Improving Efficiency for Dynamic load in Cloud Infrastructure

Ravi Shah[1], Ms. Gayatri Pandi[2]

[1] M.E. Student, Information and Technology Engineering, L.J. Institute of Engineering & Technology, Gujarat, India

[2] Prof., Information and Technology Engineering, L.J. Institute of Engineering & Technology, Gujarat, India

ABSTRACT

Nowadays cloud computing is becoming one of the most used technological solution to achieve scalability and reduce costs. Scalability is a key point for the success of any business involving the Web and providing services to end-user requests that may vary drastically from one time to another. Auto-scaling is a key feature in clouds responsible for adjusting the number of available resources to meet service demand. Resource modifications are necessary to keep performance indicators, like utilization level between user defined lower and upper bounds. There are lots of problem of Scale up and Scale down in auto Scaling. In this paper focusing on how to handle scale up in auto Scaling. In this paper focusing on on resource utilization and dynamic load handling with scale up in public cloud.

Keyword: - *Cloud computing, Auto scale, scale up, Software-as-a-Service, Platform-as-a-Service, Infrastructure-as-a-Service, Utilization, Allocation*

1. INTRODUCTION

Now a days, every organization is propagating towards the use of cloud computing. With no doubt it can be said that within few years, there will be lots of users for cloud computing. During that period of time, cloud providers will need to maintain more effective techniques for VMs to handle such huge amount of requests than the current scenario. And it is of no worth to buy new hardware just to use CPUs instead of fully utilizing current resources. Thus, an efficient auto-scaling mechanism is needed to handle much load and to maximize utilization of currently existing resources. Moreover, cloud users are charged on an hourly basis by the cloud providers. So, every VM that is being utilized, should be utilized with the maximum efficiency so that instead of adding new resources and increasing the budgets for the users, currently existing resources are utilized to their full extent. Thus auto-scaling of Virtual Machines by efficiently utilizing the currently existing ones that are customizable as per the user's requirement is a major concern for cloud providers. This motivated me to generate the rule and provide the resource dynamically.

1.1 Cloud Auto-scaling:

Auto-scaling is a cloud specific technique that allows you to change the number of currently working Virtual Machines (VMs) in a cloud network. We run multiple copies of our application on more than one VMs and the load balancer is responsible for forwarding incoming requests to appropriate VM. Based on the number of requests received on the VM, the resource utilization of that VM might increase or decrease and on the basis of this, we increase or decrease the number of VMs used in that network so that efficient and flawless resource utilization and application working is carried out. When a traffic campaign does well, generated traffic may be much higher than expected, leaving servers unprepared for spikes in activity. In such cases, there may not be enough time to

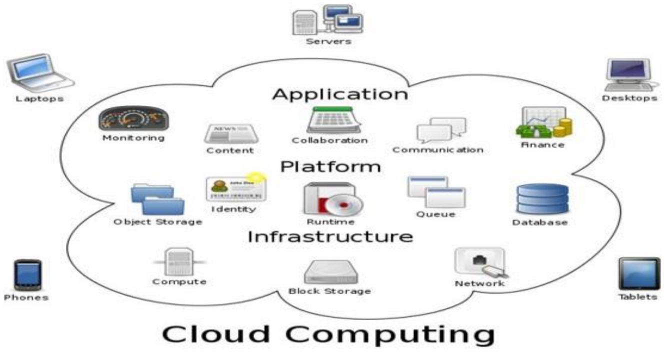


Figure 1: Architecture of Cloud Computing[3]

manually provision new instances and resources. Auto scaling is geared toward solving this common problem. Fig. 2 shows such an auto-scaling where the number of Virtual Machines increases and decreases on the basis of usage.

