An Improvised Scheduling Algorithm in Cloud Environment

Snehal Bajaj
M. Tech.
Dept. of Computer Science
Uttarakhand Technical University
Dehradun, India

Astha Sharma
Assistant Professor
Dept. Computer Science
Uttarakhand Technical University
Dehradun, India

ABSTRACT

The term scheduling implies relegating of the responsibilities to the accessible assets in some model design to finish the entire work. The target of the proposed work is to investigate the existing weighted round robin algorithm and propose a credulous methodology that defeats the downside of the existing algorithm and by consolidating both the analysis and make an enhance model which is more efficient and satisfy the user needs. The existing algorithm is not productive because of vast reaction time, high completion time, extensive turnaround time, high no. of task migration. Objective of this work is to evaluate the proposed scheduling algorithm by considering the capacities of the VM. The proposed algorithm also holds the benefits of the existing and defeating the issues. The algorithm has been compared with Weighted Round Robin(WRR) and Length based WRR it was observed that the Proposed WRR performed better than existing WRR and LWRR. Proposed WRR showed 99% improvement in Finish time over WRR. 20% and 40% improvement was observed in Task migration and Task delayed respectively over LWRR.

Keywords

Cloud computing, Scheduling, Virtual Machine, Load balancing.

1. INTRODUCTION

Cloud computing is a type of computing paradigm based on internet for providing, managing and delivering services over the internet on their demand and is defined as, “A model for enabling ubiquitous and on-demand network access to shared pool of configurable computing resources such as networks, servers, data centers, applications, and services which can be rapidly provisioned and released with minimum efforts and service provider interaction”. Cloud computing is an integrated concept of distributed and parallel computing that shares resources like hardware, software and information with computers or other devices on their demand.

Cloud computing can be defined as internet based computing in which different services like storage, servers and application are provided to consumers and organization using internet. One of the major challenges is to balance the load, as the traffic is increasing due to highly demand of services. The load balancing is defined as distribution of load among various nodes such that it improves resource utilization and job response time.

The concept of virtual machine (VM) act as a building block in cloud computing. It is considered as an execution unit which is base foundation for cloud. By VM it is meant that virtualization of host in creation, execution and managing of some application. As load balancing is the most challenging concept in cloud, load balancing means balancing of the load that is the balancing of task which are incident on a VM or network. Therefore, there is need to design an algorithm which can eliminate this load issue.

Here the objective is to assign the incoming task to the most appropriate virtual machine (VM).The request from the client is given to the data centers which are in cloud environment. These requests are then intern the directed to the host from the data centers. There are mostly two scheduling algorithm which are in use at most. One is round robin algorithm, where the allocation of task occurs to the next VM who is in the queue irrespective of the load on that VM. Hence the Round Robin does not consider the resource capabilities, priority and the length of the tasks (completion time). So, it ends up with higher response times. Therefore the need for more improved algorithm was felt and weighted round robin was discovered. Unlike round robin the weighted round robin considers the resource capabilities of the VMs and their weights and assigns higher number of tasks to the higher capacity VMs based on the weightage that each of the VMs is given.

Our work deals with this weighted round robin in more improved way so as to optimize the performance of VM and the completion time of tasks. And the most important to avoid the overloading of VM.

In this paper weighted round robin algorithm has been considered and a new improvised WRR i.e PWRR has been proposed which uses bitwise concept to optimize the completion time and reduce the overhead. The paper is further organized as the next section 2 discusses the brief survey on related work, section 3 explains the designing of scheduling and load balancing. Section 4 gives the proposed algorithm and the flow chart and the last section gives the result and conclusion.

2. LITERATURE SURVEY

[2] The paper gives the impact on performance and the significance in optimizing the performance and reduction in load and power reduction in data centers in cloud computing environment and the paper discusses the queuing model for a group of heterogeneous resources or multi-core servers with variant sizes and speeds as discussed. More importantly it focuses on addressing the problem of allocation of optimal power and how to distribute load for multiple heterogeneous multi-core server processors across the clouds and their data centers. Nevertheless, it is only a feasibility study for modeling power.

