Energy Efficient Virtual Machine Migration in Cloud Data Centers

Harshit Verma
(Roll No: 111CS0162)
Subrat Kumar Dhal
(Roll No: 111CS0123)

Department of Computer Science and Engineering
National Institute of Technology, Rourkela
Rourkela-769 008, Odisha, India
June, 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 Virtual Machine Migration in Cloud Data Centers*"** submitted by ***Subrat Kumar Dhal*** and ***Harshit Verma*** is a record of an original research work carried out by them under our supervision and guidance in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in 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.

Dr. Bibhudatta Sahoo
Assistant Professor
Department of CSE
National Institute of Technology
Rourkela-769008

Place: NIT,Rourkela-769008
Date: 04 - 05 - 2015

Acknowledgment

First of all, we would like to express our deep sense of respect and gratitude towards our supervisor Prof Bibhudatta Sahoo, who has been the guiding force behind this work. We want to thank him for introducing us to the field of Cloud Computing and giving us 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 us to complete this thesis. We are greatly indebted to him for his constant encouragement and invaluable advice in every aspect of our academic life. We consider it our good fortune to have got an opportunity to work with such a wonderful person.

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

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

Subrat Kumar Dhal

Harshit Verma

Abstract

Cloud computing services have been on the rise over the past few decades, which has led to an increase in the number of datacenters worldwide which increasingly consume more and more amount of energy for their operation, leading to high carbon dioxide emissions and also high operation costs. Cloud computing infrastructures are designed to support the accessibility and deployment of various service oriented applications by the users. The resources are the major source of the power consumption in data centers along with air conditioning and cooling equipment. Moreover the energy consumption in the cloud is proportional to the resource utilization and data centers are almost the worlds highest consumers of electricity. It is therefore, the need of the hour to devise efficient consolidation schemes for the cloud model to minimize energy and increase Return of Investment(ROI) for the users by decreasing the operating costs. The consolidation problem is NP-complete in nature, which requires heuristic techniques to get a sub-optimal solution. The complexity of the problem increases with increase in cloud infrastructure. We have proposed a new consolidation scheme for the virtual machines(VMs) by improving the host overload detection phase of the scheme. The resulting scheme is effective in reducing the energy and the level of Service Level Agreement(SLA) violations both, to a considerable extent.

For testing the performance of our implementation on cloud we need a simulation environment that can provide us an environment with system and behavioural modelling of the actual cloud computing components, and can generate results that can help us in the analysis so that we can deploy them on actual clouds. CloudSim is one such simulation toolkit that allows us to test and analyse our allocation and selection algorithms. In this thesis we have used CloudSim version 3.0.3 to test and analyse our policies and modifications in the current policies. The advantages of using CloudSim 3.0.3 is that it takes very less effort and time to implement cloud-based application and we can test the performance of application services in heterogeneous Cloud environments. The observations are validated by simulating the experiment using the CLoudSim framework and the data provided by PlanetLab.

Contents

Certificate **2**

Acknowledgment **3**

Abstract **4**

List of Figures **7**

List of Tables **8**

List of Acronyms **9**

1 Introduction **10**

1.1 Overview 10

1.2 Problem Background 14

1.3 Research Statement and Research Questions 15

1.4 Goal of the Research 16

1.5 Research Objectives 16

1.6 Research Methodology 16

1.7 Contributions 17

1.8 Organization of Thesis 17

2 System Model and VM Placement Strategy **19**

2.1 System Model 19

2.2 Heuristics for VM Placement 21

2.2.1 Introduction 21

2.2.2 Formalising the VM placement problem 22

2.2.3 Proposed Model 24

2.2.4 Simulation Results 28

2.3 Conclusion 31

3 Dynamic VM Consolidation **32**

3.1 Introduction 32

3.2 Problem Statement 33

3.3 The power consumption model and performance metrics 33

3.4 Calculating the cost of Virtual Machine migrations 35

3.5 Modeling the SLA violations and performance metrics 36

3.6 Dynamic VM placement optimization 38

3.6.1 Initial Placement of the Virtual Machines 38

3.6.2 Detecting Overloaded Hosts 38

3.6.3 Detecting Underloaded Hosts 40

3.6.4 Selecting the VMs for migration from the hosts 40

3.6.5 Optimal VM placement 41

3.7 Performance Evaluation 42

3.8 Conclusion 45

4 Conclusion and Future Work **46**

Bibliography **47**

List of Figures

1.1 Virtualization concept in cloud 12

2.1 System Model 20
2.2 Packing using BFD and FFD algorithms 23
2.3 Placement of VMs being done set-wise in the PMs 25
2.4 Number of PMs used vs Number of VMs placed (for 100 PMs) 29
2.5 Number of PMs used vs Number of VMs placed (for 150 PMs) 29
2.6 Number of PMs used vs Number of VMs placed (for 200 PMs) 29
2.7 Time taken vs VMs placed (for 100 PMs) 30
2.8 Time taken vs VMs placed (for 150 PMs) 30
2.9 Time taken vs VMs placed (for 200 PMs) 30

3.1 Energy Consumed 43
3.2 Comparison based on number of migrations 43
3.3 Comparison based on PDM metric 43
3.4 Comparison based on SLATAH metric 44
3.5 Comparison based on SLAV metric 44
3.6 Comparison based on ESV metric 44

List of Tables

3.1 Server Power consumption at different load levels 35
3.2 Percentage improvement in ESV metric over LrMMT 1.2 45

List of Acronyms

Acronym	Description
FFD	First Fit Decreasing
BFD	Best Fit Decreasing
VM	Virtual Machine
PM	Physical Machine
VMM	Virtual Machine Monitor
DVFS	Dynamic Voltage Frequency Scaling
QoS	Quality of Service
ROI	Return of Investment
CPU	Central Processing unit
NAS	Network Attached Storage
MIPS	Million instructions per second
MNM	Minimum Migration Time
SLA	Service Level Agreement
SLATAH	Service Level Agreement violation Per Active Host
PDM	Performance Degradation due to Migration
ESV	Energy and SLA Violation
MMT	Minimum Migration Time
MEANMAD	Mean of Absolute Deviation from Median

Chapter 1

Introduction

1.1 Overview

NIST defines cloud computing as ”a model for enabling ubiquitous, convenient and on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.” (25)
According to NIST, the services offered by the cloud computing model can be classified as:

Software as a Service (SaaS) - The capability provided to the consumer is to use the cloud service provider's applications running on the cloud. The applications can be accessed from client devices such as a web browser (e.g., web-based email) or a program interface. The consumer does not manage or control the underlying cloud infrastructure, including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.

Platform as a Service (PaaS) - The capability provided to the consumer is to deploy onto the cloud infrastructure, consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the cloud service provider. The consumer does not manage or control the underlying cloud infrastructure, including network, servers, operating systems, or storage, but can control the deployed applications and configuration settings for the application-hosting environment.

Infrastructure as a Service (IaaS) - The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, including operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but can control operating systems, storage, and deployed applications; and some networking components like host firewalls.

These services are made available to the cloud service users by creating instances of Virtual Machines (VMs) and then consolidating the resource allocation periodically. After virtualization, users' applications can run on the same hardware managed by their own operating system.

Traditionally, organizations have had to own and deploy the hardware, network resources and also run them efficiently. Cloud computing has changed this approach drastically. Now the organizations can outsource their computational requirements to the cloud service providers and use the services over the internet, to reduce infrastructure and maintenance costs, instead of dealing with the cost and expensive process of purchasing expensive IT infrastructure and then dealing with periodic upgrades of the same. They can now pay only for the cloud resources they actually use (10).

A cloud data center consists of a large number of servers and switches for transmitting data between servers or between servers and clients. The infrastructure energy consumed in a data center includes the energy used for computational tasks, the energy used for transmission of data and the energy required for cooling the data center. The cost incurred due to this infrastructure energy has been estimated to be much more than the IT costs (4). The rise in the use of cloud computing has resulted in the setting up of more and more number of data centers, which has led to a huge increase in the consumption of energy. According to the Environmental Protection Agency (18), data centers consume around 110 Billion kilowatt hours of energy per year. This humongous increase in the infrastructure energy consumption in recent times has resulted in a sharp increase in the CO2 emissions, which contributes towards global warming (17). Thus it is imperative that energy consumption in the cloud data centers be reduced, by improving the way in which the cloud resources are provisioned.

Some of the ways in which the resource provisioning in the cloud has been improved is by the use of the virtualization technology (20) and live migration techniques (9).
The most important difference between cloud computing and a traditional method like grid computing, is the use of large scale virtualized environments and devices. Virtualization is advantageous as it partitions the resources of a single server into several execution environments, isolated from one another. This enables a number of Operating systems to run on the same hardware (28), thus reducing the hardware cost and increasing the return on investment(ROI) for the service providers. Each of these partitioned units is known as a Virtual Machine. Without virtualization, the processing power of a physical server may not be fully utilized if it runs only a single OS. An important concept in Virtualization is that of the Hypervisor or the Virtual Machine Monitor(VMM). Hypervisor is a layer of abstraction between the hardware and the operating system and the applications running on top of it. The virtualization layer is an interface between the users and the infrastructure resources.

The Virtualization layer contains the resource managers and other components that are responsible for energy efficient consolidation and resource allocation. The Virtual Machines behave like Physical Machines(PM), and they can run simultaneously, yet remain isolated from each other, all the while sharing the same physical resources. The hypervisor is responsible for the abstraction of these Virtual Machines from the physical resources and for determining what share of resources each Virtual Machine gets to use. 1.1 shows the basic architecture of a physical machine along with the hypervisor layer and the VMs running on it.

