An Energy-Efficient Framework for Dynamic Server Consolidation in Cloud Data Centers

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

Master of Engineering

in

Computer Science & Engineering

Submitted By

Sakshi Jain
(Roll No. : 801532046)

Under the Supervision of:

Dr. Neeraj Kumar
Associate Professor
Computer Science & Engineering
Thapar University,
Patiala (Punjab), India

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY, PATIALA – 147004
July 2017

Certificate

I hereby certify that the work which is being presented in the thesis entitled, *"An Energy-Efficient Framework for Dynamic Server Consolidation in Cloud Data Centers"*, in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Computer Science & Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Dr. Neeraj Kumar*.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Signature
(Sakshi Jain)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Signature
(Dr. Neeraj Kumar)
Associate Professor
CSED
Thapar University
Patiala

Acknowledgment

I have been waiting long for this moment to acknowledge all those who contributed in building this work. It is my pleasure to thank all of them here. First of all, I offer my sincere gratitude to my supervisor, *Dr. Neeraj Kumar*, for accepting to be my supervisor. Without his help and encouraging advice, this work would have never begun. I am deeply indebted to him for providing me wonderful research atmosphere and platform to explore my research to the fullest.

I would also extend my gratitude to *Dr. Maninder Singh*, Head, CSED for providing me the opportunity to conduct my research work. I would also like to thank the Director of the institute, *Prof. Prakash Gopalan* for his continuous support.

My special thanks goes to my friends for discussing thoughts and sharing all ups and downs with me during the course of this work. At the same time I would also like to thank all my colleagues for their continuous support.

Last but not the least I would like to thank my parents and family members, who made me capable of reaching this point of life and for giving me their kind support and love. I dedicate my work to them.

Sakshi Jain
ME (CSE)
801532046

Abstract

Cloud computing is a new internet-based computing which provides computing services as utilities, which are chargeable according to the usage and available on demand over the Internet. It allows the users and enterprises to store and process their data in a cloud infrastructure to let them focus better on their business needs. Rise in popularity of cloud computing has led to the growth of large scale data centers which need significantly higher level of energy consumption. As these data centers remains underutilized most of the time, a significant amount of energy get wasted in keeping the servers active. Efficient management of this energy consumption thus becomes a key issue to address. Various techniques are being developed these days for better power management in cloud. One such method is to dynamically adjust voltage and frequency of a processor according to its workload to reduce the energy consumption of underutilized servers. Another technique is VM consolidation which uses Live VM migration to pack the maximum number of VMs on minimum number of servers to maximize utilization, and thereby, decreasing energy consumption. In case of highly dynamic workloads, these techniques may lead to resource insufficiency in VMs when there is an increased load. Migration should be done in a way that there are no violations to Service Level Agreements (SLA) decided by the cloud consumers and service providers. So overall, there is a need of method that can maintain balance between energy efficiency and performance levels of a data center. Some methods have been proposed to handle this problem but very few of them considered the cost of migrating VMs in terms of time and energy.

We have proposed a strategy for energy-efficient cloud data centers. It makes use of a prediction model based on historical data of workloads to anticipate the upcoming resource demands of applications. This enables identification of optimum number of servers required to host all the VMs, so that underutilized servers can be hibernated. This reduces the energy consumption in data centers without violating the required performance parameters. Our technique is migration cost-aware which means it takes into account the cost associated with the migration in terms of both energy and downtime. The simulation results show that our method provides a significant amount of energy conservation with minimal downtime and number of migrations.

Contents

List of Figures

List of Tables

List of Symbols

W	Watt
r	physical resource
cap_i^r	capacity of PM i for resource r
U_j^r	predicted demand of VM j for resource r
x_{ij}	indicator variable VM j to PM i placement
p_j	profit value associated with VM j
N	Normal Distribution
μ_j^r	predicted mean utilization of VM j for resource r
σ_j^r	predicted deviation of utilization of VM j for resource r
$(\sigma_j^r)^2$	predicted variance of utilization of VM j for resource r
O_j	Set of previous observation for resource utilization of VM j
ξ	Prediction Error
μ_ξ	Mean Prediction Error
σ_ξ	Mean Deviation Error
γ_{max}^r	Maximum allowed capacity of PM for resource r that can be alloacted to VMs
α	Optimum number of active servers at any time
S_i	Score value associated with PM i
$FLAG$	Boolean variable to show if VM is full or not
sf	Sorting Factor to sort VMs in VMset
AS_r	Number of released PMs
AS_b	Number of active PMs before consolidation
AS_a	Number of active PMs after consolidation
PE	Packing Efficiency
P_{DC}	Total Power consumed in DC
P_p	Power consumed by PM p

List of Abbreviations

IT	Information Technology
QOS	Quality of Service
SLA	Service Level Agreement
CSP	Cloud Service Provider
IAAS	Infrastructure-as-a-Service
PAAS	Platform-as-a-Service
SAAS	Software-as-a-Service
SOA	Service Oriented Architecture
XML	Extensible Markup Language
SOAP	Simple Object Access Protocol
WSDL	Web Services Description Language
UDDI	Universal Description, Discovery, and Integration
VM	Virtual Machine
VMM	Virtual Machine Manager
PM	Physical Machine
VHD	Virtual Hard Disk
AVHD	Automatic VHD
RAM	Random Access Memory
CPU	Central Processing Unit
TCP	Transmission Control Protocol
I/O	Input/Output
VMC	Virtual Machine Consolidation
FUSD	Fast Up Slow Down
CDC	Cloud Data Center
DC	Data Center
MKP	Multiple Knapsack problem
LR	Linear Regression
NP	Non Polynomial Time
KP	Knapsack Problem
CONKP	Consolidation algorithm based on Knapsack approach
BW	Bandwidth
OS	Operating System
HPC	High Performance Computation

Chapter 1

Introduction

Cloud computing is a popular and still an emerging computing technology which aims at providing IT resources as a utility or a service. It uses the concept of virtualization to deliver a shared pool of various resources like computing resources, network resources, storage resources, softwares, data, *etc* over the Internet in a pay per use manner. Before the emergence of cloud computing, there was a client-server computing, in which all the software applications, data and controls reside on a centralized storage on the server side. A user on client-side needs to connect to the server to gain access to any data or a program as required. After that, distributed computing emerged, with all the computers networked together which share their resources whenever needed. Both the above computing technologies influenced the emergence of cloud computing as the latter implemented concepts from both. It has taken some part from other existing IT trends as well that involves *autonomic computing*, where IT environment is able to manage itself automatically based on increasing or decreasing demands of users and *utility computing*, where computing power is provided as a 'utility' like other utilities such as water or electricity, for which customers can pay only for what they use. The term "Cloud" in cloud computing refers to a web based infrastructure of IT resources which is typically privately owned and offers access to these resources in an automated and metered fashion. The users request for resources by specifying various constraints for performance, QOS, response time, reliability and availability, pricing negotiations *etc* in an agreement, called Service Level Agreement (SLA), with the Cloud Service Providers (CSP). After this initiation, the users get their share of resources which they can access or re-provision themselves as needed from anywhere through Internet.

1.1 Key features of Cloud Computing

1. **On-Demand Self-Service** : A cloud user can access the cloud resources in a self-provisioned way, that is, once configured, provisioning of resources can be done automatically, without further human involvement by cloud providers.

2. **Ubiquitous Access** : Cloud services are widely accessible over the Internet, that is, can be accessed from any place any time using wide range of devices, interfaces, protocols and technologies.

3. **Multitenancy and Resource Pooling** : Ability to serve multiple cloud users (tenants) simultaneously, which are independent from each other, is *multitenancy*. It enables multiple cloud users to use the same resource or its instance without knowing that it may be used by others.

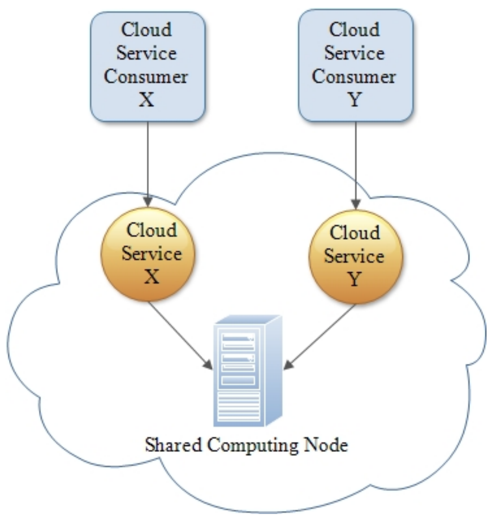

Figure 1.1: A multi-tenant environment where single instance of computing resource is shared by multiple consumers

4. **Scalability and Elasticity** : Elasticity is ability of cloud that transparently scales the IT resources automatically as required by the users in runtime conditions. Scalability is ability of resources to handle the increasing or decreasing usage demands. There are two types of scaling:

   - **Horizontal Scaling** : Adding (Scaling out) or releasing (scaling in) IT resources of same type.
   - **Vertical Scaling** : When an IT resource is replaced with another with higher (Scale up) or lower (Scale down) capacities.

5. **Measured Usage** : Cloud service providers (CSP) can only charge its users for resources they actually used and/or for the time duration they were allowed access to these resources. Monitoring service usage not only helps in billing purposes but also provides an insight to the resource usage pattern that helps in better maintenance of system.

6. **Resiliency** : In cloud, redundant implementation of IT resources across different physical locations provide great resilience against fail overs. Consumers can leverage this characteristic of cloud resources to increase both the reliability and availability of their applications.

1.2 Benefits of cloud computing

- **Reduced Investments and Proportional Costs**: CSPs set up their infrastructures by doing mass acquisition of IT resources in lower prices that they in turn, lease to cloud consumers in attractively priced packages. This helps organizations to access such large infrastructure without needing to purchase it themselves. This way it results in elimination of up front investments for hardware and software purchases. CSPs cut their costs by setting up large scale data centers in localities where real estate, network bandwidth, IT professionals can be obtained at lower costs thereby reducing both capital as well as operational costs. Also, sharing of cloud resources by its consumers results in increased resource utilizations and thus reduced costs.

- **Increased Scalability**: Cloud computing enables its users to dynamically scale cloud resources on demand when needed. Users can scale up the resources to accommodate load fluctuations or peaks in applications automatically and can release resources as workload decreases.

- **Increased reliability and availability**: by redundant implementation of resources and simplified backup and archiving capabilities.

- **Less Energy consumption**: Cloud data centers focus on sharing of resources which leads to saving of energy used by these resources and also reducing the carbon footprints of data centers.

- **Reduced Administrative responsibilities**: Business organizations by adopting cloud centers can focus more on their business-centered strategic decisions than worrying about organizing and maintaining their own IT infrastructure required for business operations.

1.3 Cloud Service Models

CSPs provides services in form of pre-packaged set of IT resources called cloud service models. There are three common models of services:

- **Infrastructure-as-a-Service (IAAS)** : This model offers an IT environment which comprises IT infrastructure centric resources like hardware, network, operating system, connectivity, and other such raw resources. These resources are offered as virtual instances. A typical resource available in IAAS environment is the *virtual server*. The cloud consumers have to specify server hardware requirements such as secondary storage, memory and processor capacity as shown in fig 1.2 below.

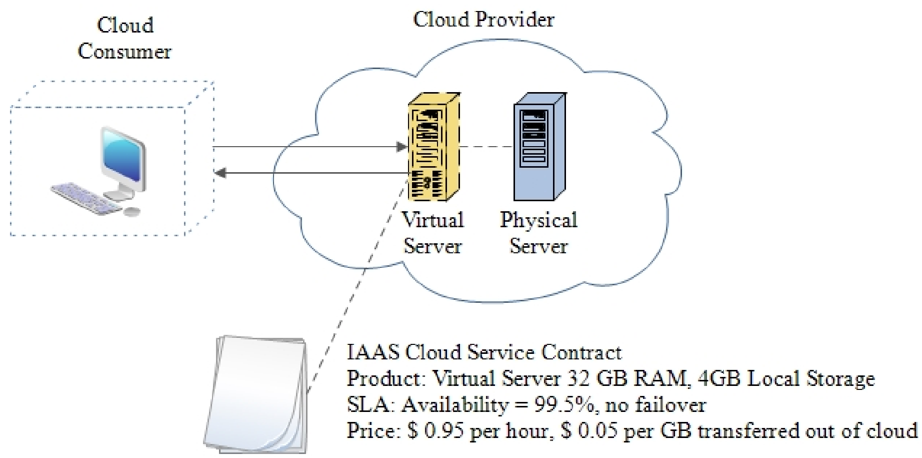

Figure 1.2: A Cloud Consumer requesting for a virtual server through a contractual agreement

- **Platform-as-a-Service (PAAS)** : It offers a ready made environment which consists of already deployed and configured hardware and software resources. It provides the tools and products required to support the life cycle of custom applications. PAAS is beneficial for users when they want to deploy on-premise environments onto the cloud to make them more scalable and reliable or when they want to completely eliminate on- premise environments for economic purposes. PAAS provides different development stacks to be used by customers. Some examples being .NET based environment provided by Microsoft Azure, JAVA and Python based environments provided by Google APP Engine. Fig 1.3 below shows the PAAS environment.

