www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 03 Issue 07 July, 2014 Page No. 7001-7009

A Multiple Queue Management Scheme to Optimize the Job Sequencing in Cloud Computing

Navdeep Singh[1], Ghanshyam Verma[2],

[1] School of Computer Science and Engineering, Galgotia's University,
Yamuna Expressway, Greater Noida, Uttar Pradesh, India
navdeepsinghbargoti@gmail.com

[2] School of Computer Science and Engineering, Galgotia's University,
Yamuna Expressway, Greater Noida, Uttar Pradesh, India
ghanshyam.verma@galgotiasuniversity.edu.in

Abstract: ***A cloud System is considered as one of most required shared distributed system in which multiple clients are connected with multiple servers in an integrated environment. The intermediate layer is formed between clients and server to arrange the order of their execution. The Server side is defined with specific cloud servers and integrated virtual machines along with memory and IO capabilities. Multiple client requests are generated under the request time parameters. As a user enters to the system, a particular VM is allocated to it for the process execution. It the VM is not capable to execute the process under defined constraints, the migration of the request is performed. In cloud system, one of the critical issues is to handle these multiple request in effective way so that the overall wait time will be reduced. In this present work, multiple queue system is suggested to reduce the wait time and to reduce the chances of request migration. The presented work is divided in three main stages. In first stage, the Virtual machines are set in order of their capacity parameters. Once the machines the ordered, in next stage, the user requests are set in specific queue based on dead line criticality. In this work, two queue middle layers are composed under dead line criticality analysis called hard deadline queue and soft deadline queue. Once the processes are allocated to these queues, the next work is to perform the allocation of the processes to virtual machines. This allocation is performed under capacity and load parameter. After this allocation, the execution of the processes on particular virtual machine is done under process time and deadline parameters. The processes that are not been executed before the deadline are considered as migrated processes. The obtained result from system shows the effective execution of processes on cloud system along with reduced deadline and process migration probability.***

Keywords: Cloud Computing, Virtual machine, Migration, Deadline, Reduced deadline, Criticality, Scheduling, Queue based scheduling, Job sequencing.

1. Introduction

Cloud computing, a new kind of computing model, is coming. This word is a new word that appears at the fourth season, 2007. It is an extend of changing with the need, that is to say the manufacturer provide relevant hardware, software and service according to the need that users put forward. With the rapid development of the Internet, user's requirement is realized through the Internet, different from changing with the need. In fact cloud computing is an extension of grid computing, distributed computing, and parallel computing. Its foreground is to provide secure, quick, convenient data storage and net computing service centered by internet. The factors that impel the occurring and development of cloud computing include: the development of grid computing, the appearance of high quality technology in storage and data transportation, and

the appearance of Web2.0, especially the development of Virtualization (Loganayagi.B, 2011).

The character of cloud computing is in the virtualization, distribution and dynamically extendibility. Virtualization is the main character. Most software and hardware have provided support to virtualization. We can virtualize many factors such as IT resource, software, hardware, operating system and net storage, and manage them in the cloud computing platform; every environment has nothing to do with the physical platform. Carries on the management, the expansion, the migration, the backup through the hypothesized platform, all sorts of operations will be completed through the virtualization level. Distributional refers to the physical node which the computation uses is distributed. Dynamic expandability is refers to through the dynamic extension virtualization level, and then achieves to above apply carries on the expansion the goal. Has broken between the physical structure barrier, represents is transforming the physical resources for logic may manage the resources the inevitable trend. In the future, all resources transparently will move in each physical platform, the resources management will carry on according to the logical way, will realize the resources automated assignment completely, but the virtualization technology realizes this ideal only tool. In view of the cloud computation, the virtualization technology's fusion and the application should face the high-quality hypothesized main engine, the application and the resources, as well as aspects and so on.

2. Scheduling

Scheduling is one of the major requirements in distributed environment to setup the order of user requests so that the execution will be done in effective way. The scheduling approaches are defined for the ordering of requests under the specification of execution time so that that the real time scheduling will be obtained. In a multi programming based time sharing system, the scheduling is having its importance to provide constraint specific communication over the network.

2.1. **Cloud Scheduling:** The cloud systems are large scale system that requires improving the performance under application program analysis, scalability analysis and faulting tolerant communication over the network. This kind of network proves the parallel communication and cost effective communication over the network. This kind of communication are effective to provide low cost communication. The main feature of scheduling in cloud environment are given here under.

2.1.1. *Good Processor Utilization:* All the processors are defined to process on their relative queues so that the effective execution will be performed over the network. These kind of networks are defined under the time constraint analysis so that the execution time and the execution speedup will be obtained. The process coordination and its effective execution all depends on process utilization.

2.1.2. *Good Synchronization effectiveness:* In these kind of distributed network, the task integration is performed under different vectors so that the fine grained interaction will be obtained.

2.1.3. *Low communication/memory-access cost*: The task based analysis in a communication domain is performed so that the time shared communication and reduced latency is obtained and minimized. The scheduling scenarios are defined under data structure specification under contention specification.

