International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 17
December 2017

Resource Allocation to Maintain the Load Balance Using Scheduling in Cloud Computing

Mr. Parveen
Department of Computer Science
E-Mail: parveenbarak1988@gmail.com

Abstract: *There are an increasing number of Cloud Services available in the Internet. Cloud services can be a component of a system and different Cloud Servers that would provide different services. In this present work we have defined a multiple cloud environment. Each cloud server is defined with certain limits in terms of memory and the CPU specifications. Now as the users enter to the system, the user request is performed in terms of processes. To represent the parallel user requests, n number of requests are been generated by the users. All these requests are to be handled by the cloud servers in parallel by using the multiple cloud concepts. A middle layer is defined between the cloud servers and the client requests that will perform the allocation of the processes to different clouds in under load and over load conditions. As user requests are performed, some parameters are also defined with each request. These parameters are the process time, deadline, input output specifications etc. In the general case, the allocations of the processes are performed in a sequential order. Each process must be executed within the deadline limit. But if more than one processes occur at same time and not get executed before the deadline, in such case the processes is switched from one cloud server to other called the process migration. In this present work, a parametric analysis is performed to identify the requirement of process migration and based on this analysis the migration will be performed on these processes. The effectiveness of the work is identified in terms of successful execution of the processes within the time limits.*

Keywords: Cloud Computing, SaaS, NIST, Process Scheduling, Time Analysis.

Introduction:

Cloud computing is a construct that allows you to access applications that actually resides at a location other than your computer or other internet-connected device. It has become one of the most talked about technologies in recent times and has got lots of attention from media as

well as analysts because of the opportunities, it is offering. The beauty of cloud computing is that another company hosts your application (or the suite of applications, for that matter). This means that they handle the costs of servers, they manage the software updates, and—depending on how you craft your contract—you pay less for the service. It's also convenient for telecommuters and traveling remote workers, who can simply log in and use their applications wherever they are. Cloud computing is combination of two terms: Cloud & Computing. Cloud is the Network. A network is a bulk of thousands of users. These users may or may not be connected. If they are connected, there will be one of model formed (IaaS, PaaS, SaaS), discussed further. The cloud also consists of Server & a Database. Server is also known as Cloud-Provider; while Database is a collection of user-details and applications to be worked upon by users. Computing is the term used for services of cloud. The US National Institute of Standards and Technology (NIST) has developed a working definition that covers the commonly agreed aspects of cloud computing. The NIST working definition summaries cloud computing as: "a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." This definition describes cloud computing as having five essential characteristics, three service models, and four deployment models. The essential characteristics are:

- ✓ On-demand self-service
- ✓ Broad network access
- ✓ Resource pooling
- ✓ Rapid elasticity
- ✓ Measured service

Evolution of Cloud Computing:

Cloud computing means different to different people, its benefits are different to different people. To IT managers, it means to minimize capital-expenditure by outsourcing most of the hardware and software resources. To ISVs, it means to reach out to more users by offering a SaaS solution. To end users, it means to access an application from anywhere using any device. The following diagram illustrates a high level overview. In the beginning of the computing era, the relationship between the user and the machine was one-to-one. One user used to access the applications that (s) he needs to use on one machine. Then the Internet era

came. In the Internet era, the relationship between the user and the machine was many-to-one. Many users could access applications running on one machine. The applications in this case were websites or client-server applications, the machine was a central server hosting the server application, web server or/and the database. In cloud computing, the relationship between the user and machine are many-to-many. Many users can access an application that is served from many machines. Now, what was the reason of this evolution? What were the driving factors behind this? The reason for the evolution from PC-based application to Internet-based application was obvious. This happened because of the need of multiple users trying to access an application from their own machines. The only way that it was possible was to have the application hosted on a central server and having separate client applications communicate to it.

