Formatted: Font: (Default) Times New Roman

Load Balancing of virtual machines on multiple hosts

-By
Prince Malik
Kavya Mugadur
Sakshi Singh

Instructor:
Prof. Ming Hwa Wang
Santa Clara University

Table of Contents

Formatted: Font: (Default) Times New Roman

Formatted: Font: (Default) Times New Roman, Not Bold, Check spelling and grammar

Formatted: Font: (Default) Times New Roman

1. Introduction

1.1 Objective

On a cloud computing platform, dynamic resources can be effectively managed using virtualization technology. Virtualization technologies enable application computation and data to be hosted inside virtual containers (e.g. virtual disks) which are decoupled from the underlying physical resources. These virtualization- based clouds provide a way to build a large computing infrastructure by assessing remote computational, storage and network resources. Since a cloud typically comprises a large amount of virtual and physical servers, to efficiently managing (load balancing) this virtual infrastructure has attracted considerable interest in recent years.

Load balancing of the entire cloud system can be handled dynamically by using virtualization technology where it becomes possible to remap virtual machines (VMs) and physical resources according to the change in load. However, in order to achieve the best performance, the virtual machines have to fully utilize its services and resources by adapting to the cloud computing environment dynamically. The load balancing and proper allocation of resources must be guaranteed in order to improve resource utility.

Thus, the important objectives of this paper are to determine

- How to achieve effective load balance in a cloud computing environment.

1.2 What is the problem?

Virtualization is a computer architecture technology by which multiple virtual machines (VMs) are multiplexed in the same hardware machine. In a cluster of multiple hosts, VMs are not always distributed equally. A few hosts may be overloaded or underutilized compared to others. Load balancing of resources across VMs is the fundamental problem of Cloud Computing.

This project provides a scheduling strategy to enable effective load balancing. The method used in this paper will compute its influence on the system in advance, when current VM resources are allocated to every physical node and will opt for the deployment that will have the least load on the system.

1.3 Why this is a project related to this class

The cloud computing technology is rapidly developing nowadays because of its flexible features of automatic configuring, freely expanding, on-demand resource allocating and power saving, which just hits the goal of various organizations on IT infrastructure construction. Meanwhile, virtualization technology has firm relationship with cloud computing technology because of the common nature of virtualization feature.

1.4 Why other approach is no good

Many researchers in the past have proposed different scheduling algorithms like static, dynamic and mixed scheduling strategies. For example *Bin Packing* algorithms used in dynamic resource scheduling, *Best Fit Decreasing* (BFD) and *First Fit Decreasing* (FFD) which are suitable for high internet speed, ignorable communication delay and small distributed environments and these algorithm does not guarantee optimal solutions.

Some dynamic resource allocation algorithms that balances and distributes computing volume by using available resources in a virtualized environment. Thus, using dynamic migration all, systems can achieve system load balance; but frequent dynamic migration would employ a large number of resources that might lead to degrading the entire system performance.

1.5 Why you think your approach is better

Our approach is avoiding unnecessary migrations, it is taking the cost of migration in consideration. Our approach is suitable for both large scale and small scale cloud infrastructures.

1.6 Area or scope of investigation

The area or scope of investigation is concentrated on the writing and implementing the efficient algorithm for loading balancing of virtual machines on multiple hosts.

Formatted: Font: (Default) Times New Roman, Bold

Formatted: Font: (Default) Times New Roman, Bold

Formatted: Font: (Default) Times New Roman, Bold

Formatted: Font: (Default) Times New Roman

Formatted: Font: (Default) Times New Roman, Bold

Formatted: Font: (Default) Times New Roman

Formatted: Font: (Default) Times New Roman, Bold

Formatted: Font: (Default) Times New Roman, Bold

2. Theoretical basis and literature overview

2.1 Definition of the problem

A virtual machine (virtual machine) is a software implementation of a machine (i.e. a computer) that executes programs like any physical machine. Virtualization technologies are used to enhance the hardware load on server systems and allow a more efficient use of those servers. Nowadays, there is a wide range of existing HA solutions which guarantee the availability of all virtual machines. There are just a few commercial solutions available for allocating virtual machines during their operation time to optimize the actual server workload (e.g. VMware DRS, VirtualIron LiveCapacity). A generic solution for all kinds of virtualization technologies is nonexistent today. In cloud computing, the system should avoid wasting resources as a result of under-utilization and avoid lengthy response times as a result of over-utilization. Load balancing of resources (CPU and Memory) across virtual machines is the fundamental problem of Cloud Computing.

2.2 Theoretical background of the problem

Load balancing is a fundamental problem of cloud computing. Applications run inside VMs and VMs run on physical or virtual hosts. When a VM is first provisioned, it has a computation requirement. Thus, it must be placed on a host that could provide the expected computational resources like CPU, Memory etc. After the initial provisioning, when a VM runs it consumes computational resources. This requirement could change with time. So would the ability of a host to provide these resources. Therefore, at run time we may need to move VMs around to other hosts so that no VMs would starve for computational resources. This problem is Dynamic Load balancing of virtual machines where an algorithm is required to identify an under-utilized or over-utilized host and to recommend a migration plan to maintain a balance.

