

Network Aware VM Migration using Community Recognition

By Abhishek Dash
710CS1035



Department of Computer Science & Engineering

National Institute of Technology

Rourkela

Network Aware VM Migration using Community Recognition

By Abhishek Dash
710CS1035

A Thesis presented for the partial fulfillment
of Dual Degree



Guide : DR. A.K. TURUK
Department of Computer Science & Engineering
National Institute of Technology
Rourkela

Dedicated to

All My Teachers, My Parents and My Friends



Computer Science and Engineering
National Institute of Technology, Rourkela

Rourkela-769 008, India www.nitrkl.ac.in

Dr. Ashok Kumar Turuk

Professor

May 20, 2015

Certificate

This is to certify that the work in the thesis entitled *Network Aware VM Migration using Community Recognition* by *Abhishek Dash*, bearing the roll number 710cs1035, is a record of novel research work carried out by him under my supervision and guidance in partial fulfillment of the requirements for the award of *Dual Degree Bachelor of Technology in Computer Science & Engineering* and *Master of Technology in Computer Science*. Neither this thesis nor any part of it has been submitted elsewhere for any degree or academic award.

Place:Rourkela

Date:May 20,2015

Ashok Kumar Turuk

Associate Professor

CSE Department

NIT, Rourkela

Network Aware VM Migration using Community Recognition

Abhishek Dash

Submitted for the partial fulfillment of Dual Degree Mtech

May,2015

Abstract

Cloud Computing is a powerful concept buzzing these days in industry by which we can avail resources as and when we require like electricity and where required softwares and information are provided based on demand. It enables us with large computing power with low-cost and hence removes the hassle of storing and maintaining servers locally. It can be basically divided into three models of business i.e. Software as a Service, Platform as a Service and Infrastructure as a Service which helps to transfer service to end user very efficiently.

VM placement belongs to the model of the Infrastructure as a Service. Basically it means that all applications have a certain need of computing power, memory storage, network bandwidth, and some power consumption to function which is abstracted as a Virtual Machine and provided by the Data Centers. Virtual Machine Migration is the method of transferring the VMs to the Physical Machines in such a way that there is efficient usage of energy, network bandwidth, etc. I have proposed a new network aware VM Migration scheme using Community Recognition which shows the candidates for migration and takes into account all other factors like energy, migration criteria, etc..

Declaration

No piece of this proposition has been submitted somewhere else for whatever other degree or capability and it is all my own work unless referenced actually in the content. The copyright of this proposition rests with the creator. No citations from it ought to be distributed without the writer's earlier composed assent; furthermore, data got from it ought to be recognized.

Abhishek Dash

Acknowledgements

I would like to thank my Guide Dr. Ashok Kumar Turuk for accepting me as my supervisor and introducing me to the topic of cloud computing .His constant presence and valuable inputs have been very helpful for me. I would like to thank all the faculty members for their constant support and cooperation during my stay here. I would also like to thank Mr. Sourav Addya for his valuable inputs and help during crisis.

I would also like to thank all my Dual Degree and B.Tech friends particularly Justine, Dev, Subash for patiently listening to me , for their support and for making my stay at NIT a unforgettable experience. Lastly I would like to thank my parents for providing me with unfathomable love and support.

Abhishek Dash

710cs1035

Contents

Abstract	v
Declaration	vi
Acknowledgements	vii
1 Introduction	1
1.1 Service Model	2
1.2 Deployment Model	4
1.3 Motivation	5
1.4 Objective	6
1.5 Outline	6
2 Bin Packing	7
2.1 Basic Algorithms	7
2.2 Partition Based Algorithms	9
2.3 Comparison	11
2.4 Genetic Algorithm	14
3 Virtual Machine Placement & Migration	17
3.1 VM Placement	17
3.2 VM Migration	19
4 Proposed Scheme	22
5 Implementation	25

6 Conclusion and Future Work	30
6.1 Conclusion	30
6.2 Future Work	30
Bibliography	31

List of Figures

1.1	Various Services in Cloud Computing(courtesy wikipedia)	2
1.2	Types of Services(courtesy of Wikipedia)	3
1.3	Deployment Models,courtesy of Wikipedia	5
2.1	Comparison of performance of partition algorithms	13
2.2	Representation of Chromosome	14
2.3	Comparison between a partition based & HGGA	16
4.1	Communities and Min-Cuts	23
4.2	A way of storing Distances in each Physical Machine	24
5.1	graph of physical machines	25
5.2	graph of virtual machines showing current network status	26

List of Tables

2.1	An example using Next Fit	8
2.2	An example using First Fit	8
2.3	An example using Best Fit	9
2.4	Total Bins Formed For Different Algorithms	11
5.1	VM Details	26
5.2	new placement	26
5.3	Comparison	28
5.4	Another Comparison	28

Chapter 1

Introduction

Cloud is considered as a virtualized group of computing means. It can help in coping with a diversity of dissimilar jobs, together with the group of back-end operations & user-oriented interactive applications, Rapidly deploy and increase workload by providing physical machines or virtual machines depending on the demand, support for redundancy, self-healing and highly scalable programming model, so that workload can be recovered from a variety of unavoidable hardware or software failure, real-time monitoring of resources usage, consolidate the allocation of resources when needed to minimize cost, etc.

Hosted applications over the web have changed greatly. The web which originally was just composed of static web pages, now serves as stage for many web applications that ranges from simple note taking tools to computation intensive scientific simulation services. One thing that makes such an approach special that users can outsource data and computation to a remote server that has enough resources to perform the task within much less time than traditionally running an equivalent application on the users machine. This is also one of the major factors that are driving the research in the cloud computing technologies.

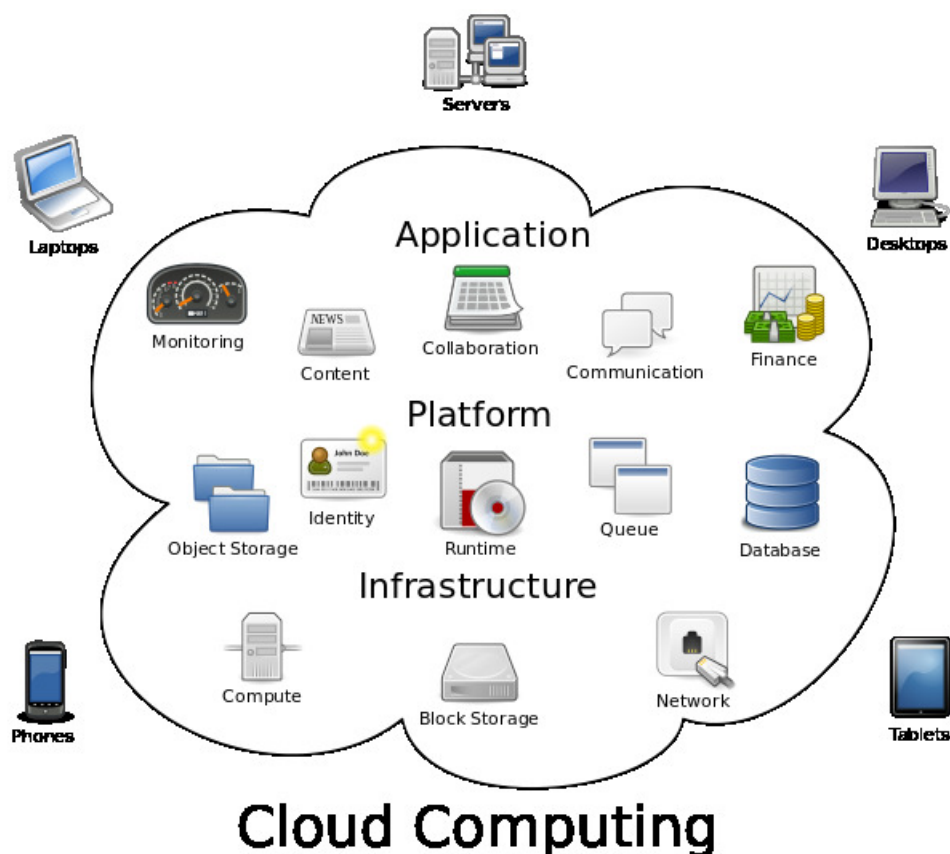


Figure 1.1: Various Services in Cloud Computing(courtesy wikipedia)

1.1 Service Model

There are basically three types of Service Models.