Proposed Objectives:

The proposed System will achieve the following objectives:

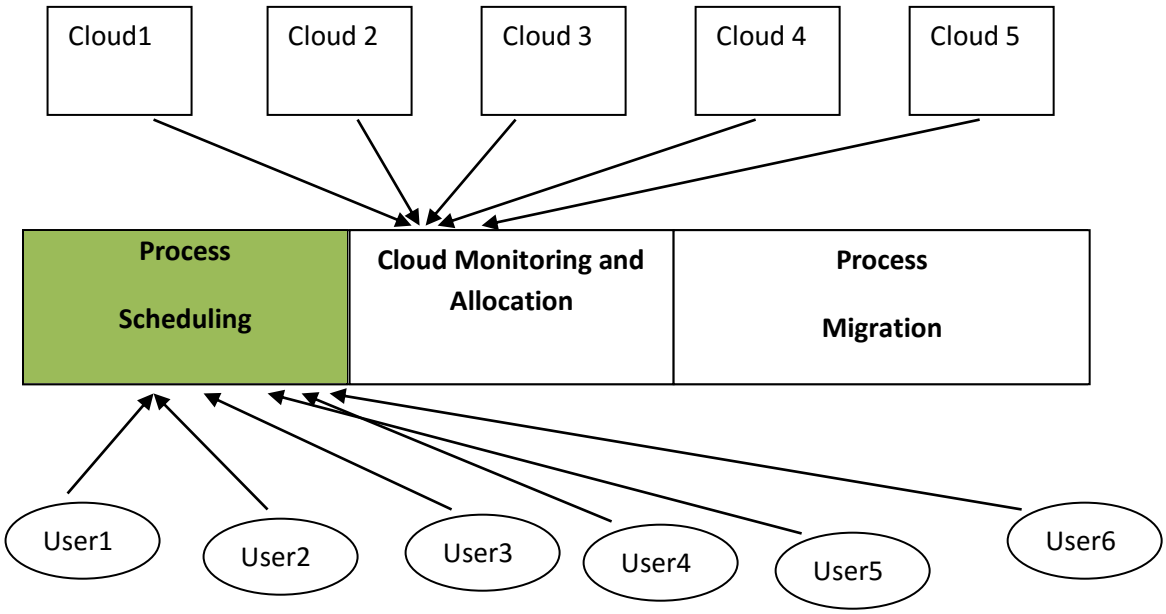
1. Create an Intermediate Architecture that will accept the user request and monitor the cloud servers for their capabilities.
2. Scheduling of the users requests is performed to identify the order of allocation of the processes.
3. Performing the effective resource allocation under defined parameters and the cloud server capabilities.
4. Define a dynamic approach to perform the process migration from one cloud to other.
5. Analysis of the work using graph under different parameters

Research Design:

The proposed system is middle layer architecture to perform the cloud allocation in case of under load and overload conditions. The over load conditions will be handled by using the concepts of process migration. The middle layer will exist between the clouds and the clients. As the request will be performed by the user this request will be accepted by the middle layer and the analysis of the cloud servers is performed by this middle layer. The middle layer is responsible for three main tasks