2.3 Related research to solve the problem

The following papers have been read and reviewed to understand various available approaches:

1. Design and Implementation of an Efficient Load-Balancing Method for Virtual Machine Cluster Based on Cloud Service - Rui Wang, Wei Le, Xuejie Zhang. Yunnan University
2. A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment - Jinhua Hu, Jianhua Gu, Guofei Sun, Tianhai Zhao. NPU HPC Center
3. Distributed Load Balancing Allocation of Virtual Machine in Cloud Data Center - Fei Ma, Feng Liu and Zhen Liu
4. The Load Balancing Algorithm in Cloud Computing Environment - Haozheng Ren, Yihua Lan, Chao Yin. Huazhong University of Science and Technology
5. Adaptive Distributed Load Balancing Algorithm based on Live Migration of Virtual Machines in Cloud - Yi Zhao, Wenlong Huang. Chinese Academy of Sciences
6. VMCTune: A Load Balancing Scheme for Virtual Machine Cluster Based on Dynamic Resource Allocation - Wenyu Zhou, Shoubao Yang, Jun Fang, Xianlong Niu and Hu Song. University of Science and Technology of China

2.4 Advantage/ Disadvantage of the related research

Advantages of the related research listed in above section can be listed as follows:

- Good for small scale applications. Less extra space compared to First fit algorithm
- More effective by first sorting the list into decreasing algorithm

However, these approaches are not best fit in present cloud requirements because of following reasons:

- Does not guarantee an optimal solution
- For large scale applications, increases the algorithm running time.

2.5 Our Solution to solve this problem

We propose an effective load balancing algorithm that considers the load of the host, the number of virtual machines, resources used by virtual machines and resources available at the host, hit count and various other important factors in selecting a virtual machine to migrate. We break the problem into two parts – Identifying the virtual machine that needs to be migrated and identifying the host where virtual machine needs to be migrated to. The selection of the target host is done in such a way that migration oscillation is minimized. When the overall datacenter load hits a global threshold, additional hosts are made active to avoid performance degradation.

Formatted: Font: (Default) Times New Roman, Bold

Formatted: Font: (Default) Times New Roman, Bold

Formatted: Font: (Default) Times New Roman, Bold

Formatted: Font: (Default) Times New Roman, Bold

Formatted: Font: (Default) Times New Roman, Bold

2.6 Where the solution is different from others

We have divided the problem into 2 parts – How to correctly estimate virtual machine resource utilization and how to apply effective virtual machine migration strategy to achieve effective resource utilization.

- Our solution runs different strategy on the hosts depending on domain where it fallstakes into account the cost of moving virtual machine from one host to another by attaching weights to the data.
- Also, it considers the quiescence introduced in the virtual machine while preparing for migrationWe avoid unnecessary migrations
- We ensure that the hosts are in optimal domain by using prediction algorithm

2.7 Why is this solution better?

Our solution is better than the other available solutions because of the following reasons:

- It does not unnecessarily runs the load balancing algorithm. Better resource utilization is considered for initial placement of the virtual machine. Thus load balancing is not required to run as a background process all the time. We can pitch in load balancing algorithm when hosts are not balanced. One scenario is when a few new hosts are added to the cluster and these new hosts are underutilized compare to the old hosts.
- It ensures that all hosts are in optimal domain. It uses Single Exponential Smoothing (SES) algorithm for prediction.
- Our solution considers all costs associated with the migration. Migrating a virtual machine from one hosts to another brings in a delay and has an overhead associated. Before preparing a migration scheme, we consider the cost of moving a virtual machine, cost of running the algorithm and considering that the new host will not be over-utilized after migration.
- This solution can cater small scale as well as large scale data centers.

Formatted: Font: (Default) Times New Roman, Bold

Formatted: Font: (Default) Times New Roman, Bold

Formatted: Font: (Default) Times New Roman, 12 pt

Formatted: Font: 11 pt

Formatted: Font: (Default) Times New Roman, 11 pt

Formatted: Font: (Default) Times New Roman

3. Project Goals

This project is focused on implementing an effective algorithm to balance resource utilization on a cluster. This goal can be explained as below:

Whenever a new virtual machine is run, it should be placed in a host which is minimal utilized. After a few new hosts join cluster, there is a need of load balancing algorithm that iterates through each host recognizing over/under utilized hosts. We then identify the virtual machine that needs to be moved to balance the anomalous host. Also, we need to identify another host that will accommodate the virtual machine identified for migration.

Formatted: Normal, Indent: Left: 0.25", No bullets or numbering

Formatted: Font: (Default) Times New Roman, Bold

4. Methodologies

4.1. How to generate/Collect input data

Load balancing algorithm is devised for effective resource utilization across cluster. Thus we need resource utilization information of each host and the VMs running on it. We collect and use this statistical data of running VMs and hosts as input data.

4.2 How to Solve the Problem

4.2.1 Algorithm Design

The algorithm will be executed in below steps:

Step 1: *Initial Placement* - When a virtual machine is powered on in the cluster, it needs to be placed on an appropriate host.

Step 2: *Distribution & Monitoring* - The algorithm distributes virtual machine workload across the hosts inside the cluster by periodically monitoring the active workload and the available resources.

Step 3: *Load balancing* - Compare the results to the normal resources distribution and perform or recommend virtual machines migration.

- VM Selection - When the resources get over-utilized/under-utilized identify the VM to be migrated.
- PM Selection – Identify the appropriate host where selected VM will be migrated.