- **Software-as-a-Service (SAAS)** : It offers a software program as a shared service and is available as a metered utility or product. This exploits the re-usability of

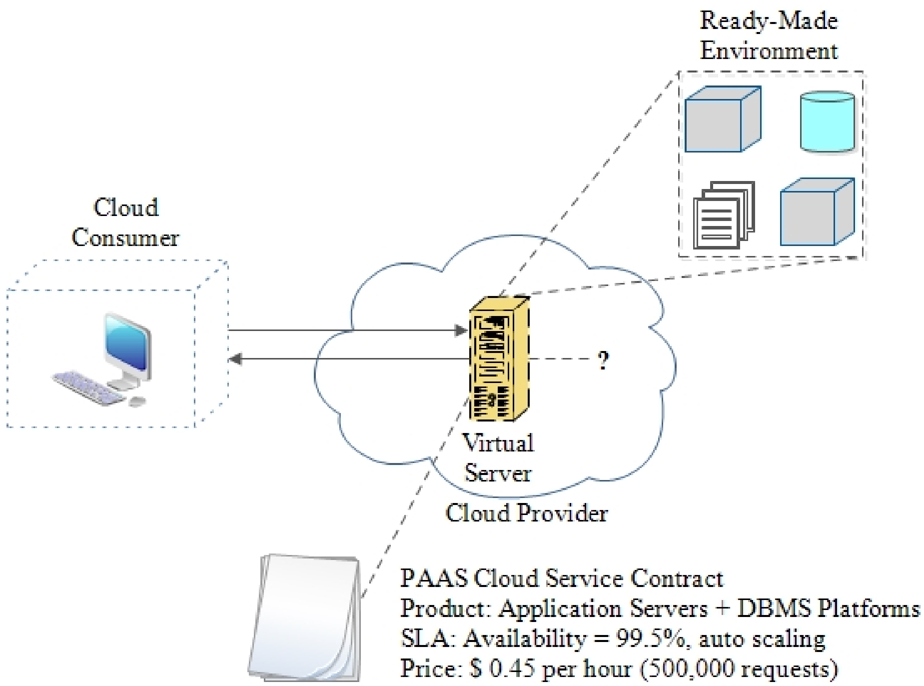

Figure 1.3: A Cloud Consumer accessing ready made environment without knowing internal implementation of platform

some software programs and make them available to a wide range of consumers. In this architecture, users have a little administrative control over the underlying infrastructure and services. Fig 1.4 below shows a SAAS implementation.

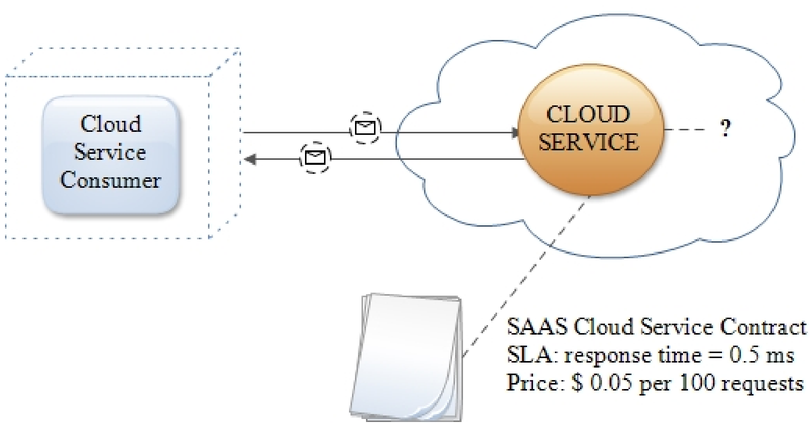

Figure 1.4: A Cloud Consumer accessing Software as cloud service without knowing internal implementation of platform

Table 1.1: Summary of Cloud Service Models

Model	Applications	Responsibility
SAAS	Google Apps, Salesforce, Microsoft Business Productivity Suite Online, Cisco	Softwares are licensed to consumers as services for on demand use. Consumer is responsible for nothing.
PAAS	Google AppEngine, Windows Azure, VMware SpringSource	Provider provides solution stack, computing platform and software development platform which includes hardware, OS, virtual instances and framework stacks. Consumer is responsible for development or deployment of applications.
IAAS	Amazon Elastic Cloud Compute (EC2), VMware vCloud Express, IBM	Provider provides Computing infrastructure and is responsible for the virtual instance and hardware. Consumer is responsible for everything from operating system to application layer.

1.4 Deployment Models

Cloud deployment models represent hosting environments which are distinguished by proprietorship, access and size of cloud infrastructure. Following are the four most common deployment models:

1. **Public Cloud**: A public cloud is a cloud hosting environment which is generally owned by a third party service provider and is open for public use. These service vendors are responsible for the setup and maintenance of the IT resources in public cloud. These services are provisioned to its users at a cost decided at the time of registration into the cloud. It is maintained on the premises of cloud provider so users do not have any idea and control over the location of infrastructure.

2. **Community Cloud**: A community cloud is typically accessible to a specific community of cloud users from different organizations that usually share common concerns (e.g., security requirements, privacy, mission, policy, performance *etc*). The cloud infrastructure may be owned, managed and operated by the members of community or a third party organization that provides a public cloud with limited access. The infrastructure may exist on-premises or off-premises. It is suitable for the organizations that work on joint ventures, tenders and/or research.

3. **Private Cloud**: A private cloud is privately owned and used by a single organization. Organizations usually deploy clouds to centralize access to their IT resources so that they can be used by different departments and locations of the organization. It reduces the risk and challenges related to the management of infrastructure.

4. **Hybrid Cloud**: It is a cloud hosting environment which is a combination of two or more deployment models (public, private or community). It is an adaptation between two environments in which the workloads moves between public and private clouds as per the need. For example, applications with more sensitive data can be deployed to a private cloud while other less sensitive can be deployed on to public cloud.

1.5 Virtualization

Cloud computing evolved over pre-existing technologies which work in a seamless manner to enable cloud to provide the services to its end users. Some of these technologies are web services, service oriented architecture (SOA), virtualization technologies, network virtualization, and shared compute resources (multicore processors) and inexpensive hardware.

1. **Service oriented architecture (SOA)**: It is a way of designing softwares where services are requested and provided among components using application components or modules through the use of some communication protocols over network. A service is a well-defined, self-contained function which is independent from the context or state of other services.

2. **Web services**: A web service is a way of enabling any exchange of data between applications or systems over computer networks by using a collection of open protocols and standards. Any application written in any programming language (say JAVA) and running on any platform (say Linux) can interact and exchange information with any other application written in other language (say .NET) and running on any other platform (say Windows) by using web services. This interoperability is due to the use of open standards like XML, SOAP, WSDL and UDDI.

3. **Virtualization technology**: In computing, virtualization means to create a software-based (virtual) representation of a device or resource such as a storage device, network, server, desktop, or even an operating system. This framework divides the resources into more than one execution environments called virtual instances. It is an effective way to reduce IT expenses by providing multiple devices or applications,

an isolated access to a single physical resource, as if each has a full access to the resource. This concept of sharing resources works because most of the applications do not need the full use of underlying hardware or services. Some of the major benefits that it provides are:

- Reduce both capital and operational costs.
- Increases IT productivity, efficiency, flexibility, scalability and responsiveness.
- Increases reliability, availability and enables disaster recovery.
- Makes provisioning of resources and applications faster and minimizes the downtime.
- Automated operations help build a real software defined data center.
- Simplified data center management.

1.6 Types of virtualization

1. **Network virtualization**: is combining hardware and software network resources into one single entity or an abstract layer as a single virtual network and then sharing it through logical segmentation, each of which is independent from other which can be assigned to any server or device in real time. It can be done by splitting up the available bandwidth into many channels, each of which can be assigned to a device, which makes it easier to manage.

2. **Storage virtualization**: It combines multiple network storage devices or physical storage into a single pool of logical storage which can be managed through a central console. It is commonly used in storage area networks (SAN).

3. **Memory virtualization**: aggregation of random-access memory (RAM) resources from multiple networked systems to create a single memory pool.

4. **Server virtualization**: It provides the ability to mask server resources which includes number of individual physical servers, operating systems, processors, *etc*, from users to increase resource sharing and utilization. The user does not have to understand or manage the complicated details of underlying server resources.

5. **Data virtualization**: It provides abstraction to technical details of data and data management, such as, data format, location or performance and provides broader access and more resiliency according to business needs.

6. **Application virtualization**: is providing an abstraction to application layer to enable it to run in an encapsulated form with isolation and no dependency on the operating system running underneath. Using it, a Linux application can run on Windows or vice-versa.

7. **Desktop virtualization**: is abstracting logical desktop away from the physical machine. It provides the user with remote access to the desktop typically using a thin client.

1.7 Virtual Machine

A virtual machine (VM) is a self-contained software container with an operating system and an application running on it. Each VM is completely independent and isolated from all others. When multiple VMs run on a single computer, they enable several operating systems and applications to run on a single physical server or a host. For example, a system running Microsoft Windows can host a VM that looks like a system which is running Ubuntu Linux operating system and an Ubuntu based application on it. Figure 1.5 shows the architecture of a virtualized server containing VMs. The actual machine on which a

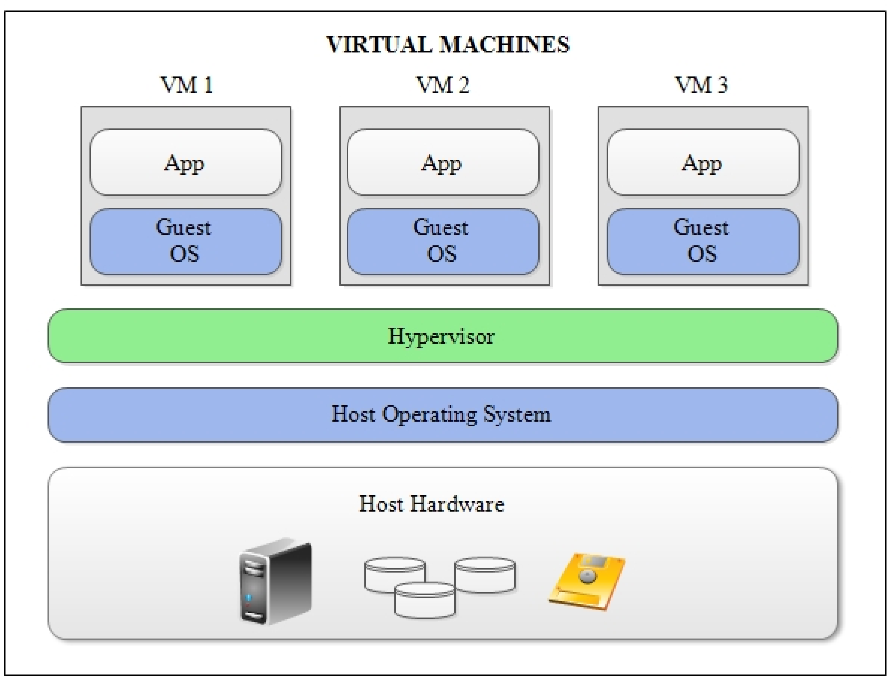

Figure 1.5: A Server Virtualization Model

VM is created is called a *host machine* while the VM which is running over it, providing a different execution environment (than the one provided by the underlying host machine) to its users, is called a *guest machine*. A thin layer of software or firmware, called a *Hypervisor* or *Virtual Machine Manager (VMM)*, is used to create a VM on the host hardware. The VMM is responsible for decoupling a virtual machine from the host and allocating the resources to each VM as needed dynamically. The following characteristics of virtual machines offer several benefits:

- Partition:
  - multiple operating systems can be run on one physical machine
  - system resources are efficiently divided among virtual machines
- Isolation:
  - fault and security isolation is provided at the hardware level
  - preserves performance by providing advanced resource controls
- Encapsulation:
  - entire state of a VM can be saved on files
  - moving and copying VMs is as easy as with files
- Hardware Independence:
  - a VM can be provisioned or migrated to any other physical server easily

1.8 Traditional datacenter vs Cloud datacenter

Huge amount of data and information is transmitted worldwide, so the question that arises is how to store data efficiently. A typical data center space, hardware, electricity and cooling. It demands that someone manages the applications, operating systems and updates within it. In traditional data centers, each application requires its own infrastructure (servers, data storage, *etc*), so if it runs a thousand applications, it would need a thousand servers. While in cloud data center, one common cloud storage hosts all the applications which reduces the costs as compared to traditional data centers.

