Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 4, April 2015, pg.863 – 872

RESEARCH ARTICLE

AN ALGORITHMIC APPROACH FOR CLOUD SERVER ALLOCATION IN PUBLIC CLOUD ENVIRONMENT

Ms. ALANKRITA AGGARWAL
Assistant Professor, Department of Computer Science & Engineering, HCTM Technical Campus, Kaithal, India, Email-id: alankrita.agg@gmail.com

RAJJU
Research Scholar M.Tech, Department of Computer Science & Engineering, HCTM Technical Campus, Kaithal, India, Email-id: 17122063rajju@gmail.com

ABSTRACT: **Cloud computing serves as the service provider globally connected, which provides service to the cloud users. The processing of the users request plays an important role in the cloud services. In order to achieve the effective service execution with lesser migration we have proposed an effective resource allocations scheme is represented in distributed cloud network. The work is defined on multiple cloud system that is connected to public user environment. User can perform the resource request and based on the request analysis that particular cloud server or service is allocated to the user. This allocation process is controlled in terms of scheduling or requests so that maximum cloud service utilization will be performed and the migration chances will be reduced. In this work, the cloud servers are defined with integrated virtual machines along with memory and resource specification. the client side requests are also scheduled initially under process time, security and dead line criticality. Based on these vectors and virtual machine capabilities, the initial level request allocation will be performed. In the later stage, the request execution on cloud server will be done under dead line criticality. If the cloud service cannot perform the request execution by its dead line, the migration is performed to other effective VM.**

Keywords: **Cloud, Scheduling, Migration, Cost.**

I. INTRODUCTION

Cloud computing has made revolution in computing as a services. With the ability to provide on-demand computing resources dynamically, with this new way of computing technology there are lots of benefits for the users such as User-centric access On-Demand service provisioning, QoS guaranteed offer, Autonomy, Scalability & flexibility [1]. Definition of cloud computing provided by the US National institute of standards and technology (NIST):
"Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, application, and services) that can be rapidly provisioned and released with minimal management effort or services provider interaction." [10].

Cloud computing service models can generally be classified into three categories: Infrastructure as a service (Iaas), platform as a service (PaaS), Software as a Service (SaaS) [12]. Scalability is a prominent quality for all these categories of system. There are mainly three deployment models of cloud computing: private cloud, public cloud and hybrid cloud [9].
Virtualization technology makes the independence of applications and servers feasible. Nowadays, computing systems heavily rely on this technology. Virtualization technology provides a new way to improve the power efficiency of the datacenters i.e., (server) consolidation, which enables the assignment of multiple virtual machine to a single physical server [2]. Virtualization is a core technique to implement the cloud computing paradigm. Virtualization provides an abstraction of hardware resources enabling multiple instantiations of operating systems to run simultaneously on a single physical machine. Another prominent advantage of the virtualization is the live migration technique which refers to the act of migrating a virtual machine from one physical machine to another even as the virtual machine continues to execute [3].

Scheduling in distributed computing is an intensely studied problem. It deals with the problem of assigning tasks, sometimes of different types, to a set of resources, sometimes with different characteristics. The tasks can be with or without dependencies, IO intensive or computationally intensive [8]. Co-scheduling defines a process of execution of more than one process simultaneously on different processors. In a cloud based system, this scheduling approach is quite common. According to a simple classification, job scheduling algorithms in cloud computing can be categorized into two main groups; Batch mode heuristic scheduling algorithms (BMHA) and online mode heuristic algorithms.

II. RELATED WORK

Scheduling is a complex challenge in cloud computing. The major factors that decide the effective scheduler is to reduce the cost under various circumstances. Cloud computing is a non-static environment it difficult to maintain a load balance in the system. The author **Pinal Salot** [11] describes FCFS, SJF, Round Robin, Min-Min and Max-Min algorithm in this regard. In the First Come First Serve job scheduling the jobs are queued in the order of which come first. In

Round Robin job scheduling jobs are dispatched in FCFS logic and the time slice of the process decide the allocation. In the Min-Min scheduling algorithm small jobs are executed first, where large jobs are waiting for more time. The Max–Min job scheduling algorithm they select the largest job to be executed first, later the small jobs are executed and takes long time. In the Most fit task scheduling algorithm select the best fit job executed first, failure to select opt job. Priority job scheduling algorithm each job is executed based on the priority. In Shortest job first scheduling the task based on priority given to small jobs.