4.2.3 Language Used

- Java

4.2.4 Tools Used

- VMware ESXi
- VMware vSphere
- viJava, Java Swing
- Eclipse IDE
- NetBeans IDE

4.3. How to generate output

Once the load balancing algorithm is executed, we check the status of all the hosts.

Formatted: Font: (Default) Times New Roman, Bold

Formatted: Font: (Default) Times New Roman

Formatted: Font: (Default) Times New Roman, Bold

Formatted: Font: (Default) Times New Roman, Bold

Formatted: Font: (Default) Times New Roman

Formatted: Font: (Default) Times New Roman, Bold

Formatted: Font: (Default) Times New Roman

Formatted: Font: (Default) Times New Roman, Bold

5. Implementation

5.1 Code

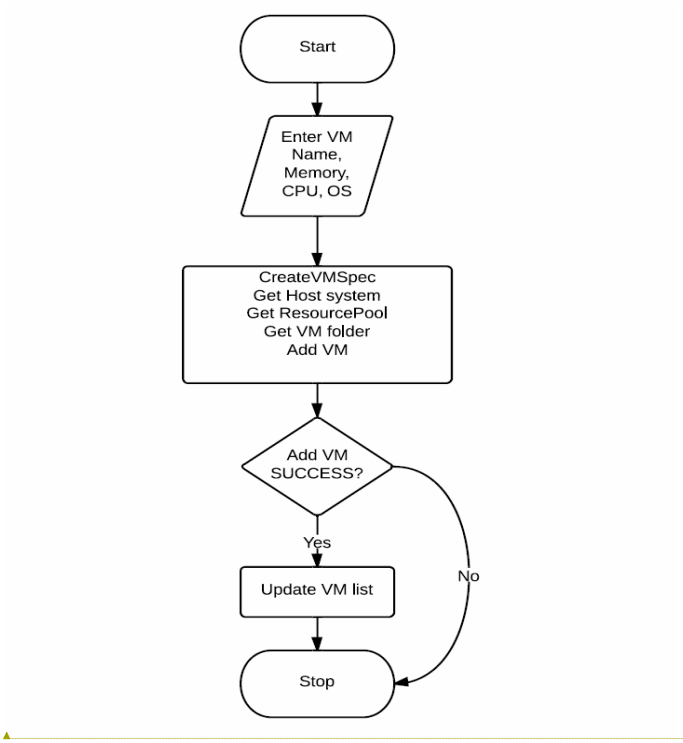
The entire set of source code pertaining to our project can be found at the Appendix section of this document.

5.2 Design and Flowchart

5.2.1 Flow Charts

Complete process covered in this project has been divided into below flowcharts:

1. *Add VM*



Formatted: Font: (Default) Times New Roman

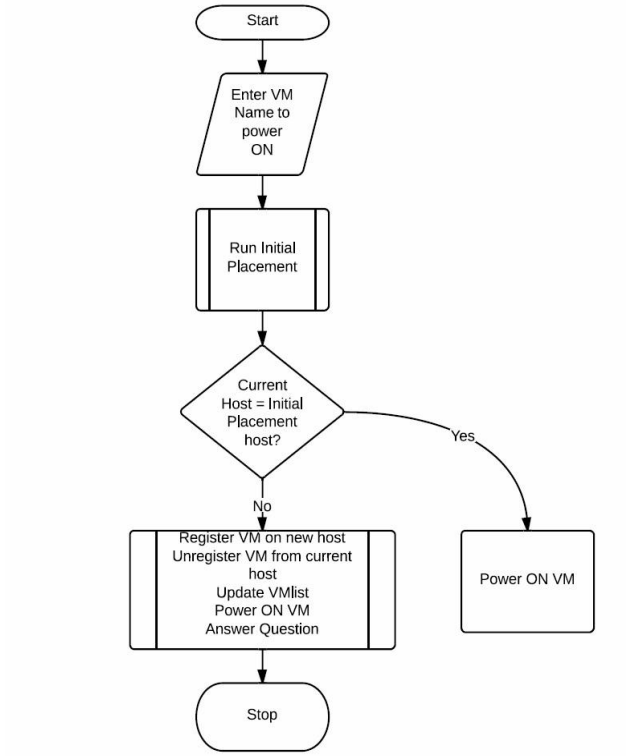
Formatted: Font: (Default) Times New Roman, Bold