- **Software-as-a-Service (SaaS):** Software as a service is a type of business model by which softwares are set up through internet by the help of browsers or which is set up behind firewall in the network. One of the major advantage of this model is that you only pay for what you use. All of its applications versions are automatically updated and needs little or no changes locally. Some of the examples of Software as a Service are Flickr, Salesforce.com (it first started the Customer Relationship Management Product), Facebook, Google Apps like Drive, Gmail, etc.
- **Platform-as-a-Service (PaaS):** Platform as a Service, this kind of cloud com-

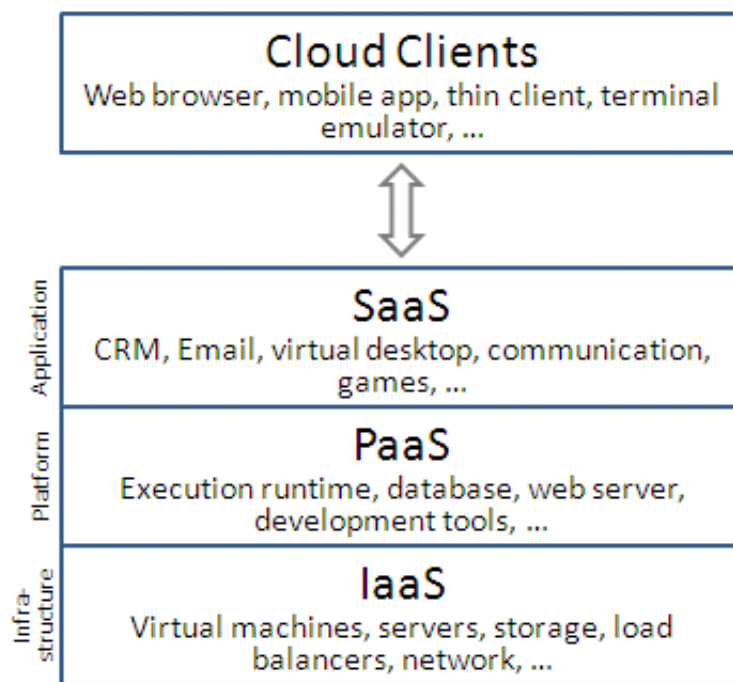


Figure 1.2: Types of Services(courtesy of Wikipedia)

puting business model provides a platform/environment for the development of applications as a service. Hence users doesnt have to install anything to run any program or applications. Only they have to provide on the basis of per-use. Some of the examples of Platform as a Service are Amazon Web Service (AWS), Google App Engine, Force.com, Microsofts Azure, etc.

- Infrastructure-as-a-Service (IaaS): Infrastructure as a Service is a cloud computing model by which computing resources are delivered as a service. You dont need to buy servers, softwares, networking gears, etc. for setting up business .Everything is outsourced i.e. only through internet connection one can have control over operating systems, storage and firewalls limited control but doesnt need to manage the infrastructure. It is a pay as you go model where according to the usage and allocation of resources one pays the vendor. Some of the examples of Infrastructure as a Service vendors are Google, VMWare, Yahoo, etc.

1.2 Deployment Model

- Public cloud: In public clouds, numerous customers have a stake in the computing resources delivered through a single service provider, Customers can quickly access these resources, and only pay for the operating resources. Although the public cloud has compelling advantages like Elasticity, Utility Pricing, Leverage Expertise ,etc. there are many concealed danger of security, regulatory compliance and quality of service(QoS).
- Private cloud: In the private cloud, computing resources and cloud infrastructure is created, used and controlled exclusively by a private enterprise. It is capital intensive as well as having a physical footprint. The main advantages of this model are that the total control, regulation and flexibility and security (in the hands of the enterprise).
- Hybrid cloud: A third type can be hybrid cloud that is characteristic composition of public and private cloud. It assists the enterprise to run its state-steady assignments in the private cloud, and asking the public cloud for intensive computing resources when peak workload occurs, then return if no longer needed. For example, an organization may store sensitive client data in house on a private cloud application, but interconnect that application to a business intelligence application provided on a public cloud as a software service.

According to SM Hasemi the service oriented, loose coupling, strong fault tolerant, business model and ease of use are some of the main characteristics of cloud computing. Server consolidation using virtualization technology has become increasingly important for improving data center efficiency. It enables one physical server to host multiple independent virtual machines (VMs), and the transparent movement of workloads from one server to another.

With increase in popularity of Cloud Computing several security and privacy concerns have been aroused. According to the Cloud Security Alliance, the top three dangers in the cloud's security are "Insecure Interfaces and API's , Data Loss & Leakage, and Hardware Failure "which accounted for 29%, 25% and 10%

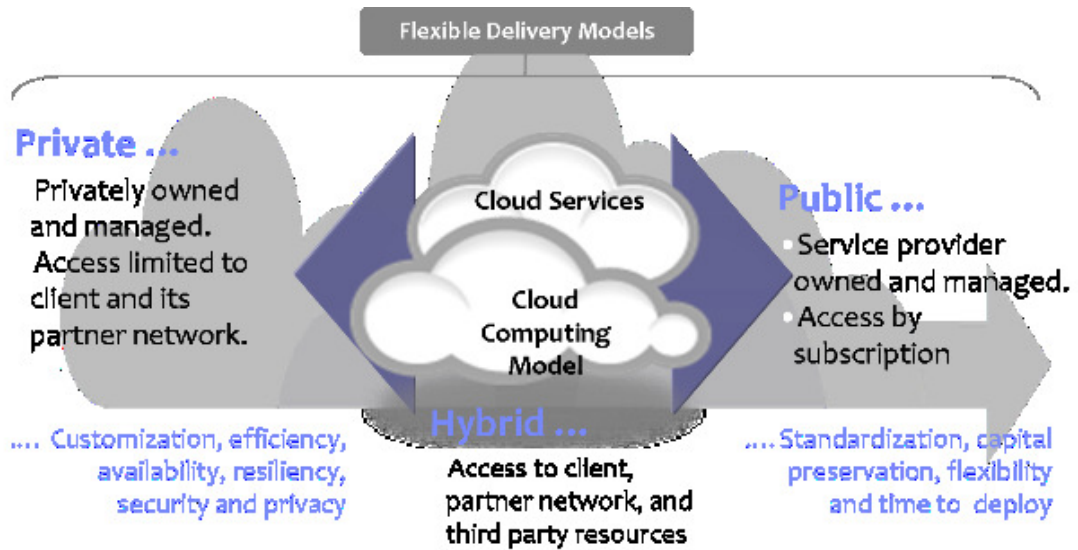


Figure 1.3: Deployment Models, courtesy of Wikipedia

of all cloud security outages respectively - together these make shared technology vulnerabilities [1]. In a platform of cloud supplier, if resources are pooled by different users there is a likelihood that information which might be of property of different users are residing on the same data server. Thus leakage of Information may creep in by slip-up when information for one consumer is given to another [1].