Traditional Data Center

The fig 1.6 shows the workflow of a traditional cloud data center, which hosts n number of applications within a network. Each application uses its own physical infrastructure. As the number of applications running in this data center increases, the need of physical storage

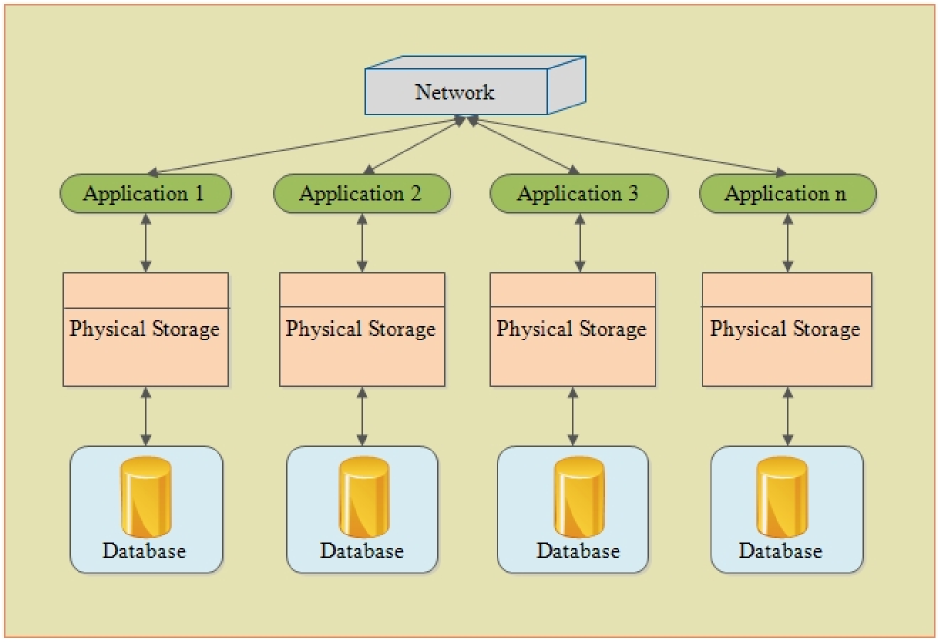

Figure 1.6: Workflow of a Traditional Data center

and thus, the complexity increases linearly.

Key features:

- It stores data within the local network of organization.
- Consists of heterogeneous hardware environment.
- It requires special equipments and knowledge.
- Uses combination of different software architectures.
- Runs complex workloads.
- Cost of running is very high as each application needs separate server. It consumes 42% of costs for maintaining hardware, power supply, networking, and disaster recovery management. Other 40% is used in cooling and labour costs. So, in all, 80% is spent in maintenance only.
- These are comparatively more secured. Within the local network, only authenticated users can have access to it, so security is ensured.
- owner has to maintain complete control over software and hardware.
- In case of failures, it demands semi-automated repair.

- Leasing is in terms of physical resources.

Cloud Data Center

The fig 1.7 gives an idea about the functioning of a cloud data center. There are n number of

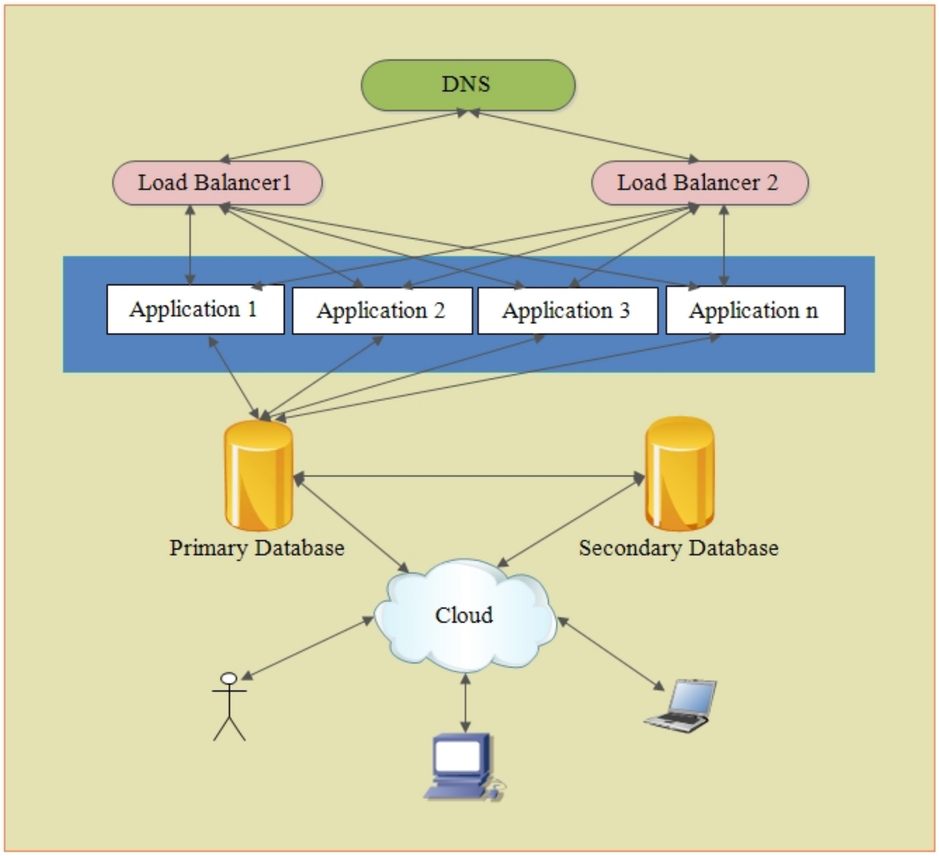

Figure 1.7: Cloud Data center Architecture

applications residing in different locations are hosted on the same cloud. New applications can be added to the cloud at any point of time as it is easily scalable. Both primary and secondary databases are on the same cloud, so even if primary database fails, there will be no loss of data. Applications hosted on cloud are used by users worldwide through the Internet.

Key features:

- It stores data on the internet.
- Consists of heterogeneous hardware environment.
- It requires no special equipments and knowledge.
- Uses one standard software architecture.

- Runs simple workloads.
- Cost of running is low as applications share servers. Overall all, 48% is spent in maintenance.
- These may be comparatively less secured. However, there are many ways available to make data secure on cloud.
- Self-service, pay-as-you-go model.
- In case of failures, it has mechanisms for automated recovery.
- It is platform independent and rents on the basis of logical usage.

1.9 VM Migration and Consolidation

The ability to migrate a virtual machine (VM) from one physical machine to another without interrupting application availability is one of the most important benefits of virtualization technology. In Live migration, a VM's complete memory state along with the state of its processor registers is captured and sent to memory space on another server. That server then loads the registers and the applications in VM proceed to run. Figure 1.8 below shows the process of Live Migration. Many technologies are available for Live migration of VMs and are provided by most major virtualization vendors such as vMotion by VMware, XenMotion by Citrix Systems and Hyper-V by Microsoft. Use of virtualized environments and VM migration in data centers provide various benefits. Some of them are:

- **Load Balancing**: Many applications are dynamic in nature and may increase or decrease their resource requirements according to the nature of workload. In such a scenario, these VMs can be moved to a less busy host machine with enough resource capacities to accommodate the VM. Also, in case of addition or removal of new physical servers, the process of VM migration makes the newly added capacity useful.
- **Maintenance**: Moving VMs is helpful when a physical server needs to be shut down for upgrades or maintenance purposes. Live migration leads to less application downtime even when the hosting server is shut down.
- **Easy backups and recovery from host failure**: With virtualization, full backup of a virtual machine can be created within minutes. These can then be moved from one host to another and VM can then be restarted.

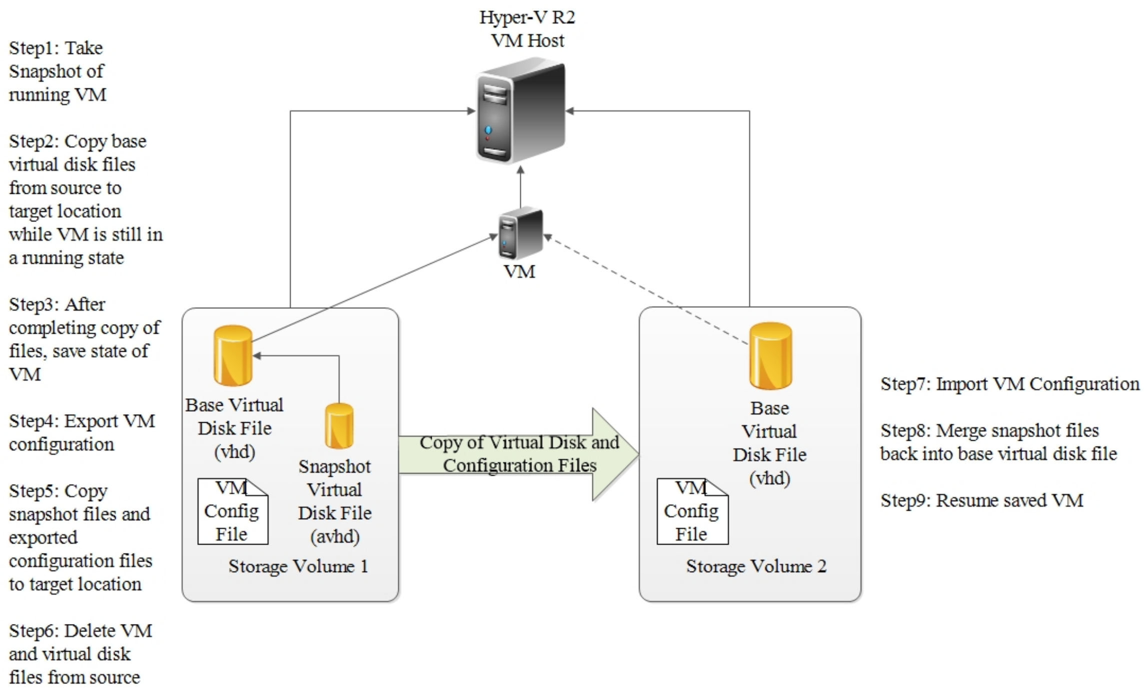

Figure 1.8: Live Migration of a virtual machine

- **Increased uptime**: It gives a VM the ability to keep running and recover quickly from any unplanned outages and thus increases the uptime of applications.

- **Server consolidation**: Using VM migration, more number of workloads can be moved to and run on a single server which reduces the number of servers in a data center.

1.9.1 Types of Migration

- **Cold Migration**: VM is shutdown first and the powered-off VM is then moved to a new host.

- **Suspended Migration**: Moves a suspended VM to a new host by copying across RAM and CPU registers. The VM can the run from the same point with a downtime of some seconds.

- **Live Migration**: Live migration moves the VM's configuration file and CPU register values to a new host while the VM is powered on without interrupting the availability of the VM. VM continues to run while RAM pages are copied to another host. The pages on which the contents are changed while VM is running are marked *dirty* and are re-copied to the other host. After successful completion of copying the VM's contents, VM is stopped at the previous host and is then run on the new host with a brief suspension of less than a second.

Live Migration Techniques

There are two main techniques available for Live migration of VMs. These are:

- **Pre-Copy memory migration**: All the memory pages are copied from source to destination while VM runs on the source before migration. VM is then stopped on the original host and the dirty pages are copied to the destination and Vm then resumes to run on destination host. The downtime ranges between few milliseconds to seconds and depends on the size of memory of applications running on the VM.

- **Post-Copy memory migration**: VM is suspended at the source and a minimal subset of execution state of VM which includes CPU state and register values, are transfered to the destination. VM then resumes to run on target and generates a page-fault for each page that it requests for and that has not yet been transfered from the source. These traps are redirected to source host which responds with the required page.

Using pre-copy technique, a page can be transfered multiple times if it is dirtied repeatedly during migration. Whereas, in post-copy technique, each page is transfered exactly once. In case of a destination fail over, the VM can be recovered in Pre-copy but it is difficult to recover when using Post-copy. This is because in pre-copy, an up-to-date state of VM is retained at the source host during migration whereas post-copy distributes the state of VM over both source and destination hosts.

1.10 Thesis Organization

The thesis is organized into following chapters:

- **Chapter 1**: This chapter introduces Cloud computing, its service models and deployment models. It provides a brief summary of enabling technologies that lay down the foundation for cloud computing. Virtualization is discussed in detail along with its role in distinguishing cloud data centers from traditional data centers and in Live VM migration and Consolidation.

- **Chapter 2**: This chapter provides a review of the state of the art in the related area.

- **Chapter 3**: In this section, problem formulation along with research objectives and assumptions are provided.

- **Chapter 4**: This chapter gives details about the proposed method along with techniques used. It is divided into two sections, one is describes the technique used for prediction of resource demands of applications which are then used in the other section that describes the method of consolidation with the help of flow chart and algorithms.

- **Chapter 5**: Results obtained from proposed method are provided in this section. This chapter evaluates the performance of the proposed method and compares its result on various parameters with three existing VMC algorithms from the literature, namely, FFD, ACO and V-MAN.