2.2. Characteristics of Cloud Computing: The major characteristics that are provided in a cloud computing system are listed under :

2.2.1. *Shared Infrastructure:* To present the available infrastructure in an effective way, the virtualization software model is required. Such software model requires the sharing of physical services and the storage and network capabilities. This cloud infrastructure model is defined regardless to the deployment model under the infrastructure based analysis independent to the number of users.

2.2.2. *Dynamic Provisioning:* Such kind of systems are presented by provisioning of service based architecture under the demand requirement analysis of different uses. This kind of system are defined with software automation and expansion so that the contradiction to the system capability will not occur. Such systems are dynamic respective to the client need and modify the offered services based on the service request analysis.

2.2.3. *Network Access:* This kind of systems is defined under different network architecture such as laptop, PCs, mobile devices and the integrated API. The work includes the service deployment in the cloud respective to the business application in an integrated distributed environment. The environment can be a smart phone environment or it can be a desktop based integrated environment.

2.2.4. *Managed Metering:* These kind of systems are defined under the service management and optimization. In includes, the reporting services and billing the services in distributed environment.

2.3. Cloud computing Challenges: The use of the cloud provides a number of opportunities: (Ristenpart, 2009)

A cloud computing system also required to deal with different challenges associated with system.

I. It enables the services to be used without specification of the infrastructure. As the cloud system is defined in virtual environment, the robustness in terms of infrastructure can be achieved easily and effectively.

II. Cloud computing system must be scalable. Scalability itself gives number of benefits. The major benefit is the low cost transform to the wider size network. These kind of system also support the concept of on-demand pricing and the revenue system. These systems are effective enough to provide the robustness in terms of ongoing system without major changes.

III. Cloud system also based on the location independent system, it means the system will be affected respective to the location of the service.

IV. Cloud computing system provides the parallel request offering to the system as well as parallel execution of the services in the system.

V. Cloud computing provides the dependence to the system under different service capabilities such as flexibility and innovations. There are number of web architectures that itself defines the cloud system in an open market. To manage the huge requests from users high computing capability based infrastructure is required to the system.

VI. Security is one of major issue that defines the safe communication in terms of product delivery, ongoing transactions and the access to the services in a secure way.

VII. These kind of systems are also defined with policy definition. This system includes the secure access to the system so that effective access to the server will be done.

VIII. Different kind of system area available to different kind of users under different constraints. These constraints specification is dynamic and respective to the user.

3. Proposed work

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.

3.1 OBJECTIVES:

The proposed System will achieve the following objectives

I. Create an Intermediate Architecture that will accept the user request and monitor the cloud servers for their capabilities.

II. Scheduling of the users requests is performed to identify the order of allocation of the processes.

III. Performing the effective resource allocation under defined parameters and the cloud server capabilities.

IV. Define a dynamic approach to perform the process migration from one cloud to other.

V. Analysis of the work using graph under different parameters

3.2 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

I. Scheduling the user requests

II. Monitor the cloud servers for its capabilities and to perform the process allocation

III. Process Migration in overload conditions.

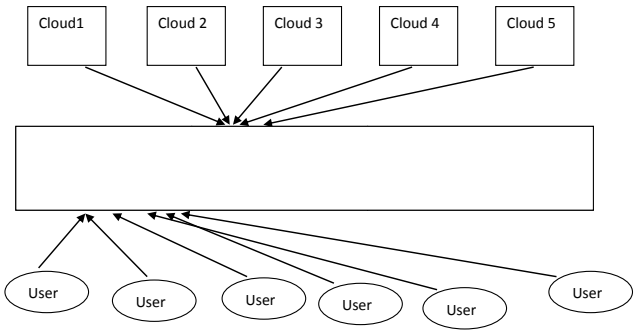

Figure 3.1 : Process Scheduling

3.3 Flow chart:

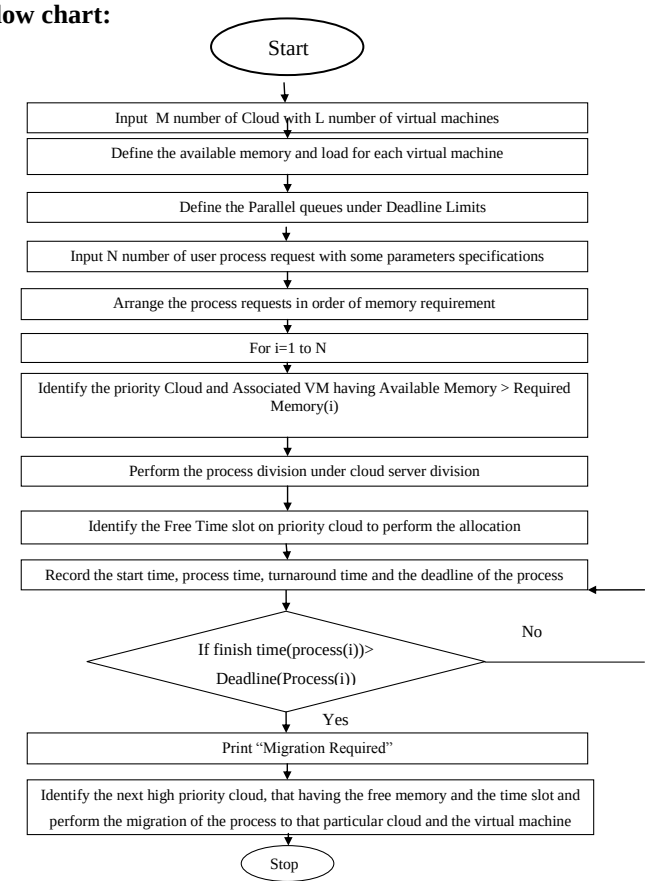

Figure 3.2: Flowchart of Algorithm

3.4 Algorithm:

/*Define a Cloud based System with intermediate layer to communicate between cloud server and clients*/