1.3 Motivation

In cloud computing model, virtualized computing, storage resources are being provided to customers as a service through the web with equitable price. Sometimes certain applications have high network usage and hence that require communication with other applications; hence there may be high network cost along with high energy usage due to a certain placement of Virtual Machines. There is a need for schemes that can reduce network cost as well as maintain energy usage by migration or placement of Virtual Machines.

1.4 Objective

The objective is to reduce the network Traffic using Virtual Machine Migration. Also at the same time we will consider the three factors i.e. CPU Usage, Memory Usage and Energy Usage during migration. Also we will try to maintain the cost of VM Migration so that it doesn 't exceed the value of reduction in network cost.

1.5 Outline

Outline of the rest of the thesis is now briefly described. Chapter 2 provides some details on Bin Packing Problem and some ways to solve it which is very similar to the problem of the placement of the Virtual machines. Chapter 3 describes about the Virtual Machine Placement and migration and various schemes used according to the ongoing research trend. In Chapter 4 my proposed approach for Virtual Machine Migration has been elaborately defined. Chapter 5 has some notes on the implementation details of my approach. Chapter 6 gives concluding remarks and future work.

Chapter 2

Bin Packing

Consider a series of items $\{1, 2, \dots, n\}$ with magnitudes $(a_1, a_2, \dots, a_n) \in (0, 1]$; where we try to find a division of the objects given into sets of a fixed magnitude (called bins) say 1, so that all bins are utilized to maximum and the number of sets is minimized. Moreover, the total sum of items stored in each bin does not exceed the size of the bin. This is the bin packing problem (in 1-dimension).

Bin packing is a combinatorial NP-hard problem. We can express Cloud Storage as a form of bin packing problem when a set of objects consisting of scalar sizes and dimensions are to be stored in the memory of Servers in the data centers in an efficient manner so that less space is used.

2.1 Basic Algorithms

Next Fit:

The easiest algorithm is Next Fit for traditional one dimensional bin packing. Next Fit is an online algorithm that is bounded space and in which the only partially filled bin that is open is the most recent one to be opened. It is initiated with one active bin inside into which the input is pushed; if the size of input is greater than remaining capacity of the current active bin then another bin is opened into which the current input is pushed and the previous bin is closed (never to be used). This is a least efficient algorithm but one benefit is there i.e. only a single bin is active at any moment of time.

Example: Suppose bin size is 1 and inputs are 0.3, 0.6, 0.2, 0.1, 0.5, 0.7, 0.2, 0.4, 0.1, 0.9

Answer:

Table 2.1: An example using Next Fit

Bins	Items
1	0.3, 0.6
2	0.2, 0.1, 0.5
3	0.7, 0.2
4	0.4, 0.1
5	0.9

First Fit:

In First Fit all non-empty bins are considered for the input. Hence an input is checked in all the bins to see whether it can fit in any one of the remaining bins, if it fits perfectly then it is stored in that bin, or else a new bin is opened. Since First Fit checks all the bins for the correct place for the bin, it has a worse running time compared to Next Fit, but it is efficient as compared to Next Fit.

Example: Suppose bin size is 1 and inputs are 0.3, 0.6, 0.2, 0.1, 0.5, 0.7, 0.2, 0.4, 0.1, 0.9

Answer:

Table 2.2: An example using First Fit

Bins	Items
1	0.3, 0.6, 0.1
2	0.2, 0.5, 0.2, 0.1
3	0.7
4	0.4
5	0.9

Best Fit:

Best Fit is the best recognized algorithm for on-line bin packing (in online bin packing we do not know all the values of the input beforehand) which becomes a front-runner when compared for simplicity and efficiency among various algorithms. Best Fit has a better worst case complexity and uniform case complexity when compared to other algorithms. Best Fit selects the bin that has the smallest possible space among all the bins which are active and places the input there. In this way all the bins are efficiently utilized for their space and also less number of bins are used.

Example: Suppose bin size is 1 and inputs are 0.3, 0.6, 0.2, 0.1, 0.5, 0.7, 0.2, 0.4, 0.1, 0.9 and 0.1

Answer:

Table 2.3: An example using Best Fit

Bins	Items
1	0.3, 0.6, 0.1
2	0.2, 0.5, 0.2, 0.1
3	0.7
4	0.4
5	0.9, 0.1

2.2 Partition Based Algorithms

There are some online bin packing algorithms that partitions the input according to some reasoning and then the partitioned input is put into a bin according to its set. Some of the partition based Algorithms are Harmonic Algorithm, K-binary Algorithm, Epstein Algorithm, etc.

Harmonic Algorithm(Lee & Lee) [2]

For a given value of n , k , and the objects $O = \{a_1, a_2, \dots, a_n\}$, where n indicates the number of items and k represents the number of partitions.

1. Suppose the input size is $(0, 1]$ which is then partitioned in the following

way: $(0, \frac{1}{k}], (\frac{1}{k}, \frac{1}{k-1}], \dots, (\frac{1}{2}, 1]$.

2. Each item a_i which is partitioned according to its value is put into a bin of that type (e.g. if the value is 0.6 it belongs to the $(\frac{1}{2}, 1]$ partition.).
3. In each partition the Next Fit is used i.e. if the current bin is not sufficient for the current input then a new bin is opened.
4. Compute the overall amount of bins that have been used.

Epstein Algorithm [2]

For a given value of n , k , and the objects $O = \{a_1, a_2, \dots, a_n\}$, where n indicates the number of items and k represents the number of partitions.

1. Suppose the input size is $(0, 1]$ which is then partitioned in the following way: $(0, \frac{1}{k}], (\frac{1}{k}, \frac{1}{k-1}], \dots, (\frac{1}{2}, 1]$.
2. Each item a_i which is partitioned according to its value is put into a bin of that type (e.g. if the value is 0.6 it belongs to the $(\frac{1}{2}, 1]$ partition.).
3. Here a constraint is put into the number of items that can be stored in a bin. If k items of k type have been put into the bin, then it is closed and a new bin is opened.
4. In each partition the Next Fit is used i.e. if the current bin is not sufficient for the current input then a new bin is opened.
5. Compute the overall amount of bins that have been used.

K-binary Algorithm [2]

For a given value of n , k , and the objects $O = \{a_1, a_2, \dots, a_n\}$, where n indicates the number of items.

1. Suppose the input size is $(0, 1]$ which is then partitioned in the following way:

$$(0, \frac{1}{2^{k-1}}], (\frac{1}{2^{k-1}}, \frac{1}{2^{k-2}}], \dots, (\frac{1}{2^{k-m}}, 1]$$

for $k = 1, 2, 3, 4, \dots, n-2, n-1$ where m denotes number of partitions.