- **Chapter 6**: Finally, conclusion and future scope of proposed method are provided in this chapter.

Chapter 2

Literature Survey

2.1 Dynamic Consolidation

Many techniques have been introduced for consolidation of servers in order to improve energy efficiency of cloud data centers. To get an overview of work done in this area some techniques are discussed in this section.

Authors in [1] proposed an integrated energy-aware framework for resource provisioning in cloud data centers. They have used a prediction approach which is a combination of clustering algorithm and stochastic theory. The k-means clustering algorithm is used to to group all similar types of VM requests into one cluster to predict the number of VM requests that the data center will receive in near future. A stochastic Weiner filter method made of k Weiner filters is then used to anticipate the workload of each VM request category/cluster. These predictions are then used to determine the right number of PMs that need to be ON in data center. The presented framework uses a modified version of Best Fit Decreasing (BFD) algorithm where they fit these predicted VM requests in PMs by preferring PMs that are ON and that have higher utilizations and more capacities. Finally they have tested the framework using traces from Google cloud.

Mishra *et al.* in [2] discusses the role of Live VM migration techniques in dynamic resource management, load balancing, hotspot mitigation and minimizing power consumption. They present the detailed approaches used by different heuristics to address the three important components of VM migration *viz.*, *when to migrate*, *which VMs to migrate* and *destination hosts for migrations*. They suggested some situations such as occurrence of a hotspot, server sprawl, load imbalance, *etc* which can trigger the process of migration. Further, the work presents some approaches to select target VMs such as a VM with smaller memory makes the process of migration easy and less time consuming or communicating

VMs that share memory together may be chosen to be placed on the same PM. Finally, the authors conclude with a discussion about the challenges faced with storage migration and network reconfigurations in wide-area migrations.

Xiao *et al.* introduced the concept of ”skewness” in [3] for quantification of unevenness in the utilizations across multiple resources of a server. They present an automated system to achieve the goal of dynamic resource management with overload avoidance and support green computing. A load prediction algorithm called Fast Up and Slow down (FUSD) is used to estimate the future resource usage of applications without looking inside a VM and is designed by introducing two new parameters in calculation of exponentially weighted moving average (EWMA) in a TCP-like scheme. This work aims at minimizing the *skewness* in resource utilizations of a server so that different types of workloads can be combined nicely on a server to provide better overall utilization. Candidate VMs and destination PMs for migration are then selected on the basis of skewness factor in a way such that both migrating a VM away from a PM and placing a VM on a PM results in reduced skewness of both PMs.

Dabbagh *et al.* [4] in year 2016, proposed an energy-efficient framework for resource management in overcommitted clouds. To achieve the optimal utilization of servers and thus to reduce the number of active PMs, this work uses resource overcommitment which means assigning VM resources to a PM in excess of its actual capacity, assuming that these VMs will not utilize their requested resources fully. An *online* predictive module using Weiner filters, which does not require any prior knowledge about VMs, is used to avoid PM overloading ahead of time and to increase utilization gains due to overcommitment of host resources. The proposed framework has migration cost aware module which takes into consideration the energy spent to move VMs and the energy spent in switching PMs ON /OFF.

Li *et al.* [5] in 2015 presented the first comprehensive evaluation of scheduling policies for VM migration in order to reduce the energy consumption. Many policies for determination of VM migration timing, selection of migrated VMs and selection of destination servers have been compared both separately and in combinations. For determination of migration timing, Double Threshold trigger policy (DTTP) which sets upper and lower thresholds on CPU utilizations and trigger migration whenever thresholds are compromised and Single Threshold policy, that is, setting the upper threshold only, are discussed. To determine the threshold values, policies like Median Absolute Deviation (MAD), Interquartile range (IQR), Local regression (LR), LR Robust (LRR) are discussed and compared in details. For selection of VMs to be migrated, policies like Minimizations of migrations (MM), Highest Potential growth (HPG), Random selection policy (RS), Minimum migration time (MMT) and maximum correlation (MC) policy are discussed and compared. Greedy heuristic (GH), modified first fit heuristic (MFFH), optimal VM allocation (OP) and best from random heuristic (BFRH) are some of the policies compared for selection of destination servers. This paper introduced a new set of performance evaluation metrics which were used to evaluate the policies and also presented an evaluation environment constructed according to these metrics.

Authors in [6] designed a power-aware application placement controller *pMapper*, which uses an energy-aware placement algorithm mPP to determine a target scheme. It then compares the target and previous schemes to classify the PMs as receivers and donors and finally, the VMs are re-allocated from the donors to the receivers. Migration costs are reduced as it keeps all the VMs in receivers unmoved. However, it limits further power savings by keeping the VM assignments in receivers fixed.

Murtazaev *et al.* [7] proposed a server consolidation algorithm called Sercon which minimizes the number of active servers while also minimizing the number of migrations. This scheme tries to either move all the VMs in one PM to the other active PM or moves none of the VMs at all, following an all-or-none strategy. On a successful attempt, this method provides a new consolidated placement scheme with one less active server. This algorithm iterates until no further improvement is found. One problem with this scheme is that it can not consolidate the servers if an extra migration is needed. For example, if virtual machine *vm1* hosted on *pm1* can be moved to *pm2* only after we move a small virtual machine *vm2* hosted on *pm2* to *pm3* and *pm1* can then be turned off. Sercon fails to consolidate PMs in this case.

Wu *et al.* in [8] presents a migration cost aware dynamic VM consolidation scheme which works on a bi-objective optimization problem. One objective is to consolidate VMs in a heterogeneous cloud data center and the other is to limit the migration costs to save maximum amount of energy. This scheme designed a consolidation score function for overall evaluation of the two conflicting objectives. This method deals with VM consolidation problem as a grouping problem and devises an improved grouping genetic algorithm (IGGA) that uses a greedy heuristic and a swap operation to optimize the consolidation score. IGGA was tested on a simulation environment based on traces from google cluster and the results show that it achieves the highest consolidation score when compared with traditional consolidation methods. however, the computation time of IGGA becomes unacceptable after number of VMs becomes extra ordinarily large.

Ghribi *et al.* in [9] proposed two Exact algorithms for optimal allocation and migration of VMs that jointly minimizes the energy consumption in cloud data centers. The VM placement algorithm uses an extended bin-packing approach by including valid conditions in form of inequalities or constraints. The objective is to pack VMs into a set of PMs. The exact migration algorithm is formulated as a linear integer program (ILP) to re-optimize the placement of VMs for consolidation in minimum number of host machines. The proposed algorithms are implemented on a JAVA based tool and results are compared with that of Best-Fit heuristic and are found to be better.

Reguri *et al.* in [10] presented three VM migration schemes for energy conservation in computing, migration and host re-activating. They have considered the communication traffic factor and hence adopted VM clustering. The VMs are divided into clusters by grouping VMs that exchange more traffic and have highly correlated services. An entire VM cluster is then selected to be migrated to another server. They have proposed three cluster placement algorithms namely, *Least Increased Power*, *Best Fit Host* and *Best Fit VM* depending upon three different criteria. Results of simulations show that VM clustering reduces migration costs dramatically and results in overall saving in energy consumption. The proposed work can be applied on wireless mobile cloud where communication cost is significant.

Beloglazov *et al.* [11] presented an architectural framework and power model for green cloud computing. They proposed energy-aware resource allocation heuristics for dynamic consolidation of VMs that take into account the energy efficiency as well as the negotiated QOS. The proposed approach is evaluated using Cloudsim toolkit and results offer significant energy and cost savings. A number of open research challenges in the area of improvement in energy efficiency in cloud data centers has been discussed in the end.

A stochastic load balancing scheme is proposed by Yu *et al.* in [12] which provides a probabilistic guarantee against resource overloading in case of VM migration with highly dynamic resource demands. They consider the stochastic properties of workload in prediction of resource demands of VMs. The proposed work makes migration decisions whenever a hotspot is detected in a probabilistic way that there is a high probability of total demand of VMs not exceeding the resource capacity of the host PM while minimizing the migration overhead. This work characterizes the resource demands of workloads as random variables and maintains a probability that the requested resource demands does not exceed a PM's capacity for each resource type. This probability is decided by the SLA agreement with consumers and it controls the degree of SLA violations which is tolerable for each PM. The proposed migration algorithm considers network topology in minimizing the migration cost for load balancing and provides an improved worst case performance of the system in presence of hotspots.

Wadhwa *et al.* in [13] presented a VM placement and migration technique for efficient management of carbon emission and energy consumption in federated cloud data centers.

This work can be applied to distributed cloud data centers, each data center having different carbon footprint rates depending on its energy source. For each new VM request, the placement algorithm finds a suitable host in most carbon-efficient data center available among all. The migration algorithm is then used to optimize current allocation of VMs inside each data center in order to reduce energy consumption. Minimization of migration (MM) policy is used for selection of VMs to be migrated from the over or underutilized hosts. For selection of suitable host to place these VMs, a modified version of Best Fit Decreasing algorithm (MBFD) is used.

A light-weight *interference-aware* live VM migration strategy *iAware* is proposed by Xu *et al.* in [14] to estimate and minimize performance interference and cost during and after VM migration on both source and destination servers. A simple demand-supply model is designed for multiple resources to measure and mitigate both VM migration interference and VM co-location interference experienced on the source and destination servers during live migrations of VM. The results of evaluation show that I/O and network throughput and runtime overheads in terms of CPU consumption are improved substantially in comparison to traditional interference unaware migration strategies.

A power management approach *Dynamic Voltage and Frequency Scaling (DVFS)* is proposed in [15, 16] which saves energy by dynamically adjusting the operating frequency and voltage of CPU in PMs according to the resource demands of workloads in the cluster. A prediction model can be used to anticipate the demands apriori and make the adjustments accordingly to ensure better management of energy efficiency in data centers. Although these techniques provide some level of energy conservation but higher energy savings can be achieved by consolidating workloads and switching off the unnecessary machines.

Authors in [17] present a novel strategy *Predictive Elastic Resource Scaling* (PRESS) for minimizing energy consumption and costs associated with resource provisioning while maintaining service level objectives (SLO). This method uses signal processing and statistical learning algorithms to predict dynamic resource requirements of applications. The resource allocations of each scheduled VM are then tuned (*VM resizing*) according to these predictions.

Tighe and Bauer in [18] combined dynamic consolidation of VMs with auto scaling of applications to match changing workload demands. VM resizing is performed to scale up and down according to the resource requirements of applications. These resized VMs are consolidated on fewer number of ON PMs using VM migrations.

EnaCloud, a novel approach proposed by Li *et al.* in [19] is used to enable live placement of applications dynamically to improve energy efficiency in a cloud platform. They consider the problem of application placement as a bin packing problem and use an energy-aware heuristic which extends the Best Fit (BF) algorithm to minimize the number of running machines. In order to cope up with the varying resource demands of applications, they use an over-provisioning approach and the results of implementation show that this approach is feasible and successfully conserves energy.

Canali and Lancellotti in [20] presents a new technique of Class-Based Placement (CBP) which groups VMs that show similar behaviours. It considers VM placement problem as a multi-dimensional bin packing problem. CBP selects some representative VMs for each class and use them as a building block to solve the global placement problem while aiming to minimize the number of physical servers to reduce energy consumption.

A fully decentralized dynamic VM consolidation model is presented by Feller *et al.* in [21] which is based on an unstructured P2P (peer-to-peer) network of PMs. The schema is evaluated using three popular VM consolidation algorithms *viz.* Sercon, First-Fit-decreasing (FFD) and V-MAN. This work improves the scalability or packing efficiency of VMs on PMs by eliminating the application of these VMC algorithms on traditional centralized and hierarchical system topologies.

In [22] Marotta and Avallone proposed a Simulated Annealing based algorithm to solve the consolidation problem in order to reduce the number of active nodes and increase overall cost-efficiency.

Elnozahy *et al.* present five policies in [23] that use various combinations of two power saving techniques namely, node vary-on/vary-off (VOVO) and dynamic voltage and frequency scaling (DVFS) for cluster-wide power management in data centers. This work estimates the total CPU frequency needed by servers to provide necessary response time and also determines the optimal number of physical servers and divide the total estimated frequency among all the nodes and set the proportional frequency accordingly to each node.

The workload consolidation problem is modeled as a Multi-dimensional Bin Packing (MDBP) problem by Feller *et al.* in [24]. It considers the problem as multi-dimensional because it takes into account multiple types of resources. This problem is then solved using a nature inspired algorithm called *Ant Colony Optimization* (ACO) which is suitable to be applied in a fully distributed environment. The ACO based approach is evaluated by comparing it with a greedy algorithm called *First Fit Decreasing* (FFD) and results show that better energy savings and lesser active machines are obtained by using the former approach.