The author **Hwanju Kim [5]** defined a task aware scheduling scheme so that effective I/O performance over the system was achieved. In this paper a scheduling approach under the I/O boundation is defined and on incoming events. **Takahiro Hirofuchi [6]** proposed an advanced live migration mechanism enabling instantaneous relocation of VMs. To minimize the time needed for switching the execution host, memory pages are transferred after a VM resumes at a destination host. **Zhongni Zheng [7]** proposed an optimized scheduling algorithm to achieve the optimization or sub-optimization for cloud scheduling problems. Author investigated the possibility to place the Virtual Machines in a flexible way to improve the speed of finding the best allocation on the premise of permitting the maximum utilization of resources. **Damien borgetto [8]** proposed an integrated approach for VM migration and reconfiguration, and PM power management. The author incorporates an autonomic management loop, where proactive actions are suggested for all three areas in a hierarchically structured way. This approach is evaluated with both, synthetic workload data and real-word monitoring data of a Next Generation Sequencing (NGS) application used for the protein folding in the bioinformatics area.

III. PROPOSED WORK

In this presented work a three layer cloud server allocation and scheduling model for public cloud environment is presented. In the first layer, user request will be performed to the available cloud servers and it will identify the effective cloud based on the perfect match, availability and the response time. Once the most appropriate, cloud servers will be identified, the next work will be to perform process scheduling on the high priority clouds. The scheduling mechanism will order the multiple requests and allocate the clouds under request analysis. In third layer, the cost analysis of the process will be done on particular cloud under the deadline criticality. If the process not gets executed on the cloud, the migration of the process will be done. The proposed model is given as under

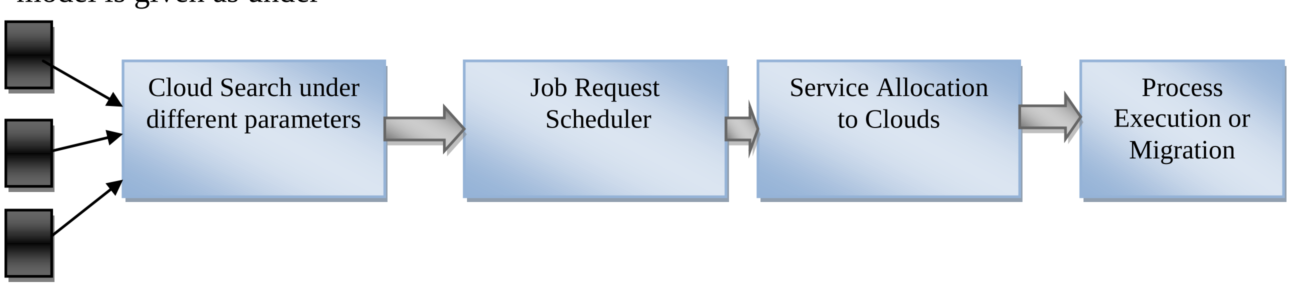

Figure 1.1: Scheduling Architecture

The scheduling process suggested in this work will be defined under three main parameters shown in figure 1.2.