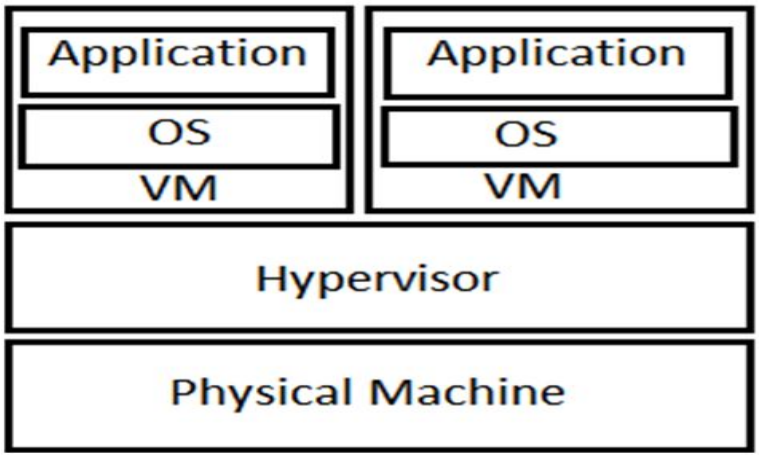

Figure 1.1: Virtualization concept in cloud

Live migration is a technique using which VMs can be transferred between PMs with nearly a zero downtime. It enables the VMs to utilize minimal number of PMs based on their present resource requirements. Huang et al. (19) in their experiment to compare the power consumed in VM consolidation schemes deploying live virtual migration, with the power consumed in regular VM placement strategies without making use of live migration, found out that live VM migration does increase some overhead involved in the consolidation process, but overall, it leads to a significant decrease in the power consumption in the data centers.

Another important concept which is made use of in reducing the energy consumption is switching the idle servers on/off or putting them to sleep (power saving state). This is because an idle server may still use up-to 70% of the peak energy consumption (14). It is therefore, highly inefficient to keep the underutilized servers running in such a state. So, based on VM selection policies, certain chosen VMs, if possible, may be migrated away from these under-utilized servers and the server can be put to sleep or switched off.

On the other hand, overloading of servers leads to performance degradation of the application workload and hence the occurrence of SLA violations. In the state of over-utilization, the servers are not able to allot the adequate amount of CPU processing power requested by the applications. As a result, the cloud service providers have to pay a previously agreed upon fine, as defined in the Service level agreement (SLA), to the cloud service users based on the level of SLA violations experienced by the applications. Thus overload of servers incurs extra cost to the cloud service providers. Live migration is made use of, to migrate some of the VMs from the overloaded servers.

It is therefore, extremely important for the cloud service providers to find a good balance between keeping the energy consumption down and reducing the SLA violations as both contribute significantly towards the cost incurred by the cloud service providers.

1.2 Problem Background

One of the earliest attempts at energy saving in cloud data centers was attempted by Horvath et al. (24). They made use of the Dynamic voltage and frequency scaling (DVFS) method. In DVFS, the CPU frequency and voltage of the hosts is reduced proportionally to the power required due to the application workloads running on them.

The DVFS method was further improved upon by Heo et al. (23) by combining it with switching the hosts on/off to further reduce energy consumption. Calheiros et al. (11) combined the DVFS approach and switching on/off of hosts with live migration of the VMs. They aggressively consolidated the VMs, migrating them after every 5 seconds, based on the CPU utilization of the hosts. Gupta et al. (16) in their work have suggested considering network infrastructure for reducing energy consumption. They suggest that Network interfaces, switches, routers and links be sent to power saving mode to reduce energy usage. Kusic et al. (24) have defined the virtual machine provisioning problem as an uncertain sequential optimization problem and they use a technique called Limited Lookahead Control (LLC) for the optimization of the same. Their objective is to reduce power usage and the level of SLA violations. In their work, they predict the future state of the system using Kalman filter and then provision the Virtual Machines, based on these predictions. Srikantaiah et al. (30) have suggested consolidation using multi-dimensional bin packing to obtain optimal energy consumption. Nathuji et al. (26) divided the data center resource management into local and global levels. The global level decides whether VM optimization needs to be done, based on the information received from all the local managers and the local manager in each of the hosts is responsible for provisioning the resources locally in the host. For optimizing the VM placements, Verma et al. (31) migrate the VMs, using live migration, periodically to minimize the power consumption and enhance resource utilization of the hosts. They have used a power-aware First Fit Decreasing heuristic for VM placements. But they do not consider SLA violations in their work. Jung et al. (21) (22) in their work of dynamic consolidation of virtual machines employed the technique of live migration and strictly adhering to the SLA. Kumar et al. (23) in their work proposed consolidating the Virtual Machines based on the stability of the current placement scenario. They estimate the current allocation's stability and then decide if further consolidation is needed or not.

Beloglazov et al. (7) proposed self-adapting heuristics which make use of a statistical analysis of the historical data of the resource usage by the virtual machines in the hosts. They have proposed the use of various measures of statistical dispersion for dynamically changing the overload threshold limit. For detecting underload of hosts, they first choose the host with the least CPU utilization and check if all VMs running on it can be migrated or not to other hosts. If it can be done, that host is switched to a power saving state and the VMs are migrated to other hosts. This process is carried on iteratively for all other hosts. For choosing which VMs to migrate they have used three algorithms namely: Minimum number of migrations (MNM), highest potential growth (HPG) and random choice (RC). For placing the VMs a modified power aware version of the Best Fit Decreasing heuristic is used. Their proposed algorithms succeed in reducing energy to a great extent while also maintaining a high level of adherence to the SLA.

1.3 Research Statement and Research Questions

The problem consists of devising an efficient VM consolidation scheme that can not only reduce the energy consumption in the data centers but also adhere to the SLA to a great extent. The problem deals with rigorous online monitoring of utilization levels of servers to check for over-utilization and under-utilization, VM selection and VM placement. The consolidation scheme can be divided into 5 phases: Initial VM placement, Host Underload detection, Host Overload detection, VM Selection and VM placement. While there exist many algorithms for each of the five phases which give good results, the performance of the algorithms can always be enhanced even further to reduce the energy consumption and also lower the level of SLA violations.

The main research questions are:

1. How to formulate the energy and QoS model for an IaaS environment with unknown workloads.

2. When to consider a Host as Overloaded and Underloaded, so that VMs can be migrated away from the host.

3. How to efficiently solve the VM placement problem.

4. How to design the scheme to reduce energy as well as SLA violations.

5. How to efficiently choose the VMs for migration.

1.4 Goal of the Research

The goal of this research is to propose an improved VM consolidation scheme which reduces the combined performance metric involving the energy consumption in the data center and the level of SLA violations.

1.5 Research Objectives

To address the goal of the research, following objectives have been identified:

1. To propose efficient algorithms for each/some of the three phases defined above.

2. To improve the performance of the VM consolidation scheme by maintaining a good balance between energy consumption and SLA violations.

3. To perform extensive evaluation of the proposed scheme, and compare and contrast it against the prevalent, most popular consolidation scheme.

1.6 Research Methodology

1. The Dynamic VM consolidation process as a whole can be improved by enhancing each/some of the five phases (7) involved in it. In these earlier stages of the research work, we have tried to improve the Overload detection scheme of the VM consolidation process, by theoretically analyzing and experimenting with the performance of various measures of statistical dispersion, both robust and non-robust, and applying it to the VM consolidation scheme.

2. We have evaluated and compared the proposed scheme with the most effective scheme, put forward by Beloglazov et al. (7), using extensive simulation through the CloudSim 2.0 toolkit (11) (open source).

The simulation and performance evaluation of the consolidation schemes has been done using CloudSim 3.0.3. A simulation tool is chosen instead of real cloud infrastructure because it is very difficult, time consuming, resource and cost intensive to use real cloud infrastructure just for evaluation of the consolidation schemes; real cloud infrastructures are too rigid to be of any use in this regard, while in the cloud simulators, the cloud parameters and the entire setup can be controlled very easily and we can develop and test

the provisioning schemes with different types of workloads and resources. CloudSim provides us the facility to actually model and test the performance of the services in large heterogeneous cloud environments with little programming and deployment effort. It provides support for modeling various features of the cloud including the broker policies, overload and underload detection schemes, allocation schemes; it supports virtualized environments; and provides option for choosing between time-shared and space-shared task allocation policies.

For the workload, we have used the PlanetLab workload traces as provided in CloudSim 3.0.3. These workload traces represent an IaaS cloud environment and correspond to the system model in question (section 2.1). The data was collected over a period of 10 days as part of the CoMon project.

1.7 Contributions

The research contributions for "Energy Efficient Virtual Machine Migration in Cloud Data Centers" is as follows:

1. We have extended the work done by Beloglazov et al. (7) and proposed the use of a non-robust measure of statistical dispersion for adaptive threshold based overload detection: the MEANMAD 2.5.

2. We have proposed a VM selection scheme Migrate Maximum MIPS (MMM).

We have used both these proposed schemes together (MEANMAD MMM 2.5). Performance evaluation has indicated that the proposed consolidation scheme has improved upon the results obtained by Beloglazov et al. in his paper, in terms of reducing both energy and the level of SLA violations as a whole.

1.8 Organization of Thesis

The rest of the Thesis is organized as follows:

Chapter 2: Introduces the system model related to our work and provides a comparative analysis of the most effective algorithms used in VM placement.

Chapter 3: Deals with the main work of the thesis which includes proposing the use

of ”Mean of absolute deviation from Median (MEANMAD)” measure of statistical dispersion and ”Migrate Maximum MIPS (MMM)”. It also presents extensive simulation, evaluation, and comparison of the proposed scheme with the best scheme in use.

Chapter 4: Discusses the conclusion derived from the entire thesis, and the future work.

Chapter 2

System Model and VM Placement Strategy

2.1 System Model

We consider the virtualized data center model with multiple cloud users (6). The cloud environment consists of N Physical hosts. P is the list of Physical Machines, where $P = \{P_1, P_2, ..., P_n\}$. The Physical hosts are homogeneous i.e. each of the hosts has identical memory, CPU performance capacity (measured in MIPS) and also has same network bandwidth. Each of the Physical hosts has a capacity of A_h. It is assumed that the servers are connected among themselves with LAN of adequate speed and also to the internet. All the hosts have access to a Network Attached Storage (NAS), which is used for enabling the virtualization technology and serves as a storage drive for the VMs.