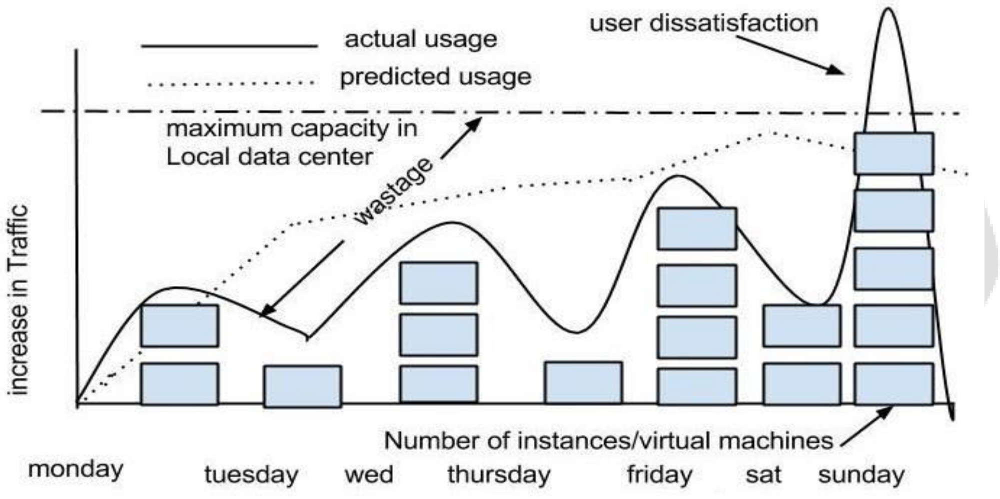


Figure 2: Cloud Auto-scaling[4]

2. RELATED WORK

A. Improved Max-Min Algorithm In Max-Min algorithm large tasks have highest priority and smaller tasks have lower priority. It means, when we have one long task, then Max-Min algorithm could execute many short tasks concurrently while executing large task. The makespan is calculated in this by the execution of long task .It would be similar to the Min-min makespan. There are many existing load balancing algorithms in cloud computing which used for load balancing. Some new algorithms are also implemented from existing algorithms, this will helps to researchers to carry out further work in this area. We combine improved max min and ant algorithm as hybrid approach. Improved max min work in different way from original max min algorithm.[1]

```
1. for all submitted tasks in meta-task; Ti
2. for all resources; Rj
3. Cij = Eij + rj
```

```
4. While meta-task is not empty
5. find task Tk costs maximum execution time.
6. Assign Tk to the resource Rj which gives minimum completion time.
7. remove Tk from meta-tasks set
8. update rj for selected Rj
9. update Cij for all j
```

B. Ant Colony Optimization Algorithm Ant Colony Optimization (ACO) algorithm is inspired from real ant colonies and it work based upon their actual behavior. Ants are live in colonies. They are work for the survival of colony. Ants always travel from their nest and food sources when they searching for food. In the initial stage ants explore the area surrounding their nest in a random way. While moving from one place to another ants deposit special substances called pheromones. Ants can smell pheromones. Ant Colony Optimization (ACO) algorithm is inspired from real ant colonies and it work based upon their actual behavior. Ants are live in colonies.[2]

```
1. begin
2. Initialize the pheromone
3. while (stopping criterion not satisfied) do Position each ant in a starting VM
4. while (stopping when every ant has build a solution) do
5. for each ant do 6. Chose VM for next task by pheromone trail intensity
7. end for
8. end while
9. Update the pheromone
10. end while
11. end
```

3. PRPOSED WORK

Cloud computing is emerging and it has gained a great deal of popularity. With this use of Cloud Computing is increased. At that point of time it is difficult to use all the resources and handle all the requests. When load is increase it is difficult to handle these load with mini-mum response time and maxi-mum resource utilization. In cloud system many tasks (cloudlets) are executed on available resources at same time period. So, in cloud environment Load balancing is required for proper utilization of all the resources and for better response time. So in this research focusing on resource utilization and dynamic load handling with scale up in public cloud.

The main problem is spin up when more request are there. To resolve this problem how to provide Load Balancing with mini-mum Response time and maxi-mum resource utilization including auto-scaling is learned by the context-level architecture.