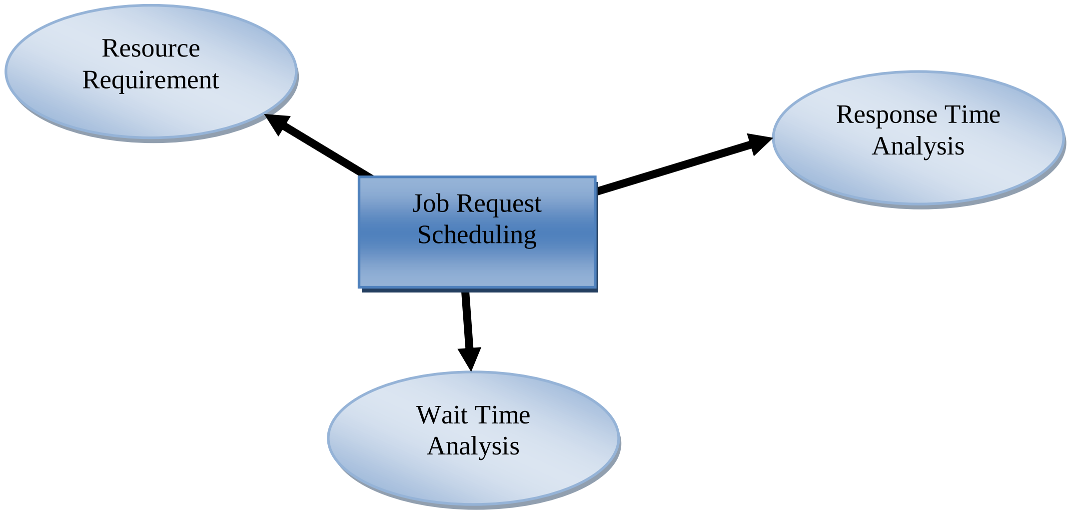

Figure 1.2: Factors for Low level scheduling

Just after the high level scheduling, the low level scheduling will be done. Based on this scheduling the process execution cost will be analyzed. Once the process cost will be obtained, the next work is to analyze the processes based on the virtual machine capacity. On the capacity analysis, the analysis of the wait time and capacity vector will be done and identify weather the allocated virtual machine is capable to execute the process or not. If it is capable, the process will be executed by the particular cloud otherwise the process migration is required.

IV. ALGORITHM OF PROPOSED WORK

Algorithm(CloudServerEnvironment)
/*Cloud Server is the Global Distributed Environment in which client and server are integrated for user request Processing*/
{
1. For i=1 to Count(Cloud)
   [Process All Cloud]
   {
2. Set Cloud(i).Security=Random
   [Assign Security Constraint to Each Cloud]
   }
3. Arrange All Clouds under Security Aspects
4. For i=1 to Count(VMS)
   [Process All Virtual Machines]
   {
5. Set the Capacity, Criticality and Availabilty Parameter on Each VM

}

6. Generate N User Request under Arrival Time, DeadLine, Process Time and Resource Request Parameters
7. For i=1 to N
   [Process All Requests]
   {
8. Cost(i)=Securityreq(i)*Weight1+DeadLine(i)*weight2+ProcessTime(i)*Weight3
   }
9. Arrange the User Request under cost parameter.
10. Group the VMs under similar Cost Analysis
11. Perform the Initial Allocation of Each User Request to Particular VM Group.
12. Perform the Second Level Scheduling under VMLoad, Process Time, Arrival Time and DeadLine Parameters
13. Identify the Request that cross DeadLine Criticality and Mark them as Migration Required Requests
14. Migrate the Mark Request on Other Server

}

V. RESULTS

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 accepts the server side parameters, user side parameter and global parameters that are used in the algorithm. These input parameters are related to the server side as well as to define the user requirements.

Here a cloud service environment is defined with following parameters

- Number of Cloud Server : 6
- Number of VMs : 4
- Number of requests : 10
- Load Threshold : 5

The result analysis is defined under different parameters such as deadline, wait time, process time etc.

