

## Efficient Job Provisioning for a Cloud Service Provider

Dharmvir Sharma<sup>1,a</sup>, Tejinderpal Singh<sup>2</sup>

<sup>1</sup>Student, CSE Department, Yadavindra College of Engineering, Talwandi Sabo

<sup>2</sup>Assistant professor, CSE Department, Yadavindra College of Engineering, Talwandi Sabo

**Abstract.** Cloud Computing is a very fast emerging technology as every enterprise is moving fast towards this system. Cloud Computing is known as a provider of dynamic services. It optimizes a very large, scalable and virtualized resource. So lots of industries have joined this bandwagon nowadays. One of the major research issues is to maintain good Quality of Service (QoS) of a Cloud Service Provider (CSP). The QoS encompasses different parameters, like, smart job allocation strategy, efficient load balancing, response time optimization, reduction in wastage of bandwidth, accountability of the overall system, etc. The efficient allocation strategy of the independent computational jobs among different Virtual Machines (VM) in a Data center (DC) is a distinguishable challenge in the Cloud Computing domain and finding out an optimal job allocation strategy guided by a good scheduling heuristic for such an environment is a mape-k loop problem. So different heuristic approaches may be used for better result and in this result we paper we implement worst fit in mape-k and evaluated the results.

### 1 Introduction

Cloud gives the administrations to the customer on the web. Cloud administrations agree to people and organizations to make utilization of equipment and programming that are over seen by outsiders at far-away areas. Illustrations of cloud administrations incorporate long range informal communication destinations, webmail, online business applications and online document stockpiling, Email customers like Yahoo! furthermore, Gmail, Wikipedia, YouTube, Skype or Bit Torrent and some more. Cloud has Pay per use measured administration and Cloud provider uses this metered service according to their utilization. Websites like Hurray, Email, Gmail, or Hotmail and so on [1]. Millions of resources are submitted to computing task to cloud system. It is challenge to cloud computing environment.

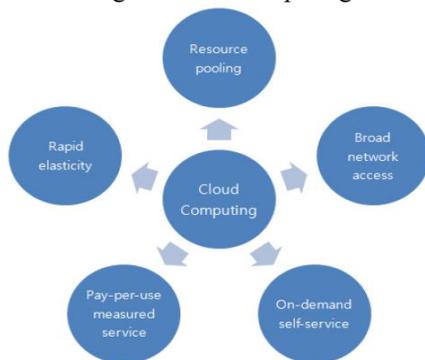


Figure 1. shows the services at clouds.

Different characteristics like On-demand self service, Ubiquitous network access, Location independent resource pooling, Rapid elasticity, Pay per use [2] and scheduling strategies are used. Cloud computing have three categorized of service [1]:

- 1). SaaS (software as a service)
- 2). PaaS (platform as a service)
- 3). IaaS (infrastructure as a service)

Allocation of resources not guaranteed Quality of service (QoS). Constraints clients via service level agreements (SLAs). Virtualization is the most essential procedure. With the help of virtualization, we can make version of a server, storage, an operating system or resources of system [3]. The cloud computing is a distributed computing system and it is a collection of virtualized and interconnected machines that provides many resources to the consumers depending upon the SLA (Service Level Agreement). Cloud suppliers give computational assets in the variety of virtual machines (VMs) to consumers on through appropriate geographically distributed data centres. According to consumer request of resources like virtual machines, storage and applications cloud distributes these resources and these assets are given through scheduling. Scheduling is a technique in which requests of tasks are scheduled according to given requirements. . The VM (virtual machine) scheduling algorithm is utilized for scheduling the virtual machines to the physical machine according to necessity of the specific data center [4].

<sup>a</sup>Dharmvir305@gmail.com

## 2 Related work

Load balancing and VM placement to host has always been a research subject whose objective is to ensure that every computing resource is distributed efficiently and fairly and in the end improves resource utility. In traditional computing environments of distributed computing, parallel computing and grid computing, researchers in and abroad have proposed a series of static and dynamic and mixed scheduling strategies.

