig. But, the rate of efficiency of NodMig may increase or decrease and the same for FFD. The NodMig always shows the better efficiency than FFD.

3.5 Summary

Virtualization enables server consolidation to reduce the number of active PSs and to maximize the utilization of pool of resources within the cloud data center. This results in the reduction of cost for power, management, and cooling. VM live migration enables dynamic placement of VMs across cluster of PMs.

Problem is formulated as a bin-packing problem where PMs replace bins and VMs replace objects. Here, NodMig algorithm is proposed for server consolidation using VM live migration. NodMig has three objectives: minimize number of active physical servers, minimize number of VM migration, and maximize resources within the cloud data center. NodMig takes properties from FFD, BFD, FF, and BF. But it differs from FFD that it has initial state of cluster. First, we simulated NodMig with 15 VMs and 6 PMs and obtained optimal value of UTC and mAllowed. Second, we simulated NodMig and FFD at the increasing rate of 25 VMs. We found NodMig is 2 to 3 times better than FFD, because NodMig has 2 more objectives than FFD.

CHAPTER **4**

CONCLUSION AND FUTURE WORK

Energy efficient cloud data center is built to optimize each and every component of the cloud data center in terms of its energy consumption. components like CPU, memory, hard disk, server, network element, and cooling system of data center are considered for optimization. In this thesis only CPU, memory, and server models and their optimization functions are created. Model and optimization functions of Cooling system, network elements, and hard disk will be created in the future work. By optimizing energy consumption of each component, cost that is incurred by the data center is significantly reduced. This results in the reduction of carbon emission by the data center and provides Green IT environment. This enables long life of the data center, at a time more requests are serviced, and the level of QoS parameters increases.

Initially, VM placement and online VM migration technique that is based on heuristic approach FF, BF, FFD, and BFD is proposed. Problem statement is formulated as a knapsack problem. Optimization functions for processor, memory and server are separately defined. We are not aware of the nature of upcoming request. For processing of request, VMs are run and at the same time VMs are migrated according to the VM migration policy in heterogeneous environment. At processing of each new request and some defined time interval, the overall energy consumption and resource utilization of the data center are calculated. A number of VM placement techniques exist. But, in this study VM placement and its migration are proposed to minimize energy consumption, maximize resource utilization, and minimize number of VM migrations, and to reach a certain limit of QoS parameters. We observed that proposed scheme shows the better result than the brute force approach in terms of energy consumption, resource utilization, number of requests processed by the data center. This result is dependent on the nature of requests within some time interval and the configuration of data center.

Further, server consolidation is performed using VM migration. We are aware of the nature of upcoming requests. We have some initial state of the cluster. VMs are migrated according to the CPU utilization of server upto the threshold value, and maximum capacity of each resource of server. VM migration stops when no further release of node is possible. VM migration is an expensive operation. VM migration is only scheduled instead of actual VM migration at each iteration, because VM migration is an expensive operation. Actual VM migration takes place if VM migration schedule releases at least one node. Another thing is that for a cluster of server some predefined no of VM migrations should be take place for server consolidation. These 2 things minimize the number of VM migrations with minimization of number of used nodes. NodMig is compared with FFD. Fig. 3.4, 3.5, 3.6, and 3.7 show that NodMig always show better result than FFD in terms of no of used nodes, no of released nodes, no of migrations, migration efficiency and overall efficiency.

So, Energy consumption by the cloud data center is dependent on the type of the request, at what time, in which situations or conditions, why, request is send, and configuration and the quantity, the quality of resources of data center, what virtualization technique is being used by the cloud data center, how big our cloud data center apart from algorithm proposed for server consolidation, and model proposed for server, processor, memory, hard disk, network elements, and cooling system.

First of all, We will further optimize the proposed model in comparison to the existing model, and my proposed model. We will model the same or different problem based on GA, or SIP, or PSO, or ILP. We will add another objective real-time scheduling, scalability, time-interval between requests, migration time for VMs, network traffic within the data center, servicing time of request, quickly changing the workload. Data will be taken as poisson distribution or t-distribution. We would make private or hybrid cloud by considering these constraint with VMs placement and their migration.