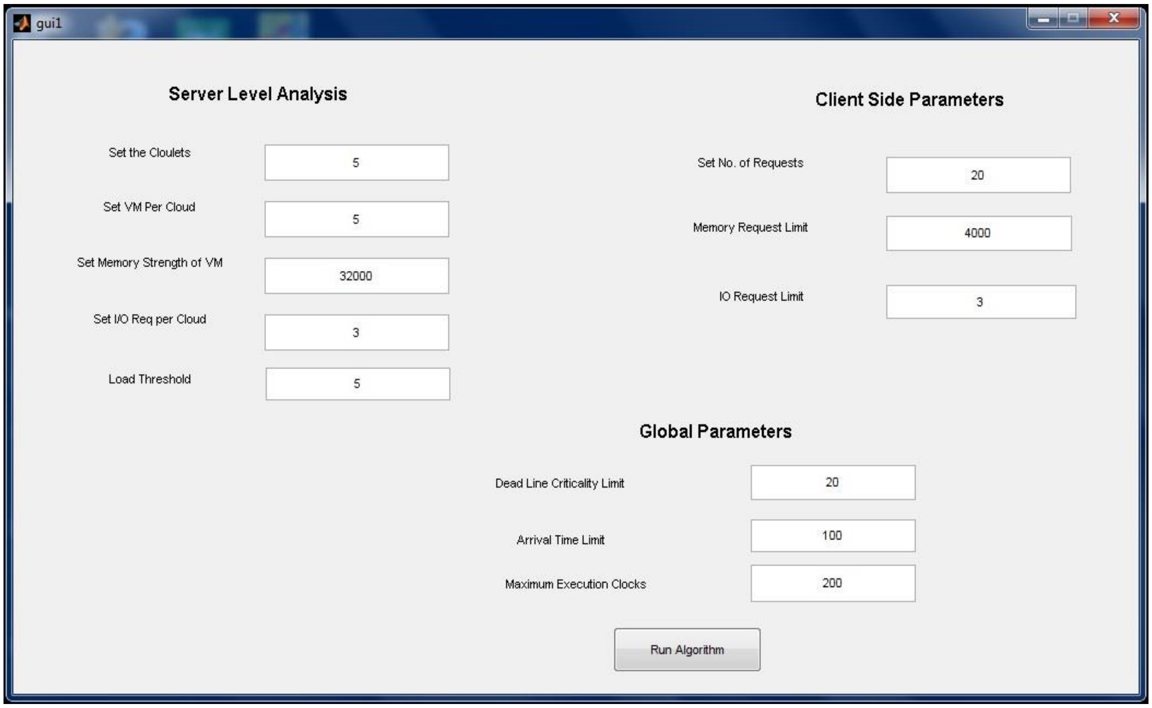

Figure 1.3: GUI of proposed work

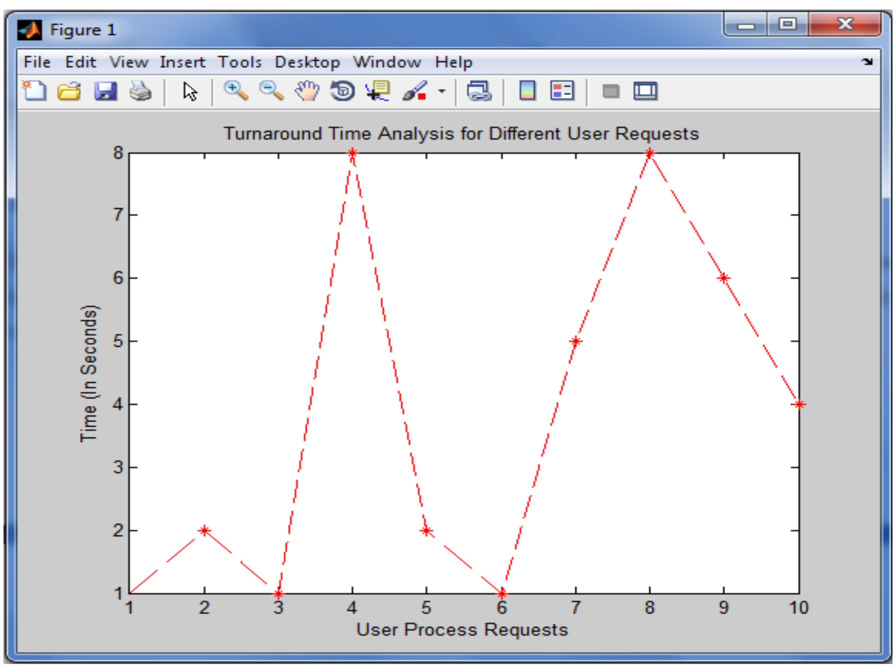

Figure 1.4: Turnaround Time Analysis

Fig. 1.4 shows the turnaround time analysis. 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.