2. Each item a_i which is partitioned according to its value is put into a bin of that type (e.g. if the value is 0.6 it belongs to the $(\frac{1}{2}, 1]$ partition.).
3. In each partition the Next Fit is used i.e. if the current bin is not sufficient for the current input then a new bin is opened.
4. Compute the overall amount of bins that have been used.

2.3 Comparison

I implemented the Harmonic Algorithm, K-binary Algorithm and Epstein Algorithm that have been described in detail in [2]. I compared the performance of all the above algorithms for various number of bins and created a table showing the comparisons.

Table 2.4: Total Bins Formed For Different Algorithms

NUMBER OF ITEMS	HARMONIC	EPSTEIN	K-BINARY
20	14	15	14
40	27	28	28
60	42	41	40
80	51	55	53
100	63	68	66
120	78	79	79
140	94	94	95
150	97	100	98
200	127	132	131

In the simulation the number of items were randomly generated; the number of

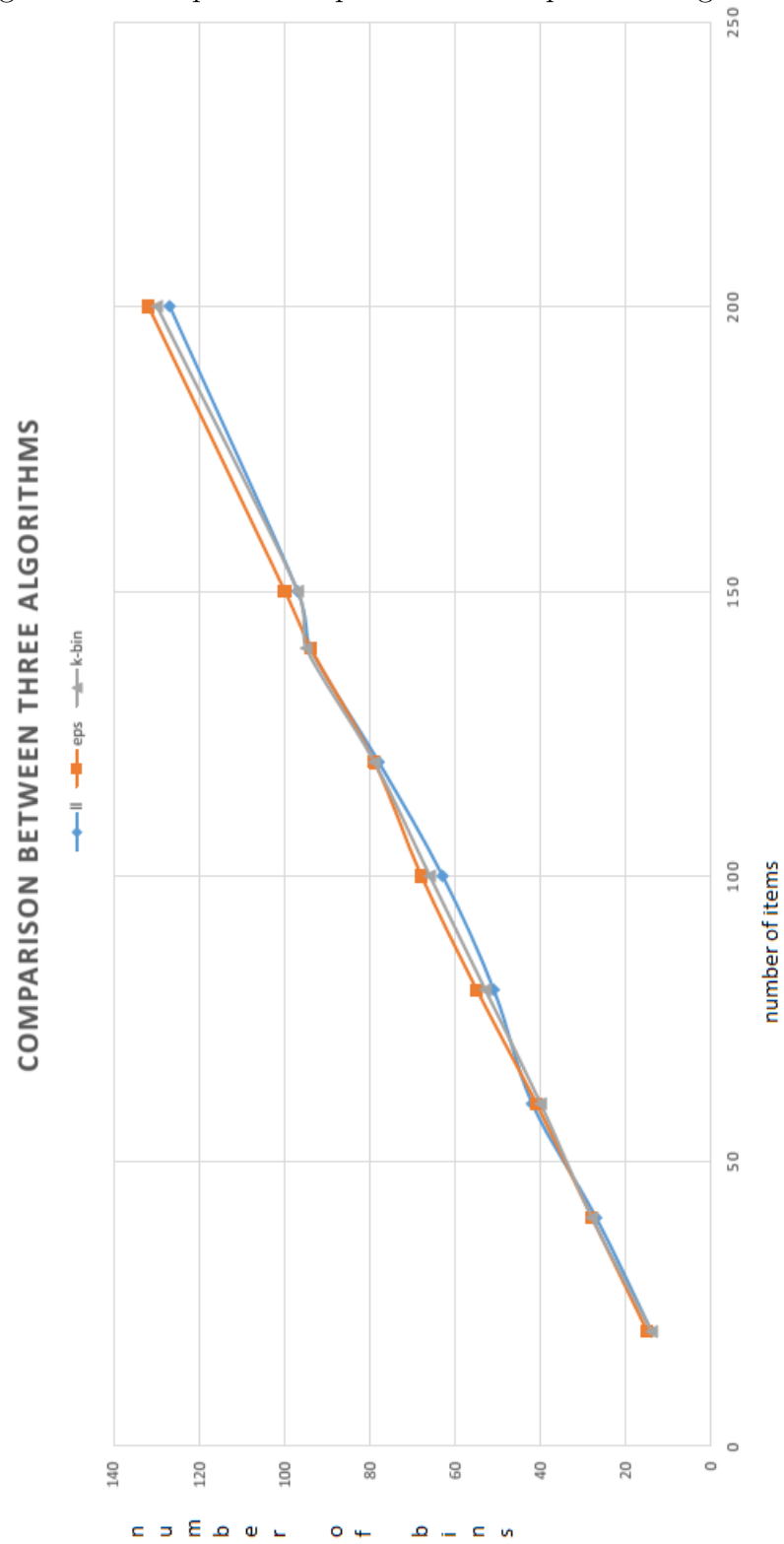
items, the number of partitions can be provided by the user. The remaining capacity available in the current active bin is also displayed.

The following are some of the snapshots of the simulations of the three algorithms. I observed that in almost all cases Harmonic Algorithm was showing the best performance in terms of less number of bins used. Both Epstein and K-binary showed that they required almost equal number of bins.

For smaller items, each have almost the same performance. For larger number of items, we note that k-Binary has a lower variance than Harmonic but slightly higher variance than Cardinality Constrained [2].

In the graph provided below all the algorithms have been compared where x-axis represents the number of items and the y-axis represents the number of bins that have been used for packing.

Figure 2.1: Comparison of performance of partition algorithms



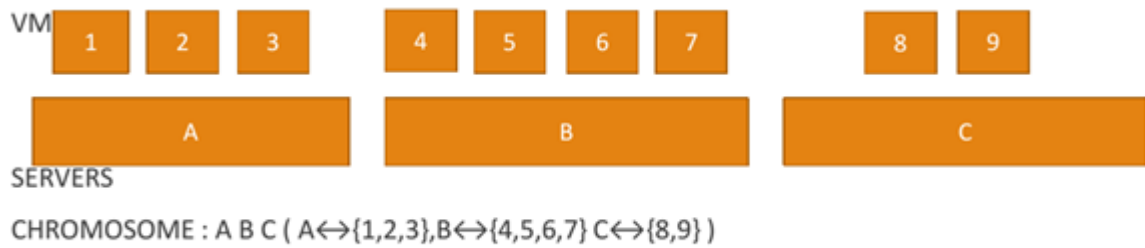


Figure 2.2: Representation of Chromosome

2.4 Genetic Algorithm

There are many operations that are used in Genetic Algorithms like mutation, crossover, etc. (Normal Genetic Operators). These are generally operated on genes. For Bin Packing Problem initially using Standard Steps of Genetic Algorithm every item was selected as a gene. This led to large disturbances.

Crossover Operation: First step in crossover operation to create off-springs is creation of selected parents clones. The selected parents are then used as base of offspring chromosomes. For a standard genetic crossover operation a chromosome like

1	2	3	4	5	6
A	C	T	T	T	T

The probability of disruption of the cluster of Ts is very large since the number of Ts is very high and there is a high chance that it will break during selection of a point for crossover.

Mutation Operation: In mutation randomly a chromosome is modified which results in a severe cost of fitness. Suppose the Chromosome is: AAAGGG and after a random change the chromosome becomes AACGGG. The mutated gene has a higher chance of being detached.