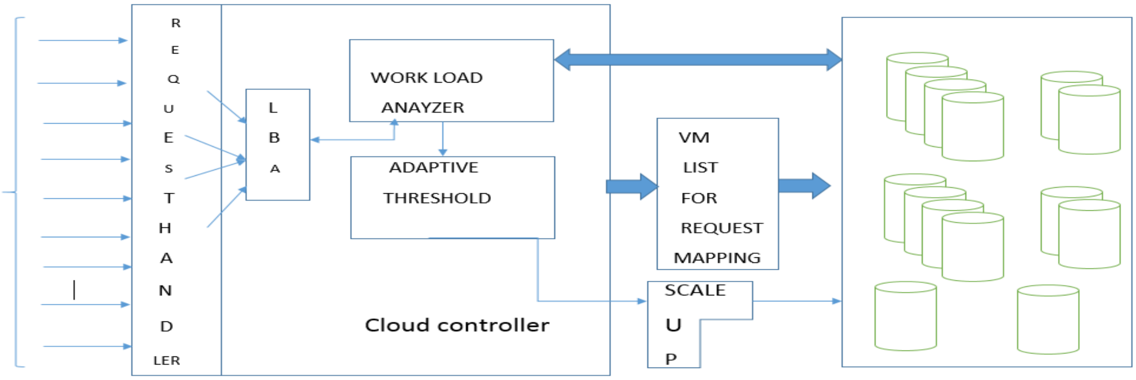


Figure 3: Context level Architecture

How to context level architecture works for storing and accessing the data is implemented using Architecture.