1.	Define C Number of Clouds along with Integrated M virtual machines in Each Cloud
2.	Assign The priorities to Each Cloud Based on Performance Parameter
3.	Arrange the Clouds Servers in Order of the Priorities Assigned to Cloud
4.	Reorder the Virtual Machines Based on IO and Memory Capabilities.
5.	Accept N Number of User Requests with Relative Parameters such as ProcessTime, Arrival Time, IORequest etc.
6.	Define the DeadLine Criticality Threshold to Generate Multiple Queues
7.	For i=1 to N [Process All Requests] {
8.	if (Request(i).DeadLine<Threshold) {

9.	HardRealTimeQueue.Add(Request(i))
	}
	else
	{

10.	SoftRealTimeQueue.Add(Request(i)) } }
11.	Arrange the HardRealTimeQueue User Members based on Memory Requirement
12.	Arrange the SoftRealTimeQueue User Members based on Memory Requirement
13.	Process HardRealTimeFirst to assign on Prioritized Clouds and VMS
14.	Perform the Initial Allocation of HardRealTimeQueue Processes Based on Memory Requirement and Capacity Analysis and VM Load Analysis
15.	Perform the Initial Allocation of SoftRealTimeQueue Processes Based on Memory Requirement and Capacity Analysis and VM Load Analysis
16.	Perform the Execution of user Request on VM under arrival time, process time and deadline analysis
17.	Perform the Process Migration under Waittime Analysis
18.	Analyze the Performance Vector respective to Client request

4. Result

4.1 Implementation Tool

MATLAB Editor is used for writing the code to implement our algorithm.The result will be shown in the command window of MATLAB.

4.2 GUI

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

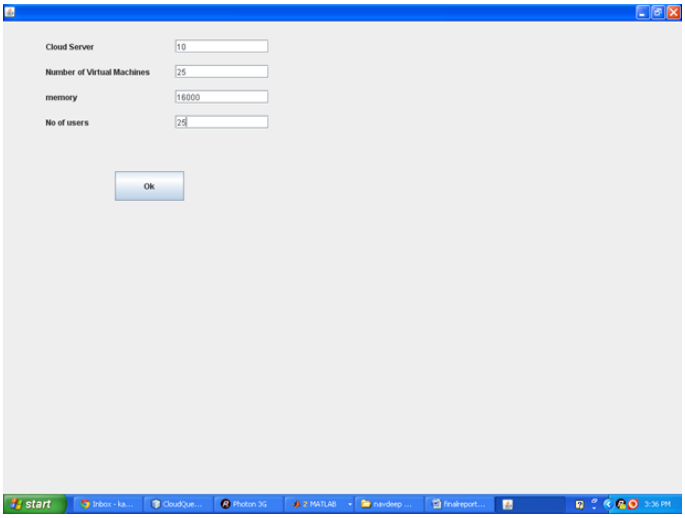

Figure 4.1 : Graphical Interface

As shown in figure 4.1, 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. By using these parameters the clouds server is configured. After that, the number of users , input to the system to represent the number of process requests. 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.

4.3 Results

Table 4.1 : List of Parameters

User ID	Arrival time	Process time	Deadline	I/O request	memrequest	Typeofservice
1	9	3	18	4	30876	0
2	97	10	112	4	4540	0
3	91	8	109	3	1142	1
4	93	7	108	3	12551	1
5	17	8	26	1	1477	0
6	82	7	93	4	1102	0
7	38	8	54	0	15672	0
8	64	8	80	1	21750	1
9	16	2	23	4	10892	1
10	22	8	33	2	22370	1

Table 4.1: values of Arrival time, Process time, Deadline, I/O request, memrequest and Typeofservice are computed to schedule requests of users.

Table 4.2 : List of Parameters (sort according to memory)

User ID	Arrival time	Process time	Deadline	I/O request	memrequest	Typeofservice
6	82	7	93	4	1102	0
3	91	8	109	3	1142	1
5	17	8	26	1	1477	0
2	97	10	112	4	4540	0
9	16	2	23	4	10892	1
4	93	7	108	3	12551	1
7	38	8	54	0	15672	0
8	64	8	80	1	21750	1
10	22	8	33	2	22370	1
1	9	3	18	4	30876	0

Table 4.3 : user process requests are sorted according to memory.

