A Multi-Level Analysis to Improve Scheduling in Distributed Cloud Environment

Sanket Mani Tiwari

Abstract— **In the Cloud System Job request scheduling is the most critical aspects in a distributed system where many request are maintain in the Queue form. This type of requests is scheduled in such way the cloud system will not get overloaded. In terms of overload a virtual machine raise the delay of the job over the system as well as raise the chances of process failure. In this proposed work, a dominant parametric scheduling mechanism is defined under two level analyses. Criticality, security and efficiency parameters are Consider in this work. In these parameters virtual machines and the user requests both are defined inside in the cloud system. In these papers work defined a multi-level load balanced scheduling mechanism. So in this paper define and categorize the work in different level. In the first level virtual machines are ordered under the capacity and criticality parameters. In the second level, the user requests are ordered under requirements of the resource and the job request time. It is based on these two parameters the cost analysis and performed on server side and client side. At next level, all virtual machines are grouped under the cost analysis so that the sharing of the load among similar kind of machines will be done. At next level, Use the cost level mapping between the server side and client side is performed to allocate the requests to particular virtual machine. So the virtual machine assigns the request and the next work is to perform the execution of the request. During the execution process, the wait time and process time analysis is performed. This analysis shows, the requests are executed in appropriate time frame and the wait time over the system is reduced.**

Index Terms— Cloud computing, Objective, Scope of work, Scheduling, multilevel Scheduling, Virtualization, Algorithm, Result, Work Stage, Comparison of Start Time and Arrival Time.

Manuscript received July 19, 2014

Sanket Mani Tiwari, Galgotias University, School of Computer Science and Engineering,Yamuna Expressway, Greater Noida, Uttar Pradesh, India

I. INTRODUCTION

Cloud computing is known as a provider the distributed environment to many application and services that are placed at one location and available to the all public users worldwide using on the basis of internet. Cloud system was developed by NIST for fulfill the requirement of global users present over the web in low cost and infrastructure requirements. A cloud user can exist in different cloud environment such as public environment, private environment, hybrid environment or the community environment. With each level, the cost specification, security specification is differ [3]. In the Cloud system the scheduling is the biggest task to handle the process on the virtual machine. This paper basically designs for multilevel scheduling concept. In this paper categorize the level of the scheduling. In the first level we describe the cost security and capability specification. The second level of cloud architecture is the application level interface. It is the upper level of the cloud architecture. Application levels interface actually the cloud to user interface. In Cloud architecture physical layer or lower level of is represented by the cloud server itself. This layer contains the database layer and it is integrated with virtual cloud. This layer is responsible to perform the actual service allocation and the execution. The cloud system defined at this level is most complex respective to the user. As the cloud system is present in web form, it is more complex but provides effective services to the users. Virtualization is actually the technique that removes the links between the software, hardware and the operating system. It connects the logical resources in abstracted way with physical resources under flexible and low cost access. The virtual environment provides the dynamic and large network scenario so that the demand level satisfaction will be obtained over the network. This kind of servers includes the dynamic cloud based infrastructure so that the product and service level isolation will be obtained under the resource sharing [2]. A software level environment is generated within the machine that works itself as a separate hardware system so that the machine level distribution will be performed [5].

II. RELATED WORK

In the proposed work mainly use for increase the number of Cloud Services which are available in the internet. These types of services are provided by different cloud servers in an integrated environment. Scheduling mechanism use for virtual machine under task based analysis. In this paper defined inference technique based analysis so that the IO bounded scheduling will be performed over the environment [1]. These papers are defined the multilevel scheduling. In multilevel scheduling we categorize the task in different level under the cloud computing system. First level we are defined and filter the task in the basis of Resource allocation and the second level the process migrate on the basis of cost and capacity. Distributed computing environment face the problem of load balancing. So with the help of multilevel scheduling easily manage the load balancing problem of the process in cloud computing system. Scheduling algorithms and resource management strategies specially designed for the cluster and grid cloud and peer-to-peer computing. In the all cloud computing we are migrating process while reducing the service access time with the help of different parameter.

A. Objective

In presented work we are define the given object here.