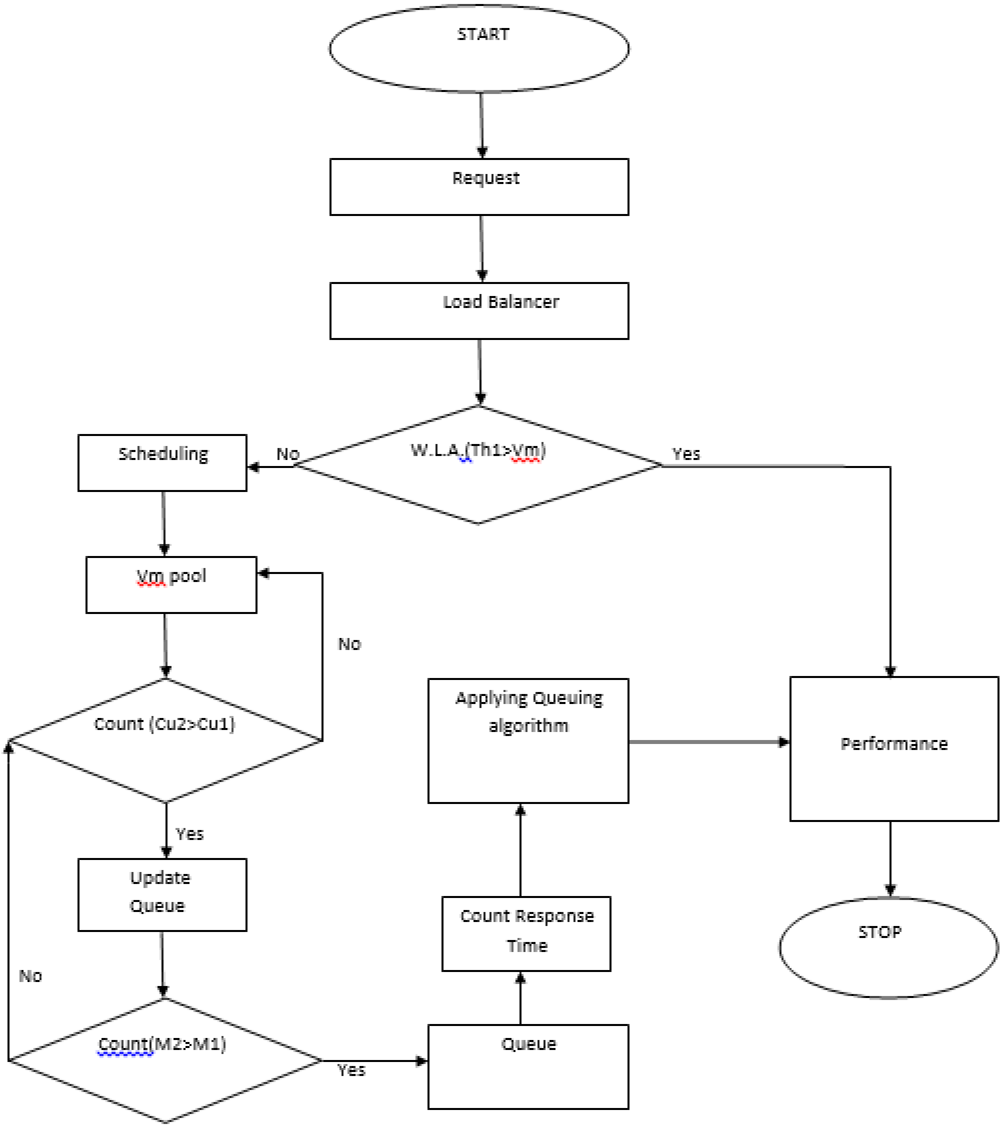
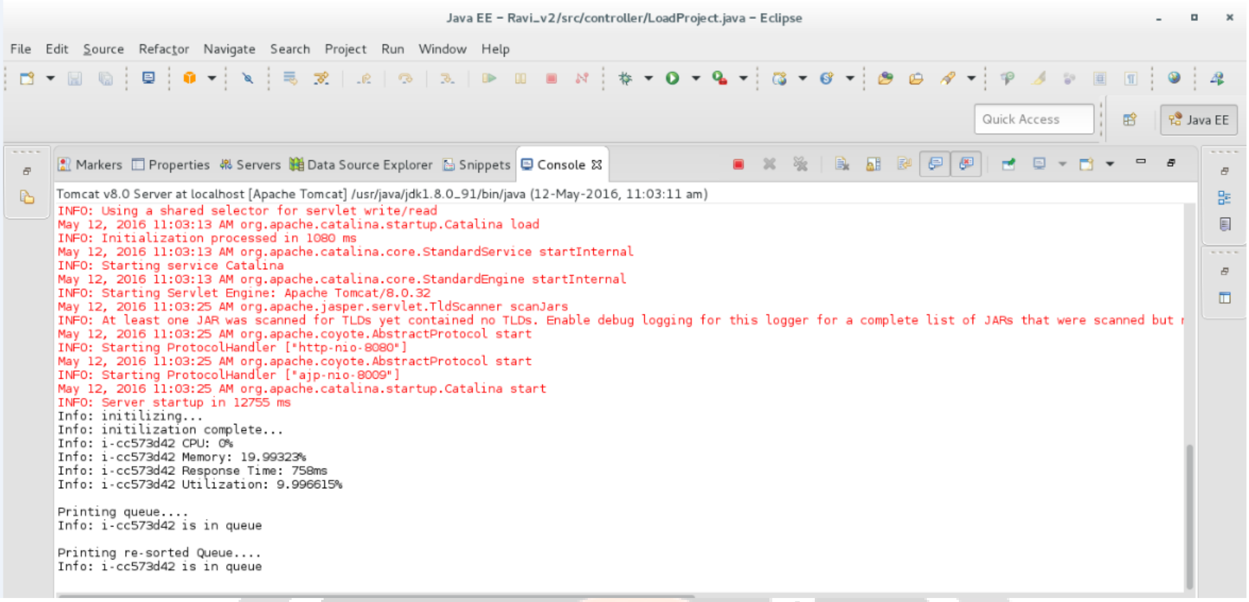


Figure 4: Work Pan

4. RESULTS

Here after counting CPU utilization of different VMs and analyzing memory, we made a queue of VMs having maximum CPU utilization and enhanced use of memory from that queue we have proposed response time of those virtual machines. Thus we can allocate the request to the VM having minimum response time. And after this we ca find new threshold value. And thus maximum resource utilization is done.