We consider work in an IaaS environment. The data center consists of an admission control manager, a cloud manager and several local managers which are local to each physical host. The admission control manager is responsible for deciding whether a new VM request (for an application) can be allocated or not and whether the Quality of service constraints can be adhered to. If it is possible, then the Service level agreements are signed and the VM request is accepted and sent to the global manager for allocation on a host. The cloud manager is present in the controller node and with data collected from the local managers; it synchronizes, handles and manages the allocation and migration of

the Virtual Machines among the Physical hosts. Each of the physical hosts consists of a local manager which is responsible for continual monitoring of all the Virtual Machines on the host and handling resource allocation to the VM and deciding whether VMs need to be migrated from the host or not. This information is then sent to the Global Manager for further action.

The following 2.1 shows the view of the datacenter:

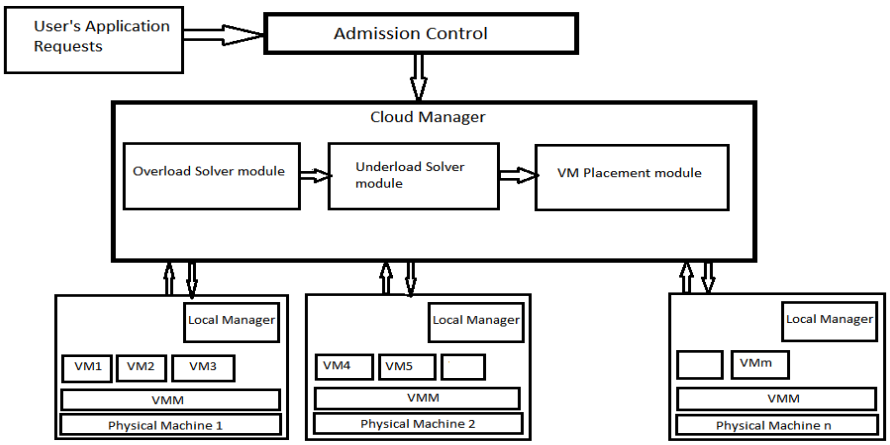

Figure 2.1: System Model

Each node can be described by the following performance parameters: the CPU performance measured in Million instructions per second (MIPS), amount of RAM available and the network bandwidth. The resource management system is not aware of the type of the applications it is managing. Various cloud service users independently submit their requests of handling M heterogeneous Virtual Machines denoted by the set $V = \{V_1, V_2, ..., V_m\}$ each of which also can be defined by the characteristics: MIPS, amount of RAM required and network bandwidth requirement. The Virtual Machines have constantly changing demands of resources i.e. the amount of CPU power it requires, but each Virtual Machine can be allotted a maximum of A_v CPU capacity. As the capacity of each physical host is A_h, a maximum of A_h/A_v number of virtual machines can be assigned to a physical host. These Virtual Machines run a wide variety of applications, which may differ greatly from one another and they are to be run simultaneously. The users and cloud service providers agree on a Service Level Agreement, which if not adhered to, will lead to a penalty on the side of the service providers.

2.2 Heuristics for VM Placement

2.2.1 Introduction

To provide fast cloud services, it all depends on how the resources are utilized in the data center especially in virtual machine placement. Virtual machine placement is the process of mapping VMs to the most suitable Physical Machine (PM) based on the requirement of VM characteristics to achieve the Quality of Service (QoS) without any violation of the SLA. VM placement is an important approach for improving power efficiency and resource utilization in cloud infrastructures. Virtual machines are of different configuration and cloud computing is a heterogeneous environment, so allocating multiple VMs to PMs has to be done wisely so that a good load balancing is achieved.

A Virtual Machine placement problem is typically a combinatorial optimization problem. Given a set of M virtual machines, each with a resource requirement specification along multiple dimensions $\{k_1, k_2, ..., k_m\}$ and a set of N physical machines, each with a capacity along m dimensions, the VM placement algorithm gives a mapping of VMs to PMs. Mapping the Virtual Machines to the available Physical Machines can be reduced to a Bin Packing Problem (9), where the Physical Machines can be considered as bins and the Virtual Machines as the objects being filled into the Physical Machines. A set of Physical Machines, not all of the same size and a set of Virtual Machines are given, sizes chosen randomly from a predefined range, the VMs must be placed into (a subset of) these Physical Machines. Naturally, the number of Physical Machines used for placing the VMs has to be minimized.

The placement of the VMs is modeled as a bin packing problem in this work. But the bin packing problem is NP-hard (13), and hence no known polynomial-time algorithm exists for this problem. While there exist other approaches to the VM placement problem like Genetic algorithms or the Stochastic Integer Programming, the Bin Packing approach is useful when dynamic VM consolidation is required where the demand is changing all the time. The Bin Packing approach being heuristic based may not give us the optimal solution always but it will produce a competitive solution in relatively lesser amount of time. It is also of use when the hosts have the same physical characteristics. The Bin Packing is a combinatorial NP-Hard problem. Energy costs of the cloud data centers can be min-

imized by efficiently packing the VMs into the least number of PMs possible. Several works have been done in this area as bin packing problem is one of the most fundamental and most studied problems in computer science history with wide range of applications in various fields. Berkey et al. (8)compared heuristics like first fit decreasing, systolic packing and harmonic packing based on their performance. They found that these heuristics performed better for smaller bins than for larger bins. Scholl et al. (29) proposed a new heuristic that incorporates tabu search and a branch-and-bound procedure which uses existing and newly proposed bound arguments and a new branching scheme. They have studied various heuristics including the First Fit Decreasing, Best Fit Decreasing, Worst Fit Decreasing and the B2F heuristic. Anily et al. (2) studied the worst case performance of the heuristics for the bin packing problem. Fleszar et al. (15) developed a new algorithm which is based on the idea of minimal bin slack. Yang et al. (33) studied a variant of the bin packing problem known as the open-end bin packing problem, in which, if there is any space remaining in the bin, new items can be further added until the total size of all items in the bin is greater than the size of the bins.

2.2.2 Formalising the VM placement problem

The one dimensional problem takes into account only a single dimension which can be any parameter among Processor usage, network bandwidth, storage capacity, memory usage and the Power usage. Broadly, there exist 2 approaches to the VM placement problem: On-line algorithms and Off-line algorithms. The on-line algorithms place the VMs into the Physical Machines as and when they appear without having any knowledge of the subsequent VMs being arrived. It is implemented on shorter durations, shorter than periods of significant variableness of the resource demand. This placement algorithm runs in the background of the application processes collecting data (9). On the other hand the off-line algorithms have knowledge of the VMs beforehand and they examine the entire list before applying their strategy to place the VMs. Two of the most popular techniques used in the bin packing problem, the First fit decreasing(FFD) and the Best fit decreasing(BFD) are compared and it is analyzed how these two differ from each other. In both the FFD and the BFD algorithms the set of Physical Machines is sorted in ascending order and the following methods are followed:

A. First Fit Decreasing

The VMs are sorted in decreasing order with respect to their size. The first VM is placed in the lowest numbered bin into which it fits i.e. if there is any partially filled i_{th} PM with capacity of $j_{th}VM$ + remaining capacity of $i_{th}PM \leq capacityof\mathrm{j}_{th}PM$ then the current VM is placed in that lowest indexed $j_{th}PM$. If it does not fit into any open PM, a new one is used and the VM is placed there. This procedure is then repeated for all of the remaining VMs, and the partially filled PMs are kept open so that they can be considered for placing the subsequent VMs into them.

B. Best Fit Decreasing

Similar to the FFD, the VMs are sorted in decreasing order. The current VM is placed into the PM which leaves the least space left over after the VM is placed in the PM. If the VM does not fit into any PM, a new one is started. Consider a case where the set of the given VMs is: $\{V_1, V_2, ..., V_{10}\}$ with capacities as $\{4, 8, 3, 1, 6, 5, 1, 4, 2, 3\}$ respectively. Here, in both the FFD and the BFD algorithms, the VMs would first be sorted in decreasing order i.e. $\{8, 6, 5, 4, 4, 3, 3, 2, 1, 1\}$. In the FFD algorithm the final packing would be as shown in the figure 2.2. Since size of PMs is fixed, BFD also achieves the exact same packing in this case.

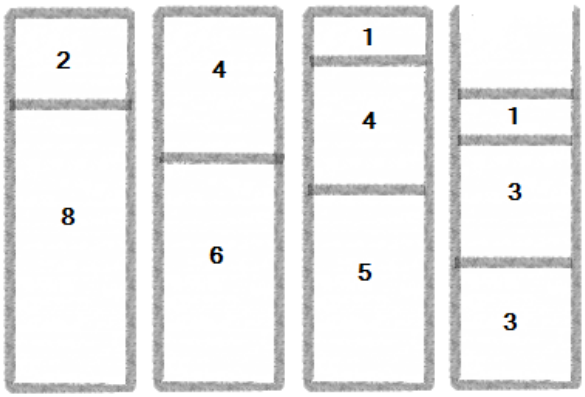

Figure 2.2: Packing using BFD and FFD algorithms

In some cases, the solution found using the BFD algorithm may be worse than that of the solution obtained using FFD. In other cases BFD finds a better or maybe an optimal solution while FFD returns a non-optimal solution.

2.2.3 Proposed Model

Given a set of n Physical Machines $\{P_1, P_2, ..., P_n\}$ of processing capacity in the range $[u_1, u_2]$ and a set of m Virtual Machines $\{V_1, V_2, ..., V_m\}$ with processing capacity in the range $[v_1, v_2]$, we need to find the number of bins B such that

$$\sum_{i \in P_k} capacity(V_i) \leq capacity(P_k); \quad \forall k \in [1, n] \tag{2.1}$$

(2.1) implies that the sum of the processing capacity of all the VMs should not exceed that of the Physical machine.

$$B = \sum_{j=1}^{n} y_j \tag{2.2}$$

Subject to

$$\sum_{i=1}^{m} capacity(V_i) t_{ij} \leq capacity(P_j) y_j; \quad \forall j \in [1, n] \tag{2.3}$$

$$\sum_{j=1}^{n} t_{ij} = 1; \forall i \in [1, m] \tag{2.4}$$

$$y_j \in \{0, 1\}; \forall j \in [1, n] \tag{2.5}$$

$$t_{ji} \in \{0, 1\}; \quad \forall j \in [1, n] \quad \forall i \in [1, m] \tag{2.6}$$

(2.2) shows the total number of PMs utilized, which is indicated by (2.5) giving 1 if a particular PM_j is used and 0 if it is not used. The presence of VM_i in PM_j is shown in (2.6) which is 1 if present else 0. A particular VM can only be placed in one PM is shown in (2.4). (2.3) indicates that the sum of the processing capacities of all the VMs

placed in a PM must not exceed that of the particular PM.