Cloud no.	Virtual machine no. associated with cloud							
	1	2	3	4	5	6	7	8
1	12847	3777	10250	9630	1124	32000	32000	32000
2	32000	32000	32000	32000	32000	32000	32000	32000
3	32000	32000	32000	32000	32000	32000	32000	32000
4	32000	32000	32000	32000	32000	32000	32000	32000
5	32000	32000	32000	32000	32000	32000	32000	32000

Table 4.4 : values of remaining memory after allocation of processes to particular VM's and clouds.

User ID	Cloud	Virtual machine	Arrival time	Start time	Turnaround time	Finish time	deadline	Wait time
6	1	1	82	82	7	89	93	0
3	1	1	91	91	8	99	109	0
5	1	1	17	20	8	28	26	3
2	1	1	97	100	10	110	112	3
9	1	1	16	16	2	18	23	0
4	1	2	93	111	7	118	108	18
7	1	2	38	39	8	47	54	1
8	1	3	64	64	8	72	80	0
10	1	4	22	29	8	37	33	7
1	1	5	9	9	3	12	18	0

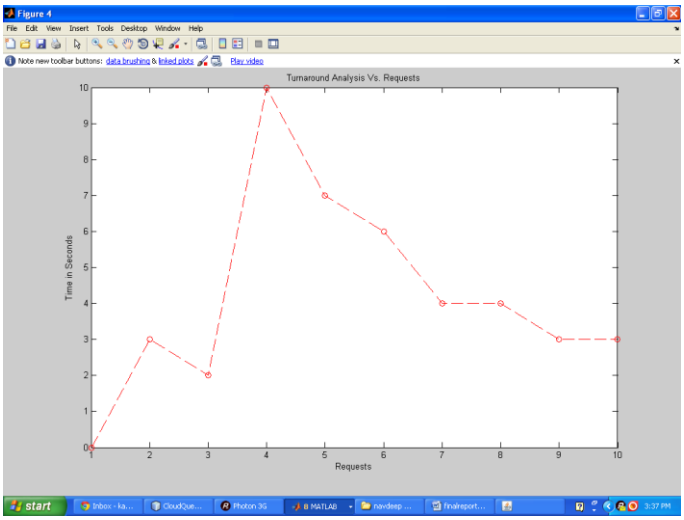

Figure 4.2 : Turn Around Time Analysis

Here figure 4.2 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.

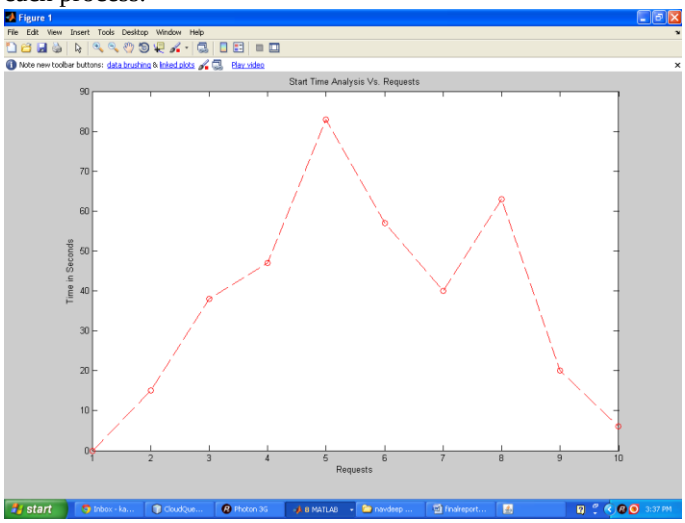

Figure 4.3 : Start Time Analysis

Here figure 4.3 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.

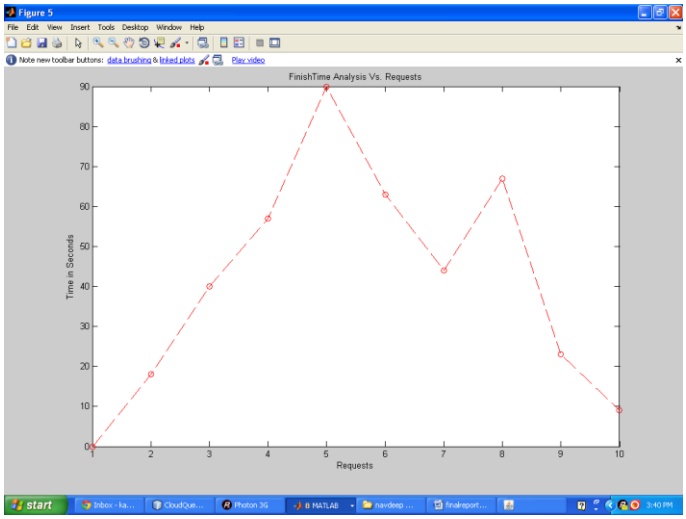

Figure 4.4 : Finish Time Analysis

Here figure 4.4 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.