1. Scheduling the user requests
2. Monitor the cloud servers for its capabilities and to perform the process allocation

3. Process Migration in overload conditions



Process Scheduling

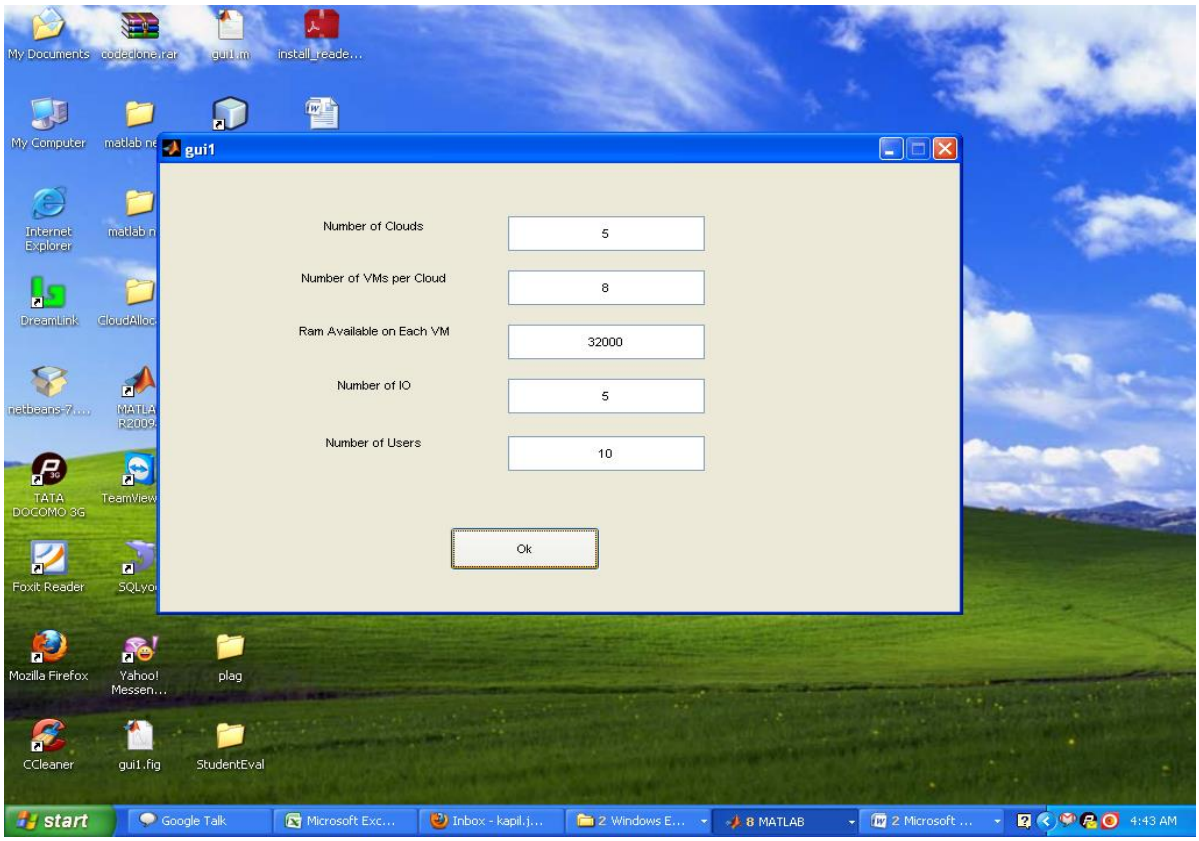
Algorithm:

1. Input the M number of Clouds with L number of Virtual Machines associated with each cloud.
2. Define the available memory and load for each virtual machine.
3. Assign the priority to each cloud.
4. Input N number of user process request with some parameters specifications like arrival time, process time, required memory etc.
5. Arrange the process requests in order of memory requirement
6. For i=1 to N
7. {
8. Identify the priority Cloud and Associated VM having Available Memory > Required Memory(i)
9. Perform the initial allocation of process to that particular VM and the Cloud
10. }
11. For i=1 to N

12. {
13. Identify the Free Time slot on priority cloud to perform the allocation. As the free slot identify, record the start time, process time, turnaround time and the deadline of the process.
14. }
15. For i=1 to N
16. {
17. If finishtime(process(i))> Deadline(Process(i))
18. {
19. Print "Migration Required"
20. Identify the next high priority cloud that having the free memory and the time slot and perform the migration of the process to that particular cloud and the virtual machine.
21. }
22. }