The placement of VMs into the Physical Machines is done set-wise i.e by considering a particular set of the VMs are considered at a time to be placed into a particular set of the PMs. This helps in enhancing the fault tolerance capabilities of the entire cloud infrastructure. 2.3 shows the list of PMs into which the VMs has to be allocated set-wise.

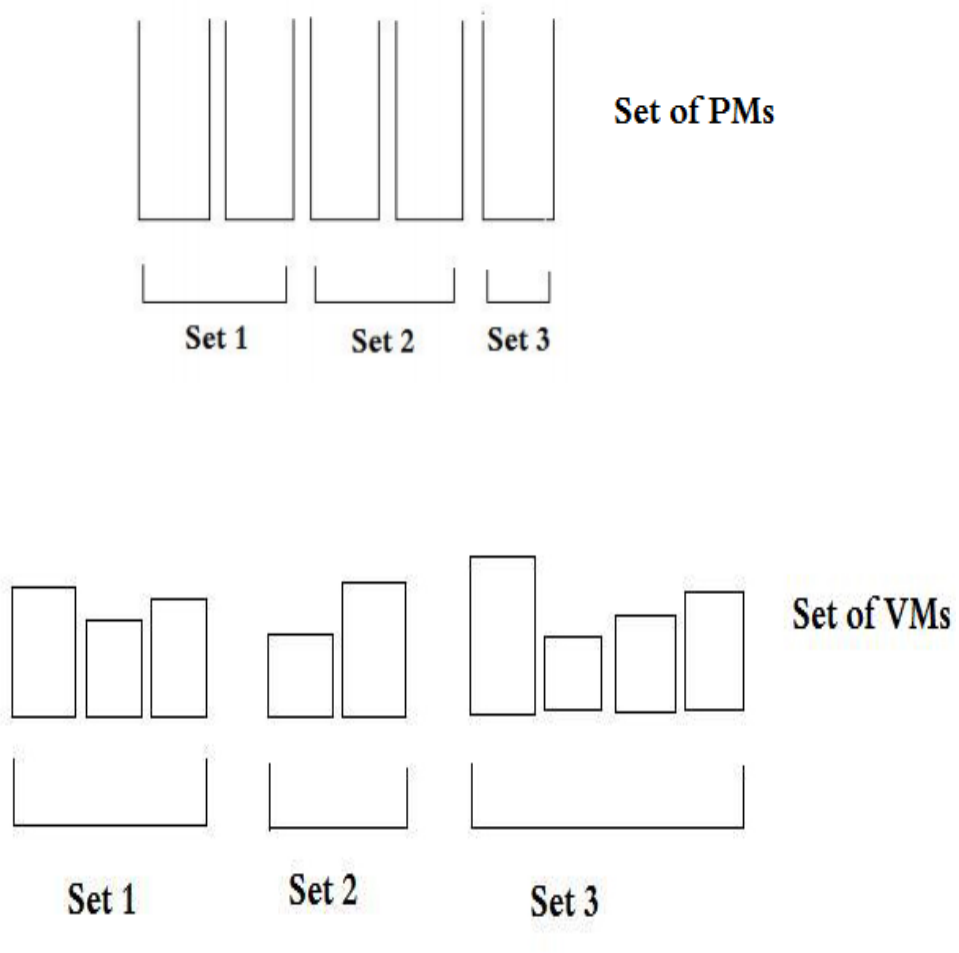

Figure 2.3: Placement of VMs being done set-wise in the PMs

The experiment is conducted by varying the number of Virtual Machines from 50 to 750 at uniform intervals of 50 units i.e 50, 100, ..., 750. The Virtual Machines have capacities in the range of $[15, 30]$ and the Physical Machines have capacities in the range of $[80, 120]$ units. The VMs are then tried to be placed in the given 100 number of Physical Machines using the modified FFD, the modified BFD and the randomized algorithm. The randomized algorithm is purely for comparative purposes, to gauge the effectiveness of FFD and BFD against random placement of the Virtual Machines.
Following present the algorithms used in the placement model:

Algorithm 1: The Global Function

```
Input: vmList, pmList
Output: Mapping of VMs to PMs
vmUnplaced ← numberofVM;
pmOpen ← numberofPM;
Call First fit algorithm;
Call Best fit algorithm;
Max heapify the VMs;
Call Randomized algorithm;
while All VMs and PMs are not considered do
    vmSet ← Generate a set of VMs;
    pmSet ← Generate a set of PMs;
    Call modified FFD algorithm;
    Call modified BFD algorithm;
    if All VMs placed or All PMs full then
        break;
return mapping;
```

Algorithm 2: Modified FFD Algorithm

```
Input: vmSet, pmSet
Output: Mapping of VMs to PMs
for each VM in vmSet do
    if VM is not placed already then
        for each PM in pmSet do
            if PM has enough resources then
                if VM can be placed into PM then
                    mapping.add(VM, PM);
                if PM is full then
                    Close PM;
                    pmOpen ← pmOpen − 1;
                if VM is placed then
                    Goto next VM;
return mapping;
```

Algorithm 3: Modified BFD Algorithm

```
Input: vmSet, pmSet
Output: Mapping of VMs to PMs
for each VM in vmSet do
    if VM_i is not placed already then
        for each PM in pmSet do
            Min Heapify the PMs;
            if PM has enough resources then
                if VM can be placed into PM then
                    mapping.add(VM, PM);
                if PM is full then
                    Close PM;
                    pmOpen ← pmOpen − 1;
                if VM is placed then
                    Goto next VM;
return mapping ;
```

Algorithm 4: Randomized Algorithm

```
Input: vmSet, pmSet
Output: Mapping of VMs to PMs
while All VMs and PMs are not considered do
    vmSet ← Generate a set of VMs;
    pmSet ← Generate a set of PMs;
    Min Heapify the PMs;
    for each VM in vmSet do
        while some PMs are open do
            pmChosen ← Generate a random PM;
            if VM can fit into pmChosen then
                mapping.add(VM, PM);
                break;
    pmOpen ← pmOpen − pmSet;
    vmUnplaced ← vmUnplaced − vmSet;
    if All VMs are placed or No PM has any resource then
        break;
return mapping;
```

Algorithm 1 is the main function which calls the modified FFD, modified BFD and Randomized algorithms. It takes the list of VMs and PMs as input and gives us the mapping of VMs onto PMs. $vmUnplaced$ denotes the virtual machines yet to be placed and $pmOpen$ denotes the physical machines that have enough resources. $vmSet$ and $pmSet$ are the set of VMs and PMs that have been generated and are to be used for mapping.

Algorithm 2 is the modified FFD algorithm which takes set of VMs and PMs as input and provides the mapping of VMs onto PMs. For each VM in the $vmSet$, it tries to place in the PM which has enough resources for VM and whichever comes first in the $pmSet$.

Algorithm 3 is the modified BFD algorithm which takes $vmSet$ and $pmSet$ as input and provides the mapping of VMs onto PMs. For each VM in the $vmSet$, it tries to place in the PM which has enough resources for the VM and leaves least amount of free resources after placement.

Algorithm 4 is the randomized algorithm that takes $vmSet$ and $pmSet$ as input and gives VM-PM mapping a output. For each VM in $vmSet$, it tries to place in a randomly chosen PM.

2.2.4 Simulation Results

The algorithms were implemented and the following results were obtained. Also it has to be said that the timing is not accurate for smaller test cases which take small running times. This is because of the $CLOCKS\_PER\_SEC$ macro which has been used for measuring the running times of the programs. Hence in some cases, same running times will appear, but actually the running times differ by a very small amount. Also, for smaller test cases the running time for the program may be 0 seconds but actually they have very small values not big enough to be measured by the macro.

From figures 2.4,2.5 and 2.6, it can be observed that the modified FFD algorithm uses lesser number of Physical Machines for placing the VMs closely followed by the modified BFD algorithm. The modified BFD algorithm takes the slightly more number of PMs than

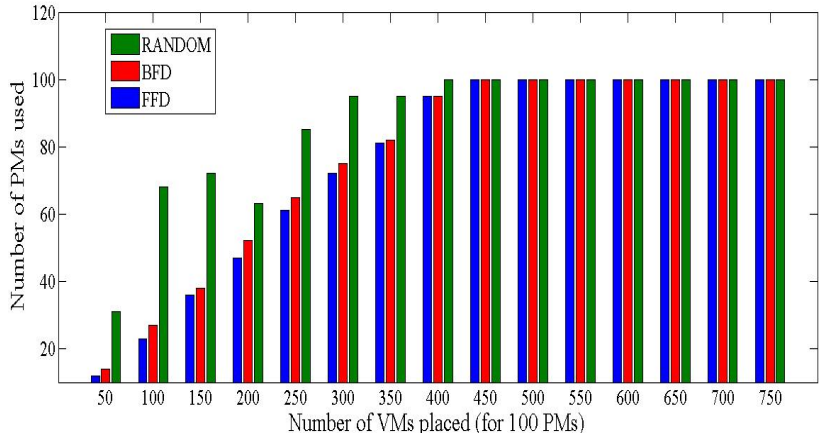

Figure 2.4: Number of PMs used vs Number of VMs placed (for 100 PMs)

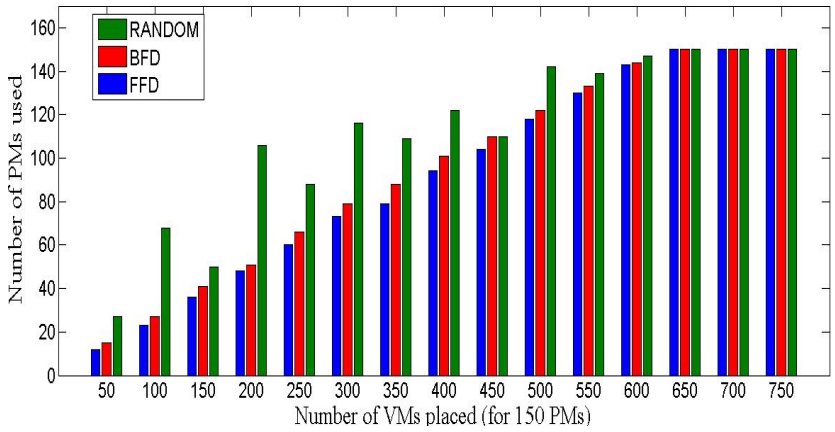

Figure 2.5: Number of PMs used vs Number of VMs placed (for 150 PMs)