In [7] a Hybrid Grouping Genetic Algorithm has been discussed to solve Bin Packing.

Encoding: In HGGA we encode the chromosome in the following way:

Each bin is represented by a gene. A D B C E B : B E C D A
 A A A B B B : A B

Here each chromosome can be of different length.

Crossover in HGGA: A B C D E F FIRST PARENT First copies of par-
 a b c d SECOND PARENT ents are made of the two parents and two crossing sites are chosen at random. In each of them:

$A|BCD|EF$

$ab|cd|$

Then the bins that are crammed between the crossing sites in the second chromosomes are added into the first parent 's chromosome at the first crossing site yielding A c d B C D E F.

Now if some of the objects appear twice in the solution, then they have to be removed i.e. A c d B D (if c and d 's objects are contained in bins of C, E and F).

New Chromosome: A c d B D x : Here x denotes the objects that are missing from the solution. They are added by the first fit.

Mutation: Given a chromosome, we select at random a few bins and eliminate them. The objects those missing from solution can be inserted back in random order.

Inversion: The work of inversion operator is to change the positions of some genes on the chromosome so that the offspring is assisted. It only affects the representation of genetic contents of a chromosome.

e.g. A D B C E B : B E C D A can be inverted to

A D B C E B : C E B D A (Hence if B and D are well filled bins , hence promising bins, they have a higher probability of being both present in next crossover.)

Fitness Function: [7]

$$F = \frac{\sum_{i=1}^n (\frac{f_i}{c})^k}{n}$$

Here F denotes fitness of the solution

n represents no. of bins.

f fill of the ith bin.

c- capacity of the bin.

k- constant greater than 1.

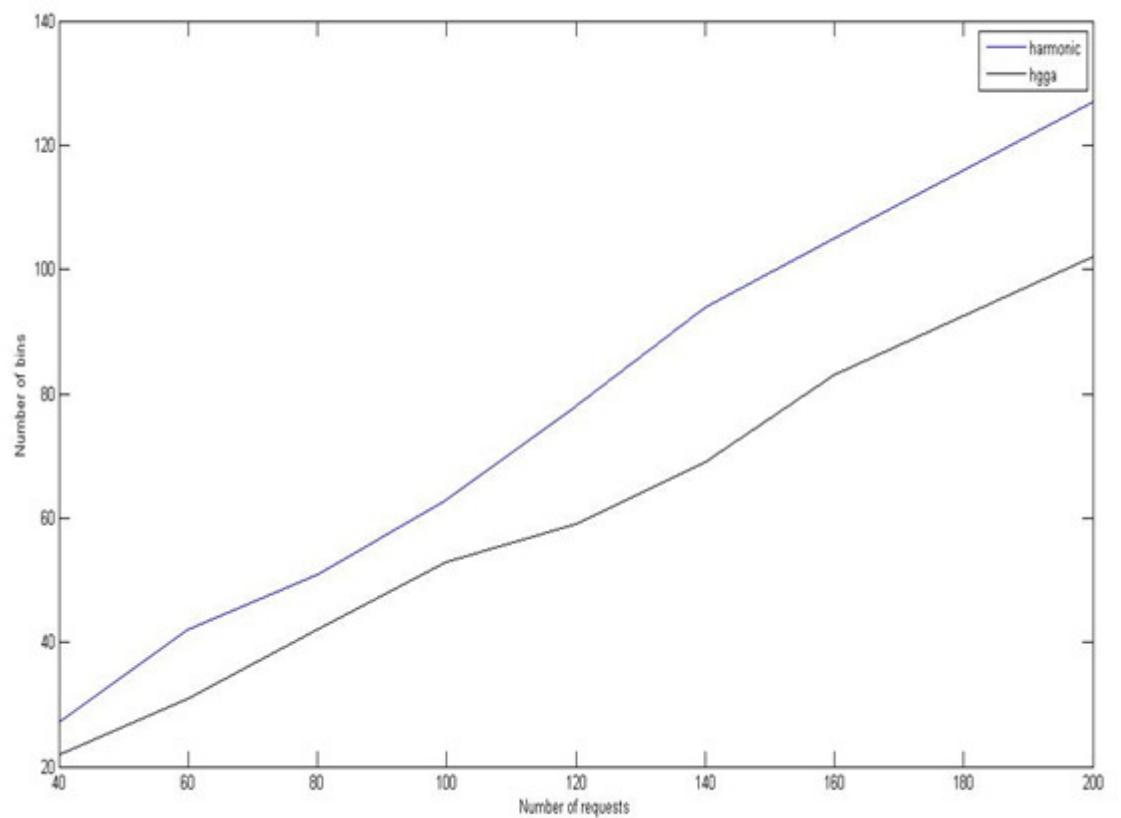


Figure 2.3: Comparison between a partition based & HGGA

One can clearly see that HGGA clearly improves upon the result of Harmonic Algorithm for 200 generations from the above graph where we have compared the performance of HGGA algorithm and partition based algorithm though HGGA has a worse running time as compared to the partition based algorithm.

Chapter 3

Virtual Machine Placement & Migration

3.1 VM Placement

Virtual Machine placement is a part of the Infrastructure as a Service Model. Basically it means that all applications have a certain need of computing power, memory storage, network bandwidth, some power consumption to function. In cloud computing these needs are provided by servers residing in the Data Centers. Virtual Machine placement means these requests/needs (these requests are all convert to virtual machine (VM) requests.) of the applications are placed in the servers in such a way that there is efficiency and less wastage i.e. less number of servers are used, less power consumption, less memory usage and efficient network bandwidth utilization. The VM placement problem can be formulated as a multidimensional bin packing problem or optimization problem.

Following are some of the papers read for Literary Survey.

In [7] following opinions were made by the Authors. The consolidation of applications and its relation with energy Efficiency is elucidated. Also the relation between operating points of the servers and usage of the resources is explained. Applications consolidation is converted to bin packing problem (multi-dimensional). Using experimental setup and plotting graphs between utilization of c.p.u power and disk

utilization during energy consumption and degradation in output, the operating points are identified at which the servers can work with maximum throughput. In the algorithm used for consolidation, first the size of bin is set according to the optimal operating point and each application is treated as a vector. Then based on Euclidean distances applications are allocated their corresponding bins/servers.

In [4] following opinions were made by the Authors. A VM Placement scheme which is network aware was discussed. Since application that are data intensive interact frequently with data hence how the performance of network input and output affects the applications is discussed. Here all applications are treated as Virtual Machines. Basically three matrices i.e. Data Distribute Matrix, Speed Matrix (speed between data storage nodes and physical machines) and Resource Requirements Matrix (each row represents the VMs resource need) are collected and multiplied and from the final matrix decision is made to where the Virtual Machine is to be transferred.

In [8] following comments were made by the author. Due to substantial differences in the patterns of traffic in the data centers and the structural differences of data center architectures that have been proposed recently, a relative analysis on the impact of different network architectures and traffic patterns is conducted. The VM placement is expressed as an optimization problem and a two-tier approximate algorithm is planned that solves the Virtual Machine placement problem for large problem sizes.

In [3] following comments were made by the author. The VM placement is treated as multi-objective optimization problem and using genetic algorithm and using fuzzy multi-objective evaluation efficient solutions are given. Also the algorithm is compared with other algorithms which treat them as bin packing problem. To cope the mappings of workloads to VMs and VMs to physical resources a two-level control system is given.