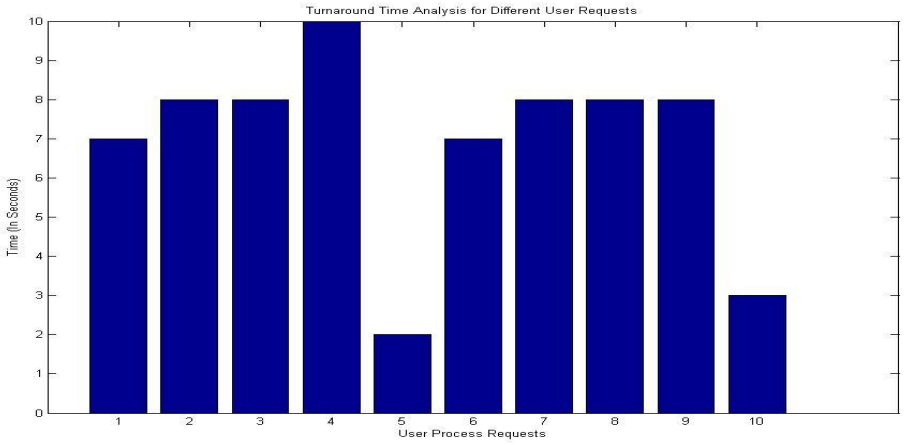
RESULT:

To present the work effectively and to accept the user input, a graphical interface is presented in this work in Matlab language. The graphical user is here to accept the input parameters related to the clouds as well as to define the number of users. The graphical screen of this work is shown in figure.



Graphical Interface

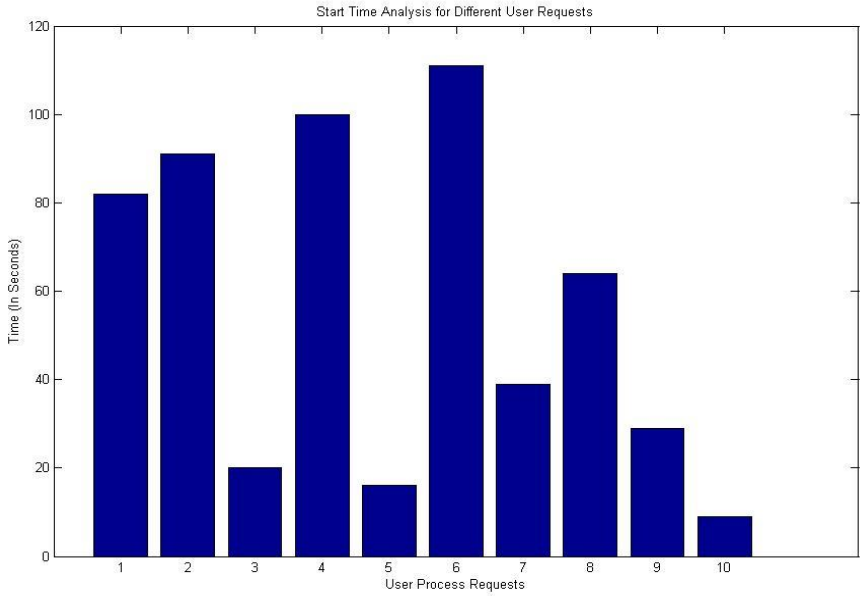
As shown in figure a graphical interface is presented to accept the main input parameters to build the cloud environment and to input the user requirement. The input taken here includes the number of clouds in the environment, number of VMs supported by each cloud, memory availability, IO availability for each virtual machine. Once these all parameters get input, the cloud system is configured. Some parameters are taken on the random basis to show the dynamic processes. These parameters include the priority assignment to each process.