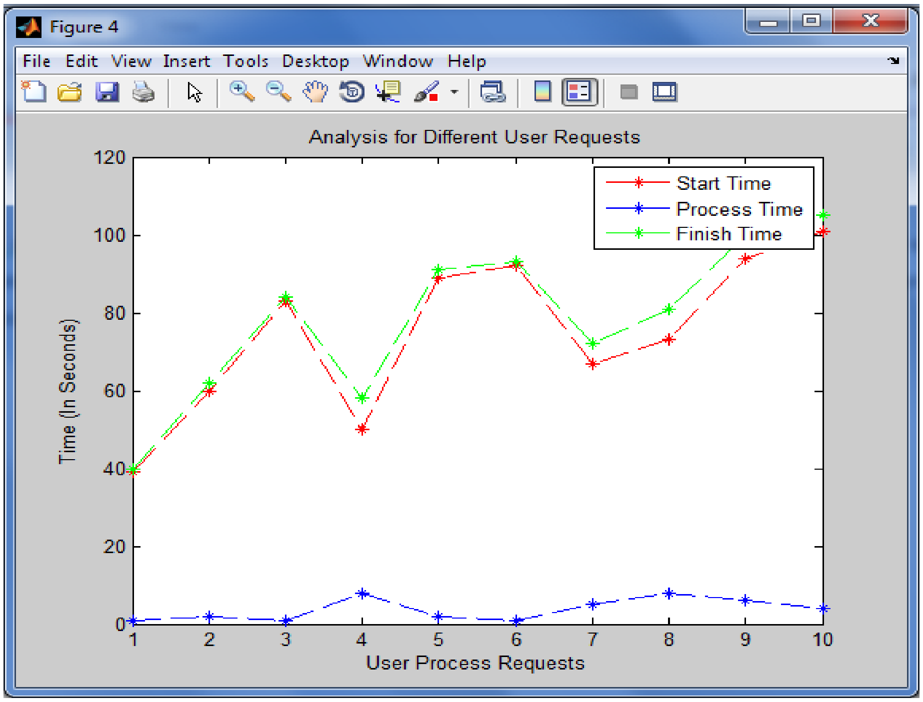

Figure 1.5: Analysis for different user Request

Here fig. 1.5 shows the analysis of user request with start time, process time and finish time. The Here x-axis represents the number of user requests and y-axis represents the time taken by these processes in seconds.

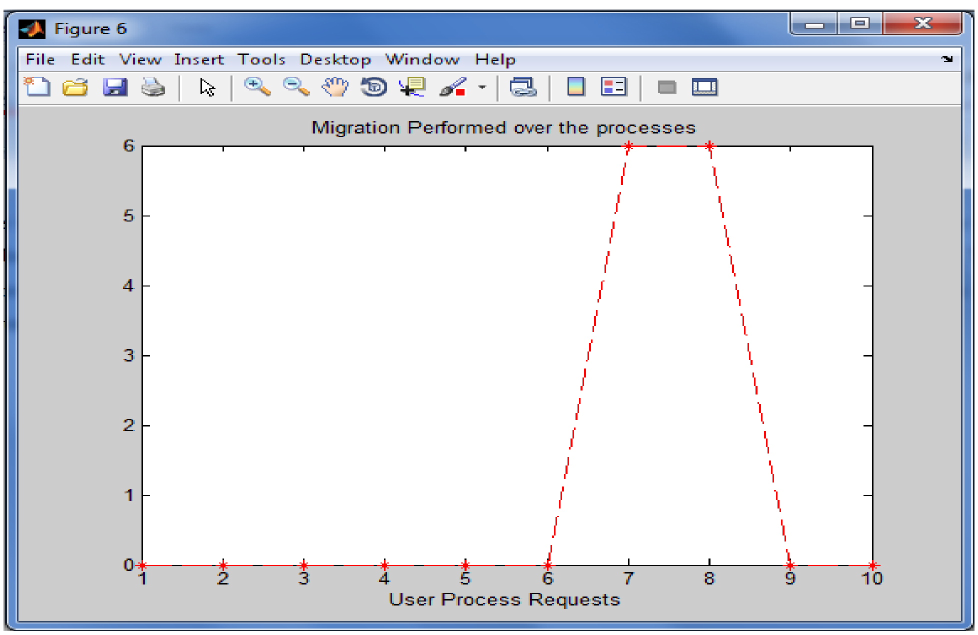

Figure 1.6: Migrated Processes

Here figure 1.6 is showing the list of migrated processes. Here x-axis represents the number of user requests and y-axis represents the clouds. As we can see, three processes 7 and 8 are migrated to cloud 6.

VI. Comparative Analysis

The comparative analysis of the presented work is defined with the base paper work. The anlaysis is here performed to identify the effectiveness of virtual machines. In the existing approach a dynamic programming approach is suggested to perform the process execution on cloud server and to reduce the execution cost and migration cost. The analysis is here performed under the performance parameters obtained in existing and proposed work.

No of Users	Avg.(Pro.)	Worst(Pro.)	Avg.(Exist)	Worst(Exist)
250	.92	1.13	1.15	1.36
500	.96	1.23	1.14	1.23
1000	.89	1.012	1.12	1.2
1500	.96	1.19	1.09	1.21
2000	.87	1.10	1.1	1.24
3000	.97	1.16	1.09	1.19
4000	.92	1.11	1.1	1.18

Table 1.1: Performance Analysis (Existing Vs. Proposed)