- The main objective of work is to define a level scheduling mechanism to set the sequence of process execution in cloud environment.
- The objective of work is to define a condition to perform the process migration in cloud environment.
- The objective of work is to reduce the wait time and migration probability.

B. Scope of work

The presented work is effective in the following ways

- Each process will be executed on some effective VM so that the wait time of the process will be reduced.
- The presented system will be able to handle the over load conditions.
- As the level scheduling is defined, each level deals with specific performance parameter so that overall cloud server performance will be improved.

C. Research Methodology

In the present work we are defined a three layer scheduling approach. In the First layer presented work on server side, and arrange the cloud server according the available reliability vector. At first level we are consider a cloud server with the higher security feature. Cloud virtual machine is setup in a specific order, and the work is to arrange the user process order. So In this layer we are performing the parametric cost estimation. Cost estimate parameter includes the process time, security and dead line criticality. Based on these parameters, client side request ordering is performed. Completing the first level the machine is defined in specific order with client side request and server side machine in a particular order. After the first level use the second level in this level allocate the user requests to the particular cloud server. And the user request is performed on the cloud server. On the basis of allocation of the request we are estimate the load and capacity is done. We allocate the request under the three parameters. After the allocation of the request we execute the request on relative virtual machine. Execution is done under arrival time, process time and dead line parameters. Execution is based on situation first is execution process is done before dead line this means the process will be executed on the particular virtual machine. We are use the another parameter is wait time. The work is about to reduce the average wait time of process. With the help of wait time we calculate the process migration if the wait time is exceeds the wait threshold, the process migration will be done. And with the help of migration we are identify the next cloud which are handle the request.

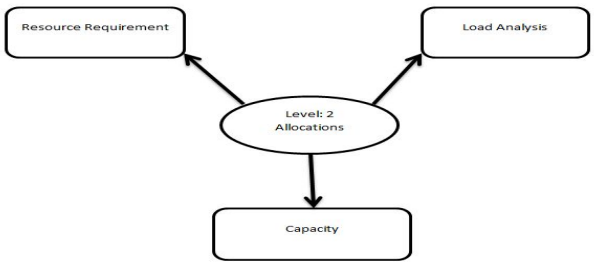
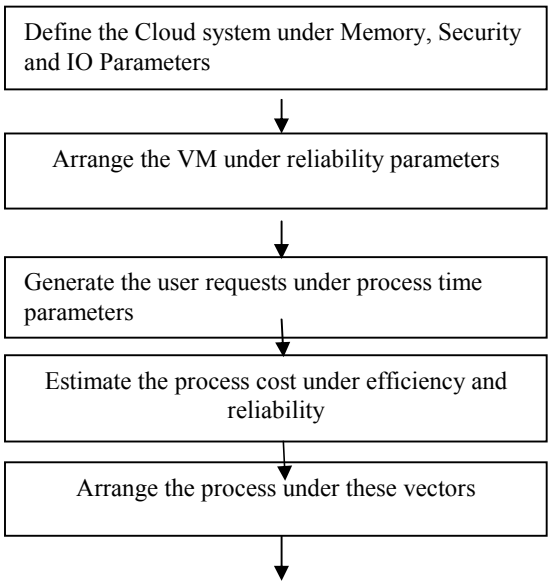


Fig: a) Factors for Low level scheduling

D.*Proposed Methodology*:



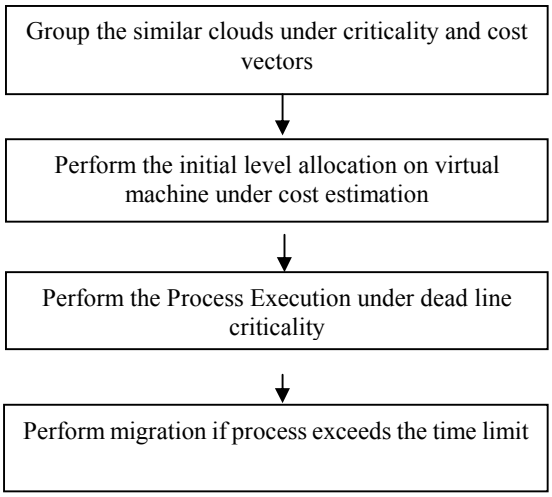


Fig: b) Flowchart of the Proposed Work

1. Cloud Environment Setup

The first stage to work with cloud computing is to build the cloud environment. While setting up the cloud environment, it is required to set the some properties such as number of physical cloud servers, number of virtual machines available on each cloud server and the capabilities of each virtual machine. The capabilities can be defined in terms of memory availability and the I/O devices associated with each virtual machine.