e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 17
December 2017

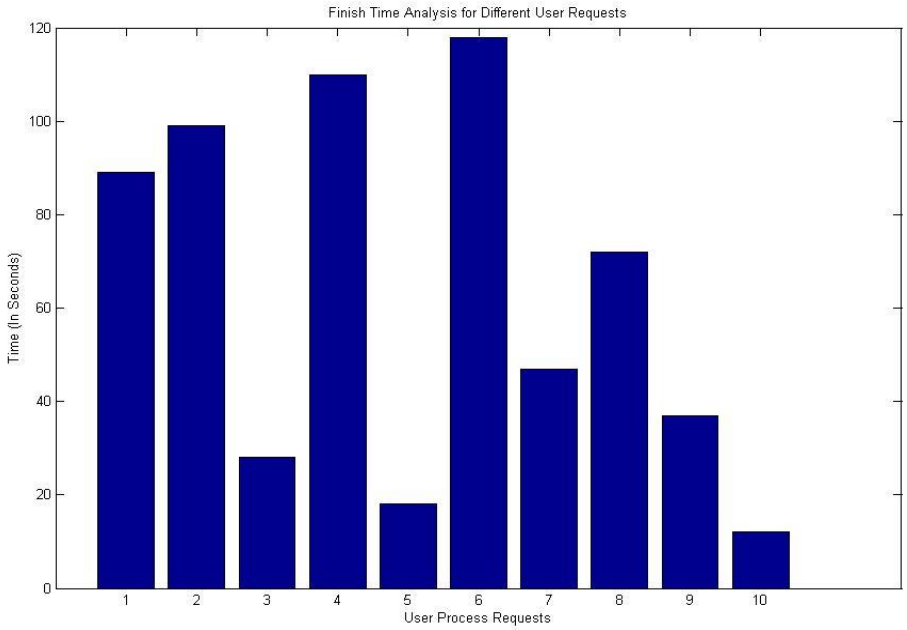
Turn Around Time Analysis

It is showing the turnaround analysis for 10 processes. Here x axis represents the number of user requests and y axis represents the time taken by these processes in seconds. The figure is showing the process time required by each process.



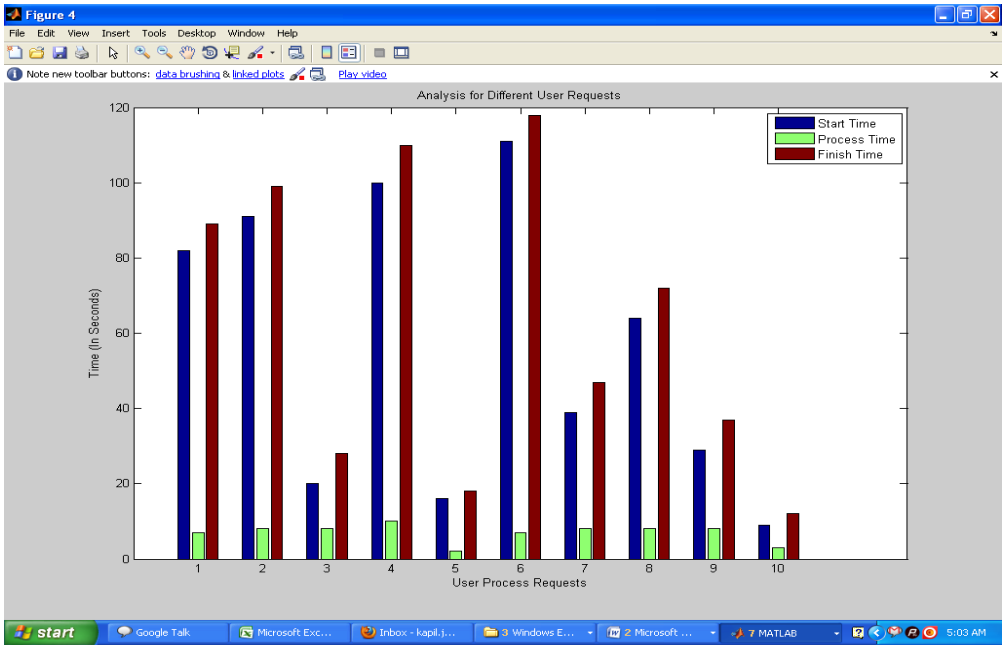
Start Time Analysis

It is showing the Start Time analysis for 10 processes. Here x axis represents the number of user requests and y axis represents the time taken by these processes in seconds. The figure is showing the Start time of each process.