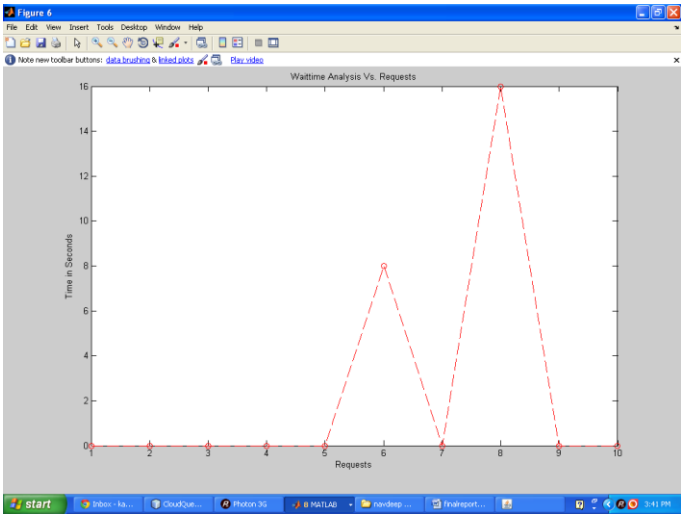

Figure 4.5 : Wait Time Analysis

Here figure 4.5 is showing the Wait 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 wait time of each process. As we can see, most of the processes are executed without any wait. And some are having the nominal wait time.

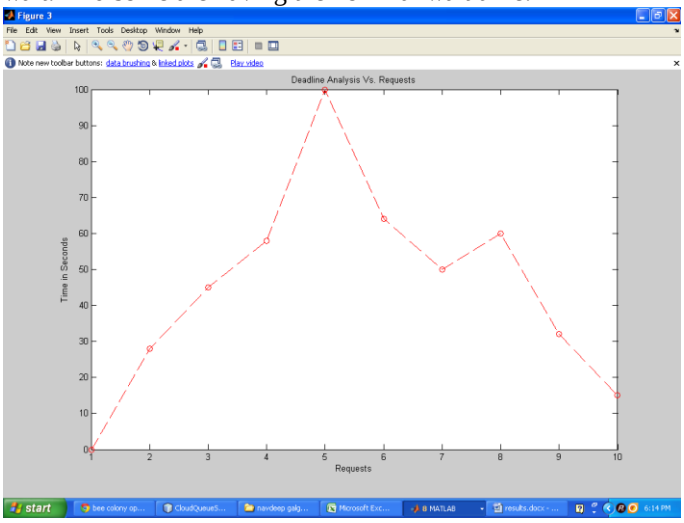

Figure 4.6 : Dead Line Analysis

Here figure 4.6 is showing the DeadLine 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 wait time of each process. As we can see, most of the processes are executed without any wait. And some are having the nominal wait time.

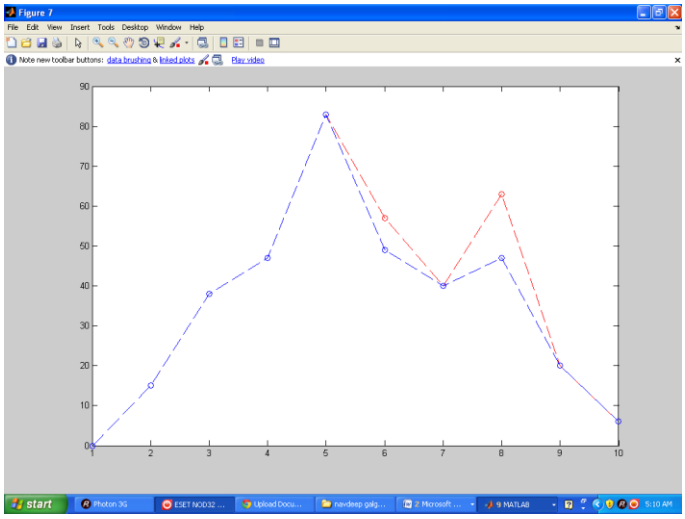

Figure 4.7 : Start Time and Arrival Time Analysis

Here figure 4.7 is showing the Start Time and Arrival Time analysis for 10 processes. Here x-axis represents the number of user requests and y-axis represents the arrival or start time of these processes in seconds. The figure is showing the most of the processes are executed as they are generated by the user i.e. no wait time occur. Only two processes stay in wait queue before the execution.

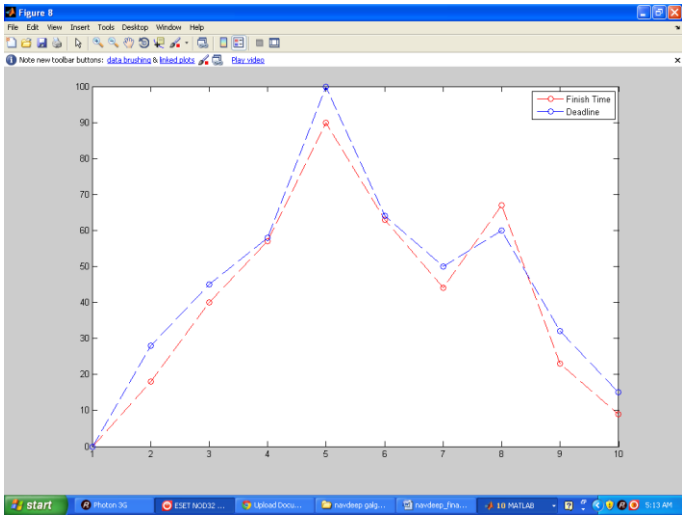

Figure 4.8 : Finish Time and Deadline Analysis

Here figure 4.8 is showing the Finish Time and Deadline analysis for 10 processes. Here x-axis represents the number of user requests and y-axis represents the arrival or start time of these processes in seconds. The figure is showing the most of the processes are executed in limits of dead line.