Khosravi *et al.* in [25] proposed a novel VM placement algorithm to reduce the power consumption and carbon dioxide emission in distributed data centers. They take into account the different carbon footprint rates, power usage effectiveness (PUE) and energy sources of these data centers to increase the environmental sustainability. This approach reduced carbon dioxide emission and power consumption in data centers while maintaining the negotiated level of QOS.

Wang *et al.* in [26] present a VM placement approach based on an improved version of Particle Swarm Optimization algorithm. This approach takes into account the heterogeneity in physical server configurations in national cloud data centers (NCDCs) and provides minimized energy consumption with a global QOS guarantee during the provisioning of data-intensive services.

Xu and Fortes in [27] proposed a two-level control system for management of workloads to VMs mapping and Vms to physical resources mapping. They formulate the problem of VM placement as a multi-objective optimization problem where the objectives are to minimize power consumption, total resource wastage and thermal dissipation costs. For efficient searching of large solution space, an improved grouping genetic algorithm is devised and is used with multi-objective evaluation. This approach provides a good balance among conflicting objectives.

Ferreto *et al.* in [28] present an LP formulation and heuristic for dynamic consolidation and attempts to reduce number of migrations by prioritizing VMs with steady capacity demands and controlling the migrations of such VMs. This work is tested on Google data center workloads and results show that avoiding the migrations of prioritized VMs reduces the number of migrations and has minimal negative impact on the consolidation of physical servers.

Tao *et al.* [29] present a binary graph matching based bucket-code learning algorithm (BGM-BLA) to solve the dynamic migration of VMs (DM-VM) problem. They make a triple objective optimization formulation for DM-VM where the three objectives are energy consumption, migration cost and communication between VMs.

Authors in [30] present a solution for power management in a data center by combining five different policies. These different approaches are coordinated by applying a feedback loop to the controller. This work deals with the management of CPU only and is independent of the type of workload.

2.2 Load Prediction

A significant amount of energy overhead is associated with turning PMs ON/OFF, so a PM should be turned OFF only if there is a surety that it will remain OFF for long enough that the overhead will be compensated. For this, there is a need to predict the resource demands of workloads to be able to make a correct decision. Many techniques have been proposed to predict the workload resource usage patterns and some of them are discussed below.

Table 2.1: Summary of Literature Survey

Techniques	**Multi-Resource**	**Prediction Module**	**Energy/ Power Saving**	**Reduced Number of Migrations**	**Migration Cost Aware**
[1]	Yes	Yes	Yes	No	No
[2]	Yes	No	Yes	Yes	Yes
[3]	Yes	Yes	Yes	Yes	No
[4]	No	Yes	Yes	No	Yes
[5]	Yes	No	Yes	Yes	No
[6]	No	No	Yes	Yes	Yes
[7]	No	No	Yes	Yes	No
[8]	No	No	Yes	Yes	Yes
[9]	Yes	No	Yes	Yes	No
[10]	No	Yes	Yes	Yes	Yes
[11]	Yes	No	Yes	Yes	No
[12]	Yes	Yes	No	Yes	Yes
[13]	Yes	No	Yes	Yes	No
[14]	Yes	No	Yes	No	Yes
[15]	No	Yes	Yes	No	No
[16]	No	Yes	Yes	No	No
[17]	No	Yes	Yes	No	No
[18]	No	Yes	Yes	No	No
[19]	Yes	Yes	Yes	No	No
[20]	Yes	Yes	Yes	No	Yes
[21]	No	No	No	No	No
[22]	No	Yes	Yes	No	No
[23]	No	Yes	Yes	No	No
[24]	Yes	No	Yes	No	No
[25]	No	No	Yes	No	No
[26]	Yes	No	Yes	Yes	Yes
[27]	Yes	No	Yes	No	No
[28]	Yes	Yes	Yes	Yes	Yes
[29]	No	Yes	Yes	Yes	Yes
[30]	No	No	Yes	No	No

Chandini *et al.* [31] discuss various load prediction approaches and models to minimize the downtime of VMs during dynamic consolidation of VMs in cloud data centers. This work discusses prediction using Bayesian model, Neural network (NN) model, Support vector (SVM) and Kalmann smoother, Phase Space reconstruction (PSR) and Group method of data handling (GMDH) based on Evolutionary Algorithm (EA). This work also provides methods to find error rates and use them to compare the performance of prediction techniques.

Ismaeel and Miri [32] designed a new prediction model to forecast future load in terms of VM requests, CPU, memory and other parameters. They combined the Extreme learning machines (ELMs) technique with the k-means clustering technique in their work to estimate future VM requests by using the traces from historical resource usage pattern of workloads in a data center.

Authors in [33] present a prediction algorithm to forecast the load on each PM by predicting the future resource demands of all VMs. This work uses a modified version of FUSD algorithm to predict the resource requirement by using the historic resource usage as observation. This method captures both the acceleration and deceleration in resource usage and helps to decide which PM is going to be overloaded next.

Liu *et al.* in [34] proposed a time series based algorithm for workload prediction of host machines. The past workload values for a host are gathered to make a history of workload. This history is used in predicting the future workload trend using the cloud model based time series workload prediction technique.

A brief summary of literature review is given in table 2 on the basis of parameters that were considered during work.

Chapter 3

Problem Statement

3.1 Research Motivation

Physical servers in cloud data centers (CDCs) consume a huge amount of energy for servicing application requests. These requests are serviced by VMs hosted on servers. These VMs are provisioned dynamically according to need and are freed periodically. This results in poor utilization of resources over time. A lightly loaded or an idle server waste about 65% of its peak power [4]. Moreover, the cooling systems require an extra 0.3 to 0.8 W of power when a computing resource consumes 1 W of power. Thus, to reduce the energy consumption in CDCs and make them more eco-friendly, server consolidation is required to turn off the idle PMs maximize the overall resource utilization within the data center. With this, it is necessary to also keep in mind the QOS parameters like throughput, response time, bandwidth, delay *etc* to provide the negotiated performance levels to cloud consumers. Also, with different types of applications like social network, enterprise and scientific applications, resource demands can vary for each application as it can be data, network or compute intensive. Most of the works done in consolidation usually considers these applications to be having a uniform workload or of one particular workload type. So, there is need for consolidation mechanism which combines these heterogeneous workloads in an energy-efficient manner while meeting SLAs.

3.2 Problem Statement

Consider a data center (CDC) containing n number of PMs P1, P2, P3, ..., Pn and m number of VMs V1, V2, V3, ..., Vm are hosted on these PMs. Each host in data center has multiple types of resources such as CPU, memory and network resources. We assume that each resource type r has a maximum capacity cap_i^r on each PM i. We assume that resource demands of applications hosted on VMs can be taken as a random variable U_j^r which follows a probability distribution as predicted by a prediction algorithm. The problem of consolidating the number of servers then can be formulated as an instance of multi-dimensional 0/1 Knapsack problem [35], where the aim is to fill n number of knapsacks with a set $I = 1,2,...,n'$ of items. Each item j has a profit p_j and weight value w_j associated with it and knapsack has capacities c_i, $i \in 1,2,...,n$. Then the knapsack problem is filling the knapsacks in such way that total profit is maximized while sum of weights in each knapsack i does not exceed c_i.

In our case, PMs act as knapsacks with multi-dimensional resource capacities cap_i^r while VMs are the items with resource demand U_j^r as multi-dimensional weight values and p_j as profit value. These profit values are used to minimize the number of migrations taking place during consolidation. We take an indicator variable x_{ij} which denote a mapping between a VM and a PM as follows:

$$x_{ij} = \begin{cases} 0, & \text{if VM j is not placed on PM i} \\ 1, & \text{if VM j is placed on PM i} \end{cases} \tag{3.1}$$

The problem of consolidating VMs can be mathematically formulated as:

$$maximize: \quad \sum_{i=1}^{n} \sum_{j=1}^{m} p_j x_{ij} \tag{3.2}$$

$$s.t. \quad \sum_{j=1}^{m} U_j^r x_{ij} \leq cap_i^r, \quad i \in \{1,..,n\}, \quad r \in \{cpu, memory, network\} \tag{3.3}$$

$$\sum_{i=1}^{n} x_{ij} \leq 1, \quad j \in \{1,..,m\} \tag{3.4}$$

where p_j, U_j^r and cap_i^r are positive integers. The objective function mentioned in equation3.2 maximizes the total profit which in our case maximizes the number of VMs packed on servers. Constraint 3.3 ensures that for each resource type, sum of total resource demands of VMs hosted on a PM does not exceed the resource capacity of that PM while Constraint 3.4 ensures that each VM is assigned only to one PM.

3.3 Assumptions

- Both PMs and VMs have multi-dimensional resource capacities and demands respectively. For our work, we are taking three types of resources namely, CPU, memory

(RAM) and network resources.

- We are assuming that servers in DC are homogeneous with respect to resource capacity, that is, each PM has approximately equal maximum available capacity for each type of resource.
- VMs are already placed on servers.
- We are aware of upcoming resource requests for each VM.
- During the process of consolidation, maximum resource capacities of VMs remain same.

3.4 Objectives

Following are the objectives that this thesis work achieves:

- Minimize overall energy consumption of data center
- Maximize resource utilization
- Minimize number of active servers in data center
- Minimize number of migrations caused due to consolidation
- Minimize SLA violation caused due to consolidation

Chapter 4

The Proposed Scheme

This chapter provides the details of approach used in solving the problem stated in previous chapter. The solution can be divided into two main sections, first is Prediction, which gives details about the algorithm used to predict the future resource demands of applications based on which the VMs can be consolidated on servers, and the second section is Consolidation, which provides the solution to find an optimum number of servers required to host all the VMs according to the results of prediction algorithm and the method used to solve the MKP problem and the problem of VM migration.

4.1 Prediction

In order to make efficient VM consolidation and migration decisions, there is a need to predict the resource utilizations of VMs such that these VMs can be placed on PMs efficiently, providing better resource management and energy conservation, and reduced SLA violations and performance degradation caused due to load imbalance or frequent migrations of VMs. There have been many attempts to predict the resource requirements of applications but due to the highly dynamic workloads running on these VMs, most of these methods result in inaccurate predictions. This leads to inefficient use of resources as over estimation in predictions result in wastage of resources while under estimation results in SLA violations. To overcome this, we have considered the resource demands of workloads as random variables which follow a probability distribution. The work in [12] shows that most of the real world applications have resource usage in a pattern that follows Normal distribution. We are taking a random variable $U_j^r \sim N(\mu_j^r, (\sigma_j^r)^2)$ to represent the resource demands of VMs for each resource type r. This way using the variance of application's resource usage with mean usage helps to avoid resource overloading and SLA violations in case of dynamic workloads. We use a prediction technique based on Linear regression

(LR) model [36] which fits a curve that satisfies all values lying in the set of previous n' observations of resource demands of VMs for each resource. This technique however, is too simple and can not determine the changing trends or uncertainties in resource usage of these VMs. So every time a new prediction is made, the mean error from past predictions is used to guide the change in resource usage pattern and variance of these errors are used to handle the uncertainties.

Algorithm 1: Prediction Algorithm

Input: O_j is the set of previous n' observations $o_{j1}, o_{j2}, .., o_{jn'}$ of resource usage of VM j for resource r

Output: μ_j predicted mean resource demand and σ_j^2 predicted variance of resource demand of VM j for resource r

1 Use Linear regression (LR) model to fit a curve to the set of n' observations of resource usage of VM j for resource r, using Least squares approach.

2 Calculate the prediction to the resource demand of VM j $\hat{o}_{n'+1}$ in next interval using the (LR) curve.

3 Calculate the errors in predictions for n recent observations:

$$\xi = o_{jk} - \hat{o}_{jk}, \quad k \in \{1, 2, ..., n\} \tag{4.1}$$

4 (I) Calculate the mean of errors $\xi_{j1}, \xi_{j2}, ..., \xi_{jn}$ as:

$$\mu_\xi = \frac{1}{n}\sum_{i=1}^{n} \xi_i \tag{4.2}$$

(II) Calculate the variance in errors $\xi_{j1}, \xi_{j2}, ..., \xi_{jn}$ as:

$$\sigma_\xi^2 = \frac{1}{n}\sum_{i=1}^{n} (\xi_i - \mu_\xi)^2 \tag{4.3}$$

5 Calculate μ_j as:

$$\mu_j = \hat{o}_{n'+1} + \mu_\xi \tag{4.4}$$

6 Calculate σ_j^2 as:

$$\sigma_j^2 = \sigma_\xi^2 \tag{4.5}$$

4.2 Consolidation

Fig 4.1 shows the scenario where consolidating VMs on lesser number of servers and then shutting down the idle servers result in a great deal of overall energy savings in a data center. We have worked with homogeneous servers in data center that have similar resource capacities but this method can be used for heterogeneous servers also. We have compared the problem of consolidation to the multiple knapsack problem (MKP), where the capacities and weights of knapsacks and items respectively are multi-dimensional, to take multiple types of resources into consideration. For MKP though, number of knapsacks should be known as an input. So, we have divided this work further into two parts, one deals with determining number of optimal servers to host all the workloads in data center and the other deals with the method of performing VM migration and finding the VM-to-PM mapping using only the optimum number of servers.