2. User Request Initialization

Once a user will enter to the system, a service request will be performed to the integrated environment. User process request will be defined under different parameters such as process time specification, memory requirement, I/O requirement, Dead line specification etc.

3. Cost Estimation

A weighted mechanism will be defined to estimate the cost of a process. The cost estimation weighted process is based on the process time, arrival time, dead line criticality and the memory requirements.

4. Scheduling

Once the weightage to each process is defined, these processes are scheduled based on the greedy algorithm. The objective function is to arrange the processes under the least cost ratio.

5. Process Allocation

Once the processes are scheduled and the cost estimation is done. The evaluation of the virtual machine will be done in order of process occurrence in the queue. This evaluation of the virtual machine will be done respective to under the capacity and the request analysis. If the process is feasible to the virtual machine capacity the process will be allocated to that particular virtual machine.

6. Migration

If the process is not feasible to the particular virtual machine, the requirement of the migration is identified. Now to migrate the process, the capacity and load on other virtual machines will be analyzed and based on it the migration of the process will be done.

7. Analysis

Analysis is the final stage of the presented work in which all the processes will be analyzed under different parameters such as wait time analysis, response time analysis etc.

E. Algorithm

/*A Multi-Level Weighted Mechanism to Schedule User Requests for Cloud Server */

1. Initialize the Cloud Server with Integrated Virtual Machines and Resource Parameter Specifications
2. For i=1 to Length (Cloud)
   {
3. For j=1 to Length (VM)
   {
4. Set VM (i, j).Criticality=Random
   Set VM (i, j).Availability=Random
   Set VM (i, j).Response Time=Random
   }
   }
   [Assign the Reliability and Efficiency Parameters for Each Virtual Machine]
5. For i=1 to Length (User)
   {
   Set User (i).Process Time=Random
   Set User (i).Deadline=Random
   Set User (i).Arrival Time=Random
   Set User (i).Resource Requirement=Random
   }

[Set the Resource Requirement and Process Specification for All Requests]

6. Estimate the Level 1 Cost for Each Request under Criticality and Security parameters
7. Arrange the Requests under cost parameter
8. Assign the Level 2 Cost Parameter for Each VM under the Load and Capacity Parameters
9. Arrange the Virtual Machines under Cost Parameter
10. Perform the cost analysis on virtual machines and divide the virtual machines in different groups under cost analysis
11. Perform the Level 1 and Level 2 Cost Analysis and Matching to Assign the Users to most effective Virtual Machines
12. Execute the allocated user request on specific virtual machine under cost matching and deadline criticality

F. Work stage

1. *Scheduling*

This is the primary stage of the proposed work where the scheduling is performed by taking three main criteria. (i) Arrival Time Analysis (ii) Process Time Analysis (iii) Priority Analysis. The prioritization here is taken as the dynamic attribute. As the priority of the process will be shifted, the order of the process execution will also modify. In this work, we have defined a limit on the number of processes assigned under the same priority. As the limit will be crossed, the processes will be shifted to next higher priority. This concept will avoid the starvation because of the higher priority process execution. As the priority will be changed rescheduling of the process will be changed.

2. *Wait Time Analysis*

The wait time analysis is here defined to avoid the starvation condition. To avoid the starvation, the concept of aging is included in this work. According to this, as the wait time of a process will exceed the certain limit. The priority of the process will be increased. An improved scheduling approach is suggested in this work. The improvement is here performed on shortest job first scheduling technique by including the two main parametric vectors called Dynamic Prioritization and Aging. The broader view of the defined work is given as figure 2.2. Each stage of the proposed work is defined here in detail Based on these all factors the scheduling of the process will be performed and relatively the resource allocation on the Cloud environment will be performed.