5. 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.

6. Future work

The presented work is about to perform the scheduling and the allocation of the processes to the clouds in case of under load and overload conditions. In case of over load condition, the migration of the processes is performed from one cloud to other. The Future enhancement of the work are possible in the following directions.

I. The presented work is defined the overload conditions in terms of deadline as well as the memory limit of the Clouds. In future some other parameters can also be taken to decide the migration condition.

II. The presented work is defined for the public Cloud environment, but in future, the work can be extended to private and the hybrid Cloud environment.

7. References

[1] Aaron J. Elmore," Zephyr: Live Migration in Shared Nothing Databases for Elastic Cloud Platforms", SIGMOD'11, June 12–16, 2011, Athens, Greece. ACM 978-1-4503-0661-4/11/06

[2] Anthony T.Velte, Toby J.Velte, Robert Elsenpeter, "Cloud Computing ,A Practical Approach"

[3] Anton Beloglazov," Energy Efficient Allocation of Virtual Machines in Cloud Data Centers", 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing 978-0-7695-4039-9/10 © 2010 IEEE

[4] Anton Beloglazov," Energy Efficient Resource Management in Virtualized Cloud Data Centers", 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing 978-0-7695-4039-9/10© 2010 IEEE

[5] Balaji Viswanathan," Rapid Adjustment and Adoption to MIaaS Clouds", Middleware 2012 Industry Track, December 3-7, 2012, Montreal, Quebec, Canada. ACM 978-1-4503-1613-2/12/12

[6] Baocun Hou," Research on Independent and Dynamic Fault-Tolerant and Migration Technology for Cloud Simulation Resources", GCMS 2011

[7] Behrooz Nobakht," Programming and Deployment of Active Objects with Application-Level Scheduling", SAC'12 March 25-29, 2012, Riva del Garda, Italy. ACM 978-1-4503-0857-1/12/03.

[8] Bogdan Nicolae," A Hybrid Local Storage Transfer Scheme for Live Migration of I/O Intensive Workloads", HPDC'12, June 18–22, 2012, Delft, The Netherlands. ACM 978-1-4503-0805/12/06

[9] Boris Danev," Enabling Secure VM-vTPM Migration in Private Clouds", ACSAC '11 Dec. 5-9, 2011, Orlando, Florida USA ACM 978-1-4503-0672-0/11/12

[10] Christina Delimitrou," Paragon: QoS-Aware Scheduling for Heterogeneous Datacenters", ASPLOS'13, March 16–20, 2013, Houston, Texas, USA. ACM 978-1-4503-1870-9/13/03

[11] Chuliang Weng, Zhigang Wang, Minglu Li, and Xinda Lu,"The Hybrid Scheduling Framework for Virtual Machine Systems", 2009 ACM

[12] Claris Castillo," Resource Co-Allocation for Large-Scale Distributed Environments", HPDC'09, June 11–13, 2009, Munich, Germany. ACM 978-1-60558-587-1/09/06

[13] Chen, Cloud Computing Research and Security Issues, 2010

[14] D Dutta," A Genetic –Algorithm Approach to Cost-Based Multi-QoS Job Scheduling in Cloud Computing Environment", International Conference and Workshop on Emerging Trends in Technology (ICWET 2011) – TCET, Mumbai, India ICWET'11, February 25–26, 2011, Mumbai, Maharashtra, India. ACM 978-1-4503-0449-8/11/02

[15] Damien Borgetto," Energy-efficient and SLA-Aware Management of IaaS Clouds", e-Energy 2012, May 9-11 2012, Madrid, Spain. ACM 978-1-4503-1055-0/12/05

[16] Dan Miao," Resource Allocation for Cloud-based Free Viewpoint Video Rendering for Mobile Phones", MM'11, November 28–December 1, 2011, Scottsdale, Arizona, USA. ACM 978-1-4503-0616-4/11/11

[17] H Mehta," Decentralized Content Aware Load Balancing Algorithm for Distributed Computing Environments", International Conference and Workshop on Emerging Trends in Technology (ICWET 2011) – TCET, Mumbai, India ICWET'11, February 25–26, 2011, Mumbai, Maharashtra, India. ACM 978-1-4503-0449-8/11/02

[18] Hadi Goudarzi," SLA-based Optimization of Power and Migration Cost in Cloud Computing", 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing 978-0-7695-4691-9/12© 2012 IEEE

[19] Hemant Kumar Mehta," Performance Enhancement of Scheduling Algorithms in Clusters and Grids using Improved Dynamic Load Balancing Techniques", WWW 2011, March 28–April 1, 2011, Hyderabad, India. ACM 978-1-4503-0637-9/11/03

[20] Herald Kllapi," Schedule Optimization for Data Processing Flows on the Cloud", SIGMOD'11, June 12–16, 2011, Athens, Greece. ACM 978-1-4503-0661-4/11/06