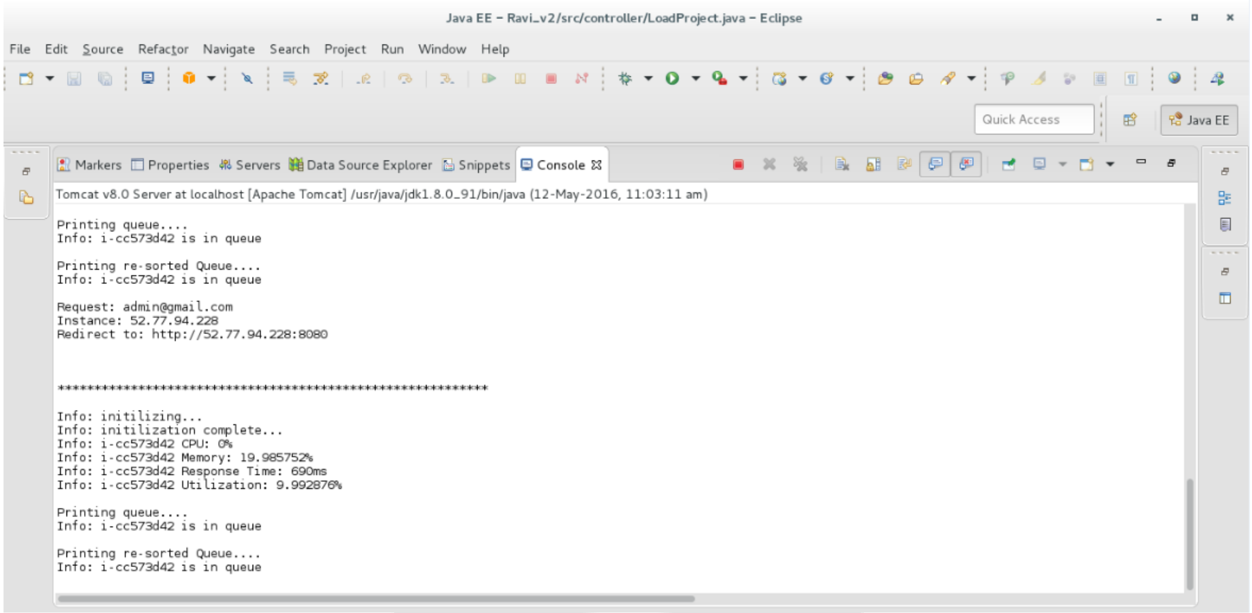

Java EE – Ravi_v2/src/controller/LoadProject.java – Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java EE
Markers Properties Servers Data Source Explorer Snippets Console

```
Tomcat v8.0 Server at localhost [Apache Tomcat] /usr/java/jdk1.8.0_91/bin/java (12-May-2016, 11:03:11 am)
INFO: Using a shared selector for servlet write/read
May 12, 2016 11:03:13 AM org.apache.catalina.startup.Catalina load
INFO: Initialization processed in 1080 ms
May 12, 2016 11:03:13 AM org.apache.catalina.core.StandardService startInternal
INFO: Starting service Catalina
May 12, 2016 11:03:13 AM org.apache.catalina.core.StandardEngine startInternal
INFO: Starting Servlet Engine: Apache Tomcat/8.0.32
May 12, 2016 11:03:25 AM org.apache.jasper.servlet.TldScanner scanJars
INFO: At least one JAR was scanned for TLDs yet contained no TLDs. Enable debug logging for this logger for a complete list of JARs that were scanned but r
May 12, 2016 11:03:25 AM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["http-nio-8080"]
May 12, 2016 11:03:25 AM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["ajp-nio-8009"]
May 12, 2016 11:03:25 AM org.apache.catalina.startup.Catalina start
INFO: Server startup in 12755 ms
Info: initilizing...
Info: initilization complete...
Info: i-cc573d42 CPU: 0%
Info: i-cc573d42 Memory: 19.99323%
Info: i-cc573d42 Response Time: 758ms
Info: i-cc573d42 Utilization: 9.996615%

Printing queue....
Info: i-cc573d42 is in queue

Printing re-sorted Queue....
Info: i-cc573d42 is in queue
```

