

Design of Simulator to Evaluate Performance Parameters of Scheduling Clouds in Virtual Machine Environment

Dr. P. K. Suri

Received: 9 December 2013 Accepted: 2 January 2014 Published: 15 January 2014

Abstract

Cloud computing acts as a vision of infinite computing resources that are provided on-demand to the cloud users as needed and are billed on pay as per usage basis. Cloud computing employs the concept of virtualization that provides an opportunity to achieve business and IT objectives. Scheduling is one of the most important challenges that a cloud computing environment faces. Scheduling process determines the order of execution of jobs and the virtual machine to which job is assigned to execute so as to improve the performance and quality of service and at the same time resources are utilized effectively. An attempt has been made in this paper to develop a simulator to schedule a job on allocated virtual machine so as to make efficient resource utilization. In the proposed work, scheduling is done on the basis of availability of allocated virtual machine, providing equal capabilities. We assumed that arrival of jobs in a cloud and their execution on a virtual machine is exponentially distributed. For analysing the proposed algorithm, the performance parameters of scheduling clouds are evaluated in Virtual Machine Environment.

Index terms— cloud computing, job scheduling, simulator, exponential distribution, response parameters.

1 Introduction

Cloud computing environment pivots on sharing of resources to handle cloud applications. Cloud computing provides all types of IT resources as a service to the users and promotes high performance computing. Cloud providers offer their services as software as a service, platform as a service and infrastructure as a service. Cloud computing system rents the computing resources on demand and employs the concept of virtualization.

Cloud computing environment allocates resources in terms of virtual machines that share the resources of single physical computer among several different virtual computing environments. For example, if the physical machine has 15 GB of RAM, then virtual machines could have 15 GB in total, and can be distributed as per user requirements. After allocation, scheduling strategy is required to schedule the jobs on allocated resources to make them execute. Hence cloud resource management is specified by both Author ? : H.C.T.M., Technical Campus, Kaithal, Haryana, 136027, India. e-mail: pksurikuk@gmail.com, goel.himanshi6@gmail.com allocating and scheduling the computing resources on virtual machines.

Scheduling is an important issue of a cloud computing environment that determines the order of execution of jobs and the system on which jobs execute. Because of the heterogeneous nature of cloud environment, the performance of applications is directly influenced by scheduling process. The scheduling of large and diversified group of jobs on virtual machines is considered as NP-complete problem. According to the scheduling strategy used, scheduler either maps the jobs over suitable resources as soon as they arrive or first places the jobs in a queue and then maps over resources for execution. There are various scheduling algorithms that are already present in cloud computing environment.

The present study proposed a simulator to schedule the jobs over allocated resources on the basis of availability of virtual machines. In the present work, a simulator is implemented to map the jobs over suitable machine on the basis of its availability to make them run.

The paper is organized into various sections; present one is the section I that briefly introduces the topic of our concern. Section II provides the overview of different scheduling algorithms that are already present in cloud computing environment. Section III provides the overview of proposed system. Section IV introduces the new cloud job scheduling algorithm that we have proposed. Section V interprets and demonstrates the results of proposed algorithm by using charts that represents response parameters. Section VI concludes the paper.

II.

2 Related Literature

Scheduling is a challenging issue in cloud computing environment that determines the order of execution and the machine to which task is assigned. In this section, an overview of various scheduling strategies that are already presented in cloud computing system is presented. In [1], scheduling is done on the basis of execution time. At first, largest job is mapped to the machine with minimum completion time and then max-min is applied to the remaining job requests. In [2], scheduling is done on the basis of priority; each and every job is having a predetermined priority. In [3], (D D D D D D D D) Year B

priority and execution time are the two main aspect on the basis of which scheduling is done. Task with highest priority is assigned to the machine that provides earliest completion time. In [4] and [5], genetic algorithm is used to map the jobs over suitable virtual machine in minimum execution time. Later one is influenced by bees' colony life that is aspired by two main behaviors, one is reproduction and other one is food source searching. In [6], the parameter that is used to schedule jobs on resources is cost, best suited for the heterogeneous computing environment. In [7], an algorithm for scheduling jobs is proposed to meet the deadline and reliability requirements of the user. In [8], selection is made between max-min and min-min algorithm, the one that helps in minimizing the total completion time and to meet Qos is selected. In [9], scheduling strategy is based on priority that is assigned to both jobs and virtual machines. In [10], a workflow scheduling algorithm is presented to meet Qos requirements on the basis of negotiation between user and server. In [11], load balancing algorithm is presented that is based on the min-min algorithm and to satisfy user requirements more appropriately an improved user aware algorithm is presented. In [12], algorithm is presented to schedule real-time tasks nonpre-emptively. In [13], users select a method on the basis which scheduling is done and then are prioritized. In [14], location-aware scheduling is performed.