[21] Hui Kang," Enhancement of Xen's Scheduler for MapReduce Workloads", HPDC'11, June 8–11, 2011, San Jose, California, USA. ACM 978-1-4503-0552-5/11/06

[22] Hwanju Kim," Task-aware Virtual Machine Scheduling for I/O Performance", VEE'09, March 11–13, 2009, Washington, DC, USA. ACM 978-1-60558-375-4/09/03

[23] Ilango Sriram, Ali Khajeh-Hosseini ,"Research Agenda in Cloud Technologies",2010

[24] J. Brandt," Using Cloud Constructs and Predictive Analysis to Enable Pre-Failure Process Migration in HPC Systems", 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing 978-0-7695-4039-9/10© 2010 IEEE

[25] Jeongseob Ahn, Changdae Kim, Jaeung Han, Young-ri Choi†, and Jaehyuk Huh,"Dynamic Virtual Machine

Scheduling in Clouds for Architectural Shared Resources", 2011 IEEE
[26] Jia Rao," Self-Adaptive Provisioning of Virtualized Resources in Cloud Computing", SIGMETRICS'11, June 7–11, 2011, San Jose, California, USA. ACM 978-1-4503-0262-3/11/06.
[27] Jie Zheng," Workload-Aware Live Storage Migration for Clouds", VEE'11, March 9–11, 2011, Newport Beach, California, USA. ACM 978-1-4503-0501-3/11/03.
[28] Jie Zheng," Workload-Aware Live Storage Migration for Clouds", VEE'11, March 9–11, 2011, Newport Beach, California, USA. ACM 978-1-4503-0501-3/11/03
[29] Jithesh Sathyan," Job Management in Mobile Grid Computing", MoMM2009, December 14–16, 2009, Kuala Lumpur, Malaysia. ACM 978-1-60558-659-5/09/0012
[30] Junliang Chen," Tradeoffs between Profit and Customer Satisfaction for Service Provisioning in the Cloud", HPDC'11, June 8–11, 2011, San Jose, California, USA. ACM 978-1-4503-0552-5/11/06
[31] Kejiang Ye," VC-Migration: Live Migration of Virtual Clusters in the Cloud", 2012 ACM/IEEE 13th International Conference on Grid Computing 1550-5510/12 © 2012 IEEE
[32] Kento Sato," A Model-Based Algorithm for Optimizing I/O Intensive Applications in Clouds using VM-Based Migration", 9th IEEE/ACM International Symposium on Cluster Computing and the Grid 978-0-7695-3622-4/09© 2009 IEEE
[33] Loganayagi.B, S.Sujatha,"Enhanced Cloud Security by Combining Virtualization and Policy Monitoring Techniques", 2011 Published by Elsevier Ltd.
[34] Madhan Kumar Srinivasan," State-of-the-art Cloud Computing Security Taxonomies -A classification of security challenges in the present cloud computing environment", ICACCI '12, August 03 - 05 2012, CHENNAI, India ACM 978-1-4503-1196-0/12/08
[35] Marcos Dias de Assuncao," Evaluating the Cost-Benefit of Using Cloud Computing to Extend the Capacity of Clusters", HPDC'09, June 11–13, 2009, Munich, Germany.
[36] Masaya Yamada, Yuki Watanabe, Saneyasu Yamaguchi,"An Integrated 1/0 Analyzing System for Virtualized Environment,2011
[37] Michael Menzel," CloudGenius: Decision Support for Web Server Cloud Migration", WWW 2012, April 16–20, 2012, Lyon, France. ACM 978-1-4503-1229-5/12/04.
[38] Minqi Zhou, Security and Privacy in Cloud Computing: A Survey, 2010
[39] Mohammad Hajjat,'Cloudward Bound: Planning for Beneficial Migration of Enterprise Applications to the Cloud", SIGCOMM'10, August 30–September 3, 2010, New Delhi, India. ACM 978-1-4503-0201-2/10/08
[40] Muhammad Ali Babar," A Tale of Migration to Cloud Computing for Sharing Experiences and Observations", Waikiki, Honolulu, HI, USA. ACM 978-1-4503-0582-2/11/05
[41] Navendu Jain," Near-Optimal Scheduling Mechanisms for Deadline-Sensitive Jobs in Large Computing Clusters", SPAA'12, June 25–27, 2012, Pittsburgh, Pennsylvania, USA. ACM 978-1-4503-1213-4/12/06
[42] Omer Khalid, Ivo Maljevic, Richard Anthony, Miltos Petridis, Kevin Parrott, Markus Schulz, "Dynamic Scheduling of Virtual Machines Running HPC Workloads in Scientific Grids" 2009 IEEE.
[43] Omer Khalid, Ivo Maljevic, Richard Anthony, Miltos Petridis, Kevin Parrott, Markus Schulz,"Deadline Aware Virtual Machine Scheduler for Grid and Cloud Computing",2010 IEEE
[44] Patrizia Scandurra," A Layered Coordination Framework for Optimizing Resource Allocation in Adapting Cloud-based Applications", SAC'12 Mar 26-30, 2012, Riva (Trento), Italy ACM 978-1-4503-0857-1/12/03
[45] Ripal Nathuji," Q-Clouds: Managing Performance Interference Effects for QoS-Aware Clouds", EuroSys'10, April 13–16, 2010, Paris, France. ACM 978-1-60558-577-2/10/04
[46] Ruijin Zhou," Optimizing Virtual Machine Live Storage Migration in Heterogeneous Storage Environment", VEE'13, March 16–17, 2013, Houston, Texas, USA. ACM 978-1-4503-1266-0/13/03
[47] Ristenpart, 199-212. Electronic Edition • CiteSeerX • Google scholar, 2009
[48] Samer Al-Kiswany," VMFlock: Virtual Machine Co-Migration for the Cloud", HPDC'11, June 8–11, 2011, San Jose, California, USA. ACM 978-1-4503-0552-5/11/06
[49] Sean Barker," "Cut Me Some Slack": Latency-Aware Live Migration for Databases", EDBT 2012, March 26–30, 2012, Berlin, Germany. ACM 978-1-4503-0790-1/12/03
[50] Shigeru Imai," Elastic Scalable Cloud Computing Using Application-Level Migration", 2012 IEEE/ACM Fifth International Conference on Utility and Cloud Computing 978-0-7695-4862-3/12 © 2012 IEEE
[51] Sudipto Das," Albatross: Lightweight Elasticity in Shared Storage Databases for the Cloud using Live Data Migration", The 37th International Conference on Very Large Data Bases, August 29th September 3rd 2011, Seattle, Washington. Proceedings of the VLDB Endowment, Vol. 4, No. 8 VLDB Endowment 21508097/11/05
[52] Sumit Kumar Bose," CloudSpider: Combining Replication with Scheduling for Optimizing Live Migration of Virtual Machines Across Wide Area Networks", 2011 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing 978-0-7695-4395-6/11© 2011 IEEE
[53] Schridde, et al An Identity-Based Security Infrastructure for Cloud Environments, 2010
[54] Takahiro Hirofuchi," Enabling Instantaneous Relocation of Virtual Machines with a Lightweight VMM Extension", 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing 978-0-7695-4039-9/10© 2010 IEEE
[55] Timothy Wood," CloudNet: Dynamic Pooling of Cloud Resources by Live WAN Migration of Virtual Machines", VEE'11, March 9–11, 2011, Newport Beach, California, USA. ACM 978-1-4503-0501-3/11/03
[56] Van Tran," Application Migration to Cloud: A Taxonomy of Critical Factors", SECLOUD '11, May 22, Waikiki, Honolulu, HI, USA ACM 978-1-4503-0582-2/11/05
[57] Wes Lloyd," Migration of Multi-tier Applications to Infrastructure-as-a-Service Clouds: An Investigation Using Kernel-based Virtual Machines".
[58] WANG, A Key-policy Attribute-based Encryption Scheme with Constant Size Ciphertext, 2012
[59] WANG, Chang-Ji WANG, A Key-policy Attribute-based Encryption Scheme with Constant Size Ciphertext, 2010
[60] Xiaoli Wang, Yuping Wang ,Hai Zhu,"Energy-Efficient Multi-Job Scheduling Model for Cloud Computing and Its