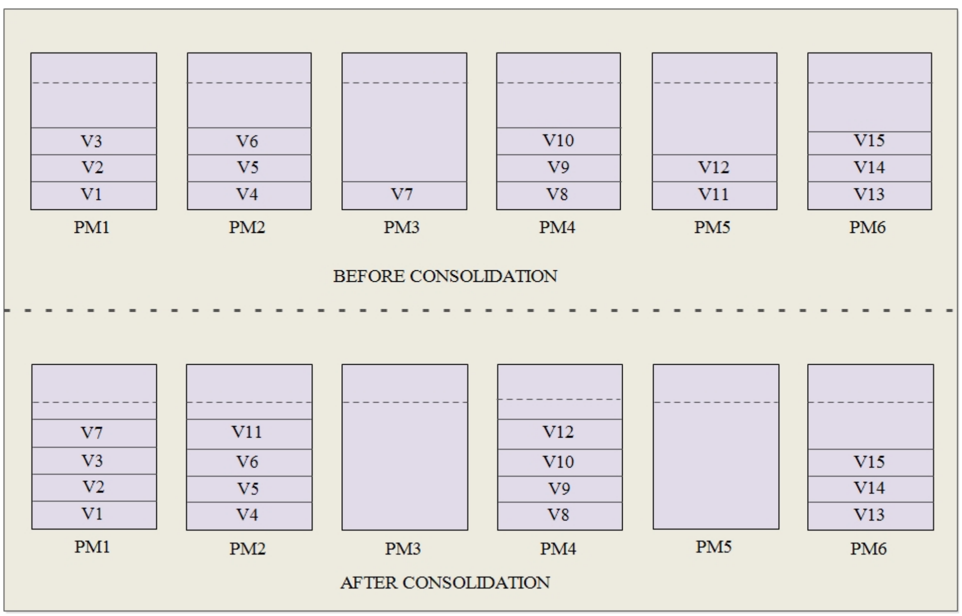

Figure 4.1: A view of cloud data center with and without consolidation

4.2.1 Optimum number of servers

Equation 4.6 below gives the optimum number of servers α required to host all the VMs in a data center at a given point of time. Calculation in equation uses the results of prediction of future resource demands of VMs to avoid resource overloading and SLA violations.

$$\alpha = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} U_j^r}{\gamma_{max}^r} \tag{4.6}$$

where, $U_j^r \sim N(\mu_j^r, (\sigma_j^r)^2)$ is the predicted resource demand for resource r of VM j, γ_{max}^r is the maximum capacity of each PM for resource r, $j \in \{1, 2, .., m\}$ represents number of VMs and $i \in \{1, 2, .., n\}$ represents the number of PMs in data center.

As, U_j^r is a random variable that follows a normal distribution [37] $N(\mu_j^r, (\sigma_j^r)^2)$, the values

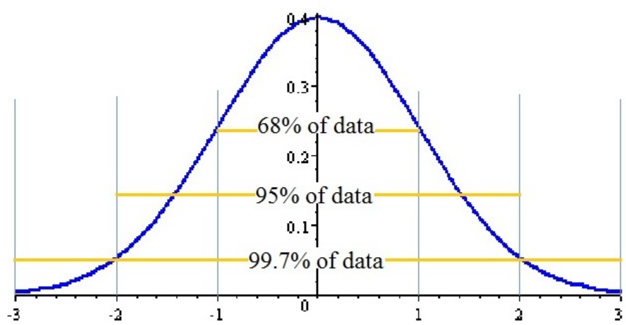

Figure 4.2: Standard Normal Distribution Curve

of μ_j^r and $(\sigma_j^r)^2$ can not be combined directly. We use the distribution chart [37] given in figure 4.2 to determine the overall value of U_j^r as follows:

$$\alpha = \frac{\sum_{i=1}^{n}(\sum_{j=1}^{m} \mu_j^r + 4 * \sum_{j=1}^{m} \sigma_j^r)}{\gamma_{max}^r} \tag{4.7}$$

where μ_j^r is the mean predicted value and σ_j^r is the predicted variance of resource demands of VMs. As shown in figure, four levels of deviation when added to mean value in normal distribution graph makes the curve closer to 100% and gives a probability of 1. We have included the deviation of applications to reduce the chances of overloading in case of dynamic workloads. With workloads that have lesser deviations the U_j^r value will be closer to μ_j^r.

In our work, the equation 4.7 is applied to all three resource types, cpu, memory and network resources. The calculation of optimum number of servers is done in all three dimensions and the maximum value among three is picked to determine the optimum number of servers. This is done to satisfy the need for resources of VMs according to all resource types.

$$\alpha_{cpu} = \frac{\sum_{i=1}^{n}(\sum_{j=1}^{m} \mu_j^{cpu} + 4 * \sum_{j=1}^{m} \sigma_j^{cpu})}{\gamma_{max}^{cpu}} \tag{4.8}$$

$$\alpha_{mem} = \frac{\sum_{i=1}^{n}(\sum_{j=1}^{m}\mu_j^{mem} + 4*\sum_{j=1}^{m}\sigma_j^{mem})}{\gamma_{max}^{mem}} \tag{4.9}$$

$$\alpha_{net} = \frac{\sum_{i=1}^{n}(\sum_{j=1}^{m}\mu_j^{net} + 4*\sum_{j=1}^{m}\sigma_j^{net})}{\gamma_{max}^{net}} \tag{4.10}$$

4.2.2 VM Consolidation

We have formulated the consolidation problem as MKP which is an NP hard problem. One solution is the brute force approach which solves the problem in polynomial time only if the problem space is decent in size. We have solved the problem by taking single knapsack one by one until the number of knapsacks equals the number of optimum servers. Following figure 4.3 shows the flowchart of our solution.

Choosing the Donor server

We assign a score value with each server in data center and the server with least score is selected as a donor server. If once a donor selected, consolidation is performed and results in emptying the donor server, then that server is shutdown and next donor server with next least score value is selected.

Score function

The score function is based upon the utilization of physical servers and is the ratio of server resources utilized to the resource capacity of server. The Score function is given in equation 4.11 as:

$$S_i = \frac{\sum_{j=1}^{m} U_j^r x_{ij}}{cap_i^r} \tag{4.11}$$

where U_j^r represnt the resource utilization of VM j for resource r, x_{ij} is used to sum the utilization of VMs that are placed on PM i only and cap_i^r is the total capacity of PM i for resource r.

Choosing the Knapsack server

The server whose score value is the highest represents the server which is maximum filled. So, the receiver or knapsack server is selected based upon the score value and the server which has highest score is chosen. If more than one servers have the same score values then the tie is broken by choosing the most power-efficient server out of them as the Knapsack or receiver server. By choosing the most utilized server, we are reducing the number of

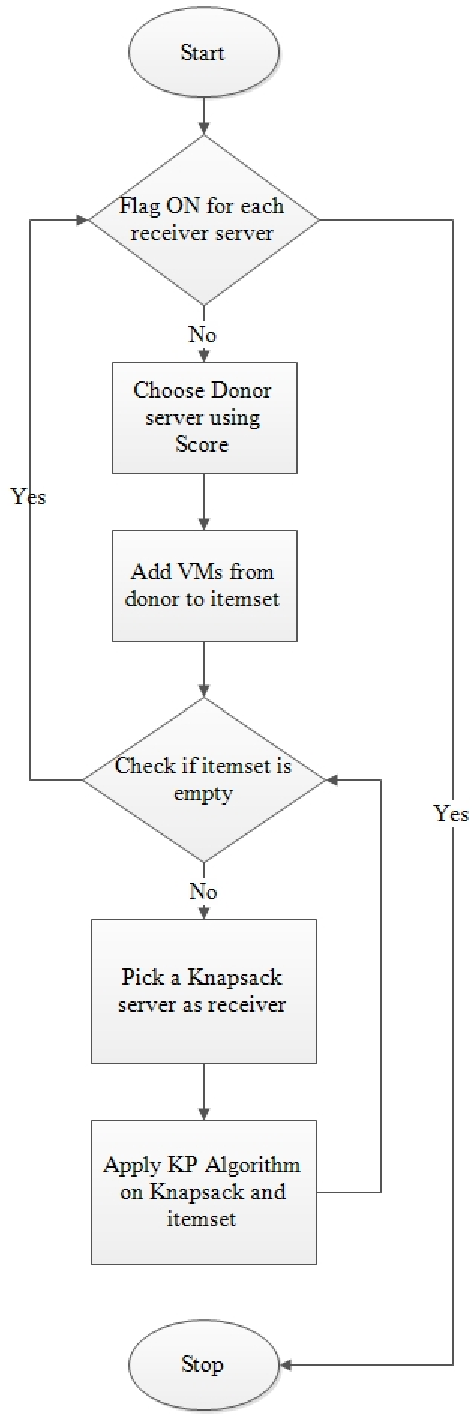

Figure 4.3: Flowchart of proposed work

migrations that would take place during consolidation if this server had made to be a donor server. So, number of migration are reduced by picking the least filled servers as donors and most filled servers as receivers first.

Assigning profit values to VMs

When assigning profit values to VMs, the VMs that are already placed on knapsack server are given a higher value (2) as compared to the VMs (1) that are added to the VM itemset by the donor PM. This again helps in minimizing the number of migrations by increasing the chances of VMs which are already placed on receiver to remain there and does not have to be migrated. Also, when the knapsack problem is solved, maximizing the profit values indicate maximizing the number of VMs on a PM. The profit values are assigned to VMs as follows:

$$p_j = \begin{cases} 1, & \text{if VM j is not placed on knapsack PM i} \\ 2, & \text{if VM j is placed on knapsack PM i} \end{cases} \tag{4.12}$$

Setting Flag Values

While consolidating, there is a need to determine if the servers are full to terminate the process. This is done using a boolean variable FLAG. It is set to 1 for each PM that can not host any more VMs and 0 for PMs that still have the resource capacities to be allocated to VMs.

$$FLAG_i = \begin{cases} 0, & \text{if PM i has resource capacity to host a VM} \\ 1, & \text{if PM i is full} \end{cases} \tag{4.13}$$

Consolidation Algorithm using knapsack approach

The algorithm 1 provides the required process of iteration for each knapsack and uses the GreedyKP algorithm 2 to solve the single knapsack problem. In greedykp algorithm, VMs in VMset are sorted according to the sorting factor sf. By, using sf, we are giving priority to the VMs with higher profit values and VMs that are smaller in size, so that maximum number of VMs can be packed in a server.

Algorithm 1 Consolidation Algorithm using knapsack approach

Input: S_i score values of all PMs, V set of all VMs, P set of all PMs

Output: x_{ij} the mapping between VM and PM after consolidation

```
for i=1 to n do
    if FLAG_i==0 then
        /* Choose a Donor server with least score */
        Lscore ← MAX
        for j=1 to n do
            if S_i ≤ Lscore then
                Lscore ← S_i
                Donor ← PM_i
            end if
        end for
        /* Make a set of VMs from Donor server */
        VMset ← φ
        for j=1 to m do
            if VM_j ∈ Donor then
                VMset ← VM_j
                p_j ← 1
            end if
        end for
        while VMset && FLAG_i is OFF do
            /* Pick the knapsack PM */
            Hscore ← Min
            for j=1 to n do
                if S_i ≥ Hscore then
                    Hscore ← S_i
                    Knapsack ← PM_i
                end if
            end for
            /* Add VMs from knapsack PM to VMset */
            for j=1 to m do
                if VM_j ∈ Knapsack then
                    VMset ← VM_j
                    p_j ← 2
                end if
            end for
            for VM_j ∈ VMset do
                Calculate sf = p_j / U_j^max
            end for
            /* Sort VMs in VMset based on sorting factor sf */
            Sort.decreasing(VMset,sf)
            Apply Greedy KP on Knapsack-VMset
            Update S_knapsack /* Update Score of Knapsack VM */
```

Update $FLAG_{knapsack} \leftarrow ON$

$/* if\ no\ more\ VM\ from\ VMset\ can\ be\ placed\ on\ knapsack */$

end while

end if

end for

Algorithm 2 GreedyKP

Input: Vmset is sorted set of VMs in decreasing order of sf, k is knapsack PM, cap_k^r is resource capacity of knapsack PM for resource r, p_j profit for VMs, U_j^r is utilization of VM j for resource r

Output: VM-PM mapping for single knapsack PM, z_k is total profit of knapsack, $\hat{c}_k^r$ is remaining resource capacity of knapsack

$z_k \leftarrow 0$

$\hat{c}_k^r \leftarrow cap_k^r$

for j=1 to m **do**

$/* for\ each\ VM\ in\ VMset */$

$x_j \leftarrow 0$

if $U_j^{cpu} < \hat{c}_k^{cpu}$ && $U_j^{mem} < \hat{c}_k^{mem}$ && $U_j^{net} < \hat{c}_k^{net}$ **then**

$x_j \leftarrow 1$

$\hat{c}_k^{cpu} \leftarrow \hat{c}_k^{cpu} - U_j^{cpu}$

$\hat{c}_k^{mem} \leftarrow \hat{c}_k^{mem} - U_j^{mem}$

$\hat{c}_k^{net} \leftarrow \hat{c}_k^{net} - U_j^{net}$

$z_k \leftarrow z_k + p_j$

end if

end for

Chapter 5

Performance Evaluation

This chapter presents the performance evaluation of proposed algorithm CONKP for VM consolidation in cloud data center. The proposed work is evaluated through simulation based implementation and the results obtained from implementation are compared to other VMC algorithms from the existing literature. This section presents details of simulation environment followed by the comparison of results from implementation of algorithms to be compared, and the observations obtained from these results. We have taken both migration cost-unaware and migration cost-aware approaches for comparison with our work.