The paper considers the elasticity of cloud infrastructures which gives a suitable platform for execution of application with deadline-constrained workflow [9]. The paper deals with the issues how to mitigate the effects of performance in variation of resources on soft deadlines given by the workflow applications hence, design of an effective algorithm that uses the idle time of the provided resources and surplus the budget to replicate the proposed tasks. The result of which reduces the total execution time of the applications and the budget for replication increases. Static job arrival is also modeled whereas, overhead of duplicate executions and jobs runtime is not taken into account.

“Skewness” [10] in paper is the metric which is used to measure the unevenness of server along with multidimensional resource utilization. By minimizing the skewness in the paper, the workloads of variant that is different type have been combined to improve the overall utilization of server resources. The work in this paper contributes that developed a resource allocation system can avoid the overloading of the system effectively and the system that can also minimize the number of servers used. The paper has proposed a load prediction algorithm that can capture the future resource requirement of applications accurately without looking of details inside the VMs. The algorithm captures the trend of resource usage patterns and it helps reduce the churn significantly but QoS parameters like response time or completion time of tasks are not discussed. Load balancing in Cloud computing using Stochastic Hill Climbing- A Soft Computing Approach [1] Here, in this paper a Stochastic hill climbing approach was used for distribution of load in cloud computing, in which the soft computing based approach has been compared to round robin and First Come First Serve. There is a local optimization approach which is used here called Stochastic Hill climbing that was used here for allocation of incoming jobs to the servers or virtual machines (VMs). The performance here is analyzed by both qualitatively and quantitatively. Honey bee behavior is inspired load balancing of tasks in cloud computing environments [3]. In this paper, it has been examine the behavior of Honey Bee which inspired load balancing algorithm that was proposed with the aim to achieve well balanced load across virtual machines to maximize the throughput as well as to balance the priorities of the incoming tasks on the VMs. So as to minimize the amount of waiting time of the incoming tasks in the queue. By using this technique the average execution time and reduction in the waiting time of the incoming tasks on queue were improved.

3. DESIGN ASPECTS OF SCHEDULING AND LOAD BALANCING

The below Fig.1 explains the scheduling design where the scheduler contains the logic to find the suitable VM to assign the task to VMs. Load balancer is responsible for the migration of task from a heavily loaded VM to an idle VM during run time. It finds the idle VM by utilizing the resource monitor which contains current status information, it communicates with all VM and collects the VM capabilities, current load on each VM and no. of jobs in waiting queue in ach VM. The task requirement contains the length of task to be executed and transfer this to scheduler.

The user gives the job request through the interface to the task manager which checks the dependent and independent task analysis. If it is dependent task, then it verifies the interdependency between the multiple tasks. The dependent tasks are notified to the scheduler so that the parent tasks are scheduled only after the child tasks are executed. Dependent task queue contains those tasks which depend on the other tasks present in the VMs. Once the entire child tasks present completes its execution the parent task will be taken for the execution by assigning it to the VM. Independent task queue and dependent task are input to the scheduler. The scheduler selects the suitable VM based on WRR algorithm.

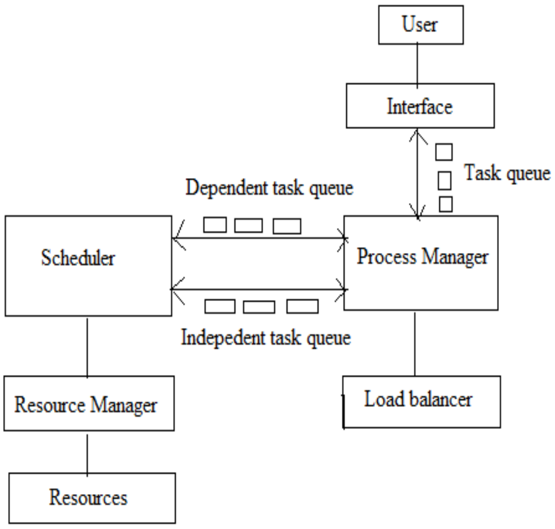

Fig.1: Load Balancing Design

The sum of loads of all virtual machines is given as –

$$S = \sum_{n=1}^{k} L_n \qquad (1)$$

Where, n represents the number of VMs in a data center.