3 III.

4 Proposed Work

Cloud resource management is concerned with allocation and sharing of resources. Different scheduling strategies are already present in cloud environment that are based on various parameters. We have proposed a scheduling strategy on the basis of availability of virtual machine. When user makes a request for resources to run an application, services provider allocates the resources in a form of virtual machine as per user needs. To run an application, a user may consist of multiple jobs which runs in virtual machines to make use of available resources required for its execution. Hence an optimal scheduling strategy is required to map these jobs over suitable virtual machine so as to increase the performance and to make resource utilization effective. Scheduling strategy that we have proposed in this paper is based on the availability of virtual machines. The objective of proposed simulator is to make effective resource utilization.

In the proposed work, it is assumed that n numbers of virtual machines (having same capabilities) are already allocated to a user to make use of resources so as to run an application. Jobs arrive at a cloud using any probability distribution for interarrival times. When a job arrives at a cloud, each of the machines is checked to determine if any of them is free. The job is assigned directly to whichever machine is free and if two or more than that are free then the job is assigned to the virtual machine that has been ideal for a longer period so as to make effective resource utilization. If all of the machines are busy in executing jobs, then the job requests joins the common queue. The jobs requests in a queue are serviced on a first-come-first-served basis. The service times for each servicing facility can be viewed as independent samples from some specific distribution.

5 IV. Cloud_sch_sim Algorithm

Step 1: Generate AT's and ST's for all jobs arrives at a cloud from exponential distribution $(1-e^{-\lambda t})$. These samples can be obtained by transforming uniform random numbers in the interval (0,1), according to eq. $T_k = -(\log e u_k) / \lambda$

Step 2: Do the initialization: [Set:-AT 0 =0, CAT 0 =0, QLT=0 For (K=1,..N) CAT K = AT K +CAT K-1 For (L=1,..S) NDT L = CDT K,L = CAT L +ST L]

Step 3: Find machine (VM) m having minimum next departure time and hence executes the suitable job.

Step 4: If all jobs are already arrived at a cloud then go to step 9 else calculate the difference between next arrival time and minimum departure time.

Step 5: If that calculated value is less than 0 then go to step 6, if equal to 0 then go to step 7 else go to step 8

Step 6: In this step, job that arrives at a cloud joins the common queue and set: QLT+= 1, I= I+1 and NAT= CAT I . Now go to step 3.

102 Max [Serv k] +1, k=1,?S Now add ST of corresponding job to NDT m and then the corresponding job departs.
103 At the same time, new job arrives at a cloud and joins the queue. Now go to step 10.
104 Step 8: If (QLT >0) then go to step 4 else the job that arrives at a cloud is executed by machine m. And
105 update the next departure time from machine m and go to step 10.
106 Step 9: In this step, the job leaves the queue on first come first serve basis and is executed by machine m and
107 its next departure time is updated.
108 Step 10: If the job that is executed by virtual machine m is the las Print the computed statistics Else go to
109 step 3.
110 Step 11: End

111 6 V. Results

112 In this section, we have analyzed the performance of the cloud by implementing "CLOUD_SCH_SIM" Simulator.
113 The performance is evaluated in three parts by using number of response parameters: Mean queue length (QL)
114 ? Average waiting time of jobs in a cloud (WT) ? Throughput (TH) ? System Utilization (SU)
115 In first part, response parameters are analyzed for constant service rate (say, ?=12), at varying arrival rate (?).
116 For the following analysis, we assume that 100 jobs are required to run an application and 2 VM's are allocated
117 on which job executes. The resultant values of response parameters on the basis of following assumptions are
118 recorded in table I The job that executes on machine m is:
119 Step 7: Both execution and arrival takes place at this It depicts that increase in arrival rate results into an
120 increase in parameter Mean queue length.

121 7 It depicts that increase in arrival rate results into an increase 122 in parameter Average waiting time of jobs in a cloud

123 It depicts that increase in arrival rate results into an increase in parameter System Utilization.
124 In Second part, response parameters are analyzed for constant arrival rate (say, ?=19), at varying service
125 rate (?). For the following analysis, we assume that 100 jobs are required to run an application and 2 VM's are
126 allocated on which job executes. The resultant values of response parameters on the basis of following assumptions
127 are recorded in table II In Second part, response parameters are analyzed for constant arrival rate (say, ?=3) and
128 service rate (say, ?=4), at varying number of allocated VM's (N). The resultant values of response parameters
129 on the basis of following assumptions are recorded in table III. It depicts that as the number of allocated VM's
130 increases, Throughput decreases.

131 8 VI. Conclusion

132 The simulator "CLOUD_SCH_SIM" is designed and implemented successfully in this paper to schedule the jobs
133 arrive at a cloud over suitable virtual machine to make them run so as to make effective resource utilization.
134 In this paper, the performance parameters of the scheduling clouds such as throughput, utilization are also
135 evaluated in Virtual Machine Environment. This simulator will be an asset for cloud service providers/ software
136 development houses to analyze the performance of clouds and thereby in managing the resources in various
137 computing environment particularly in clouds.