5.1 Simulation Setup

5.1.1 Data Center Setup

We simulated a cloud data center with n number of homogeneous physical machines each having multi dimensional resource capacities. We have taken three dimensions of resources along CPU, memory and network I/O. The total resource capacity of CPU is set as 4.0 GHz, memory as 10 GB and network bandwidth is 1 Gbps for each machine in simulated data center. We have varied the number of machines and VMs within data center to test the results better with scalable environment. The maximum BW capacity allowed for VM migrations between PMs is assumed to be equal to the total BW capacity of each PM and is 1 Gbps.

We consider that m number of VMs are placed on these machines and are randomly distributed among them in a way that overall load across the DC is balanced. As in real Cloud data centers, heterogeneous workloads are handled by VMs which require different amounts of resources depending upon types of applications which can be CPU-intensive, memory-intensive or network I/O intensive. Also, as with PMs, we consider that these applications (VMs) request for three types of resources, that are CPU, memory and network BW. We generated the resource demands of VMs across the three dimensions synthetically by generating random numbers using normal distribution. The available network bandwidth of PMs for migrating VMs are also generated synthetically at run time using normal distribution.

5.1.2 Simulation Environment

We implemented the algorithms using a simulation tool cloudsim with JAVA (JDk and JRE version 1.7.0) and used a workstation with configuration Intel Core i5-7200 3.1 GHz CPU with 2 cores, 1 TB storage and 4 GB RAM hosting Windows 10 as OS.

5.2 Benchmark Algorithms

We are comparing the following existing works from the literature with our proposed CONKP algorithm:

- **First Fit Decreasing (FFD) Algorithm**: VM consolidation is regarded as a bin packing problem by most of the works and many greedy heuristics have been used to solve it. FFD is one of the greedy heuristics that has been used as a VMC approach. It is a migration cost oblivious algorithm which is used as a baseline algorithm to compare with our work.

- **Ant Colony Optimization (ACO) Algorithm**: A modified version of ACO meta heuristic has been used for consolidation of VMs in a fully decentralized schema [21] and is a dynamic migration cost-aware algorithm.

- **V-MAN Algorithm**: A decentralized algorithm for VM consolidation using a gossip protocol [38].

5.3 Performance metrics

We have used following metrics to evaluate the quality of VM placement decisions taken by evaluated algorithms and their impact on overall performance of data center:

- **Number of Released Nodes**: Reduced number of active servers result in reduced energy consumption and costs. It can be calculated as number of active servers before consolidation minus number of active servers after consolidation.

$$AS_r = AS_b - AS_a \tag{5.1}$$

where
AS_r is the number of active physical servers released after consolidation.
AS_b is the total number of active physical servers before consolidation.
AS_a is the total number of active physical servers after consolidation.

- **Migration per VM**: Number of VM migrations is an important metric to evaluate the effect of consolidation algorithm on usual working of data center. It should be less to reduce the downtime as well as performance degradation faced by consumers due to consolidation process.

- **Packing Efficiency**: It is important to evaluate the success of VMC algorithms in consolidating the VMs on minimum number of active servers. It is calculated as the ratio of number of released nodes to the total number of nodes.

$$PE = \frac{AS_r}{n} \tag{5.2}$$

where
PE is Packing Efficiency of a VMC algorithm.
n is total number of nodes in the system.

- **Power Consumption**: It evaluates the energy efficiency provided by evaluated algorithms and is calculated by adding the power consumed by each individual server.

$$P_{DC} = \sum_{p \in PM} P_p \tag{5.3}$$

where
P_{DC} is the total power consumption of DC in K watts.
P_p is the power consumed by server p.
PM is the set of all PMs in DC.

- **Total VM Downtime**: It captures the degree of performance degradation caused by VMC algorithm. It is calculated as sum of downtimes of all the migrated VMs.

5.4 Results

We have evaluated the algorithms with varying number of PMs and VMs, taking number of PMs from 120 to 1008 and VMs from 720 to 6048 and obtaining the results for each metric. The results for *Number of released nodes* is shown in table 5.1 with graphical view shown in figure 5.1.

Table 5.1: Number of Released PMs

Number of PMs	Number of VMs	FFD	ACO	V-MAN	CONKP
120	720	29	36	39	42
240	1440	58	77	79	83
504	3024	124	161	122	163
1008	6048	246	322	323	325

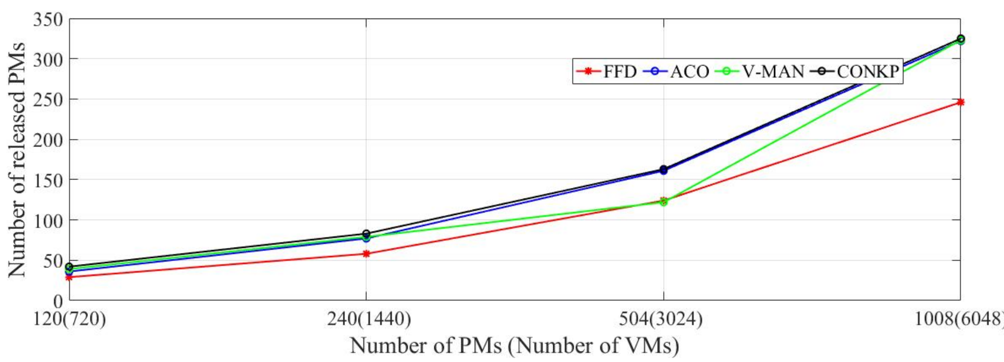

Figure 5.1: Plot of Number of Released PMs vs Number of PMs

As can be seen from Table 5.1, the performance of FFD is worst of all as it released the minimum number of nodes, that is, 246 while ACO and V-MAN closely followed our algorithm CONKP which achives the best result with 325 nodes.

Table 5.2: Number of migrations per VM

Number of PMs	Number of VMs	FFD	ACO	V-MAN	CONKP
120	720	26	5	3	1
240	1440	26	7	5	1
504	3024	27	8	4	1
1008	6048	26	9	4	1

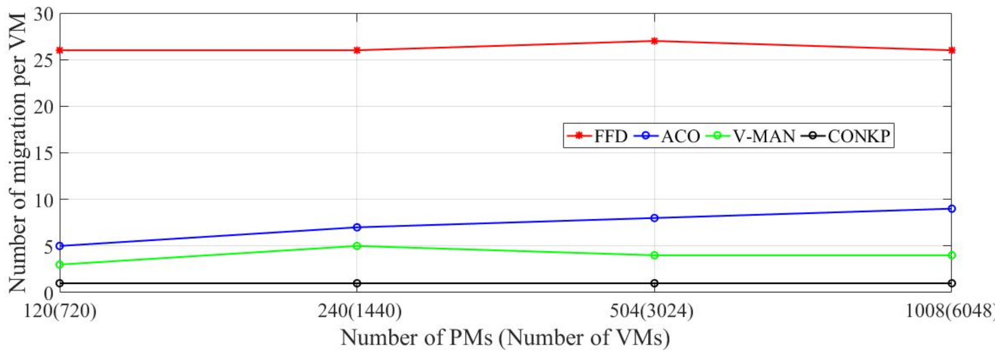

Figure 5.2: Plot of Number of migrations per VM vs Number of VMs

As can be observed from Table 5.2 and figure 5.2, the number of migrations are very less with ACO, V-MAN and CONKP, as compared to FFD. The reason for this result with FFD is that it is not aware of current VM-to-PM assignment and assumes that it has a set of VMs and a set of PMs and it has to move VMs to PMs in each iteration. CONKP gives best results in this case as it is designed to give more weightage to VMs that are already placed on receiver PMs in order to result in less number of migrations. The results of

Table 5.3: Packing Efficiency (%)

Number of PMs	**Number of VMs**	**FFD**	**ACO**	**V-MAN**	**CONKP**
120	720	24.1	30	32.5	32.7
240	1440	24.1	32	32.9	32.8
504	3024	24.6	31.9	24.2	32
1008	6048	24.4	31.9	32	33

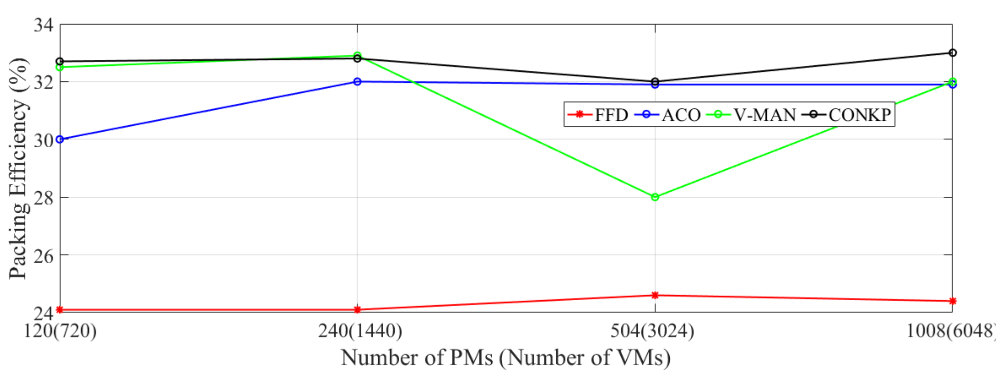

Figure 5.3: Plot of Packing Efficiencies vs Number of PMs

packing efficiency are summarized in Table 5.3 and a graphical view is shown in figure 5.3. In contrast to the results of ACO and V-MAN, the packing efficiency of our algorithm

CONKP remains almost same with increasing number of PMs, which means our algorithm has good scalability. Also, it gives the best efficiency as compared to others.

Table 5.4: Total VM Downtime (in hours)

Number of PMs	Number of VMs	FFD	ACO	V-MAN	CONKP
120	720	3	1	1	0
240	1440	9	3	2	0
504	3024	18	5	3	1
1008	6048	26	9	5	2

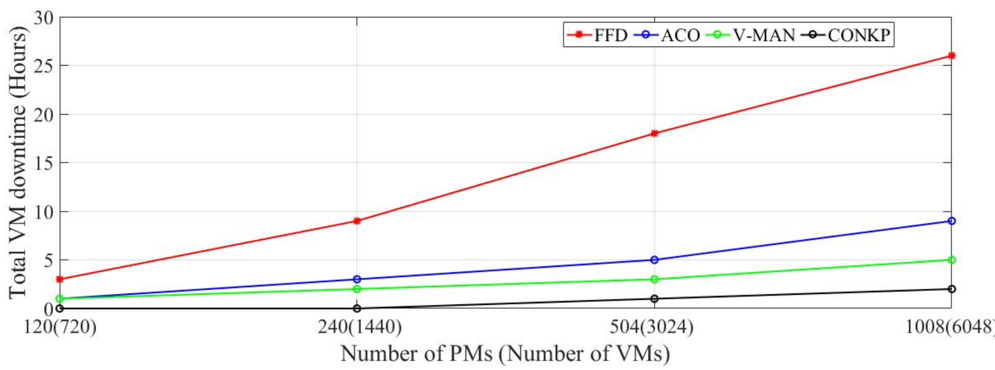

Figure 5.4: Plot of Total VM Downtime vs Number of VMs

Since total VM downtime is sum of downtime of all VMs, it is obvious that CONKP gives best results in Table 5.4 as it reduces the number of VM migrations. ACO and V-MAN requires 77% and 57% less total downtime respectively as compared to FFD. This is because FFD is migration cost unaware and does not take into consideration the memory size of VMs while migrating. CONKP and FFD give similar results for power consumption

Table 5.5: Total Power Consumption (in K Watts)

Number of PMs	Number of VMs	FFD	ACO	V-MAN	CONKP
120	720	20	25	21	21
240	1440	35	40	37	35
504	3024	77	83	80	75
1008	6048	120	140	131	118

in Table 5.5 and figure 5.5, both save more energy as compared to ACO and V-MAN. As number of PMs increase, power consumption with ACO and V-MAN increases gradually. With increased number of servers CONKP manages to give a lead to the two migration cost-aware algorithms while gives same results as migration cost oblivious FFD.

