

# A Dynamic Priority Based Self Scheduling of Virtual Machines for Cross-Cloud Computing Environment

Srilatha.M

M. Tech. Computer Science and Engineering  
SIT, JNTU Hyderabad, India

G.Praveen Babu

Associate Professor, Department of CSE  
JNTU, Hyderabad, India

**Abstract**—Cloud computing is the set of distributed computing nodes. The distribution of virtual machine (VM) images to a set of distributed compute nodes based on priority value in a Cross-Cloud computing environment is main issue considered in this paper. This paper will be dealing with the problem of scheduling virtual machine (VM) images to a set of distributed compute nodes in a Cross-Cloud computing environment.

**Keywords:** *Virtualization; cloud computing; virtual machine (VM) image; cross-cloud computing; scheduling, Eucalyptus, Dynamic priority, Cloud.*

## I. INTRODUCTION

Cloud computing is the use of computing resources that area delivered as a service over network.. Cloud computing and virtualization are two of the major IT trends of the decade. The scheduling of virtual machines in a cloud computing environment has become crucial due to the increase in the number of users. This is usually done to load balance a system effectively [1]. There are so many areas of implications provided by cloud computing which includes, enhanced data computation, cost-effective, decrement in operational resources, optimized resource deployment, enhanced software development, mobility in data access, security, etc [2]. Thus to provide cost effective computations, cloud environment needs to be operated in an efficient way.

After virtualization it has been possible to present compute resources in the form of Virtual Machine (VM) Images. Virtualization plays a special role in cloud computing. Typically VMs are offered in different types, each type have its own characteristics which includes number of CPU cores, amount of main memory, etc. and cost. To satisfy the needs of such a high performance computations evolves the use of federated clouds and multi-cloud deployments [5], [6], [7]. A Cross-Cloud computing environment is the interaction of two or more Cloud computing sites offering different computations. If the Cloud paradigm is to be extended to facilitate cross provider utilization of resources, several challenges need to be solved [8]. One of the requirements for Cross-Cloud computing discussed in this paper is the cost effective and secured scheduling of virtual machine images. The

deployment mechanism should be able to cope with both cluster environments and more heterogeneous, distributed Cross-Cloud computing environments, which also includes desktop systems as found in private Clouds. The main purpose of scheduling here is to find a best match of compute resources with the task the job consist of.

This paper is organized as follows. First the literature review on self-scheduling algorithms and dynamic priority scheduling is given in section 2. Then new proposed algorithm based on Heterogeneous self scheduling and dynamic priority based scheduling [3][4] in section 3. Section 4 consist of proposed system and then section 5 concludes the paper with experimental result.

## II. EXISTING SCHEDULING ALGORITHMS

This section describes different scheduling algorithms best suitable for the distributed and highly heterogeneous environment for VM image scheduling.

### A. Self scheduling algorithms

Efficient scheduling on distributed systems such as heterogeneous cluster and grid computing is mostly affected by load balancing. This section presents a general approach toward different self scheduling schemes. Several best performing self scheduling schemes are reviewed here as below:

**Guided Self-Scheduling (GSS):** It is characterized by its scheduling strategy. It can dynamically change the number of iterations assigned to each processor [8].

It sometimes offer good load balance with small scheduling overhead, by assigning large size chunks at the beginning of a parallel loop.

**Factorizing Self-Scheduling (FSS):** Factorizing self-scheduling results from statistical analysis. Tasks are scheduled in batches of P chunks of equal size.

FSS offers better load balance as compared to the GSS when execution time changes widely and randomly [9].

**Trapezoid Self-Scheduling (TSS)** It maintains reasonable load balance while trying to reduce the need for synchronization [9]. It linearly decreases the chunk size. Large chunks of iterations are allocated to the first few processors and successively smaller chunks to the last few processors.

B. Dynamic priority scheduling

It schedules the VMs to the nodes depending upon their priority value, which varies dynamically based on their load factor. This dynamic priority concept leads to better utilization of the resources. Priority of a node is assigned depending upon its capacity and the load factor [4].