Formatted: Font: (Default) Times New Roman, Bold

Formatted: Font: (Default) Times New Roman

Formatted: Font: (Default) Times New Roman

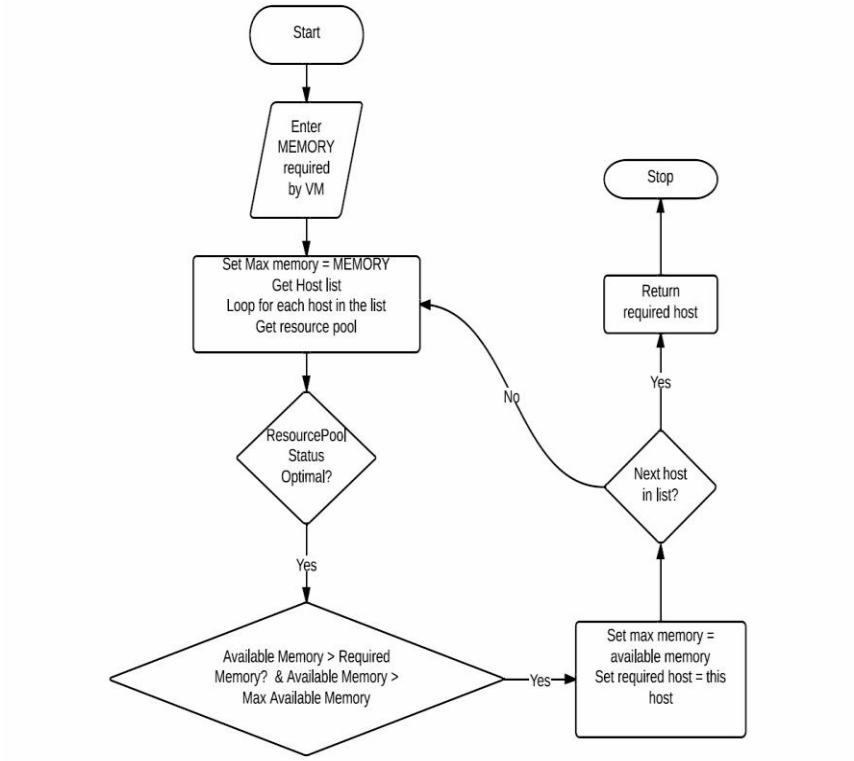
2. *VM Power ON*



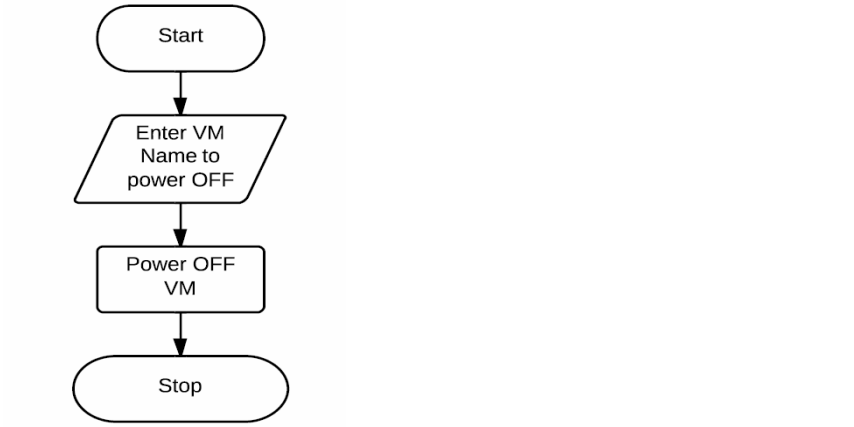
Formatted: Font: (Default) Times New Roman

Formatted: Font: (Default) Times New Roman

3. *Initial Placement*



4. *VM Power off*



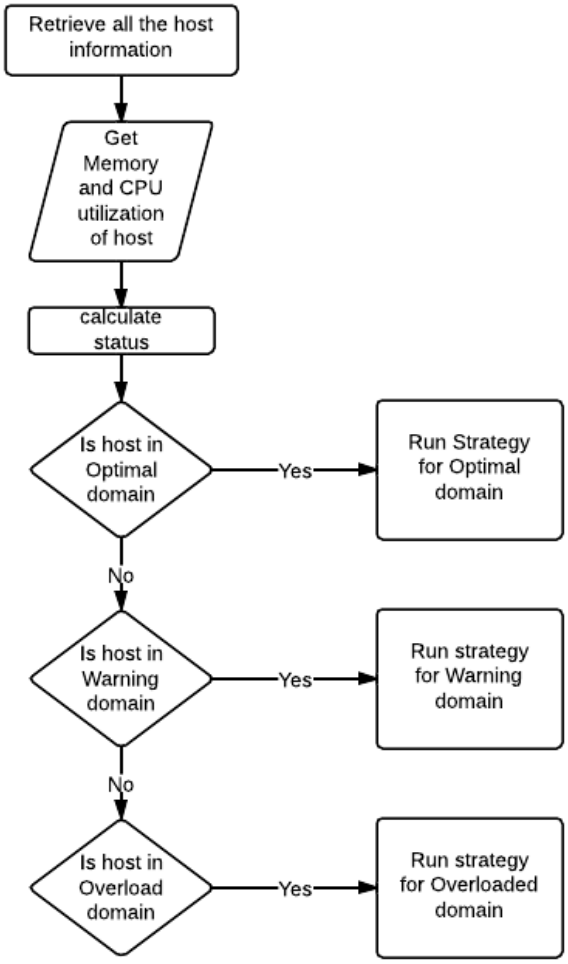
Formatted: Font: (Default) Times New Roman

Formatted: Font: (Default) Times New Roman

Formatted: Font: (Default) Times New Roman

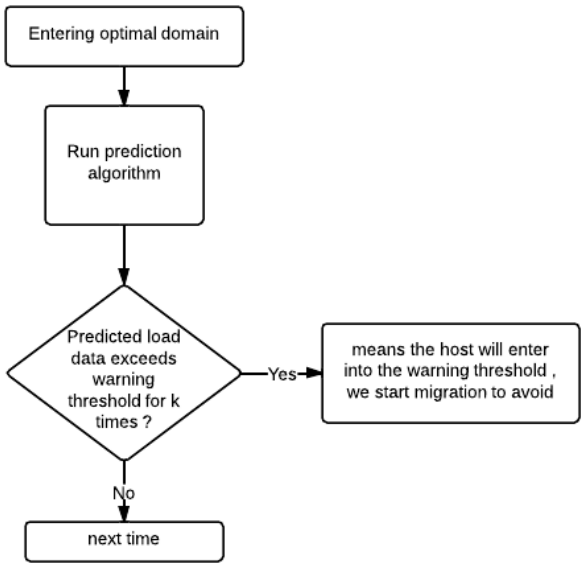
Formatted: Font: (Default) Times New Roman