Live migration of VMs permits administrator of server to move a running virtual machine or application among diverse physical machines without separating the customer or the application. The total migration time and the aggregate downtime are two key execution metrics that the customers of a VM administration stress over the most. This paper shows live virtual machine migration as a fundamental tool for dynamic resource management in modern day data centers. In this paper a survey of the current research efforts on live virtual machine migration is presented [5].

Evaluated configurations which is based on historical data and a utility function that ranks them according to the costs incurred. The proposed adaptive framework has been integrated into the Vienna Cloud Environment (VCE) and the evaluation by means of a data-intensive application is presented here in [6]. An important feature provided by the Virtual Machine (VM) advancements is its capacity to move the running operating system instances over the particular physical nodes. This paper studies the proposed and implemented methods for the VM migration [7]. The data centers are a collection of many servers; allocation of VM to Host is known as VM placement. VM placement problem is examined in this paper with focus for maximum utilization of the resources and energy reduction [1]. An algorithm is formulated that will keep the migration time least and also minimizing the number of migrations [7]. Tuning instrument used for proficient asset provisioning guarantying required quality of service. The proposed tuning mechanism also considers load balancing into account [7].

The title is set in bold 14-point Arial, flush left, unjustified. The first letter of the title should be capitalised with the rest in lower case. You should leave 35 mm of space above the title and 6 mm after the title.

## 3 VM placement

In this portion an approach in made is for VM placement which effectively solves the problems of minimizing the response time, load balancing, and balance resource utilization and server consolidation in cloud datacenter. In this work three dimensional problems are solved a) optimizing network latency b) improving utilization and c) improving the retrieval system from database.

In this algorithm the hosts are classified according to their resource availability. In general there are three types of performance parameters of any system i.e. CPU, B/W and Memory. The allocation of the VM is typically done by Mape-k loop in which the sensor to sense the VM

status and create a plan to allocate job requests to VMs. However after allocation a large number of resources remain under utilized. The algorithm shows the total remaining underutilized resources is accumulated and is allocated to next requests. Let us assume that when request for VM arrived to scheduler place this VM on the satisfying host in the list in which all host have remaining resource capacity in order. In other words if a VM having CPU requirement greater than or equal to Memory requirement and Memory requirement greater than or equal to Bandwidth is placed on the satisfying host in the list, in which all host have remaining resource capacity of CPU greater than or equal to Memory and Memory requirement greater than or equal to Bandwidth. The detail is Shown in Algorithm 1. To improve the arranging and retrieving system we are putting worst fit and binary search to make retrieving fast.

---

### Algorithm 1 : Improved mape-K Loop

- 0: Input: A set of Virtual machines, Resource Demand
  - 1: Output: Resource Provision solution R
  - 2: Max = Estimate\_the\_total number\_of\_VMs
  - 3: Lat\_matrix = Create a latency matrix (vm\_locations)
  - 4: Optimize\_utility (CPU\_rem, MEM\_rem, BW\_rem)
  - 5: If the request  $\epsilon$  CPU\_rem
  - 6: Total\_rem\_power =  $\sum_n CPU_i$  // Get all CPUs\_rem in descending order CPU>MEM>BW or CPU>BW>MEM
  - 7: If the request  $\epsilon$  MEM\_rem
  - 8: Total\_rem\_power =  $\sum_n MEM_i$  // Get all MEMs\_rem in descending order MEM>BW>CPU or MEM>BW
  - 9: If the request  $\epsilon$  BW\_rem
  - 10: Total\_rem\_power =  $\sum_n BW_i$  // Get all BWs\_rem in descending order BW>CPU>MEM or MEM>BW
  - 11: Allocate\_request to VM\_rem (Total\_rem\_power)
  - 12: Manage\_VM\_database (arrange\_by\_worst fit, sort\_by\_binary\_serach)
- 

In the Worst Fit algorithm hosts in each list in datacenter are sorted in descending order according to remaining capacity of resources. When request of a new VM or already running VM for VM placement arrives at the cloud datacenter, VM scheduler find the appropriate list and apply the binary search on the selected list to find host that is the worst fit in remaining resource capacity.