Genetic Algorithm", Hindawi Publishing Corporation Mathematical Problems in Engineering, Volume 2012
[61] Xiulei Qin," Optimizing Data Migration for Cloud-based Key-Value Stores", CIKM'12, October 29–November 2, 2012, Maui, HI, USA. ACM 978-1-4503-1156-4/12/10
[62] Xu Cheng," Load-Balanced Migration of Social Media to Content Clouds", NOSSDAV'11, June 1–3, 2011, Vancouver, British Columbia, Canada. ACM 978-1-4503-0752-9/11/06
[63] Xudong Wang," Migration and Execution of JavaScript Applications between Mobile Devices and Cloud", SPLASH'12, October 19–26, 2012, Tucson, Arizona, USA. ACM 978-1-4503-1563-0/12/10.
[64] Xun Xu, "From cloud computing to cloud manufacturing", 2011 Elsevier Ltd
[65] Yanbin Liu, Norman Bobroff, Liana Fong, Seetharami Seelam, Javier Delgado,"New Metrics for Scheduling Jobs on a Cluster of Virtual Machines" , 2010
[66] Yang Ge," A Game Theoretic Resource Allocation for Overall Energy Minimization in Mobile Cloud Computing System", ISLPED'12, July 30–August 1, 2012, Redondo Beach, California, USA. ACM 978-1-4503-1249-3/12/07
[67] Ye Hu," Resource Provisioning for Cloud Computing".
[68] Zheng Hu, Kaijun Wu, Jinsong Huang,"An Utility-Based Job Scheduling Algorithm for Current Computing Cloud Considering Reliability Factor",2012 IEEE

Author Profile

Navdeep Singh received B.E. degree in Computer Science and Engineering from All Saints College of Technology, Bhopal, M.P., India (University Rajiv Gandhi Proudyogiki Vishwavidyalaya) and Perusing M.Tech degree in Computer Science and Engineering from Galgotias University, Greater Noida, U.P., India.