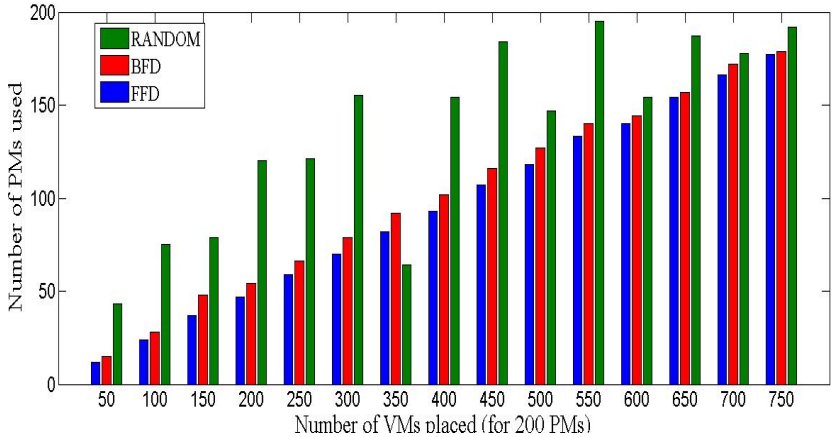

Figure 2.6: Number of PMs used vs Number of VMs placed (for 200 PMs)

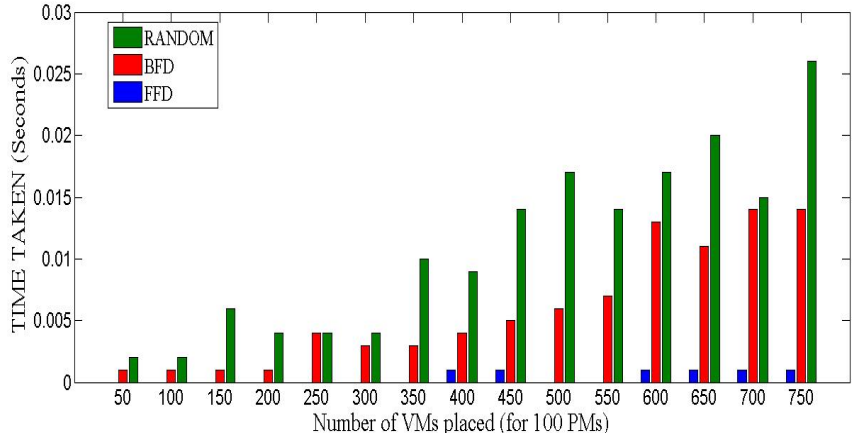

Figure 2.7: Time taken vs VMs placed (for 100 PMs)

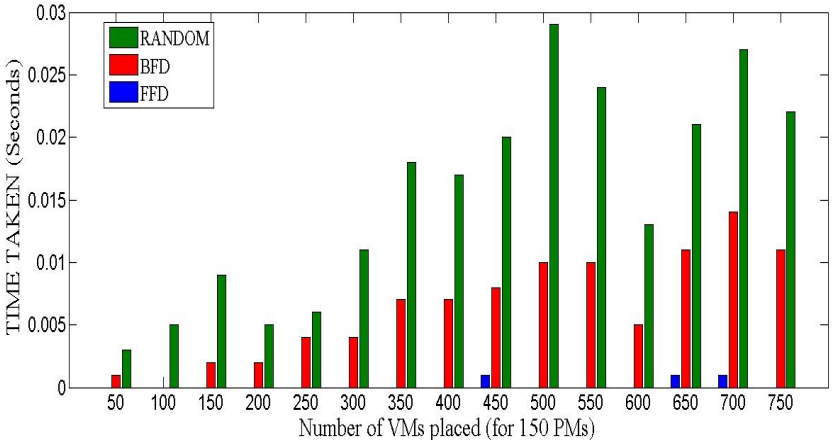

Figure 2.8: Time taken vs VMs placed (for 150 PMs)

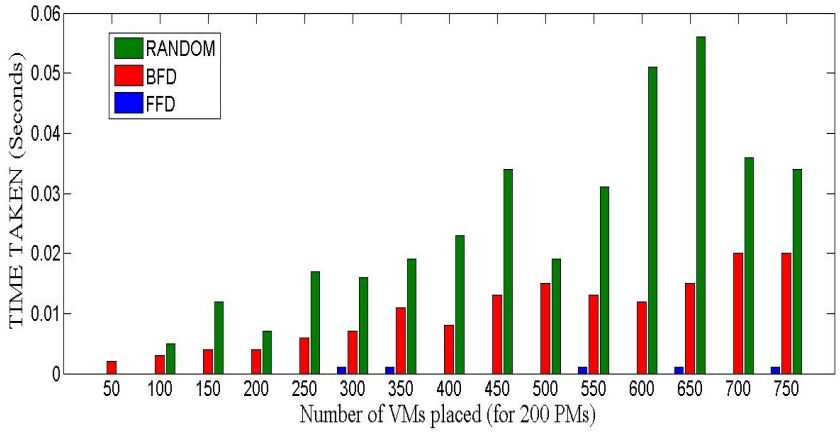

Figure 2.9: Time taken vs VMs placed (for 200 PMs)

the modified FFD algorithm in each case because in BFD the Physical Machine list is to be sorted in each iteration for placement. From figures 2.7, 2.8 and 2.9, it can be observed that the modified FFD algorithm takes the least amount of time for placing the VMs into the Physical Machines closely followed by the modified BFD algorithm. This is because in modified BFD, we are sorting the Physical Machines in ascending order according to the space remaining in them. Consequently in modified BFD, smaller bins are utilized first, so even though modified BFD uses larger number of bins, the bins used are smaller in size and it was found that the utilization or the packing efficiency in the modified BFD is generally higher than that of modified FFD.

2.3 Conclusion

From the observations in the previous section we conclude that the BFD algorithm for VM placement provides better results than the FFD algorithm. Therefore, it would be more feasible to use the BFD algorithm for placing the VMs, in our consolidation scheme.

Chapter 3

Dynamic VM Consolidation

3.1 Introduction

After initial placement of the VMs, dynamic consolidation is required. The demands of the VMs constantly change over time. It is thus imperative that the underlying resources be provisioned from time to time to serve the VMs adequately while conserving energy and keeping the level of SLA violations down.
The Dynamic VM consolidation problem can be broken down into 5 parts:

1. Initial VM placement: Initially, the VMs need to be placed on the hosts.
2. Detecting Overloaded hosts: The overload detection algorithm checks all hosts for overload. If any of the host is overloaded, the VMs need to be migrated away from the hosts.
3. Detecting underloaded hosts: The underload detection algorithm checks all hosts for underload so that the hosts can be switched to a power conserving state by migrating all the VMs away to other hosts.
4. Selecting the VMs for migration from the hosts: The VM selection algorithm returns the combined migration map for the overloaded and underloaded hosts, which indicates where to place the VMs chosen for migration.
5. VM placement: Finally, the VM placement is done according to the migration map.

3.2 Problem Statement

The problem of dynamic VM consolidation has been presented as minimization problem, to minimize the total cost incurred due to SLA violations and power consumption of data centers. The problem in this thesis assumes that the centralized cloud is hosted on a data center that has a large number of heterogeneous servers. Each of the servers may be assigned to carry out different or similar functions. A cloud computing infrastructure can be modelled with PM as a set of physical servers/machines $PM_1, PM_2, PM_3, ..., PM_n$. The resources of cloud infrastructure can be used by the virtualization technology, which allows one to create several virtual machines $VM_1, VM_2, ..., VM_m$ on a physical machine and therefore, reduces the amount of hardware in use and improves the resource utilization. So with the help of virtual machines, cloud resources are utilized.

The problem addressed in this thesis is to minimize the total energy consumption of data centers as well as to minimize the amount of SLA violations. The metric that has been used to measure it is Energy and SLA violation (ESV) metric, which is to be minimized.

3.3 The power consumption model and performance metrics

It has been shown that the power consumed by the nodes can be estimated by taking into consideration, the CPU utilization alone (14)(24). So to reduce the energy consumption the CPU utilization in the datacenters needs to be improved. For calculating the power consumption, the following model is generally assumed:

$$P(u) = k * P_{full} + (1 - k) * P_{full} * u \tag{3.1}$$

Here, u is the CPU utilization, P_{full} is the power consumption of the node when it is fully utilized which signifies the maximum power consumed and k denotes the fraction of the power that the idle server consumes. The value of k generally is around the range 0.7 (6), so the value of k is chosen as 0.7.

The modified Power consumption model then becomes:

$$P(u) = P_{full}(0.7 + 0.3 * u) \tag{3.2}$$

the Energy consumed by all the hosts for computation (18) is given by :

$$E_{comp} = \sum_{i=1}^{N} \int P(u_i(t))dt \tag{3.3}$$

The computation cost depends on the energy consumed by all the servers. Apart from the computation cost, we can also consider costs related to Virtual Machine migration and the switching costs.

The energy consumed for migration (18) is given by:

$$E_{migr} = 4 * \sum_{j=1}^{MV} P_m * \frac{C_j}{BW_j} \tag{3.4}$$

MV denotes the number of migrated Virtual Machines. P_m denotes a unit power consumption for migrating a Virtual Machine (P_m is a constant (3)). C_j denotes the memory size of migrated VM j and BW_j denotes the Bandwidth available for migrating the VM j.

The switching cost is incurred when a server is switched on from sleep state and is given by (1) (24):

$$E_{swit} = \sum_{i=1}^{K} \frac{(P_{si} * T_{si})}{2} \tag{3.5}$$

Here, K denotes the number of servers that are rebooted. P_{si} denotes the difference in power consumed by the server i when in sleep mode and in active mode. T_{si} denotes the time taken by the server i to turn on and start functioning, from sleep mode.