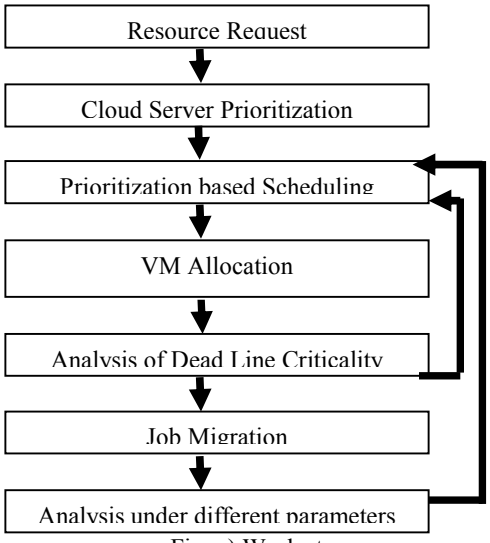


Fig: c) Work stage

III. RESULT

In the result section we perform work and find the result with the help of CloudSim because it provides the good simulation platform. CloudSim Support modeling of on-demand virtualization enabled resource and request managing.

A. Graphical Interfaces

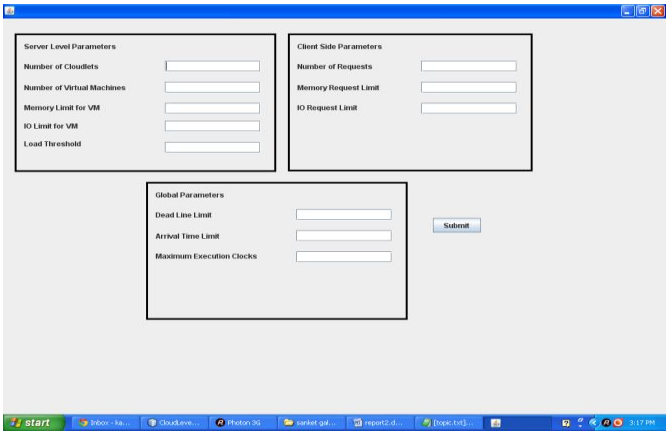


Fig: a) Graphical Interfaces

Here figure a) is showing the graphical interface to work on cloud system to perform the process scheduling. Here the input parameters are taken for server side settings and to generate the user requests

B. User Service Request

U ID	C	V M	S T	A T	D L	Ta T	F T	W T
3	0	3	52	46	56	5	57	6
7	0	0	81	76	88	7	88	5
8	0	0	89	69	90	7	96	20
1	0	4	31	27	34	2	33	4
6	1	2	69	54	71	10	79	15
4	1	2	42	42	54	8	50	0
2	0	2	22	22	33	8	30	0
5	1	0	58	48	69	10	68	10

Table: b) User Service Request

- U ID=User id
- C=Cloud
- V M=Virtual machine
- S T=Start Time
- A T=Arrival Time
- D L=Dead Line
- Ta T=Turnaround Time
- F T=Finish Time
- W T=Wait Time

Here, table b) is showing the input request parameters respective to the service requirements. The request parameter is performed at different time interval for different processes. According to these requests, the parameters are setup such request time, process time, dead line etc. These parameters can be used collectively for user request generation.

C. Start Time Analysis

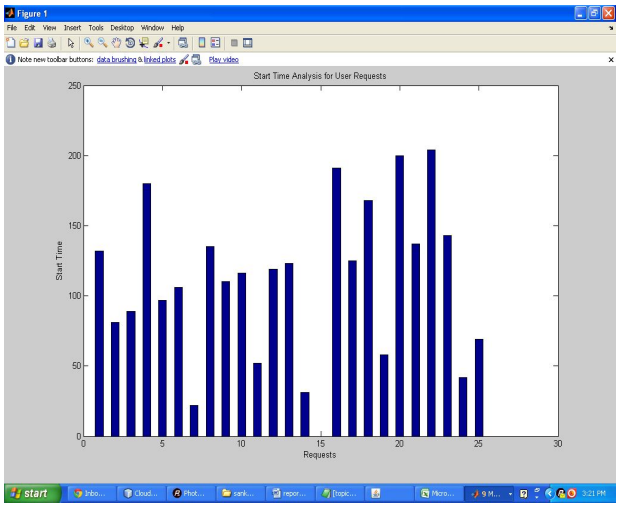


Fig: c) Start Time Analysis

Here figure c) is showing the start time anlaysis for 20 input user requests. Here x axis represents the number of user requests and y axis represents the start time in seconds. The figure shows the start time is between 0 and 120.