In [9] following comments were made by the author. In Cloud Computing people can allocate resources dynamically i.e. amount of resources needed depending upon

usage can be allocated and only that much will be charged by the provider. Using stochastic integer programming a virtual machine placement algorithm is suggested which minimizes the expenditure of cost for hosting virtual machines in a multiple cloud provider environment under future demand and price uncertainty.

3.2 VM Migration

VM Migration is the further advancement of Virtual Machine Placement whereby the VMs are/can be transferred to different physical machine depending upon the type of migration and other factors like network latency, energy usage, server consolidation, etc. Also it might help to rebalance the workload across all physical machines. There are some challenges of VM Migration like checking the total migration time, avoiding the interference of services that are currently running, minimizing the downtime, etc. There are 2 types of VM Memory Migration depending on the way memory state of a VM is transferred from source to destination.

1. Pre-copy memory migration

Warm-up phase:

Even while the Virtual Machine is running, Hypervisor copies from source server to destination server all the memory pages. If there are some changes in memory pages (dirty pages) while copying, then they are recopied till page dirtying rate is smaller than rate of re-copied pages. [11]

Stop-and-copy phase:

After the warm-up phase, the VM is stopped on the source server, the residual dirty pages are copied to the destination server, and the Virtual Machine is recommenced on the destination server [12]. The difference of time between restarting the VM at destination server and stopping the VM at the source is called "down-time", and varies from a few milliseconds to seconds depending upon the size of memory and applications running on the Virtual Machines.

2. Post-copy memory migration

First the Virtual machine is suspended at the source during Post-copy VM

migration [13]. A minimal subset of the VMs execution state i.e. its registers, CPU state, etc. is transferred to the destination. The Virtual Machine is then restarted at the destination. Simultaneously, pre-paging is done by the source i.e. the remaining memory pages are transferred to the destination. At the destination a network fault happens if the virtual machine attempts to access a page which hasnt been sent yet resulting in page fault. They are trapped and resent to the source for response. Large number of network faults can reduce performance of the VM. Therefore, by sending pages closer to last fault the pre-paging can dynamically adapt the page transmission order to network faults. In Post-copy Migration each page is sent only once over the network. But in pre-copy the same page can be transferred many times if its dirtied repeatedly at the original host during migration. But pre-copy has an advantage. It maintains a current state of the Virtual Machine at the host during migration. But with post-copy, the Virtual Machine's state is distributed over original host and its destination host. Pre-copy can easily recover if the destination fails during migration whereas the post-copy cannot.

Seamless live migration:

In seamless live migration the down time of a Virtual Machine is unnoticeable by the end users.

Following are some of the papers read for Literary Survey.

In [10] following comments have been made by the Authors. Due to virtualization of Servers efficient and dynamic management of workload in Data Centers is possible. But Virtual Machine Migration can lead to significant cost on hosts and network and can cause SLA violations. The authors introduced scheduling models which are network topology aware for live migrations of Virtual Machines. Also some methods were introduced to classify Virtual Machines into various categories depending upon their workload.

In [5] following comments have been made by the Authors. Due to Infrastructure as a Service model of cloud computing resources are delivered to customers via web as a service with dynamic pricing. Since users dont have to worry about the management of physical servers and other hardwares they can focus their concentration on

business. Due to rise in popularity of Cloud computing the demand for virtualized resources is increasing which has led to increase in energy consumption. To save and conserve energy from all servers which have low utilization virtual machines are transferred and these servers are allowed to shut down which can sometimes lead to increase in network latency for applications that are data intensive. An algorithm is given which uses the concept of clustering used in data mining to manage (migrate) Virtual Machines to reduce network cost and simultaneously maintain the energy cost.

Chapter 4

Proposed Scheme

NETWORK AWARE VM MIGRATION USING MIN-CUT

Our Objective is to reduce the Network Traffic using VM Migration but at the same time considering the 3 factors i.e. CPU Usage, Memory Usage, Energy Usage. Also we have maintained the cost of VM Migration at the same time. This algorithm is used inside a data center to choose where a particular Virtual Machine should migrate.

Following are the steps involved in my method:

- Initial VM Placement is assumed to be already done using any earlier method like Best Fit, First Fit, etc. Here it is assumed that each VM is a three dimensional vector (c.p.u, memory, network) where the third value is provided in the form of a matrix.
- Following are the available inputs: Distance Matrix between various physical machines is given. Speed Matrix between various physical machines is given. Frequency of communication between VMs is given.
- Cost of Communication between VM to different PMs can be calculated from the above inputs.
- Which VMs to choose and how to choose for the migration?
Undirected weighted Graph of VMs can be made on the basis of cost of communication.

- This graph can be divided into various communities (For an undirected graph the community is a strongly connected component).
- Say 'N' communities are formed after applying the algorithm of Ford Fulkerson algorithm to calculate the min-cut in the graph. A minimum cut of a graph is a division of the vertices of a graph such that it is divided into two disjoint subsets that are joined by at least one edge and its cut set has the fewest number of edges or smallest total of weights possible. Suppose the min-cut algorithm is applied k-times to create disjoint graphs.
- In a community, each VM is ranked in increasing order on the basis of $[a1 * CPU + a2 * MEM]$.

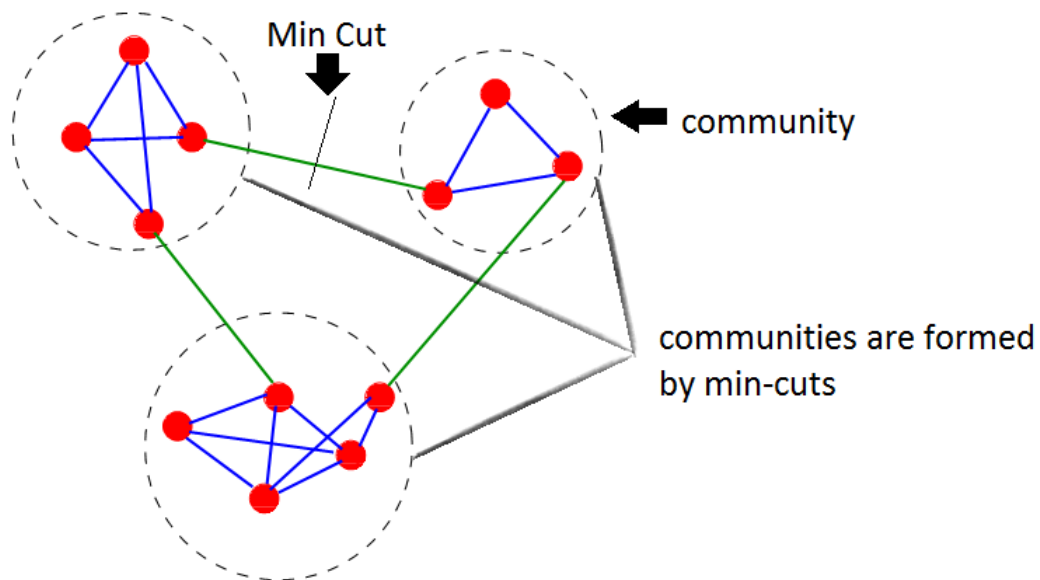


Figure 4.1: Communities and Min-Cuts

- Our motive is that for each VM present in a community, we try to pack them in the physical machines available/used in the community. We try to pack them in the physical machines which have the higher usage VMs that cannot be migrated.