Figure 3: Count CPU, Memory and Response time


Java EE – Ravi_v2/src/controller/LoadProject.java – Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java EE
Markers Properties Servers Data Source Explorer Snippets Console

```
Tomcat v8.0 Server at localhost [Apache Tomcat] /usr/java/jdk1.8.0_91/bin/java (12-May-2016, 11:03:11 am)

Printing queue....
Info: i-cc573d42 is in queue

Printing re-sorted Queue....
Info: i-cc573d42 is in queue

Request: admin@gmail.com
Instance: 52.77.94.228
Redirect to: http://52.77.94.228:8080



***********************************************************

Info: initilizing...
Info: initilization complete...
Info: i-cc573d42 CPU: 0%
Info: i-cc573d42 Memory: 19.985752%
Info: i-cc573d42 Response Time: 690ms
Info: i-cc573d42 Utilization: 9.992876%

Printing queue....
Info: i-cc573d42 is in queue

Printing re-sorted Queue....
Info: i-cc573d42 is in queue
```

Figure 3: Scheduling Algorithm

4. CONCLUSION AND FUTURE SCOPE

In cloud computing we have presented Auto Scaling by scheduling which is performed by Amazon Web Services. It helps in better load balancing and auto Scaling. The load can be CPU load, Memory capability, and response time. Load balancing is the process of distributing the jobs between all VMs of the cloud system to improve resource utilization and response time. From which we can avoid a situation where some VMs are heavily loaded and some are idea or doing nothing. In this we have concentrated on CPU, Memory and Response time. We considered limited number of cloudlets because it effect on cost. As for future work, there are some important point on which further work is needed. In this we have taken limited VMs. We are not considered cost model in our work.

5. REFERENCES

[1] Navtej Singh Ghumman, Rajwinder Kaur "Dynamic Combination of Improved Max-Min and Ant Colony Algorithm for Load Balancing in Cloud System " Cse Department SBS State Technical Campus Ferozepur, India ,2015 IEEE , ISSN: 1669-7624, 2014 IEEE, 2014

[2] Reena Panwar, Prof. Dr. Bhawna Malick "Load Balancing in Cloud Computing Using Dynamic Load Management Algorithm "Department of Computer Science and Engineering , Galgotias College of Engineering and Technology Greater Noida, India ,2015 IEEE,DOI 978-1-4673-7910-6/15.2015 IEEE,DOI 1O.1109/COMPSACW.2014.116, 2014

[3] Jing Jiang, Jie Lu, Guangquan Zhang "optimal cloud Resource Auto-scaling for web applications", *Facility of Engineering* and information Technology ,Australia, 2013 IEEE ,DOI 10.1109/CCGrid.2013.73, 2013

[4] Marco A. S. Netto, Carlos Cardonha, Renato L. F. Cunha "Evaluating Auto-scaling Strategies for Cloud Computing Environments" Department of IBM Research, 2014 IEEE, DOI10.1109/ICSC.2014.43, 2014

[5] Thoung Van Vu, Thi Mai Trang Nguyen, Guy Pujolle, Nadia Boukhatem "DODEX+ A New Network Coding Scheme for Mesh Network in Mobile Cloud Computing" IEEE IFPI Wireless Days, ISBN 978-1-4799-6606-6, pp 1-6, November 2014

[6] Mohammad Halloush , Hayder Radha "The Unequal Protection of Network Coding with Multi-generation Mixing" IEEE 2014 International Conference on innovations Information Technology, ISBN 978-1-4577-0311-9 pp.29-34,Aug 2014

[7] Martan Sipos , Frank H.P. Fiztek, Daniel E. Lucani ,Morten V. Pedersen " Distributed Cloud Storage Using Network Coding" IEEE 11th Consumer Communications and Networking Conference", ISBN 978-1-4799-2356-4, pp 127-132, January 2014

[8] Ronald L. Krutz ,Rusell Dean Vines "Cloud Security" , Wiley Publishing, Inc. Indianapolis, Indiana, ISBN: 978-0-470-58987-8, 2010