The load per unit capacity is defined as –

$$LPC = \frac{S}{\sum_{n=1}^{k} C_n} \qquad (2)$$

$$\text{Threshold}, T_n = LPC * C_n$$

Where, $C_n$is the capacity of the node.

The load imbalance factor of a virtual machine is given by -

$$\text{If, VM} \begin{cases} < T_n - \sum_{i=1}^{n} S_i \;, & \text{Underloaded} \\ > T_n - \sum_{i=1}^{n} S_i \;, & \text{Overloaded} \\ = T_n - \sum_{i=1}^{n} S_i \;, & \text{Balanced} \end{cases} \qquad (3)$$

The reason of load imbalance factor is due to the migration of task from the overloaded VM to under-loaded VM, that occurs when the load on the overloaded VM drops below the threshold and it is under-loaded when the sum of loads is below the threshold value as mentioned in the equation (3).

4. PROPOSED SCHEDULING ALGORITHM

```
Step 1: Initialize all required variables
Step 2: Define Virtual machines, queues,    weights
Step 3: Lastfinished← list of last virtual finished instance
Step 4: Receive packets
                T← 1
                minTime = infinite
Loop while Number of packets in queue
```

```
            queue←queues(T)
              If
queueHeadVirtualTime<minTime
                                    \
Mintime←queueHeadVirtualTime

qNumber←T

            else
      terminate loop
                  end if
     T=T+1
            Endfor

Step 5: Update Timer
      vStart←max(now(),Lastfinished(qNumber))
packetVirtulFinish←PacketSize + VStart
     Lastfinished[qNumber]←packetVirtulFinish
```

The modeling of completion time is computationally intensive and feasible. The algorithm recomputed every time when a new task arrives into any queue. To reduce the computational load the concept of virtual time is introduced. Completion time for each task is computed on virtual timescale. The virtual time does not accurately model the time that task completes, it only accurately model the order in which the task must assign to meet the objectives of the full-featured model. By virtual time concept it is unnecessary to recomputed the finished time for previously queued tasks. Although the finish time, for existing task can be affected by new arrivals but finish time on the virtual time line is unchanged. The virtual finish time for a newly queued process is given by the sum of the virtual start time and also the process's size. The virtual start time can be defined as the maximum of the previous virtual finish time and the current instant. The task with the minimum virtual finishing time is allocated the suitable VM.

This algorithms goal is to emulate the process of bitwise round-robin algorithm by sharing of the pool of resources among competing process. Packet-based flows, however, must be transmitted packet wise and in sequence. The queuing choose the task order for the processes by modeling the finish time for each of the task as if they would be transmitted bitwise round robin. The process with the earliest finish time according to this modeling is the next selected.

The complexity of this algorithm is O(log(n)), where n is the number of flows.

5. FLOW CHART

The flow chart shows the flow of working of the algorithm diagrammatically all the factors. The list of elements requires are listed in table below –

Table.1: List of entities required

Serial No.	Entity	Quantity
1	Data Center	2
2	No. of Hosts	600
3	No. of Process Elements	8/16/32
4	Host RAM Capacity	32/64 GB
5	No. of VM	10-10-100
6	VM RAM Capacity	4 GB

