Global Journal of Computer Science and Technology: B
Cloud and Distributed
Volume 14 Issue 2 Version 1.0 Year 2014
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

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

By Dr. P. K. Suri & Himanshi Goyal

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.

Keywords: cloud computing, job scheduling, simulator, exponential distribution, response parameters.

GJCST-B Classification : D.4.2

Strictly as per the compliance and regulations of:

Open Association of Research Society, USA
Research | Diversity | Ethics

© 2014. Dr. P. K. Suri & Himanshi Goyal. This is a research/review paper, distributed under the terms of the Creative Commons Attribution-Noncommercial 3.0 Unported License http://creativecommons.org/licenses/by-nc/3.0/), permitting all non-commercial use, distribution, and reproduction inany medium, provided the original work is properly cited.

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

Dr. P. K. Suri [α] & Himanshi Goyal [σ]

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.

Keywords: cloud computing, job scheduling, simulator, exponential distribution, response parameters.

I. 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 offers their services as software as a service, platform as a service and infrastructure as a service. Cloud computing system rent 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 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 executes. 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. 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],

Author α σ: H.C.T.M., Technical Campus, Kaithal, Haryana, 136027, India. e-mail: pksurikuk@gmail.com, goel.himanshi6@gmail.com

© 2014 Global Journals Inc. (US)

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 non-pre-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.

III. 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.

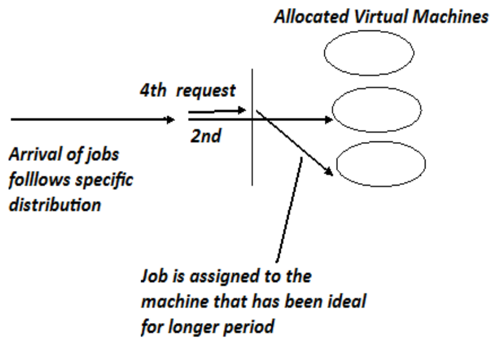

Allocated Virtual Machines
4th request
2nd
Arrival of jobs folllows specific distribution
Job is assigned to the machine that has been ideal for longer period

Figure 1 : When one or more machine is available

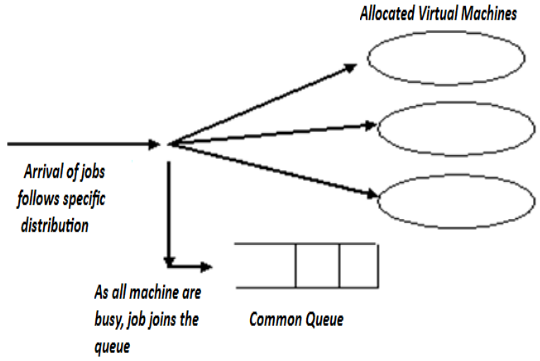
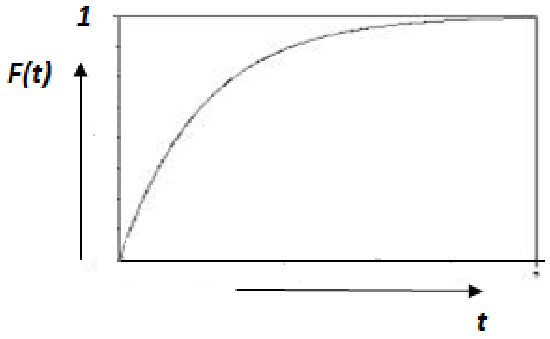

Allocated Virtual Machines
Arrival of jobs follows specific distribution
As all machine are busy, job joins the queue
Common Queue

Figure 2 : When all machines are busy in executing other requests

a) Assumptions

- Arrival of jobs and their execution on virtual machines follows exponential distribution. The probability distribution function of this distribution can be shown as:



- Jobs arrive at a cloud one at a time with an average inter-arrival time, say alpha.
- At a time, only one job can execute on a single virtual machine with service rate say beta.

© 2014 Global Journals Inc. (US)

- The jobs in the queue are served on first-come-first-serve basis.
- Virtual Machines are having same capabilities.