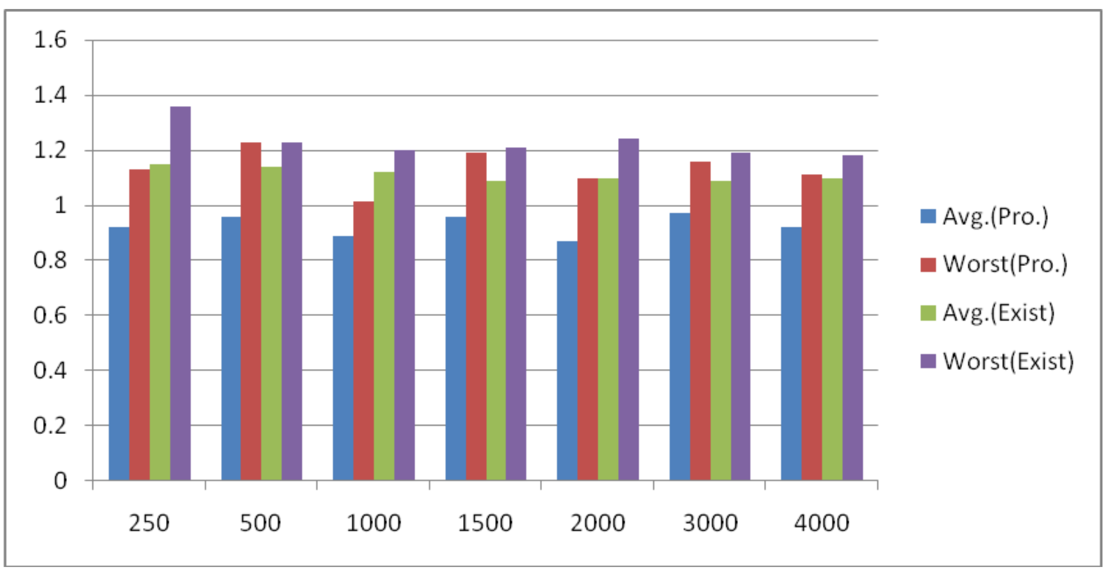

Figure 1.7: Graphical Representation of Comparative Analysis

VII. CONCLUSION

In this present work, an effective cloud service allocation mechanism is defined under multiple parameters. The work is presented on public cloud integrated environment where each cloud is defined with multiple VMs. Each virtual machine is defined under the load, capacity and resource parameters. In same way, the user requests are also defined under the process time parameters. The work requests are performed, the VM and the request mapping is performed under the resource and reliability parameters. Based on this mapping, initial level request assignment to particular cloud is performed. Once the initial level allocation is performed, the next work is to execute the process under deadline criticality on specific VM. If the VM is overloaded and not capable to handle the request, the migration of request is performed to next effective VM. The analysis of these requests is performed under wait time and migration probability parameters. The obtained results show the effective service execution with lesser migration.

VIII. Future Work

The presented work is about to perform the scheduling and the allocation of the processes to the clouds in case of under multiple reliability and efficiency parameters. If the integrated VM is not capable to handle the process in its dead line limits, the migration of the processes is performed from one VM to other. The Future enhancement of the work is possible in the following directions:

1. The presented work is defined the scheduling mechanism under deadline as well as the memory limit of the Clouds. In future some other parameters can also be taken to decide the migration condition.

2. 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.

References

[1] D.Daniel, "A Novel Approach for Scheduling Service Request in Cloud with Trust Monitor", 978-1-61284-653-8/11, 2011 IEEE.

[2] Hadi Goudarzi, Mohammad Ghasemazar, and Massoud Pedram, "SLA-based Optimization of Power and Migration Cost in Cloud Computing", 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing.

[3] Kejiang Ye, Xiaohong Jiang, "VC-Migration: Live Migration of Virtual Clusters in the Cloud", 2012 ACM/IEEE 13th International Conference on Grid Computing.

[4] Florin Pop, Ciprian Dobre, Valentin Cristea, "Scheduling of Sporadic Tasks with Deadline Constrains in Cloud Environments", 2013 IEEE 27th International Conference on Advanced Information Networking and Applications.

[5] 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.

[6] 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.

[7] Zhongni Zheng, "An Approach for Cloud Resource Scheduling Based on Parallel Genetic Algorithm", 978-1-61284-840-2/11, 2011 IEEE.

[8] 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.

[9] R. Santhosh," Pre-emptive Scheduling of On-line Real Time services With Task Migration for Cloud Computing", 2013 IEEE conference on PRIME.

[10] 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.

[11] Pinal Salot," A SURVEY OF VARIOUS SCHEDULING ALGORITHM IN CLOUD COMPUTING ENVIRONMENT",IJRET 2013.