## 4 Experimental setup and result

To setup our experiments we use following configuration where the proposed algorithm is implemented is done using CloudSim as a framework in the simulator environment. This work considers various Datacenter, Virtual Machines (VM), host and cloudlet components from CloudSim for execution analysis of the algorithms. NetBeans is an application platform framework for Java

applications. It integrated development surroundings (IDE) for developing primarily with Java, however conjointly with alternative languages like PHP, C/C++, and HTML5. NetBeans is associate degree ASCII text file integrated development surroundings (IDE) for developing with Java, PHP, C++, and alternative programming languages. NetBeans is additionally noted as a platform of standard parts used for developing Java desktop applications.

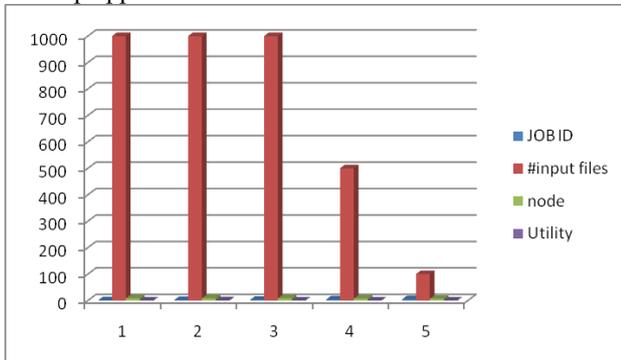


Figure 2: Results without worst case

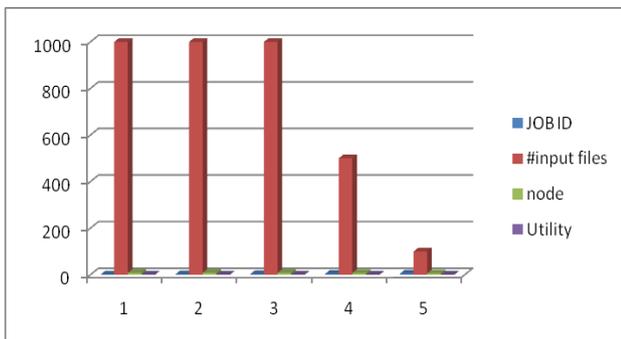


Figure 3: Results with worst case

It has been observed at fig 2 and fig 3 that with the introduction of worst case algorithm there is an increase of utility by average of 4%. Algorithm apply to broker and we test the results give broker job ids are different, nodes are different, input file are different then we compare utility without worst fit with worst fit MAPE-K loop algo. Results shows increase utility with worst fit-MAPE K loop.

## 5 Conclusion and future work

In this paper, we have discussed about the job allocation to the different VMs inside a Cloud Data Center with the help of the mape k loop, MKL is a classical scheduling policy. It is a part of our whole work. It helps to map the jobs to the appropriate VMs, which will reduce the overall response time and waiting time of the jobs and with the introduction of worst case algorithm we observe a further improvement in performance. We are now trying to develop our system in such a way that will help the module to identify intelligently that how many jobs may be served by a single VM at a certain time stamp. This will improve the overall make span of the Cloud Service Provider.

## References

1. N. R. RamMohan, E. Baburaj, International Conference on Computational Intelligence and Communication Networks, 556 (2012).
2. Y. Chawla and M. Bhonsle, International Journal of Emerging Trends & Technology in Computer Science(IJETTCS), 12 (2012).
3. N. Pasha, Dr. A. Agarwal and Dr. R. Rastogi, International Journal of Advanced Research in Computer Science and Software Engineering, 4 (5), 34 (2014).
4. J. Xiao, Z. Wang, International Conference on Cloud Computing and Service Computing, IEEE, 50 (2012).
5. A. Singh. N and M. Hemalatha, Journal of Theoretical and Applied Information Technology, 57(3), 1 (2013).
6. M. Koehler, Koehler Journal of Cloud Computing Advances, Systems and Applications, 1 (2014).
7. S. Banerjee, M. Adhikari, U. Biswas, (2014), International Conference on Business and Information Management (ICBIM), IEEE, 114 (2014).
8. A. Mohan, Shine S, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 2(1), 155 (2013).
9. D. Verma, R. K. Somani, International Journal of Engineering Science and Innovative Technology (IJESIT), 3 (6), 193 (2014).
10. S. Venkatesha, S. Sadhu, S. Kintali, Spring Department of Computer Science University of California, Santa Barbara, 1 (2009).