IV. Cloud_sch_sim Algorithm

Symbol Used	Description
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

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 next 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:

$QLT+= 1$, $l= l+1$ and $NAT= CAT_l$. Now go to step 3.

Step 7: Both execution and arrival takes place at this place as follows:-

The job that executes on machine m is:

$$\text{Max } [Serv_k] +1, \ k=1,\ldots S$$

Now add ST of corresponding job to NDT_m and then the corresponding job departs.

At the same time, new job arrives at a cloud and joins the queue. Now go to step 10.

Step 8: If (QLT >0) then go to step 4 else the job that arrives at a cloud is executed by machine m. And update the next departure time from machine m and go to step 10.

Step 9: In this step, the job leaves the queue on first come first serve basis and is executed by machine m and its next departure time is updated.

Step 10: If the job that is executed by virtual machine m is the last job then Print the computed statistics

Else go to step 3.

Step 11: End

V. Results

In this section, we have analyzed the performance of the cloud by implementing "CLOUD_SCH_SIM" Simulator. The performance is evaluated in three parts by using number of response parameters:

- Mean queue length (QL)
- Average waiting time of jobs in a cloud (WT)
- Throughput (TH)
- System Utilization (SU)

In first part, response parameters are analyzed for constant service rate (say, $\mu=12$), at varying arrival rate (λ). For the following analysis, we assume that 100 jobs are required to run an application and 2 VM's are allocated on which job executes. The resultant values of response parameters on the basis of following assumptions are recorded in table I.

Table I : Response Parameter Values for Constant **(Say, M=12)**, at Varying (Λ).

λ	QL	WT	SU
7	0.15	0.004800	0.291667
9	0.5	0.017988	0.375000
11	0.72	0.034038	0.458333
15	2.32	0.106794	0.625000
19	7.53	0.379477	0.791667

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

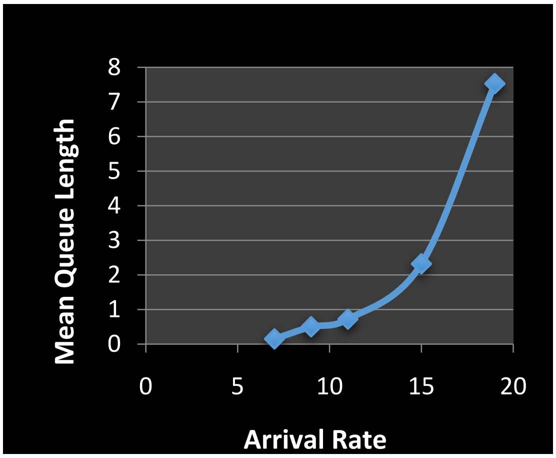


Figure 3 : Analysis of Mean Queue Length for constant (**μ**=12), at varying (λ).

It depicts that increase in arrival rate results into an increase in parameter Mean queue length.

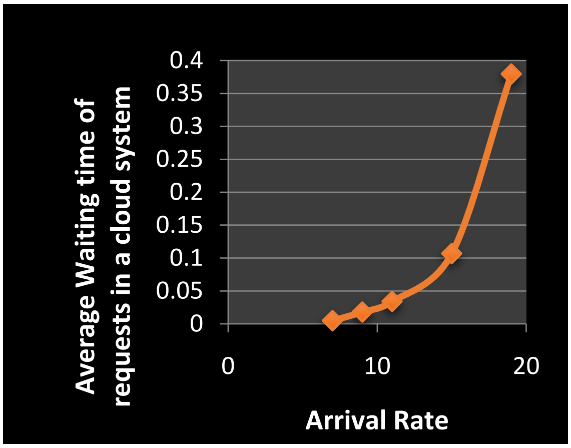


Figure 4 : Analysis of Average waiting time of jobs in a cloud for constant (**μ**=12), at varying (λ).