D. Arrival Time Analysis

Here figure d) is showing the arrival time anlaysis for 20 input user requests. Here x axis represents the number of user requests and y axis represents the arrival time in seconds. The figure shows the arrival time is between 0 and 120.

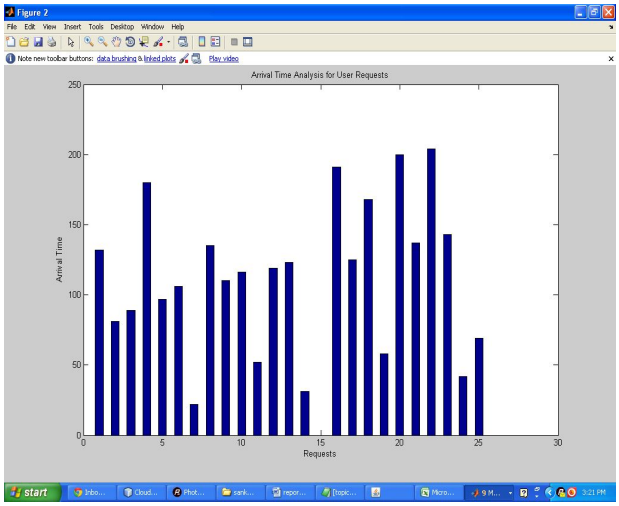


Fig: d) Arrival Time Analysis

E. TurnAround Time Analysis

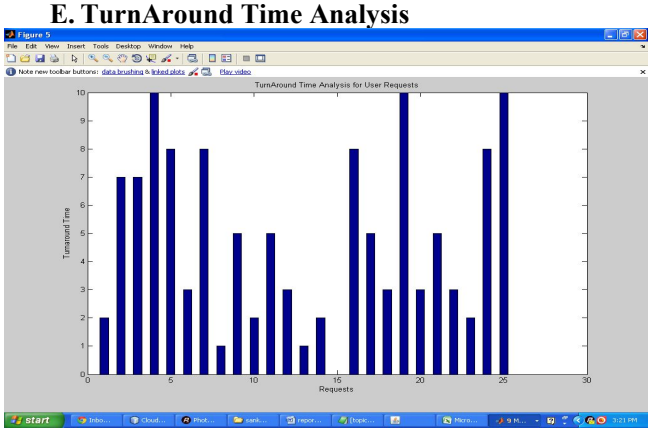


Fig: e) TurnAround Time Analysis

Here figure e) is showing the turnaround time anlaysis for 20 input user requests. Here x axis represents the number of user requests and y axis represents the turn around time in seconds. The figure shows the process time is between 0 and 10.

F. Finish Time Analysis

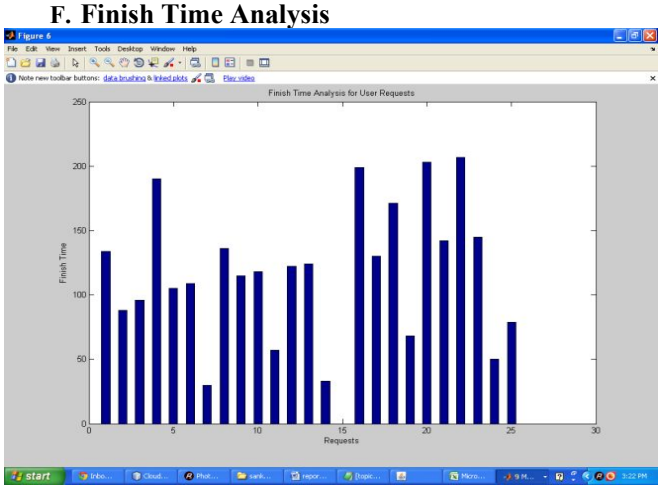


Fig: f) Finish Time Analysis

Here figure f) is showing the turnaround time anlaysis for 20 input user requests. Here x axis represents the number of user requests and y axis represents the finish time in seconds. The figure shows the finish time is between 0 and 140.