¹© 2014 Global Journals Inc. (US)



Figure 1: Figure 1 :

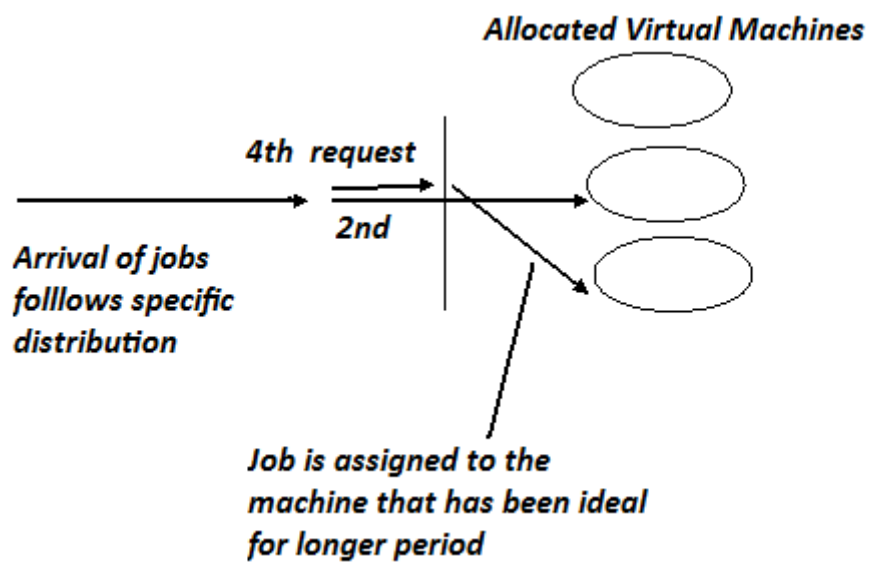


Figure 2: Figure 2 :?

3

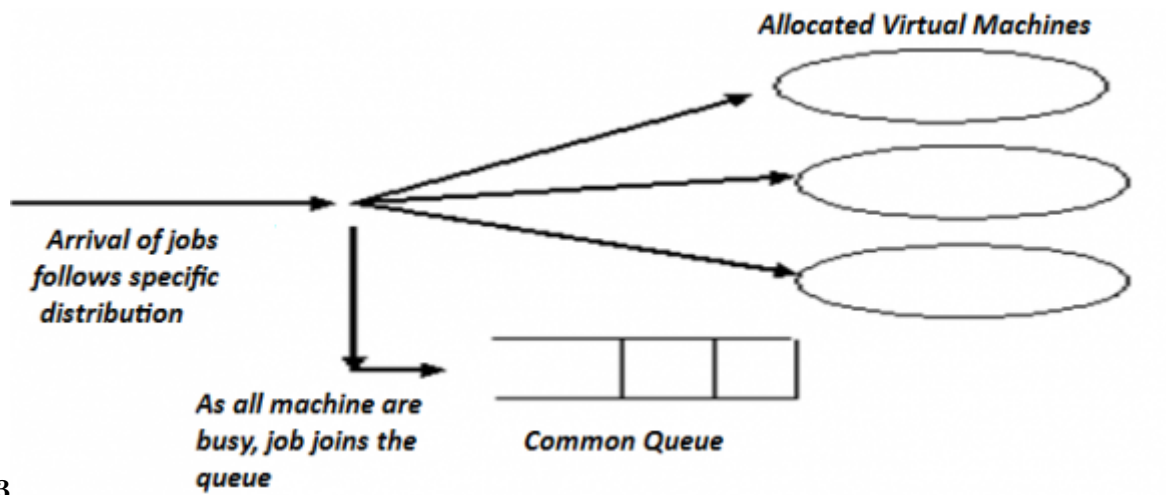


Figure 3: Figure 3 :

4

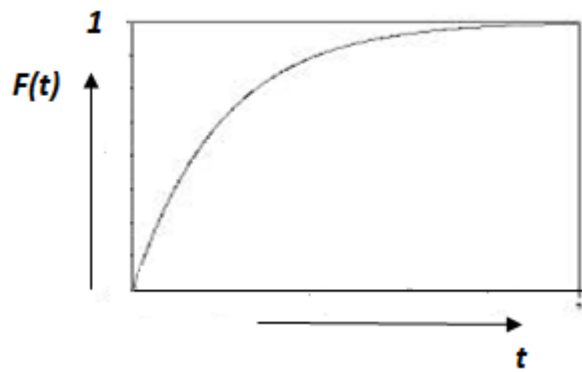


Figure 4: Figure 4 :

657



Figure 5: Figure 6 :Figure 5 :Figure 7 :