- Also each server can use only $[1*(CPU), 2*(MEM)]$ so that servers are not overloaded and hence efficient energy usage is there.
- To maintain the VM Migration cost only those VMs are allowed to migrate if their usage follows the following criteria: $v1*(CPU) + v2*(MEM) \leq i$ (the threshold value).
- How to pack the VMs in a single community?

For each VM present in the community that are allowed to migrate, calculate the cost when we place the VM in all the available physical machines and find the best machine to place. If every Physical Machine has a hash-map of cost of communication of every other Physical machine, then we can easily know the cost of communicating two VMs after a VM is migrated to a new machine. Cost between every pair can be calculated using Floyd-Warshall algorithm (All pair shortest path) or Link State Routing.

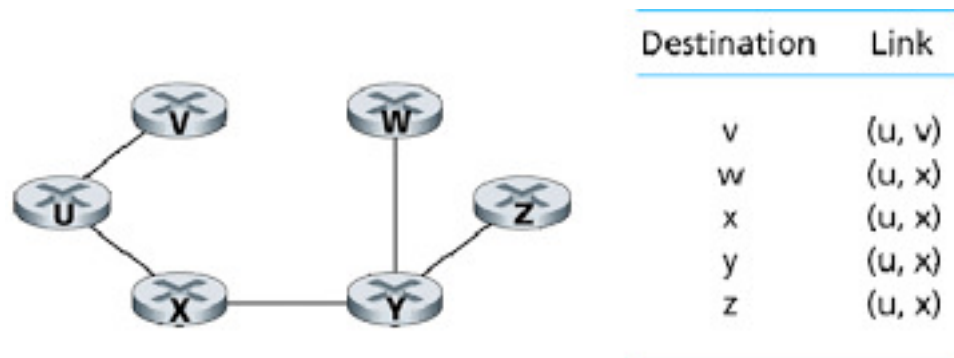


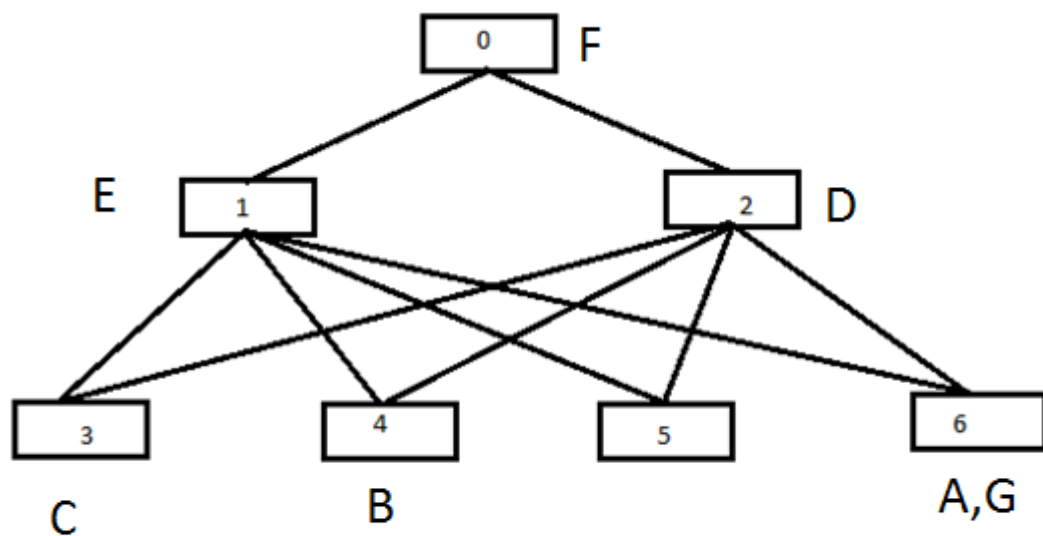
Figure 4.2: A way of storing Distances in each Physical Machine

- Hence we can get suitable candidates for VM Migration, suitable Physical Machines where VM can be migrated. Also the total network traffic is reduced when cluster of VMs which talk frequently with each other are placed in the same physical machine or nearby.

Chapter 5

Implementation

I tested my algorithm for the following simple structure of 7 PMs where the VMs are assumed to be distributed over all nodes. In the diagram given below numbers denote the Physical Machines and Alphabets denote the Virtual Machines.



Current Physical Machine Graph with VM Placement

Figure 5.1: graph of physical machines

I have assumed that there are 7 VMs which have been placed already according to the above diagram and are now considered for migration. Value of Constants for

the following particular example of 7 Physical Machines in a data center and for subsequent simulations.

Penalty of $VMMigration = 1.2 * VMMemorySize$.

Threshold Value=0.5.

Value of $a1 = 0.4$ and $a2 = 0.6$

Maximum C.P.U Utilization($\beta1$) = 0.9

Maximum Memory Utilization($\beta2$) = 0.8

Number of Min-Cuts(k) = 1

It is assumed that cost of network is calculated in Mbps.

Table 5.1: VM Details

VM	A	B	C	D	E	F	G
Capacity	0.5,0.6	0.2,0.4	0.6,0.5	0.6,0.2	0.7,0.7	0.3,0.7	0.4,0.2
Placement	6	4	3	2	1	0	6

Table 5.2: new placement

VM	A	B	C	D	E	F	G
Placement	6	2	3	2	1	0	6

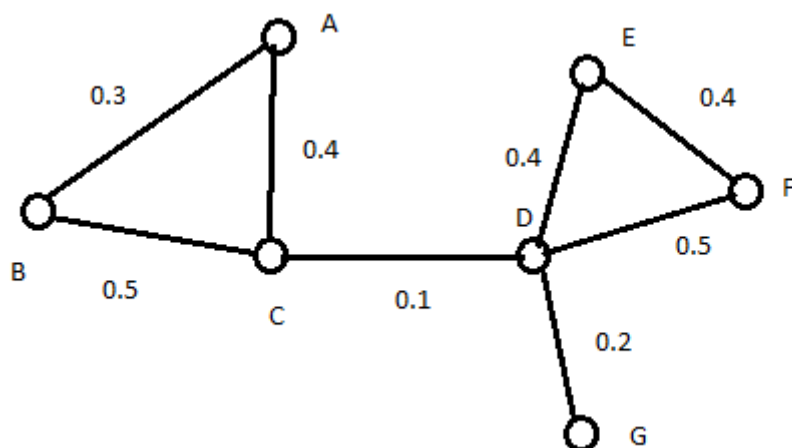


Figure 5.2: graph of virtual machines showing current network status

Number of *Migrations* = 1

Cost of *Migration* = 1.28

Number of Nodes *Released* = 2

In Existing Greedy VM Migration Policy all Physical Machines are sorted on the basis of their Remaining Capacity and then placed.

Number of Migrations=3

Cost of Migration=3.96

Number of Nodes Released=1

VM	A	B	C	D	E	F	G
Placement with existing algorithm	6	2	3	5	1	0	2

In the ideal case if all the VMs are allowed to migrate and every host has large capacity then in the case of the proposed Network Aware Migration all VMs will be packed in physical hosts 1 and 3(VMs of same community remain in a single host). In the ideal case if all the VMs are allowed to migrate and every host has large capacity then in the case of Existing Migration Policy all VMs will be packed in a single physical host 1(though cost of moving them may be very high).