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

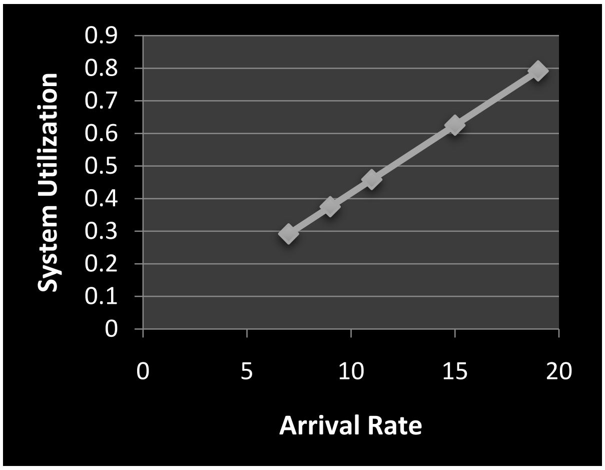


Figure 5 : Analysis of System Utilization for constant (**μ**=12), at varying (λ).

It depicts that increase in arrival rate results into an increase in parameter System Utilization.

In Second part, response parameters are analyzed for constant arrival rate (say, λ=19), at varying service rate (μ). For the following analysis, we assume that 100 jobs are required to run an application and 2 VM's are allocated on which job executes. The resultant values of response parameters on the basis of following assumptions are recorded in table II.

μ	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

Table 2 : Response Parameter values for constant (say, **λ** =19), at varying (**μ**).

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

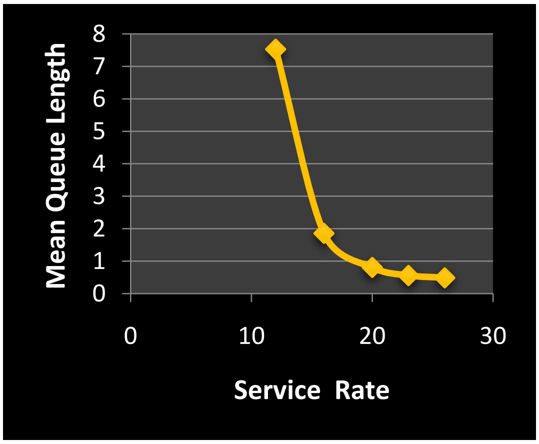


Figure 6 : Analysis of Mean Queue Length for constant (**λ** =19), at varying (**μ**).

It depicts that boost in service rate lowers the Mean queue length.

© 2014 Global Journals Inc. (US)

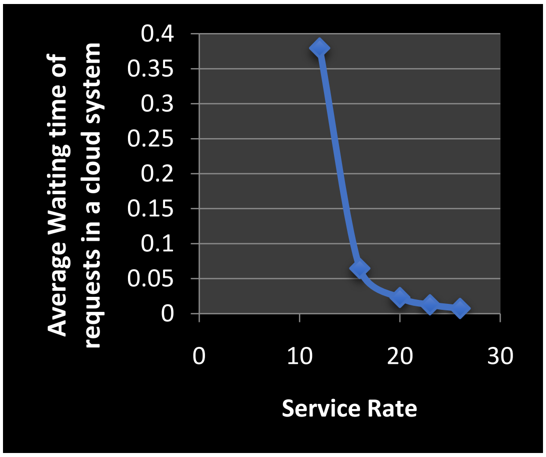


Figure 7 : Analysis of Average waiting time of jobs in a cloud for constant (**λ=19**), at varying (**μ**).

It depicts that boost in service rate lowers the Average waiting time of jobs in a cloud.

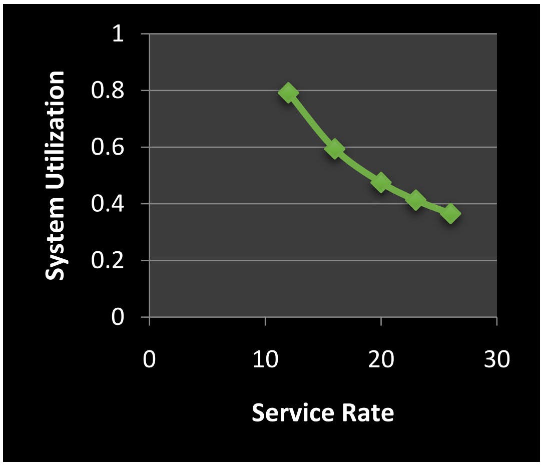


Figure 8 : Analysis of System Utilization for constant (**λ** =19), at varying (**μ**).