G. Start Time Vs. Arrival Time

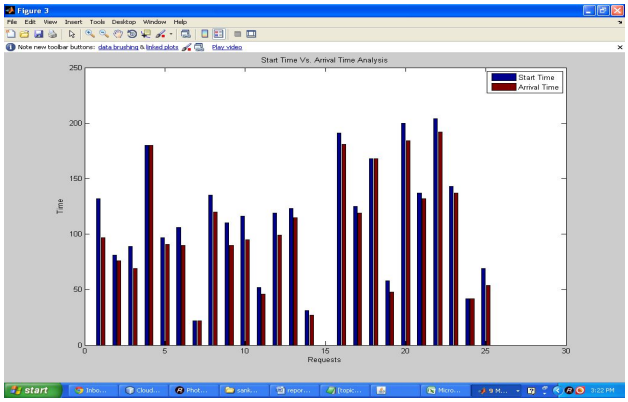


Fig: g) Start Time Vs. Arrival Time

Here figure g) is showing the start time and arrival time anlaysis for 20 input user requests. Here x axis represents the number of user requests and y axis represents the start and arrival time differnce in seconds. The figure shows the most or processes are executed without much delay.

CONCLUSION

In this present work the resource allocation scheme is based on multiple Clouds in both the under load and the over load conditions. As the request is performed by the user, certain parameters are defined with each user request, these parameters includes the arrival time, process time, deadline and the input output requirement of the processes. The Cloud environment taken in this work is the public cloud environment with multiple clouds. Each Cloud is here defined with some virtual machines. To perform the effective allocation, we have assigned some priority to each cloud. The virtual machines are here to perform the actual allocation. These are defined with certain limits in terms of memory, load etc. As the allocation begins, at first the scheduling of the processes is performed respective to the memory requirements. And along with it, the allocation of the process is done to the Cloud based on the requirement and the availability analysis. If the allocated process cannot be executed in its required time slot, in such case the migration of the process is required. The migration of the processes is here defined in case of overload conditions. The overload condition is defined in terms of simultaneous processes that are required to execute at particular instance of time. The analysis of the work is done in terms of wait time, process time of the processes. The obtain results shows the successful execution of all the processes within time limit. The work is performed on a generic system that can have n number of Clouds.

FUTURE WORK

The presented work is about to perform the scheduling and the allocation of the processes to the clouds in case of under load and overload conditions. In case of over load condition, the migration of the processes is performed from one cloud to other. The Future enhancement of the work is possible in the following directions

1. The presented work is defined the overload conditions in terms of 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. Hwanju Kim, Task-aware Virtual Machine Scheduling for I/O Performance, Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments, pp 101-110, 2009
2. Zheng Hu, Kaijun Wu, Jinsong Huang, An Utility-Based Job Scheduling Algorithm for Current Computing Cloud Considering Reliability Factor, International conference on Software Engineering and Service Science, pp 296 – 299, 2012.
3. Anthony T.Velte, Toby J.Velte, Robert Elsenpeter, Cloud Computing, A Practical approach, CRC Press/Taylor & Francis, 304p, ISBN: 978-1466593169.
4. Anton Beloglazov, Energy Efficient Allocation of Virtual Machines in Cloud Data Centers, IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, pp 577 – 578, 2010.
5. Xun Xu, from cloud computing to cloud manufacturing, Robo Compute. Integr, Manuf, 28, PP. 75–86, 2011.
6. Kento Sato, A Model-Based Algorithm for Optimizing I/O Intensive Applications in Clouds using VM-Based Migration, Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, pp 466-471 , 2009.
7. Takahiro Hirofuchi, Enabling Instantaneous Relocation of Virtual Machines with a Lightweight VMM Extension, Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, pp 73-83, 2010.
8. J. Brandt, Using Cloud Constructs and Predictive Analysis to Enable Pre-Failure Process Migration in HPC Systems, IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, pp 703 – 708, 2010.