5. *Optimized control strategy for Load Balancing*



6. *Strategy for optimal domain*

If the predicted load of host is going up to across the warning threshold (K), we should choose a suitable virtual machine to migrate out. The procedure can be described as the following
Step1: When the host lies on the optimal domain, we predict the load values in the future n time interval.
Step2: We count the number k that how many of the predicted load values exceeds the warning threshold.
Step3: If the ratio *k* is greater than the set K, we start the migration. Otherwise, we return to the step1 and continue predicting.

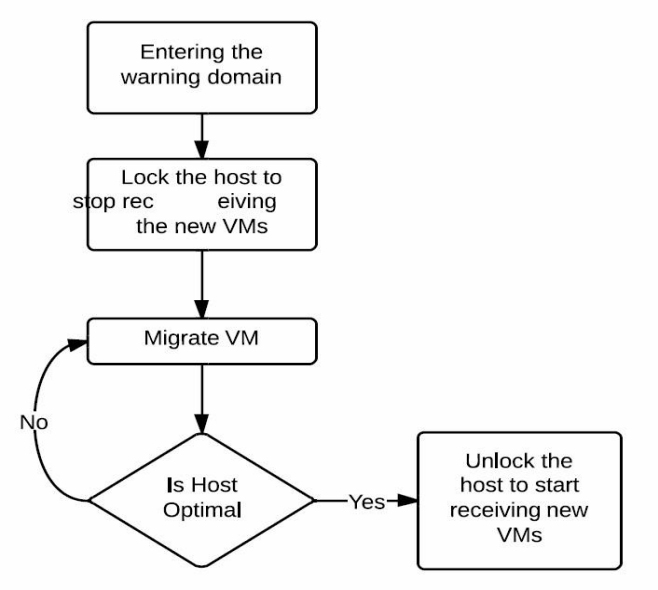


7. *Strategy for warning domain*

Formatted: Font: (Default) Times New Roman

At this moment, the host load has been so high that the system performance is going to reduce. We adopt the following strategy as the following
Step1: When the host lies on the warning domain, firstly we lock the host and prevent it from receiving the new tasks.
Step2: Virtual machine is migrated on the host.
Step3: Check whether is in optimal state
Step4: unlock the host and make it able to receive the new VM again.

Formatted: Font: (Default) Times New Roman

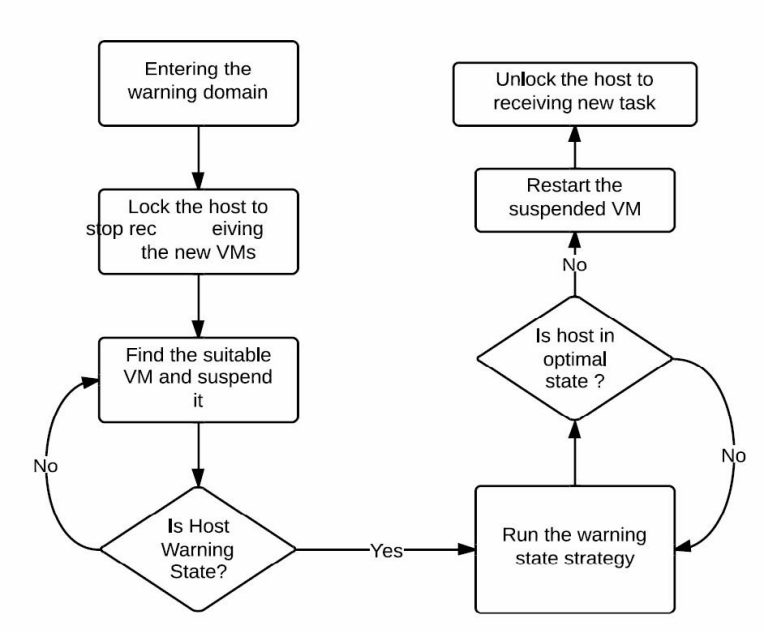


8. *Strategy for overloaded domain*

Formatted: Font: (Default) Times New Roman

Through the strategy above, we try the best to avoid that the host enters into the overload domain. However, if the host is unlucky to enter into the overload domain, we adopt the following strategy to prevent the host breakdown.