It depicts that boost in service rate lowers the System Utilization.

In Second part, response parameters are analyzed for constant arrival rate (say, λ=3) and service rate (say, μ=4), at varying number of allocated VM's (N). The resultant values of response parameters on the basis of following assumptions are recorded in table III.

N	QL	WT	SU	TH
1	6.31	1.886198	0.75	3.0
2	0.5	0.053900	0.375	1.5
3	.02	0.000530	0.25	1.0

Table 3 : Response Parameter values for constant arrival rate (say, **λ=3**) and service rate (say, **μ=4**), at varying number of allocated VM's (N).

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

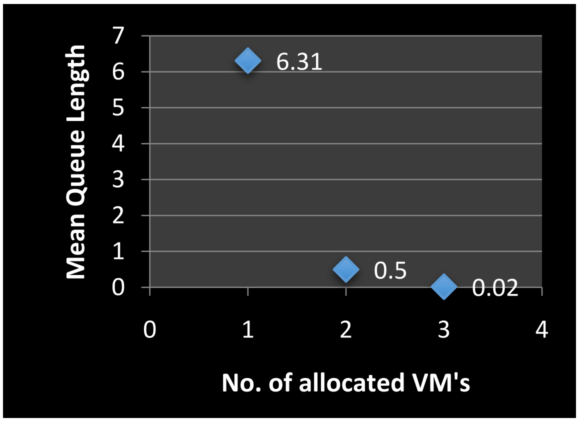


Figure 9 : Analysis of Mean Queue Length for constant arrival rate (**λ=3**) and service rate (**μ=4**), at varying number of allocated VM's (N).

It depicts that as the number of allocated VM's increases, Mean queue length decreases.

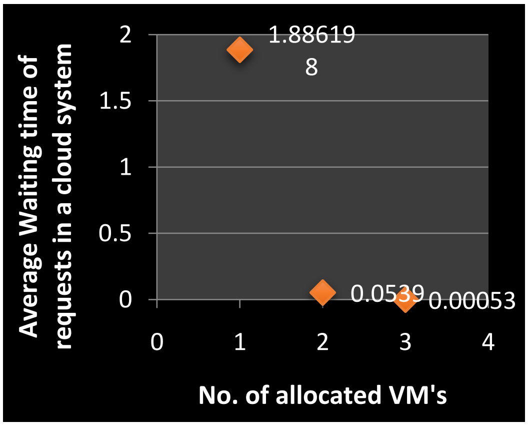


Figure 10 : Analysis of Average waiting time of jobs in a cloud for constant arrival rate (**λ=3**) and service rate (**μ=4**), at varying number of allocated VM's (N).

It depicts that as the number of allocated VM's increases, Average waiting time of jobs in a cloud decreases.

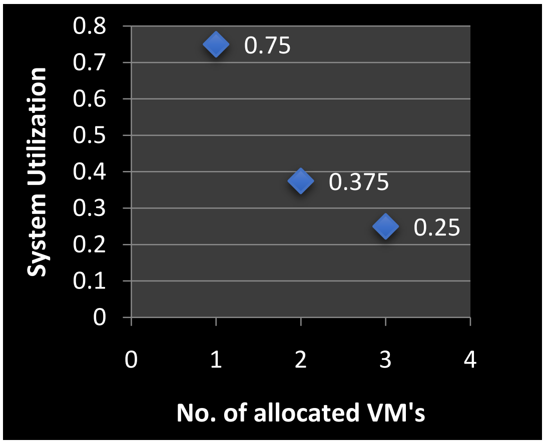


Figure 11 : Analysis of System Utilization for constant arrival rate (**λ=3**) and service rate (**μ=4**), at varying number of allocated VM's (N).

© 2014 Global Journals Inc. (US)

It depicts that as the number of allocated VM's increases, System Utilization decreases.