So the total energy consumed is given by:

$$E_{total} = E_{comp} + E_{migr} + E_{swit} \tag{3.6}$$

In their work, Beloglazov et al. (7) point out that large amount of memory being used by the physical hosts these days, have begun to dominate the power consumption by the hosts and that it is also very difficult to develop an accurate power consumption model for multi-core CPU architectures. So, instead of formulating a complex analytical model for power consumption, they have used real data provided by the SPECpower benchmark and we adopt the same. The data-set defines the power consumed by each host at various workloads based on collected data and uses these values to calculate the energy consumed by the hosts. The power consumption by the hosts at different load levels is bench-marked as follows:

Server	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
HP ProLiant G4	86	89.4	92.6	96	99.5	102	106	108	112	114	117
HP ProLiant G5	93.7	97	101	105	110	106	121	125	129	133	135

Table 3.1: Server Power consumption at different load levels

The table 3.1 represents the power consumed by the servers at different load levels, measured in Watts. From the table, we can observe that the load level and the power consumed follow a linear relationship.

3.4 Calculating the cost of Virtual Machine migrations

Live VM migration is required to migrate the VMs from the overloaded and underloaded hosts with only a small downtime. However, VM migration when carried out while the applications are running leads to performance degradation to a considerable extent. According to work done in (32), the performance degradation experienced by the applications is directly related to the number of memory pages it updates while executing. But it was also found out in these studies that the performance degradation including the downtime in the case of web applications can be estimated to be nearly 10% of the CPU utilization. Thus the performance degradation experienced by a VM_j is modeled in (6) as:

$$U_{dj} = 0.1 * \int_{t_0}^{t_0+T_{mj}} u_j(t)dt \tag{3.7}$$

$$T_{mj} = \frac{M_j}{B_j} \tag{3.8}$$

where, U_{dj} denotes the degradation in performance experienced by the j_{th} VM, t_0 is the time of start of the migration of j_{th} VM and $t_0 + T_{mj}$ is the ending time of the migration of j_{th} VM. $U_j(t)$ denotes the CPU utilization of the j_{th} VM, M_j denotes the memory used by the j_{th} VM , and B_j denotes the network bandwidth available to the j_{th} VM.

3.5 Modeling the SLA violations and performance metrics

An SLA violation will occur when less CPU capacity is being provided than what is demanded. The total SLA violation is defined as the ratio of the sum of unallocated MIPS to the sum of the requested MIPS. Hence, the Overall SLA violations is given by:

$$\frac{\sum_{i=1}^{m} \mathrm{Re}questedMIPS(i) - AllocatedMIPS(i)}{\sum_{i=1}^{m} AllocatedMIPS(i)} \tag{3.9}$$

where, $RequestedMIPS(i)$ denotes the MIPS requested by the i_{th} VM for running the application, and $AllocatedMIPS(i)$ denotes the actual MIPS that were allotted to the i_{th} VM. The SLA is assumed to be violated when for a particular VM, the Requested MIPS is less than the actual allocated MIPS.

Two kinds of SLA violation metrics are used, to estimate the level of SLA violations (7) (6) (18).

The first one is the SLA violations due to over-loaded hosts and will be denoted as $SLATAH$. It denotes the ratio of total time spent experiencing SLA violations to the total active time of hosts 3.10.

$$SLATAH = \frac{\frac{1}{N}\sum_{i=1}^{N} T_{si}}{\sum_{i=1}^{N} T_{ai}} \tag{3.10}$$

T_{si} denotes the time for which Host i has experienced SLA violations. T_{ai} denotes the total active time of Host i. N denotes the number of hosts.

The second SLA violation metric quantifies the performance degradation of the Virtual Machines due to migration. This is considered since SLA violations are also caused by VM migrations (32). This is denoted in (7) as:

$$PDM = \frac{1}{M}\sum_{j=1}^{M} \frac{C_{d_j}}{C_{r_j}} \tag{3.11}$$

C_{d_j} is an estimate of the performance degradation caused due to migration of VM_j, C_{r_j} denotes the total CPU MIPS requested by VM_j. The value of C_{d_j} is taken as the 10% of the CPU MIPS during migrations of VM_j.

The following metric denotes the total SLA violations taking into account both the above types of SLA violations, since both $SLATAH$ and PDM with equal importance denote the SLA violation level of the cloud:

$$SLAV = SLATAH * PDM \tag{3.12}$$

The metric combining both Energy consumption and SLA violations is given by (7):

$$ESV = E_{total} * SLAV \tag{3.13}$$

3.6 Dynamic VM placement optimization

The overall process of VM consolidation can be divided into the following five steps (18):

3.6.1 Initial Placement of the Virtual Machines

Initially the VMs are placed on the Physical Machines, based on the resources they demand. But their demands may change while running, so they are consolidated dynamically using the following steps.

3.6.2 Detecting Overloaded Hosts

This phase of the VM consolidation process is used to reduce the load on the overloaded hosts. The traditional approach is the STA (Static Threshold Algorithm). The STA defines an upper threshold for the hosts beforehand and the provisioning schemes have to keep the total utilization of the CPU under the threshold limit. If the threshold limit is exceeded, some VMs have to be migrated from the host to reduce the load on the CPU so that a SLA violation can be prevented. But, since STA defines static threshold limits, they are not suitable for dynamically changing workloads and conditions, so Overload detection schemes which can handle the dynamically changing environments are needed. Beloglazov et al. (7) in their study proved that local regression method, which was proposed by Cleveland (12) achieves most suitable results with dynamic, variable workload. It was shown by Beloglazov et al. that local regression can obtain the best results when compared to other popular methods of detection like, Median Absolute Detection, and Interquartile range. Beloglazov et. al in their work have shown that LrMMT 1.2 is the best detection algorithm for detecting overloaded hosts.

To migrate the VMs from the physical machines, we need to set an upper threshold, such that when the CPU utilization of the particular host reaches above this value, the VMs are migrated iteratively from the host, until the host is no longer overloaded. But in an environment where the workload is dynamic and is changing by the minute, we need a more accurate estimate of the threshold value and it cannot be fixed to a particular value.

We have proposed a new algorithm MEANMAD MMM 2.5 by improving upon the

existing algorithm proposed by Beloglazov et al.. MEANMAD MMM uses MEANMAD(Mean of absolute deviation from median) for dynamically varying the upper threshold limit for overload detection based on a statistical analysis of the historical data of CPU utilization by application workloads on the hosts.The MEANMAD MMM is used to estimate the threshold value by measuring the deviation of previous values of the CPU utilization of the host. If the deviation of these values is large enough, it is more likely that the CPU utilization will reach 100% and the CPU will get overloaded. So, for a larger deviation, we need to lower the upper threshold and migrate the VMs from the host.
MEANMAD is defined as:

$$MEANMAD = mean(|CPUutilization_i - median(CPUutilization[])|) \tag{3.14}$$

where, $CPUutilization[]$ denotes the list of all CPU utilizations, $CPUutilization_i$ denotes the CPU utilization value of the i_{th} element in the list.
And the upper threshold is given by:

$$T_u = 1 - s.MEANMAD \tag{3.15}$$

where, s is a safety parameter and we can define it as per our requirement. If we want to focus more on the energy conservation, then s can be assigned a larger value so that migrations are more and energy consumed is also less. By varying s suitably, we found that for the workload of the nature used, 2.1 comes out to be a good value and is effective in reducing the overall energy and SLA violation metric to a great extent.

MEANMAD is a measure of statistical dispersion, but it is not a robust statistic unlike the MAD used by Beloglazov et al. (7). The main idea behind the MEANMAD we do not want to entirely discard the outlier data, as the outliers do also play an important role in deciding the overload threshold as even a few instances of high CPU utilization could cause the CPU to be overloaded and cause SLA violations. This is why our algorithm succeeds in reducing the SLA violations to a great extent and even the combined metric of Energy and SLA violations is reduced greatly.

3.6.3 Detecting Underloaded Hosts

We have already seen that idle hosts may consume up-to 70% of the peak energy consumption. It will therefore be feasible to migrate all the Virtual Machines from an underloaded host and switch it off or put it in a power saving state. The procedure to be followed for handling under-loaded hosts is:
(i) First, the host with least utilization is considered for migrating all VMs from it.
(ii) The selected VMs are migrated (7) to other servers while not over-loading them.
(iii) This process is then repeated for all under-loaded hosts.

3.6.4 Selecting the VMs for migration from the hosts

After the list of overloaded and underloaded hosts is determined, the next logical step is to determine which Virtual Machines to migrate from these hosts. Various schemes exist, including the Minimum Migration Time (MMT) policy (7) which migrates a VM that needs the minimum time to migrate compared to the other VMs in the same host. The algorithm runs iteratively to migrate the VMs one by one until the host is no longer overloaded. Zhang et al. proposed the MNM policy (34) which selects the minimum number of the VMs to migrate from a host so that the CPU utilization falls below the specified upper threshold.

We have proposed a new VM selection scheme ”Migrate Maximum MIPS (MMM)”. The overload detection scheme has already decided that a host is overloaded and then we need to choose some VMs to migrate from the host so that the host may no longer be overloaded. According to this scheme, for choosing which VMs to migrate we take the CPU power (MIPS) of each VM the host is using, into consideration. Among all the VMs running in the host, we choose that VM for migration which is consuming the maximum MIPS. After the first VM is chosen for migration, the host is again checked for overload, if it is still overloaded the selection scheme is applied again and this process is carried out iteratively until the host in consideration is no longer overloaded.

The idea behind choosing MIPS for selecting the VMs is that we desire to migrate those VMs first which consume the maximum amount of the host’s CPU power. When VMs consuming larger CPU power are migrated first, chances are, the number of VM migra-

tions will go down and ultimately lead to a reduction in the consumption of energy, which is the case as demonstrated in the experiments.