Computing Node	Priority
1	2
2	3
3	4
4	1

Before arrival of new request

III. PROPOSED ALGORITHM

Dynamic priority Self Scheduling:

At Scheduler:

- Accept VM request of user at scheduler
- Scheduler will detect corresponding servers
- Partition  $\alpha\%$  of the load to computing node and assign priority values to each node based on following measures.
  - If ( P1 not set)
    - {
    - P1= max available resource node with max through put, less response time and waiting time.
    - If (P1 is turned off)
    - TURN ON P1.
    - If( load factor of P1 < Threshold value)
    - Assign VM to P1.
    - If( P2 is set and load factor < Threshold value)
    - Swap P1 and P2.
    - Assign VM to P1.
    - }
  - Assign remaining load to different computing nodes based on above measure.

Example:

Computing Node	Priority
1	1
2	3
3	4
4	2

Before arrival of new request

IV. PROPOSED ARCHITECTURE

Fig 1. Cross cloud computing Environment

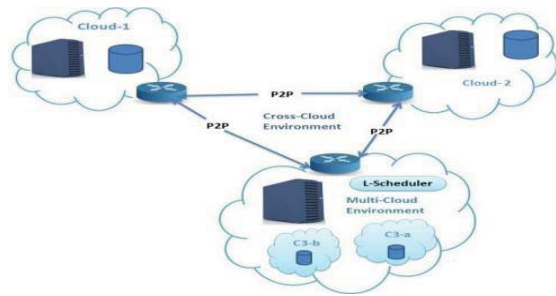
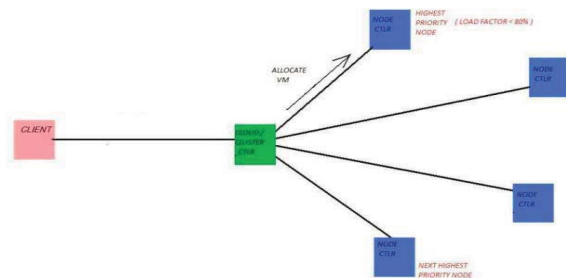
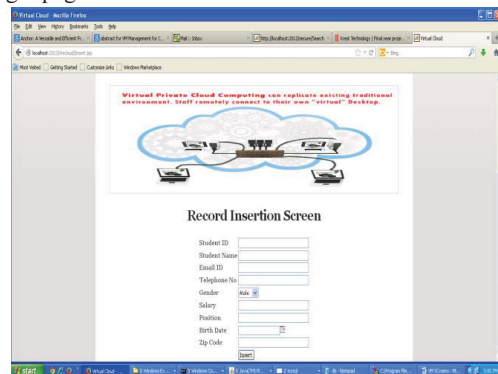