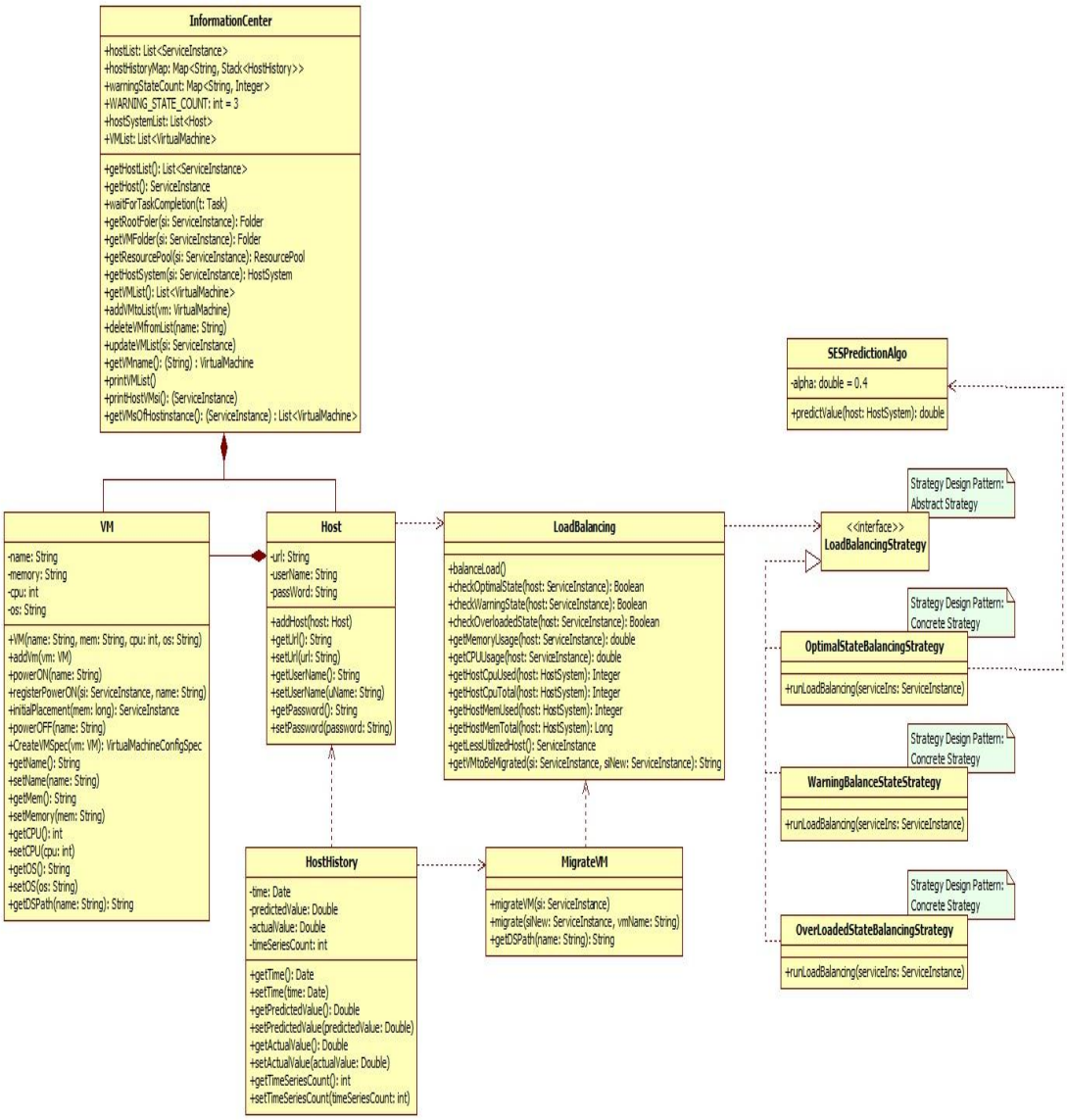
Step1: Firstly, we lock the host to stop receiving the new tasks.
Step2: We find the virtual machine with the highest load and suspend this virtual machine.
Step3: If the host load has been lower than the overload threshold, we migrate out some virtual machine as the strategy for the warning domain
Step4: When the migration is finished, we reset the virtual machine suspended.
Step5: We unlock the host to start receiving the new tasks.


Entering the warning domain
Lock the host to stop rec eiving the new VMs
Find the suitable VM and suspend it
Is Host Warning State?
No
Yes
Run the warning state strategy
Is host in optimal state ?
No
No
Restart the suspended VM
Unlock the host to receiving new task

5.2.2 Class Diagram

Formatted: Font: (Default) Times New Roman, Bold

Formatted: Font: (Default) Times New Roman



6. Data Analysis and Discussion

6.1 Output Generation

~~/* System config needs to be added */~~ We have setup the following system to evaluate our algorithm:

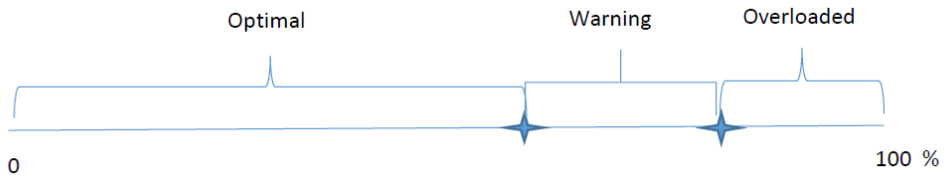
1. Hosts: We have 3 ESXi hosts (VM) I nour system. Each host has 2GB RAM, 8GB HDD and 2 CPU.
2. NFS – We installed NFS openfiler on another VM having 1 GB of RAM and 15GB HDD. We configured it with running NFS service and use it as a shared data storage amongst all the 3 hosts.
3. We have also installed vSphere clients connected to each of the above mentioned hosts and are used for GUI based monitoring purpose.

We ~~ran~~create and then power ON ~~-~~multiple virtual machines independently on a single host. Host for each VM is selected using FFD algorithm during Initial Placement. Each host includes resource components (CPU, memory). We calculated these resource usage. CPU utilization is calculated as follows:

CPU utilization = ((total CPU – CPU used by host - CPU used by VMs))/total CPU * 100 %

Similarly, we calculated the memory usage also.