3.6.5 Optimal VM placement

The VM placement problem is an NP hard one. However, various heuristics have been developed to approach closer to the optimal solution.
Algorithms used for the placement of the Virtual Machines can be broadly categorized as using the:
(1) Power Based approach and
(2) Application QoS based approach.
The first approach tries the placement in such a way so as to utilize servers to their maximum efficiency by server consolidation and hence minimize energy consumption. While the second approach aims at enhancing the Quality of Service (QoS) as defined in the Service Level Agreements (SLA) by maximizing the resources given to the applications. This work tries to take both the approaches into account and try to arrive at a tradeoff between the two which will be most beneficial to the Cloud service providers.

The Virtual Machine placement problem is modelled as a bin packing problem. The bin packing problem is NP-Hard, so no polynomial time algorithm exists to solve this problem. So various heuristics have been used in placement models which give good results in a short period of time. The most popular algorithm used for this purpose is the constraint based Best Fit Decreasing modified suitably to take the energy model into consideration. Best Fit decreasing is the most suitable choice in scenarios where the workload is dynamically changing. Though it may not always give optimal results, but it makes up for it in speed of achieving the placements. While these may not give optimal solutions, they perform the placements in quicker time, which is an acceptable trade-off in cloud computing environment, where time is of utmost importance. One such algorithm is the LIP algorithm (7) which was named so in (18). This algorithm first sorts the selected Virtual Machines in the decreasing order of CPU utilization and then for each VM, it allocates it to that particular host which will lead to the least increased power in the host. The LIP with host sort was proposed in (18) which also sorts the host in descending order of current CPU utilization. BHF algorithm was proposed (18), where the authors tried to allocate VM in that host which would lead to the highest utilization in the host. This

was predicted using local regression techniques. Zhang et al. (34) have used an improved version of the Best-Fit decreasing algorithm which is power aware (PBFDH). We use the Power aware version of the Best Fit decreasing algorithm as proposed by Beloglazov et al. (5), which is what is implemented in the CloudSim 3.0.3 toolkit which is used to conduct our experiments.

3.7 Performance Evaluation

CloudSim toolkit (11) 3.0.3 has been used to simulate the existing and proposed algorithms. The cloud consists of a data center consisting of 800 heterogeneous physical nodes. Of these, half are HP ProLiant ML110 G4 servers, having 1860 MIPS capacity on each core and other half are HP ProLiant ML110 G5 servers, having 2660 MIPS capacity on each core. Each server has a network bandwidth capacity of 1 GBPS. Both type of hosts have 4GB of RAM.

The types of VMs used are $1)$ 2500 MIPS, 0.85 GB(RAM) , $2)$ 2000 MIPS,3.75 GB, $3)$ 1000 MIPS, 1.7 GB, $4)$ 500 MIPS, 613 MB . The objective is to minimize both energy and SLA violation costs, so the metrics: Energy and SLAV, both have to be taken into account. Thus, the metric $ESV = Energy$ x $SLAV$ is used to compare the consolidation schemes which gives equal weight-age to both energy and the level of SLA violations.

The following plots show the comparison between LrMMT 1.2, the best VM consolidation scheme used by Beloglazov et al. (7) and our proposed scheme: MEANMAD MMM 2.5. The algorithm is run for 10 sets of data as provided in the CloudSim toolkit, which denotes the CPU utilization values from PlanetLab collected on the 10 different dates. The data is provided as part of the CoMon project (27).

Fig.3.2 shows the number of VM migrations, which are a bit less in comparision to LrMMT 1.2. From fig.3.1 we observe that our proposed scheme consumes lesser energy while consolidating the VMs, and from fig. 3.3, 3.4 and 3.5 we observe that the level of SLA violations is much less in our scheme than in LrMMT 1.2. But our objective is to minimize both Energy and cost incurred due to SLA violations. So the most important

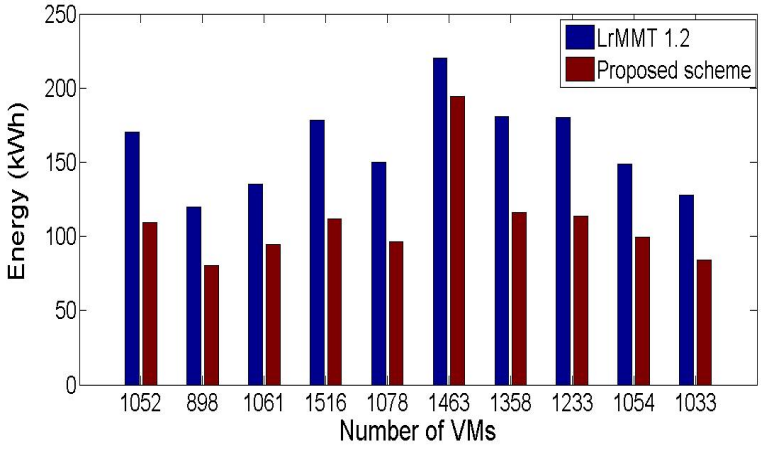

Figure 3.1: Energy Consumed

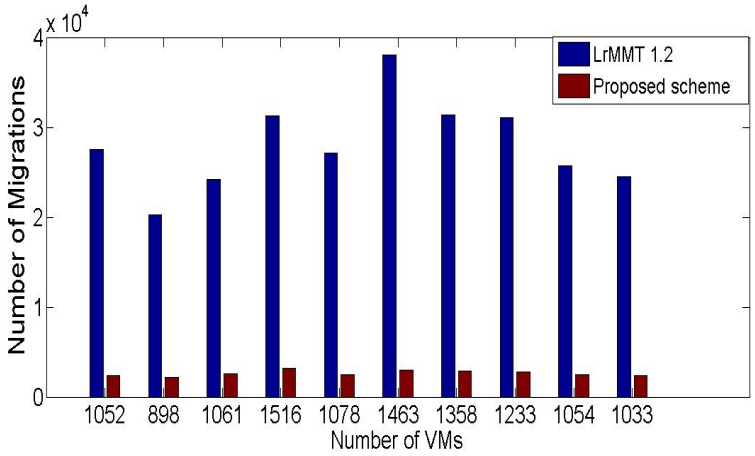

Figure 3.2: Comparison based on number of migrations

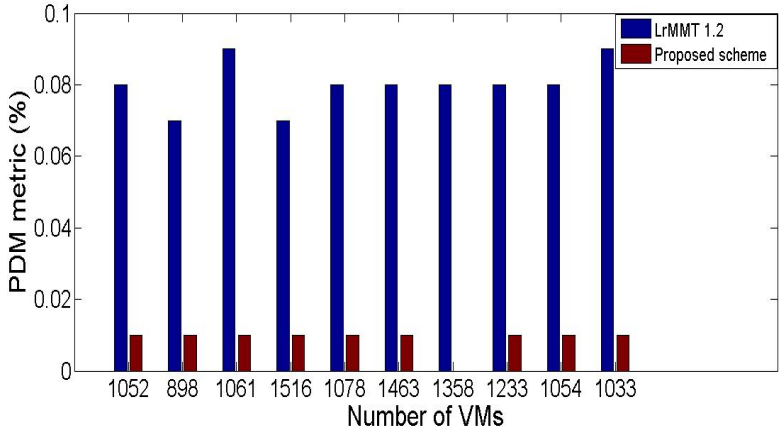

Figure 3.3: Comparison based on PDM metric

metric is the ESV metric (fig. 3.6), which clearly shows that our proposed scheme gives much better results for minimizing both Energy and SLA violations.

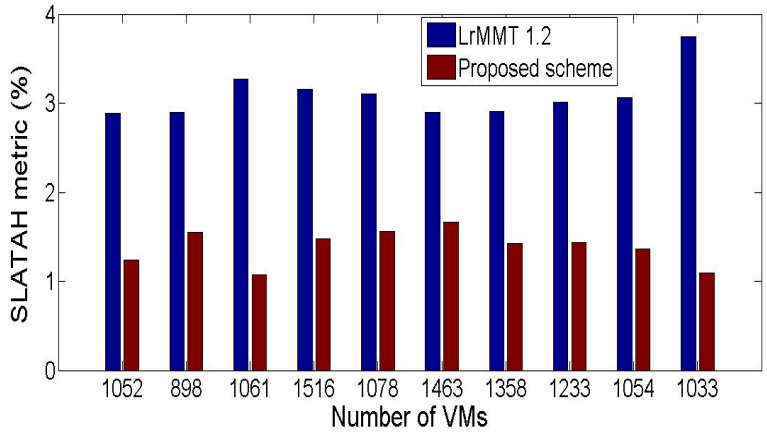

Figure 3.4: Comparison based on SLATAH metric

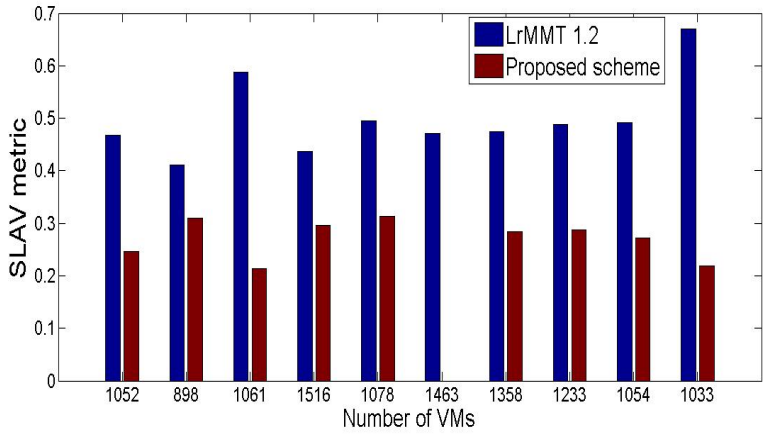

Figure 3.5: Comparison based on SLAV metric

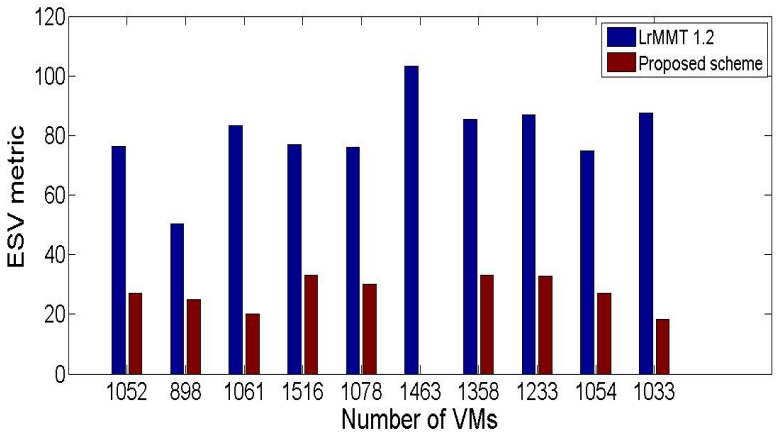

Figure 3.6: Comparison based on ESV metric

The following table 3.2 shows the percentage improvement in the ESV metric in each of the cases obtained in our scheme, over LrMMT 1.2. The entry corresponding to 1463 VMs is Not Available as the SLAV obtained in this case was 0.00 because only 2 decimal places were considered for measuring the values :

Number of VMs	1052	898	1061	1516	1078	1463	1358	1233	1054	1033
% improvement in ESV metric	63.48	49.06	74.06	53.63	58.16	NA	61.06	61.32	61.37	77.37

Table 3.2: Percentage improvement in ESV metric over LrMMT 1.2

3.8 Conclusion

1. Performance evaluation has indicated that our proposed consolidation scheme, MEAN-MAD MMM 2.5 has improved upon the results obtained by Beloglazov et al. in their paper, in terms of reducing both energy and the level of SLA violations as a whole. This will in-turn lead to a better ROI for the cloud service users.