Now I simulated the proposed algorithm and existing algorithm in a real data center topology.It is a Tree Topology.There are 16 physical nodes in it.Here the maximum bin value is 9(cpu) and 8(mem) respectively.Number of Min-Cuts allowed is number of Virtual machines divided by seven.All other constants are same.Following table shows the comparison between the algorithms where the two values separated by comma in the two columns denote the number of migrations and migration cost.

Now if the capacities of Physical Machines are halved i.e. its bin values are 4.5(CPU) and 4(Memory) respectively and it is simulated then we see that there is large difference in network cost which is shown in the following table.

Also it is seen that in almost all cases the same number of nodes are suggested for shutdown which proves that it also takes care of energy.

Table 5.3: Comparison

Number of VMs	Proposed Algorithm	Existing Algorithm
30	(16 , 28.051)	(16 , 32.635)
40	(22 , 58.787)	(23 , 62.999)
50	(21 , 77.276)	(24 , 89.542)
60	(28 , 147.001)	(28,142.919)
70	(38 , 206.877)	(39 , 248.466)
80	(42 , 261.589)	(44 , 319.48)
90	(45 , 346.996)	(45 , 340.6)
100	(51 , 432.1)	(51 , 456.533)

Table 5.4: Another Comparison

Number of VMs	Proposed Algorithm	Existing Algorithm
40	(19 , 61.3916)	(22, 78.4772)
50	(22 , 89.2288)	(22 , 105.641)

Result: Returns the CurrentVMPlacement,migrations and costSaved

Data: graph_ VM is network status matrix,graph_ PM is datacenter's graph,movable,currentPlacement,usagePMRemaining,k,VMList;

floydWarshall(graph_ PM);

for *each* i *in* *numberVM* **do**

if $(a1 * VMList[i][0] + a2 * VMList[i][1]) < threshold$ **then**

 movable[i]=1;

else

 movable[i]=0;

end

end

for *each* i *in* k **do**

 minCut(graph_ VM, 0,PV-1);

end

for *each* i *in* *cutIndex* **do**

 graph_ VM[Cut[i][0]][Cut[i][1]]=0 ;

 graph_ VM[Cut[i][1]][Cut[i][0]]=0 ;

end

Algorithm 1: pseudo code for proposed scheme-part 1

```

for each i in numberVM do
  if movable[i]==1 then
    mincost=9999.0,indexmin=9999,changed=0 ;
    for j in PV & & currentPlacement[i]!=j do
      if (usagePMRemaining[j][0] > VMList[i][0]) & &
        (usagePMRemaining[j][1] > VMList[i][1]) then
        float cost=0 ;
        for each k in numberVM & & i!=k do
          if graph_ VM[i][k] > 0 then
            cost+=graph_ VM[i][k]*dist[j][currentPlacement[k] ;
          end
        end
        if mincost > cost & & cost!=0 then
          mincost=cost ;
          indexmin=j ;
          changed=1 ;
        end
      end
    end
  if changed==1 then
    usagePMRemaining[currentPlacement[i]][0]+=VMList[i][0] ;
    usagePMRemaining[currentPlacement[i]][1]+=VMList[i][1] ;
    currentPlacement[i]=indexmin ;
    usagePMRemaining[indexmin][0]-= VMList[i][0] ;
    usagePMRemaining[indexmin][1]-= VMList[i][1] ;
  end
end

```

Algorithm 2: pseudo code for proposed scheme-part 2

Chapter 6

Conclusion and Future Work

6.1 Conclusion

A new algorithm for migration of virtual machines was developed which is network-aware which works inside a data center and also considers other factors like c.p.u usage, memory usage and energy usage. The method tries to reduce network traffic by placing the virtual machines (applications) nearer or in the same server which interact frequently with each other (if data intensive applications are placed nearer then less network i/o cost and less network latency) and at the same time it sets the bin/server size according to the optimal energy point of servers so that less energy usage is there. The algorithm determines result on the basis of distance hence it is independent of topology of network. It was observed by testing on a smaller example of set of virtual machines and a small data center graph that a prudent and logical choice of migration was taken by the proposed algorithm

6.2 Future Work

If possible better algorithms to recognize communities can be used so that the complexity of the proposed algorithm can be improved. Also a better virtual machine placement algorithm to go with this migration algorithm can be used. Also if possible better network aware migration schemes can be developed using the traffic, topologies of data centers. The algorithm could be tested in real time data centers to test its practicality.

Bibliography

- [1] https://en.wikipedia.org/wiki/Cloud_computing
- [2] Bein, Doina, Wolfgang Bein, and Swathi Venigella, *Cloud storage and online bin packing*, In Intelligent Distributed Computing V, pp 63-68, Springer Berlin Heidelberg, (2012)
- [3] Xu, Jing, and José Fortes, *A multi-objective approach to virtual machine management in datacenters*, In Proceedings of the 8th ACM international conference on Autonomic computing, pp 225-234, ACM, (2011)
- [4] Mohammadi, Eslam, Mohammadbager Karimi, and Saeed Rasouli Heikalabad, *A novel virtual machine placement in cloud computing*, Australian Journal of Basic and Applied Sciences 5, no. 10, pp 1549-1555, (2011)
- [5] Vu, Hieu Trong, and Soonwook Hwang, *A Traffic and Power-aware Algorithm for Virtual Machine Placement in Cloud Data Center*, International Journal of Grid & Distributed Computing 7, no. 1, pp 350-355, (2014)
- [6] Srikantaiah, Shekhar, Aman Kansal, and Feng Zhao, *Energy aware consolidation for cloud computing*, In Proceedings of the 2008 conference on Power aware computing and systems, vol no. 10, IEEE, (2008)
- [7] Falkenauer Emanuel, *A Genetic algorithm for Grouping*, In Procs of the Fifth Int'l Symposium on Applied Stochastic Models and Data Analysis, vol no. 10, Granada, Spain, World Scientific Publishing Co. Pte. Ltd., (1991)
- [8] Xiaoqiao Meng, Vasileios Pappas, Li Zhang (IBM T.J. Watson Research Center), *Improving the Scalability of Data Center Networks with Traffic-aware Virtual Machine Placement*, INFOCOM, 2010 Proceedings IEEE, IEEE, (2010)

- [9] Chaisiri, Sivadon, Bu-Sung Lee, and Dusit Niyato, *Optimal virtual machine placement across multiple cloud providers*, In Services Computing Conference, APSCC 2009, IEEE Asia-Pacific, pp. 103-110, IEEE, (2009)
- [10] Stage, Alexander, and Thomas Setzer, *Network-aware migration control and scheduling of differentiated virtual machine workloads*, In Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing, IEEE Computer Society, pp. 9-14, IEEE, (2009)
- [11] Hacking, Stuart, et al., *Improving the live migration process of large enterprise applications*, Proceedings of the 3rd international workshop on Virtualization technologies in distributed computing, ACM, (2009)
- [12] Clark, Christopher, et al., *Live migration of virtual machines*, Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation Volume 2, USENIX Association, (2005)
- [13] Hines, et al. , *Post-copy live migration of virtual machines*, ACM SIGOPS operating systems review 43.3, pp. 14-26, (2008)
- [14] http://en.wikipedia.org/wiki/Live_migration