8



Figure 6: Figure 8 :

9



Figure 7: Figure 9 :

1011



Figure 8: Figure 10 :Figure 11 :



Figure 9:

12

Figure 10: Figure 12 :

Symbol	Description
Used	
N	Number of jobs requires to run a cloud application
S	Number of allocated virtual machines
AT k	Inter-arrival times of arriving jobs
ST k	Service time of k th job
CAT K	Cumulative arrival time of k th job
CDT K,L	Cumulative departure time at machine l of job k
NDT L	Next departure time from machine l
NAT	Next arrival time
QLT	Current Queue length
Serv l	Describes which job is currently being executed by machine m

QL	WT	SU
0.15	0.004800	0.291667
0.5	0.017988	0.375000
0.72	0.034038	0.458333
2.32	0.106794	0.625000
7.53	0.379477	0.791667

Graphs representing parameter values recorded in above table are shown below:

Figure 11:

I

t job then
 ?=12), at Varying (?).

Figure 12: Table I :

?	QL	WT	SU
12	7.53	0.379477	0.791667
16	1.86	0.064698	0.593750
20	0.81	0.023105	0.475000
23	0.56	0.012749	0.413043
26	0.49	0.007488	0.365383

Figure 13:

2

Graphs representing parameter values recorded in above table are shown below:

Figure 14: Table 2 :

3

[Note: Parameter values for constant arrival rate (say, $\lambda=3$) and service rate (say, $\mu=4$), at varying number of allocated VM's (N).]

Figure 15: Table 3 :

-
- [Ghanbari and Othman ()] ‘A Priority based Job Scheduling Algorithm in Cloud Computing’. Shamsollah Ghanbari , Mohamed Othman . *International Conference on Advances Science and Contemporary Engineering*, 2012.
- [Zhao et al. ()] *A Resource Minimizing Scheduling Algorithm with Ensuring the Deadline and Reliability in Heterogeneous Systems*, Laiping Zhao , Yizhi Ren , K Sakurai . 2011. IEEE.
- [Kaur and Verma ()] ‘An Efficient Approach to Genetic Algorithm for Task Scheduling in Cloud Computing Environment’. Shaminder Kaur , Amandeep Verma . *International Journal of Information Technology and Computer Science* 2012. 10 p. .
- [Jin et al. ()] ‘BAR: An Efficient Data Locality Driven Task Scheduling Algorithm for Cloud Computing’. Jiahui Jin , Junzhou Luo , Aibo Song , Fang Dong , Runqun Xiong . *11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, 2011.
- [Bitam ()] ‘Bees Life Algorithm for Job Scheduling in Cloud Computing’. Salim Bitam . *2nd International Conference on Communications and Information Technology*, 2012.
- [Dr et al. ()] ‘Efficient Optimal Algorithm of Task Scheduling in Cloud Computing Environment’. Amit Dr , Saloni Agarwal , Jain . *International Journal of Computer Trends Technology* 2014. 9 (7) .
- [Dr et al. ()] ‘Enhanced Max-min Task Scheduling Algorithm in Cloud Computing’. . V Dr , R Vaithiyanathan , S Kumar , B Vignesh , B Thamocharan , Karthikeyan . *International References Références Referencias 1. Upendra Bhoi*, 2013. 2. (An Efficient TPD Scheduling Algorithm for Cloud Environment)
- [Mrs et al. ()] *Improved Cost-Based Algorithm for Task Scheduling in Cloud Computing*, S Mrs , Dr G Selvarani , Sudha , Sadhasivam . 2010. IEEE.
- [Liu et al. ()] *On-Line Scheduling of Real-Time Services for Cloud Computing*, Shuo Liu , Gang Quan , Shangping Ren . 2010. IEEE.
- [Jayadivya and Bhanu ()] ‘QoS Based Scheduling of Workflows in Cloud Computing’. S K Jayadivya , S. Mary Saira Bhanu . *International Journal of Computer Science and Electrical Engineering* 2012. 1 (1) .
- [Singh and Suri ()] ‘QPS MAX-MIN<>MIN-MIN : A Qos Based Predictive Max-Min, Min-Min Switcher Algorithm for Job Scheduling in a Grid’. M Singh , P K Suri . *International Journal of Information and Technology* 2008. 7 (8) p. .
- [Wieczorek et al. ()] ‘Scheduling of Scientific Workflows in ASKALON Grid Environment’. M Wiecezorek , R Prodan , T Fahringer . *SIGMOD Rec.* 2005. 34 (3) p. .
- [Chen et al. ()] ‘User-Priority Guided Min-Min Scheduling Algorithm for Load Balancing in Cloud Computing’. Huankai Chen , Professor Frank Wang , Dr Na Helian , Gbola Akanmu . *Parallel Computing Technologies National Conference*, 2013. p. .