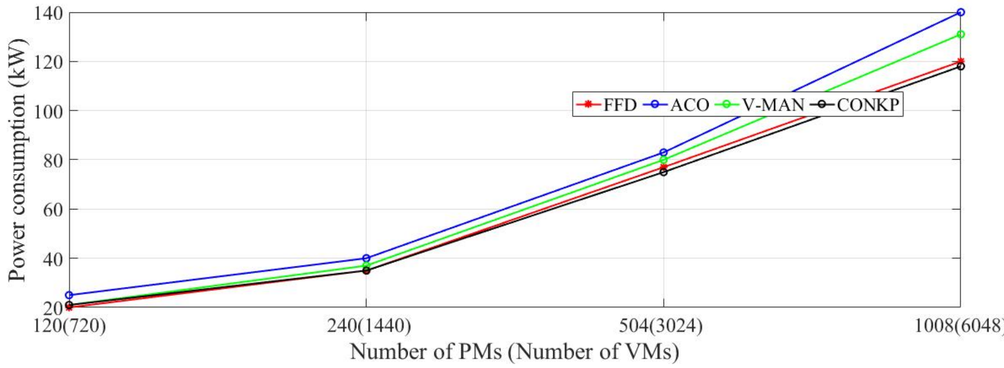

Figure 5.5: Plot of Total Power Consumption vs Number of PMs

5.5 Summary

- CONKP gives slightly better results than ACO and V-MAN in number of released PMs while FFD performs worst.
- FFD achieved worst number of migrations as it does not consider the given VM-to-PM mapping while making placement decisions. CONKP results in minimal number of migrations and minimam downtime as it gives weightage to current VM-to-PM mapping.
- CONKP is scalable as it provides almost uniform packing efficiency with increased number of servers.
- CONKP is better than other two migration cost aware algorithms in case of power consumption and provides similar results as that of FFD.

Chapter 6

Conclusion

6.1 Concluding Remarks

With emergence of cloud computing, the problem of optimized resource utilization has become tractable as compared to traditional data centers. It is due to the virtualization technologies that separates the run time environment of applications from the underlying physical resources, making them more scalable and easy to manage. However, in large scale data centers, the applications in form of jobs come and go dynamically leaving most of the resources underutilized which can not be controlled by a good VM placement strategy only. Moreover, this underutilization of resources leads to significant wastage of energy as well as emission of carbon dioxide making cloud computing non eco-friendly. So, there is a need to control the energy consumption in these data centers while providing services with negotiated levels of performance. VM consolidation is one way to reduce resource underutilization by dynamically rearranging placement of VMs on the physical servers and turning off the idle ones to save energy consumed by idle or underutilized servers.

This thesis presents a general introduction to cloud computing, its key features, virtualization and VM migration. The multi-objective VM consolidation problem is addressed in our work to increase the resource utilization of data centers and reduce the energy consumption and promote green cloud computing. This is done with minimum performance degradation and meeting SLA by reducing the number of migrations in dynamic placement of these VMs. We have compared the problem of dynamic VM consolidation to the problem of multiple Knapsaks (MKP), where physical machines represent knapsacks with resource capacities as knapsack capacities and virtual machines are taken as items with resource demands as item weights and a profit value associated with each VM. The goal is to fit VMs into PMs with total weights less than total capacities while maximizing the profit of each knapsack. We have considered the three-dimensional capacities and demands for resources, CPU, network bandwidth and memory and formulated the mathematical model for the problem. As, it is a NP hard problem, we have proposed a migration overhead aware VMC algorithm CONKP to solve it, which decides a donor and a receiver server based on server score values and take the receiver server as a knapsack. This way the problem size reduces for each iteration where the problem is to fill a single knapsack with a small number of VMs. Profit values are assigned to VM in a way that gives more weightage to already placed PMs over new PMs to reduce the number of migrations wherever possible. Finally, CONKP is evaluated using a simulation based environment with number of PMs and VMs varying from 120 and 720 respectively to 1008 and 6048 respectively. A number of performance metrics are defined namely, number of released nodes, number of migrations per VM, packing efficiency, total downtime and total power consumption. The results of our algorithm are compared to that of three other algorithms FFD, ACO and V-MAN and observations are noted for each parameter. To conclude, our algorithm CONKP outperforms all three overall.

6.2 Future Research Scope

With a novel VM consolidation method proposed, we look forward to come up with an integrated platform to test the results of algorithms in a better way which provides an easy way to implement and compare these scheduling algorithms. Secondly, we look forward to optimize our algorithm further to make it suitable for real-time scheduling by adding more parameters like dependency among jobs, division of jobs into multiple VMs, placing the jobs based on type of applications like HPC, social networking and scientific applications, and we will consider types of virtual machine sizes available in real clouds like EC2.

Bibliography

[1] M. Dabbagh, B. Hamdaoui, M. Guizani, and A. Rayes, "Energy-efficient resource allocation and provisioning framework for cloud data centers," *IEEE Transactions on Network and Service Management*, vol. 12, no. 3, pp. 377–391, 2015.

[2] M. Mishra, A. Das, P. Kulkarni, and A. Sahoo, "Dynamic resource management using virtual machine migrations," *IEEE Communications Magazine*, vol. 50, no. 9, pp. 34–40, 2012.

[3] Z. Xiao, W. Song, and Q. Chen, "Dynamic resource allocation using virtual machines for cloud computing environment," *IEEE Transactions on parallel and distributed systems*, vol. 24, no. 6, pp. 1107–1117, 2013.

[4] M. Dabbagh, B. Hamdaoui, M. Guizani, and A. Rayes, "An energy-efficient vm prediction and migration framework for overcommitted clouds."

[5] D. Li, W. Wang, Q. Li, and J. Cheng, "A comprehensive evaluation of scheduling methods of virtual machine migration for energy conservation," *IEEE Systems Journal*, 2015.

[6] A. Verma, P. Ahuja, and A. Neogi, "pmapper: power and migration cost aware application placement in virtualized systems," in *Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware*. Springer-Verlag New York, Inc., 2008, pp. 243–264.

[7] A. Murtazaev and S. Oh, "Sercon: Server consolidation algorithm using live migration of virtual machines for green computing," *IETE Technical Review*, vol. 28, no. 3, pp. 212–231, 2011.

[8] Q. Wu, F. Ishikawa, Q. Zhu, and Y. Xia, "Energy and migration cost-aware dynamic virtual machine consolidation in heterogeneous cloud datacenters," *IEEE Transactions on Services Computing*, 2016.

[9] C. Ghribi, M. Hadji, and D. Zeghlache, “Energy efficient vm scheduling for cloud data centers: Exact allocation and migration algorithms,” in *Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on*. IEEE, 2013, pp. 671–678.

[10] V. R. Reguri, S. Kogatam, and M. Moh, “Energy efficient traffic-aware virtual machine migration in green cloud data centers,” in *Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS), 2016 IEEE 2nd International Conference on*. IEEE, 2016, pp. 268–273.

[11] A. Beloglazov, J. Abawajy, and R. Buyya, “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.

[12] L. Yu, L. Chen, Z. Cai, H. Shen, Y. Liang, and Y. Pan, “Stochastic load balancing for virtual resource management in datacenters,” *IEEE Transactions on Cloud Computing*, 2017.

[13] B. Wadhwa and A. Verma, “Energy and carbon efficient vm placement and migration technique for green cloud datacenters,” in *Contemporary Computing (IC3), 2014 Seventh International Conference on*. IEEE, 2014, pp. 189–193.

[14] F. Xu, F. Liu, L. Liu, H. Jin, B. Li, and B. Li, “iaware: Making live migration of virtual machines interference-aware in the cloud,” *IEEE Transactions on Computers*, vol. 63, no. 12, pp. 3012–3025, 2014.

[15] S. K. Garg, C. S. Yeo, A. Anandasivam, and R. Buyya, “Environment–conscious scheduling of hpc application on distributed cloud–oriented centers,” *ScienceDirect, May*, 2010.

[16] M. Shojafar, N. Cordeschi, D. Amendola, and E. Baccarelli, “Energy-saving adaptive computing and traffic engineering for real-time-service data centers,” in *Communication Workshop (ICCW), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1800–1806.

[17] Z. Gong, X. Gu, and J. Wilkes, “Press: Predictive elastic resource scaling for cloud systems,” in *Network and Service Management (CNSM), 2010 International Conference on*. Ieee, 2010, pp. 9–16.

[18] M. Tighe and M. Bauer, "Integrating cloud application autoscaling with dynamic vm allocation," in *Network Operations and Management Symposium (NOMS), 2014 IEEE*. IEEE, 2014, pp. 1–9.

[19] B. Li, J. Li, J. Huai, T. Wo, Q. Li, and L. Zhong, "Enacloud: An energy-saving application live placement approach for cloud computing environments," in *Cloud Computing, 2009. CLOUD'09. IEEE International Conference on*. IEEE, 2009, pp. 17–24.

[20] C. Canali and R. Lancellotti, "Scalable and automatic virtual machines placement based on behavioral similarities," *Computing*, vol. 99, no. 6, pp. 575–595, 2017.

[21] E. Feller, C. Morin, and A. Esnault, "A case for fully decentralized dynamic vm consolidation in clouds," in *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on*. IEEE, 2012, pp. 26–33.

[22] A. Marotta and S. Avallone, "A simulated annealing based approach for power efficient virtual machines consolidation," in *Cloud Computing (CLOUD), 2015 IEEE 8th International Conference on*. IEEE, 2015, pp. 445–452.

[23] E. N. Elnozahy, M. Kistler, and R. Rajamony, "Energy-efficient server clusters," in *PACS*, vol. 2325. Springer, 2002, pp. 179–196.

[24] E. Feller, L. Rilling, and C. Morin, "Energy-aware ant colony based workload placement in clouds," in *Proceedings of the 2011 IEEE/ACM 12th International Conference on Grid Computing*. IEEE Computer Society, 2011, pp. 26–33.

[25] A. Khosravi, S. K. Garg, and R. Buyya, "Energy and carbon-efficient placement of virtual machines in distributed cloud data centers," in *European Conference on Parallel Processing*. Springer, 2013, pp. 317–328.

[26] S. Wang, A. Zhou, C.-H. Hsu, X. Xiao, and F. Yang, "Provision of data-intensive services through energy-and qos-aware virtual machine placement in national cloud data centers," *IEEE Transactions on Emerging Topics in Computing*, vol. 4, no. 2, pp. 290–300, 2016.

[27] J. Xu and J. A. Fortes, "Multi-objective virtual machine placement in virtualized data center environments," in *Proceedings of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing*. IEEE Computer Society, 2010, pp. 179–188.

[28] T. C. Ferreto, M. A. Netto, R. N. Calheiros, and C. A. De Rose, "Server consolidation with migration control for virtualized data centers," *Future Generation Computer Systems*, vol. 27, no. 8, pp. 1027–1034, 2011.

[29] F. Tao, C. Li, T. W. Liao, and Y. Laili, "Bgm-bla: a new algorithm for dynamic migration of virtual machines in cloud computing," *IEEE Transactions on Services Computing*, vol. 9, no. 6, pp. 910–925, 2016.

[30] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, and X. Zhu, "No power struggles: Coordinated multi-level power management for the data center," in *ACM SIGARCH Computer Architecture News*, vol. 36, no. 1. ACM, 2008, pp. 48–59.

[31] M. Chandini, R. Pushpalatha, and R. Boraia, "A brief study on prediction of load in cloud environment," *International Journal of Advanced Research in Computer and Communication Engineering.–2016.–5 (5).–pp*, pp. 157–162.

[32] S. Ismaeel and A. Miri, "Using elm techniques to predict data centre vm requests," in *Cyber Security and Cloud Computing (CSCloud), 2015 IEEE 2nd International Conference on*. IEEE, 2015, pp. 80–86.

[33] D. Saddar and P. Ghosh, "Future load prediction of virtual machines in cloud."

[34] Y. Liu, B. Gong, C. Xing, and Y. Jian, "A virtual machine migration strategy based on time series workload prediction using cloud model," *Mathematical Problems in Engineering*, vol. 2014, 2014.

[35] M. E. Lalami, M. Elkihel, D. El Baz, and V. Boyer, "A procedure-based heuristic for 0-1 multiple knapsack problems," *International Journal of Mathematics in Operational Research*, vol. 4, no. 3, pp. 214–224, 2012.

[36] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to linear regression analysis*. John Wiley & Sons, 2015.

[37] W. J. Dixon, F. J. Massey *et al.*, *Introduction to statistical analysis*. McGraw-Hill New York, 1969, vol. 344.

[38] M. Marzolla, O. Babaoglu, and F. Panzieri, "Server consolidation in clouds through gossiping," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011 IEEE International Symposium on a*. IEEE, 2011, pp. 1–6.