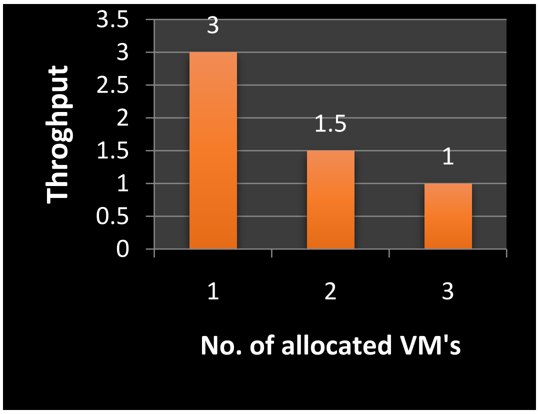


Figure 12 : Analysis of System Throughput for constant arrival rate ($\lambda=3$) and service rate ($\mu=4$), at varying number of allocated VM's (N).

It depicts that as the number of allocated VM's increases, Throughput decreases.

VI. Conclusion

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

References Références Referencias

1. Upendra Bhoi, "Enhanced Max-min Task Scheduling Algorithm in Cloud Computing", International Journal of Application or Innovation in Engineering and Management, Vol. 2, No. 4, 2013.
2. Shamsollah Ghanbari, and Mohamed Othman, "A Priority based Job Scheduling Algorithm in Cloud Computing", International Conference on Advances Science and Contemporary Engineering, 2012.
3. Wieczorek M., Prodan R. and Fahringer T., ''Scheduling of Scientific Workflows in ASKALON Grid Environment'', SIGMOD Rec., Vol. 34, No. 3, pp. 56–62, 2005.
4. Shaminder Kaur and Amandeep Verma, "An Efficient Approach to Genetic Algorithm for Task Scheduling in Cloud Computing Environment, International Journal of Information Technology and Computer Science, Vol. 10, pp. 74-79, 2012.
5. Salim Bitam, "Bees Life Algorithm for Job Scheduling in Cloud Computing", 2nd International Conference on Communications and Information Technology, 2012.
6. Mrs.S.Selvarani and Dr.G.Sudha Sadhasivam, "Improved Cost-Based Algorithm for Task Scheduling in Cloud Computing", IEEE, 2010.
7. Laiping Zhao, Yizhi Ren, and Sakurai, K., "A Resource Minimizing Scheduling Algorithm with Ensuring the Deadline and Reliability in Heterogeneous Systems", IEEE, 2011.
8. M. Singh and P.K. Suri, "$\text{QPS}_{\text{MAX-MIN<>MIN-MIN}}$: A Qos Based Predictive Max-Min, Min-Min Switcher Algorithm for Job Scheduling in a Grid", International Journal of Information and Technology, Vol. 7, No. 8, pp. 1176-1181, 2008.
9. Dr. Amit Agarwal and Saloni Jain, "Efficient Optimal Algorithm of Task Scheduling in Cloud Computing Environment", International Journal of Computer Trends and Technology, Vol. 9, No. 7, 2014.
10. Jayadivya S K and S. Mary Saira Bhanu, "QoS Based Scheduling of Workflows in Cloud Computing", International Journal of Computer Science and Electrical Engineering, Vol. 1, No. 1, 2012.
11. Huankai Chen, Professor Frank Wang, Dr Na Helian, and Gbola Akanmu, "User-Priority Guided Min-Min Scheduling Algorithm for Load Balancing in Cloud Computing", Parallel Computing Technologies National Conference, pp. 1-8, 2013.
12. Shuo Liu, Gang Quan, and Shangping Ren, "On-Line Scheduling of Real-Time Services for Cloud Computing", IEEE, 2010.
13. Dr.V.Vaithiyanathan, R.Arvindh Kumar, S.Vignesh, B.Thamotharan, and B.Karthikeyan, "An Efficient TPD Scheduling Algorithm for Cloud Environment", International Journal of Engineering and Technology, Vol. 5, No. 3, 2013.
14. Jiahui Jin, Junzhou Luo, Aibo Song, Fang Dong and Runqun Xiong, "BAR: An Efficient Data Locality Driven Task Scheduling Algorithm for Cloud Computing", 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, 2011.

Global Journals Inc. (US) Guidelines Handbook 2014

www.GlobalJournals.org