Bibliography

[1] Norman Bobroff, Andrzej Kochut, and Kirk Beaty. Dynamic placement of virtual machines for managing sla violations. In *Integrated Network Management, 2007. IM'07. 10th IFIP/IEEE International Symposium on*, pages 119–128. IEEE, 2007.

[2] Alberto Caprara and Paolo Toth. Lower bounds and algorithms for the 2-dimensional vector packing problem. *Discrete Applied Mathematics*, 111(3):231–262, 2001.

[3] Manogna Chebiyyam, Rashi Malviya, Sumit Kumar Bose, and Srikanth Sundarrajan. Server consolidation: Leveraging the benefits of virtualization. *Infosys Research, SETLabs Briefings*, 7(1):65–75, 2009.

[4] Jian-Jia Chen, Heng-Ruey Hsu, Kai-Hsiang Chuang, Chia-Lin Yang, Ai-Chun Pang, and Tei-Wei Kuo. Multiprocessor energy-efficient scheduling with task migration considerations. In *Real-Time Systems, 2004. ECRTS 2004. Proceedings. 16th Euromicro Conference on*, pages 101–108. IEEE, 2004.

[5] Jian-Jia Chen, Heng-Ruey Hsu, and Tei-Wei Kuo. Leakage-aware energy-efficient scheduling of real-time tasks in multiprocessor systems. In *Real-Time and Embedded Technology and Applications Symposium, 2006. Proceedings of the 12th IEEE*, pages 408–417. IEEE, 2006.

[6] Hui Cheng and Steve Goddard. Online energy-aware i/o device scheduling for hard real-time systems. In *Proceedings of the conference on Design, automation and test in Europe: Proceedings*, pages 1055–1060. European Design and Automation Association, 2006.

[7] Christopher Clark, Keir Fraser, Steven Hand, Jacob Gorm Hansen, Eric Jul, Christian Limpach, Ian Pratt, and Andrew Warfield. Live migration of virtual machines. In *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2*, pages 273–286. USENIX Association, 2005.

[8] EN Mootaz Elnozahy, Michael Kistler, and Ramakrishnan Rajamony. Energy-efficient server clusters. In *Power-Aware Computer Systems*, pages 179–197. Springer, 2003.

[9] Tiago C Ferreto, Marco AS Netto, Rodrigo N Calheiros, and César AF De Rose. Server consolidation with migration control for virtualized data centers. *Future Generation Computer Systems*, 27(8):1027–1034, 2011.

[10] Daniel Gmach, Jerry Rolia, Ludmila Cherkasova, and Alfons Kemper. Resource pool management: Reactive versus proactive or let's be friends. *Computer Networks*, 53(17):2905–2922, 2009.

[11] Taliver Heath, Bruno Diniz, Enrique V Carrera, Wagner Meira Jr, and Ricardo Bianchini. Energy conservation in heterogeneous server clusters. In *Proceedings of the tenth ACM SIGPLAN symposium on Principles and practice of parallel programming*, pages 186–195. ACM, 2005.

[12] Chris Hyser, Bret Mckee, Rob Gardner, and Brian J Watson. Autonomic virtual machine placement in the data center. 2008.

[13] Ravindra Jejurikar, Cristiano Pereira, and Rajesh Gupta. Leakage aware dynamic voltage scaling for real-time embedded systems. In *Proceedings of the 41st annual Design Automation Conference*, pages 275–280. ACM, 2004.

[14] Si-Yuan Jing, Shahzad Ali, Kun She, and Yi Zhong. State-of-the-art research study for green cloud computing. *The Journal of Supercomputing*, 65(1):445–468, 2013.

[15] Kyong Hoon Kim, Anton Beloglazov, and Rajkumar Buyya. Power-aware provisioning of cloud resources for real-time services. In *Proceedings of the 7th International Workshop on Middleware for Grids, Clouds and e-Science*, page 1. ACM, 2009.