Finish Time Analysis

It is showing the Finish Time analysis for 10 processes. Here x axis represents the number of user requests and y axis represents the time taken by these processes in seconds. The figure is showing the finish time of each process.



Process Analysis

It is showing the Process analysis for all 10 processes. Here x axis represents the number of user requests and y axis represents the time taken by these process in seconds for all three

parameters called process time, finish time and the start time.. The figure is showing these three vectors collectively.

CONCLUSION:

In this present work, a resource allocation scheme on multiple clouds in both the under load and the over load conditions, As the request is performed by the user, certain parameters are defined with each user request, these parameters includes the arrival time, process time, deadline and the input output requirement of the processes. The cloud environment taken in this work is the public cloud environment with multiple clouds. Each cloud is here defined with some virtual machines. To perform the effective allocation, we have assigned some priority to each cloud. The virtual machines are here to perform the actual allocation. These are defined with certain limits in terms of memory, load etc. As the allocation begins, at first the scheduling of the processes is performed respective to the memory requirements. And along with it, the allocation of the process is done to the cloud based on the requirement and the availability analysis. If the allocated process cannot be executed in its required time slot, in such case the migration of the process is required. The migration of the processes is here defined in case of overload conditions. The overload condition is defined in terms of simultaneous processes that are required to execute at particular instance of time. The analysis of the work is done in terms of wait time, process time of the processes. The obtain results shows the successful execution of all the processes within time limit. The work is performed on a generic system that can have n number of clouds.

REFERENCES:

[1] T. Hirofuchi, H. Ogawa, H. Nakada, S. Itoh, and S. Sekiguchi, "A Storage Access Mechanism for Wide-Area Live Migration of Virtual Machines," *In Summer United Workshops on Parallel, Distributed and Cooperative Processing*,2008, pp. 19–24.

[2] I. Foster, T. Freeman, K. Keahey, D. Scheftner, B. Sotomayor, and X. Zhang, "Virtual Clusters for Grid Communities," *In IEEE Int. Symp. On Cluster Computing and the Grid,* 2006, pp 513– 520.

[3] M. Strasser and H. Stamer, "A Software-Based Trusted Platform Module Emulator," *In Trust '08 Proc. of the 1st Int. Conf. on Trusted Computing and Trust in Information Technologies,* 2008, pp. 33–47.

[4] Y. Luo, B. Zhang, X. Wang, Z. Wang, Y. Sun, and H. Chen, "Live and incremental whole-system migration of virtual machine using block-bitmap," *In IEEE Int. Conf. on Cluster Computing*, 2008.

[5] H. Nishimura, N. Maruyama, and S. Matsuoka, "Virtual Clusters on the Fly - Fast Scalable and Flexible Installation," *In IEEE Int. Symp. On Cluster Computing and the Grid*, 2007,pp. 549–556.

[6] M. Tatezono, H. Nakada, and S. Matsuoka, "Mpi environment with load balancing using virtual machine," In *Symp. On Advanced Computing Systems and Infrastructures SACSIS*, 2006, pp. 525–532.

[7] S. Venugopal, R. Buyya, and K. Ramamohanarao. A taxonomy of Data Grids for distributed data sharing, management, and processing. In *ACM Computing Surveys*, volume 38,2006.

[8] A. Gopalakrishnan. 2009. Cloud Computing Identity Management. *SET Labs Briefings*. Vol. 7, No. 7. pp. 45-55.

[9] M. K. Srinivasan, P. Rodrigues, "A roadmap for the comparison of identity management solutions based on state-of-the-art IdM taxonomies," *Springer Communications in Computer and Information Science*,2010, pp. 349-358. Springer-Verlag Berlin Heidelberg, New York, USA.

[10] M. K. Srinivasan, P. Rodrigues, "Analysis on identity management systems with extended state- of-theart IdM taxonomy factors," *International Journal of Ad hoc, Sensor & Ubiquitous Computing*. (December 2010), Vol.1, No.4. pp. 62-70. DOI=10.5121/ijasuc.2010.1406.