According to the loading condition of host, we divide three status domains as shown below:



~~where~~Where Optimal is between 0 to 60%, Warning is ~~65~~60% to 85% and Overloaded is 85 to 100%. We ran different strategies depending on the domain where host lies. We saw that the load was equally balanced and optimized in our multiple host infrastructure.

6.2 Output Analysis

~~6.2 Output Analysis~~

Optimized Control Strategy for Load Balancing:
As explained above, we divide three status domains.

Optimal domain: The host lies on the optimal status. We should observe the load trend and adopt suitable strategy from going across the warning threshold.

Warning domain: This is a buffer domain between optimal domain and overload domain. In this domain, we should adopt suitable strategy to avoid the host from entering into the overload domain.

Overload domain: The unlucky host in overload domain has poor performance. If nothing to do, the host is possible to down.

Candidate VM choice
When the host load is getting too high, we choose the time for migrating according to the strategy above. Then we will consider that which virtual machine should be migrated out. We choose the VM which is used maximum memory.

Migration destination choice
After determining the virtual machine migrated, we should look for the suitable migration destination. We choose the host which is less utilized.

Formatted: Space After: 0 pt, Line spacing: Exactly 12 pt
Formatted: Font: (Default) Times New Roman, Bold
Formatted: Justified, Space Before: 0 pt, Line spacing: Exactly 12 pt
Formatted: Space Before: 0 pt, Line spacing: Exactly 12 pt
Formatted: Space After: 0 pt, Line spacing: Exactly 12 pt
Formatted: List Paragraph, Line spacing: Exactly 12 pt, Numbered + Level: 1 + Numbering Style: 1, 2, 3, … + Start at: 1 + Alignment: Left + Aligned at: 0.25" + Indent at: 0.5"
Formatted: Font: (Default) Times New Roman, 12 pt
Formatted: List Paragraph, Line spacing: Exactly 12 pt
Formatted: Font: (Default) Times New Roman
Formatted: Font: (Default) Times New Roman
Formatted: Font: (Default) Times New Roman
Formatted: Font: (Default) Times New Roman
Formatted: Line spacing: Exactly 12 pt
Formatted: Space After: 0 pt, Line spacing: Exactly 12 pt
Formatted: Font: (Default) Times New Roman
Formatted: Line spacing: Exactly 12 pt
Formatted: Indent: Left: 0", Space Before: 0 pt, Line spacing: Exactly 12 pt
Formatted: Font: (Default) Times New Roman
Formatted: Space Before: 0 pt, Line spacing: Exactly 12 pt
Formatted: Font: (Default) Times New Roman, 12 pt, Font color: Auto
Formatted: Indent: Left: 0", Space Before: 0 pt, Line spacing: Exactly 12 pt
Formatted: English (U.S.)
Formatted: Indent: Left: 0", Line spacing: Exactly 12 pt
Formatted: Line spacing: Exactly 12 pt
Formatted: Font: (Default) Times New Roman

~~6.3 Compare output against hypothesis~~

As mentioned in section 3, we have achieved our goals. We implemented an effective algorithm to balance resource utilization on multiple host infrastructure.

Whenever a new virtual machine is created and powered on, it is placed in a host which is minimal utilized. After a few new hosts join cluster or there is dynamic changes in existing VMs resource utilization, load balancing algorithm executed that iterates through each host recognizing domain in which host lies. We then used different strategy for each domain to keep the environment balanced. While migrating, we identified the virtual machine that needs to be moved to balance the anomalous host. Also, we identified another host that will accommodate the virtual machine identified for migration.

6.4 ~~6.4~~ Abnormal Case Explanation

From the strategy explained above, we try to avoid the host to enter into overloaded state. However if the host enters into overloaded domain, we run the strategy for overloaded domain which is explained in section 5.2

~~6.5 Discussion~~

Formatted: Font: 13 pt, Bold, Font color: Accent 1, English (U.S.)

Formatted: Font: (Default) Times New Roman, Bold

Formatted: Space Before: 0 pt, Line spacing: Exactly 12 pt, No bullets or numbering

Formatted: Font: (Default) Times New Roman

Formatted: Indent: Left: 0", Space After: 0 pt, Line spacing: Exactly 12 pt

Formatted: Font: (Default) Times New Roman

Formatted: Space After: 0 pt, Line spacing: Exactly 12 pt

Formatted: Indent: First line: 0", Space Before: 0 pt, Line spacing: Exactly 12 pt

Formatted: Font: (Default) Times New Roman, Bold

Formatted: Font: (Default) Times New Roman

Formatted: Space After: 0 pt, Line spacing: Exactly 12 pt

Formatted: Font: (Default) Times New Roman, Bold

Formatted: Indent: First line: 0", Space Before: 0 pt, Line spacing: Exactly 12 pt

Formatted: Font: (Default) Times New Roman

7. Conclusions and Recommendations

7.1 Summary and Conclusion

In the project, we implemented an optimized control strategy which combines multi-strategy mechanism with the prediction mechanism. We adopt different control strategy for different domains which are divided according to the host resource utilization. We design and implement optimized control strategy for load balancing based on ~~live~~ disruptive migration of virtual Machine.