2. As per Beloglazov et al. (7) the local regression based overload detection schemes significantly outperform the adaptive threshold based schemes. But the results obtained in our work indicate otherwise. The adaptive threshold based consolidation schemes outperform the local regression based consolidation schemes when an appropriate measure of statistical dispersion is used.

Chapter 4

Conclusion and Future Work

In this work, the problem of an effective Virtual Machine consolidation scheme to reduce energy consumption is undertaken. Reducing Energy consumption is one of the most serious concerns for the cloud service providers. Even in idle mode the data center is able to consume about 70% of the peak energy. Effective VM consolidation scheme can help the cloud data centers save more energy and hence reduce their costs of operation and increase the ROI. We have proposed a new algorithm for overload detection: the MEANMAD 2.5 and combined with the minimum migration time policy of selecting the VMs for migration, we get the method MEANMAD MMM 2.5. MEANMAD MMM 2.5 has improved upon the results obtained by Beloglazov et al. in their paper, in terms of reducing both energy and the level of SLA violations as a whole leading to a better ROI for the cloud service users. Also, as per Beloglazov et al. (7) the local regression based overload detection schemes significantly outperform the adaptive threshold based schemes. But the results obtained in our work indicate otherwise. The adaptive threshold based consolidation schemes outperform the local regression based consolidation schemes when an appropriate measure of statistical dispersion is used.

The future direction of this project includes modifying other phases of the VM provisioning scheme which can reduce energy consumption and SLA violations even further than what is already achieved.

Bibliography

[1] ADDIS, B., ARDAGNA, D., PANICUCCI, B., and ZHANG, L., "Autonomic management of cloud service centers with availability guarantees," in *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, pp. 220–227, IEEE, 2010.

[2] ANILY, S., BRAMEL, J., and SIMCHI-LEVI, D., "Worst-case analysis of heuristics for the bin packing problem with general cost structures," *Operations research*, vol. 42, no. 2, pp. 287–298, 1994.

[3] BALIGA, J., AYRE, R. W., HINTON, K., and TUCKER, R., "Green cloud computing: Balancing energy in processing, storage, and transport," *Proceedings of the IEEE*, vol. 99, no. 1, pp. 149–167, 2011.

[4] BELADY, C. L., "In the data center, power and cooling costs more than the it equipment it supports," *Electronics cooling*, vol. 13, no. 1, p. 24, 2007.

[5] BELOGLAZOV, A., ABAWAJY, J., and BUYYA, R., "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future generation computer systems*, vol. 28, no. 5, pp. 755–768, 2012.

[6] BELOGLAZOV, A. and BUYYA, R., "Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers," in *Proceedings of the 8th International Workshop on Middleware for Grids, Clouds and e-Science*, p. 4, ACM, 2010.

[7] BELOGLAZOV, A. and BUYYA, R., "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," *Concurrency and Computation: Practice and Experience*, vol. 24, no. 13, pp. 1397–1420, 2012.

[8] BERKEY, J. O. and WANG, P. Y., "A systolic-based parallel bin packing algorithm," *IEEE transactions on parallel and distributed systems*, vol. 5, no. 7, pp. 769–772, 1994.

[9] BOBROFF, N., KOCHUT, A., and BEATY, K., "Dynamic placement of virtual machines for managing sla violations," in *Integrated Network Management, 2007. IM'07. 10th IFIP/IEEE International Symposium on*, pp. 119–128, IEEE, 2007.

[10] BUYYA, R., YEO, C. S., VENUGOPAL, S., BROBERG, J., and BRANDIC, I., "Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation computer systems*, vol. 25, no. 6, pp. 599–616, 2009.

[11] CALHEIROS, R. N., RANJAN, R., BELOGLAZOV, A., DE ROSE, C. A., and BUYYA, R., "Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011.

[12] CLEVELAND, W. S., "Robust locally weighted regression and smoothing scatterplots," *Journal of the American statistical association*, vol. 74, no. 368, pp. 829–836, 1979.

[13] COFFMAN JR, E. G., GAREY, M. R., and JOHNSON, D. S., "Approximation algorithms for bin packing: a survey," in *Approximation algorithms for NP-hard problems*, pp. 46–93, PWS Publishing Co., 1996.

[14] FAN, X., WEBER, W.-D., and BARROSO, L. A., "Power provisioning for a warehouse-sized computer," in *ACM SIGARCH Computer Architecture News*, vol. 35, pp. 13–23, ACM, 2007.

[15] FLESZAR, K. and HINDI, K. S., "New heuristics for one-dimensional bin-packing," *Computers & operations research*, vol. 29, no. 7, pp. 821–839, 2002.

[16] GUPTA, M. and SINGH, S., "Greening of the internet," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 19–26, ACM, 2003.

[17] HANNE, F. Z., "Green-it: Why developing countries should care?," *IJCSI International Journal of Computer Science Issues*, vol. 8, no. 4, 2011.

[18] HUANG, J., WU, K., and MOH, M., "Dynamic virtual machine migration algorithms using enhanced energy consumption model for green cloud data centers," in *High Performance Computing & Simulation (HPCS), 2014 International Conference on*, pp. 902–910, IEEE, 2014.

[19] HUANG, Q., GAO, F., WANG, R., and QI, Z., "Power consumption of virtual machine live migration in clouds," in *Communications and Mobile Computing (CMC), 2011 Third International Conference on*, pp. 122–125, IEEE, 2011.

[20] HWANG, K., DONGARRA, J., and FOX, G. C., *Distributed and cloud computing: from parallel processing to the internet of things*. Morgan Kaufmann, 2013.

[21] JUNG, G., JOSHI, K. R., HILTUNEN, M. A., SCHLICHTING, R. D., and PU, C., "Generating adaptation policies for multi-tier applications in consolidated server environments," in *Autonomic Computing, 2008. ICAC'08. International Conference on*, pp. 23–32, IEEE, 2008.

[22] JUNG, G., JOSHI, K. R., HILTUNEN, M. A., SCHLICHTING, R. D., and PU, C., "A cost-sensitive adaptation engine for server consolidation of multitier applications," in *Middleware 2009*, pp. 163–183, Springer, 2009.

[23] KUMAR, S., TALWAR, V., KUMAR, V., RANGANATHAN, P., and SCHWAN, K., "vmanage: loosely coupled platform and virtualization management in data centers," in *Proceedings of the 6th international conference on Autonomic computing*, pp. 127–136, ACM, 2009.

[24] KUSIC, D., KEPHART, J. O., HANSON, J. E., KANDASAMY, N., and JIANG, G., "Power and performance management of virtualized computing environments via lookahead control," *Cluster computing*, vol. 12, no. 1, pp. 1–15, 2009.

[25] MELL, P. and GRANCE, T., "The nist definition of cloud computing," 2011.

[26] NATHUJI, R. and SCHWAN, K., "Virtualpower: coordinated power management in virtualized enterprise systems," in *ACM SIGOPS Operating Systems Review*, vol. 41, pp. 265–278, ACM, 2007.

[27] PARK, K. and PAI, V. S., "Comon: a mostly-scalable monitoring system for planetlab," *ACM SIGOPS Operating Systems Review*, vol. 40, no. 1, pp. 65–74, 2006.

[28] SAHOO, J., MOHAPATRA, S., and LATH, R., "Virtualization: A survey on concepts, taxonomy and associated security issues," in *Computer and Network Technology (ICCNT), 2010 Second International Conference on*, pp. 222–226, IEEE, 2010.

[29] SCHOLL, A., KLEIN, R., and JÜRGENS, C., "Bison: A fast hybrid procedure for exactly solving the one-dimensional bin packing problem," *Computers & Operations Research*, vol. 24, no. 7, pp. 627–645, 1997.

[30] SRIKANTAIAH, S., KANSAL, A., and ZHAO, F., "Energy aware consolidation for cloud computing," in *Proceedings of the 2008 conference on Power aware computing and systems*, vol. 10, San Diego, California, 2008.

[31] VERMA, A., AHUJA, P., and NEOGI, A., "pmapper: power and migration cost aware application placement in virtualized systems," in *Middleware 2008*, pp. 243–264, Springer, 2008.

[32] VOORSLUYS, W., BROBERG, J., VENUGOPAL, S., and BUYYA, R., "Cost of virtual machine live migration in clouds: A performance evaluation," in *Cloud Computing*, pp. 254–265, Springer, 2009.

[33] YANG, J. and LEUNG, J. Y.-T., "The ordered open-end bin-packing problem," *Operations Research*, vol. 51, no. 5, pp. 759–770, 2003.

[34] ZHANG, X., YUE, Q., and HE, Z., "Dynamic energy-efficient virtual machine placement optimization for virtualized clouds," in *Proceedings of the 2013 International Conference on Electrical and Information Technologies for Rail Transportation (EITRT2013)-Volume II*, pp. 439–448, Springer, 2014.