Fig 2. VM allocated to Highest Priority node

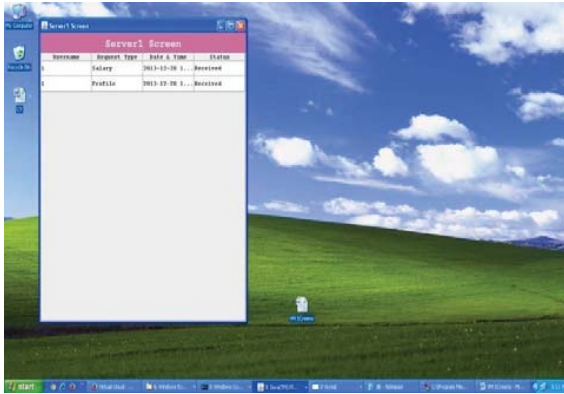


V. EXPERIMENTAL RESULT

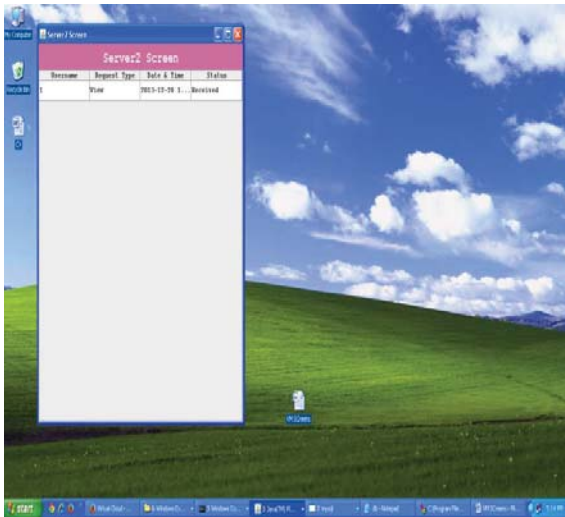
Login page



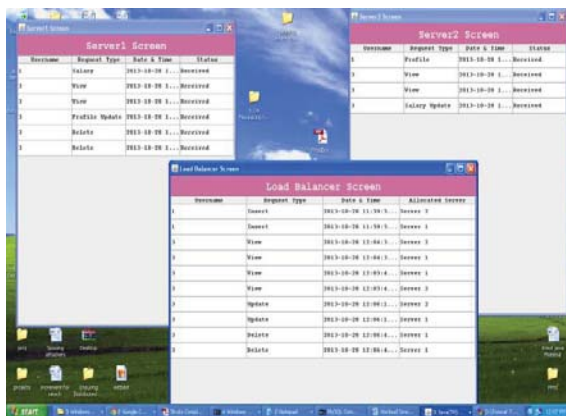
Server1 screen



Server 2 screen:



Load Balance Screen:



In this paper an existing loop scheduling algorithms and dynamic priority scheduling are presented. From previous study a dynamic priority Self-Scheduling algorithm is proposed and considered for the implementation for virtual machine scheduling in cross-cloud environment.

This scheduling algorithm load factor is kept constant for independent tasks. Further extension to this algorithm can be done by varying the maximum load factor with task dependency scheduling.

References

- [1] Scheduling: <http://en.wikipedia.org/wiki/Scheduling> (computing)
- [2] Ammar Khalid, Husnain Mujtaba, "Data Processing Issues in Cloud Computing", Second International Conference on Machine Vision, 2009.
- [3] Deepti Teng, K.N.Hande "VM management on cross cloud computing", International conference on communication systems and Network Technologies.
- [4] An Adaptive Alogtithmm for Dynamic priority based Virtual Machine scheduling in cloud", IJCSI issues, Vol 9, Issue 6, No 2, Nov-2012
- [5] R. Moreno-Vozmediano, R. S. Montero, I. M. Llorente, "Multicloud Deployment of Computing Clusters for Loosely Coupled MTC Applications", IEEE Transaction on Parallel and Distributed Systems, Vol 22, no 6, pp. 924-930, June, 2011.
- [6] B. Rochwerger, D. Beritgand, E. Levy, A. Galis and others, "The Reservoir Model and Architecture for Open Federated Cloud Computing", IBM Journal of Research and Development, Vol 53, no 4, pp. 1-11, July, 2009.
- [7] Jose Luis Lucas Simarro, Rafael Moreno-Vozmediano, Ruben S. Montero and I. M. Llorente, "Dynamic placement of virtual machines for cost optimization in multi-cloud environment", International Conference on High Performance Computing and Simulation (HPCS), 2011, pp. 1-7.
- [8] Matthias Schmidt, Niels Fallenbeck, Matthew Smith, Bernd Freisleben, "Efficient Distribution of Virtual Machines for Cloud Computing", 18th Euromicro Conference on Parallel, Distributed and Network-based Processing, 2010.
- [8] Polychronopoulos, C.D. and D.J. Kuck, "Guided self-scheduling: A practical scheduling scheme for parallel supercomputers", 1987, IEEE Computer Society. p. 1425-1439.
- [9] Yang, C.T., K.W. Cheng, and K.C. Li, "An Efficient Parallel Loop Self- Scheduling Scheme for Cluster Environments", Journal of Supercomputing, 2005. 34: p. 315-335.
- [9] Yang, C.T. and S.C. Chang, "A Parallel Loop Self-Scheduling on Extremely Heterogeneous PC Clusters", Journal of Information Science and Engineering, 2004. 20(2): p. 263-273.