7.2 Recommendation for future studies

In this project, we use FFD for Initial placement and an optimized control strategy for load balancing based on live migration of virtual machine. But we can improve to find resource in scheduling and improving of scheduling algorithm is the key work in future research work. Besides, there are four more interesting issues to be addressed:

~~1) Network Resources: Through adopting the prediction and multi-migration strategy, we avoid the unnecessary costs caused by the instantaneous peak of resource utilization. This will save network resources and optimize system performance.~~

~~2~~1) ~~Disk I/O balancing: This project can be improved by scheme using high performance shared device to provide Disk I/O. So we can use distributed file system in next solution to get better scalability and disk I/O balancing.~~Migration based on cost: Before making a migration scheme we can consider the cost or migrating a VM from one host to another.

~~3~~2) ~~Power saving~~Dynamic Power Management: As power saving and green computing become more and more important now a days. It is good way to add power saving policy into our scheme. When plenty of nodes are in light load status, we can migrate VMs to fewer nodes and ~~shutdown~~ Power OFF/ Suspend nodes without VM running.

~~4~~3) QoS mechanism: As an addition to fair scheduling policy mentioned in disk I/O, we can plan to lead QoS mechanism into a scheme for some VMs that host critical applications and services which need stable performance. These VMs will be allocated adequate resource that cannot be seized by other low priority VMs.

Formatted: Font: (Default) Times New Roman, Bold

Formatted: Justified

Formatted: Font: (Default) Times New Roman, Bold

8. Bibliography

1. Design and Implementation of an Efficient Load-Balancing Method for Virtual Machine Cluster Based on Cloud Service - Rui Wang, Wei Le, Xuejie Zhang. Yunnan University
2. A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment - Jinhua Hu, Jianhua Gu, Guofei Sun, Tianhai Zhao. NPU HPC Center
3. Distributed Load Balancing Allocation of Virtual Machine in Cloud Data Center - Fei Ma, Feng Liu and Zhen Liu
4. The Load Balancing Algorithm in Cloud Computing Environment - Haozheng Ren, Yihua Lan, Chao Yin. Huazhong University of Science and Technology
5. Adaptive Distributed Load Balancing Algorithm based on Live Migration of Virtual Machines in Cloud - Yi Zhao, Wenlong Huang. Chinese Academy of Sciences
6. VMCTune: A Load Balancing Scheme for Virtual Machine Cluster Based on Dynamic Resource Allocation - Wenyu Zhou, Shoubao Yang, Jun Fang, Xianlong Niu and Hu Song. University of Science and Technology of China
7. Load Balancing Algorithm based Virtual Machine Dynamic Migration Scheme for Datacenter Application with Optical Networks - Xinyu Zhang, Yongli Zhao, Xin Su, Ruiying He, Weiwei Wang, Jie Zhang. Beijing University of Posts and Telecommunications, Beijing
8. An Optimized Control Strategy for Load Balancing based on Live Migration of Virtual Machine - Ke Yang, Jianhua Gu,Tianhai Zhao,Guofei Sun. Northwestern Polytechnical University, Xi'an China.
9. Programming Guide – VMware Infastructure SDK 2.5
10. http://vijava.sourceforge.net/vSphereAPIDoc/ver5/ReferenceGuide/
11. http://pubs.vmware.com/vsphere-50/index.jsp#com.vmware.sdk.doc_50/GUID-E0BBED4C-8C5A-4C01-B21A-575F85E9A650.html
12. http://vijava.sourceforge.net/doc/getstarted/tutorial.htm
13. http://www.vmware.com/support/developer/vc-sdk/visdk25pubs/visdk25programmingguide.pdf
14. http://searchvmware.techtarget.com/tutorial/VMware-vSphere-beginners-guide
15. https://devcentral.f5.com/articles/intro-to-load-balancing-for-developers-ndash-the-algorithms
16. http://sourceforge.net/projects/vijava/
17. http://vmg.pp.ua/books/%D0%9A%D0%BE%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D1%8B%D0%98%D1%81%D0%B5%D1%82%D0%B8/%D0%92%D0%B8%D1%80%D1%82%D1%83%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%82%D0%BE%D1%80%D1%8B/VMware/Jin%20S.VMware%20VI%20and%20vSphere%20SDK.Managing%20the%20VMware%20Infrastructure%20and%20vSphere.Prentice.%5BENG,647p.,2010%5D.pdf

Formatted: Font: (Default) Times New Roman, Bold

Formatted: Font: (Default) Times New Roman

Formatted: Default Paragraph Font, Font: (Default) +Body (Calibri), 11 pt

Formatted: Default Paragraph Font, Font: (Default) +Body (Calibri), 11 pt

Formatted: Default Paragraph Font, Font: (Default) +Body (Calibri), 11 pt, English (U.S.)

Formatted: Default Paragraph Font, Font: (Default) +Body (Calibri), 11 pt, English (U.S.)

Formatted: Default Paragraph Font, Font: (Default) Times New Roman

Formatted: Default Paragraph Font, Font: (Default) Times New Roman

Formatted: Default Paragraph Font, Font: (Default) Times New Roman

Formatted: Default Paragraph Font, Font: (Default) Times New Roman

Formatted: Font: (Default) Times New Roman

9. Appendices

9.1 Program Source code

LoadBalanceVM.zip

9.2 Input/ Output Listing

Input File

Input File.docx

Output File

Formatted: Font: (Default) Times New Roman, Bold

Field Code Changed

Formatted: Font: (Default) Times New Roman

Formatted: Font: (Default) Times New Roman, Bold

Formatted: Normal

Field Code Changed