[16] Dara Kusic, Jeffrey O Kephart, James E Hanson, Nagarajan Kandasamy, and Guofei Jiang. Power and performance management of virtualized computing environments via lookahead control. *Cluster computing*, 12(1):1–15, 2009.

[17] Young Choon Lee and Albert Y Zomaya. Energy efficient utilization of resources in cloud computing systems. *The Journal of Supercomputing*, 60(2):268–280, 2012.

[18] David Meisner, Brian T Gold, and Thomas F Wenisch. Powernap: eliminating server idle power. *ACM SIGARCH Computer Architecture News*, 37(1):205–216, 2009.

[19] Eslam Mohammadi, Mohammadbager Karimi, and Saeed Rasouli Heikalabad. A novel virtual machine placement in cloud computing. *Australian Journal of Basic and Applied Sciences*, 5(10):1549–1555, 2011.

[20] Aziz Murtazaev and Sangyoon Oh. Sercon: Server consolidation algorithm using live migration of virtual machines for green computing. *IETE Technical Review*, 28(3):212–231, 2011.

[21] M. Hemalatha N Ajith Singh. and. Basip: A virtual machine placement technique to reduce consumption in cloud data center. *Journal of Theoretical and Applied Information Technology*, 59(2):426–435, 2014.

[22] Eduardo Pinheiro, Ricardo Bianchini, Enrique V Carrera, and Taliver Heath. Load balancing and unbalancing for power and performance in cluster-based systems. In *Workshop on compilers and operating systems for low power*, volume 180, pages 182–195. Barcelona, Spain, 2001.

[23] Meikang Qiu, Laurence T Yang, Zili Shao, and Edwin H-M Sha. Dynamic and leakage energy minimization with soft real-time loop scheduling and voltage assignment. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 18(3):501–504, 2010.

[24] Ivan Rodero, Juan Jaramillo, Andres Quiroz, Manish Parashar, Francesc Guim, and Stephen Poole. Energy-efficient application-aware online provisioning for virtualized clouds and data centers. In *Green Computing Conference, 2010 International*, pages 31–45. IEEE, 2010.

[25] Jyothi Sekhar, Getzi Jeba, and S Durga. A survey on energy efficient server consolidation through vm live migration. *International Journal of Advances in Engineering & Technology*, 5(1), 2012.

[26] Ying Song, Yanwei Zhang, Yuzhong Sun, and Weisong Shi. Utility analysis for internet-oriented server consolidation in vm-based data centers. In *Cluster Computing and Workshops, 2009. CLUSTER'09. IEEE International Conference on*, pages 1–10. IEEE, 2009.

[27] Frits CR Spieksma. A branch-and-bound algorithm for the two-dimensional vector packing problem. *Computers & operations research*, 21(1):19–25, 1994.

[28] Jan Stoess, Christian Lang, and Frank Bellosa. Energy management for hypervisor-based virtual machines. In *USENIX Annual Technical Conference*, pages 1–14, 2007.

[29] Akshat Verma, Puneet Ahuja, and Anindya Neogi. pmapper: power and migration cost aware application placement in virtualized systems. In *Middleware 2008*, pages 243–264. Springer, 2008.

[30] Gregor Von Laszewski, Lizhe Wang, Andrew J Younge, and Xi He. Power-aware scheduling of virtual machines in dvfs-enabled clusters. In *Cluster Computing and Workshops, 2009. CLUSTER'09. IEEE International Conference on*, pages 1–10. IEEE, 2009.

[31] Frances Yao, Alan Demers, and Scott Shenker. A scheduling model for reduced cpu energy. In *Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on*, pages 374–382. IEEE, 1995.

[32] Andrew J Younge, Gregor Von Laszewski, Lizhe Wang, Sonia Lopez-Alarcon, and Warren Carithers. Efficient resource management for cloud computing environments. In *Green Computing Conference*, pages 357–364, 2010.

[33] Dakai Zhu, Rami Melhem, and Bruce R Childers. Scheduling with dynamic voltage/speed adjustment using slack reclamation in multiprocessor real-time systems. *Parallel and Distributed Systems, IEEE Transactions on*, 14(7):686–700, 2003.