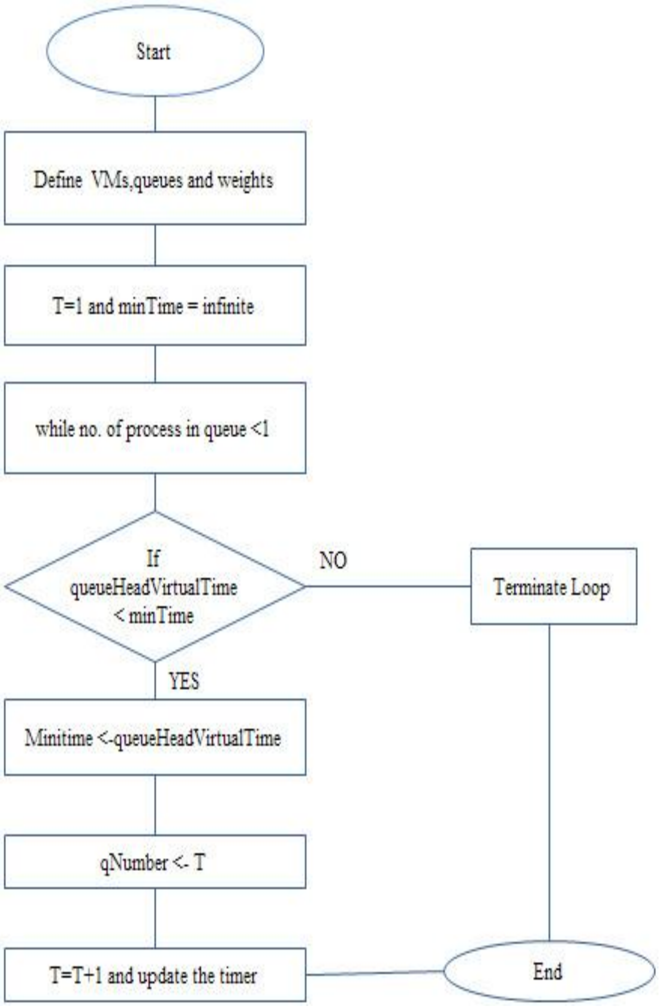
Start
Define VMs,queues and weights
T=1 and minTime = infinite
while no. of process in queue <1
If queueHeadVirtualTime < minTime
NO
Terminate Loop
YES
Minitime <-queueHeadVirtualTime
qNumber <- T
T=T+1 and update the timer
End

Fig.2: Flow chart of proposed algorithm

6. RESULT AND PERFORMANCE ANALYSIS

The performance of this algorithm has been analyzed based on the results of simulation

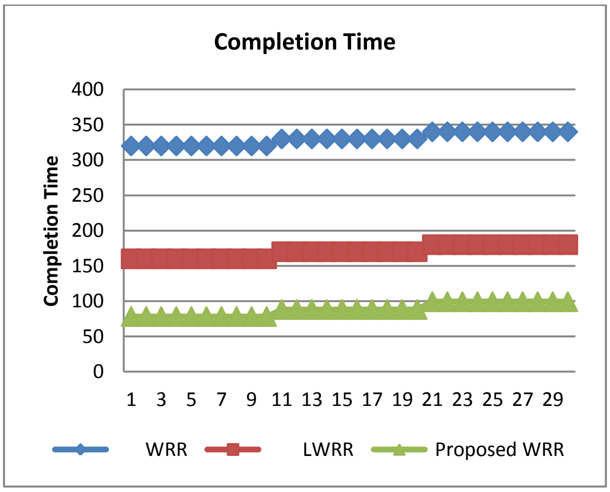

Fig.3: Overall Completion Time

The above Fig.3 proves that the proposed WRR (PWRR) delivers a faster completion time than the other two load balancing algorithms that is WRR and LWRR in the heterogeneous resources. In this algorithm more number of jobs gets assigned to the higher capacity VMs in the homogenous jobs on heterogeneous environments which helps in completing the job in a shorter time.

6.1 Comparison of Task Migration

As due to capability of static and dynamic scheduler in allocating the most appropriate VM to each task, there is much more migrations of task in the proposed WRR (PWRR) as well as the length based WRR algorithm also gives much task migration compare to WRR. Whereas, in WRR the load balancer shows less task migration in order to complete the work as the Fig 4 below shows that the WRR shows the maximum task migration and the proposed algorithm shows the zero migration.

6.2 Comparison of Delayed Tasks

Here, the proposed WRR (PWRR) algorithm shows lesser number of task delayed than the other two algorithms, as analyzed from Figure.5. This result is due to the allocation of more and more tasks to the lesser capacity of VMs then they need not have to wait in the queue.

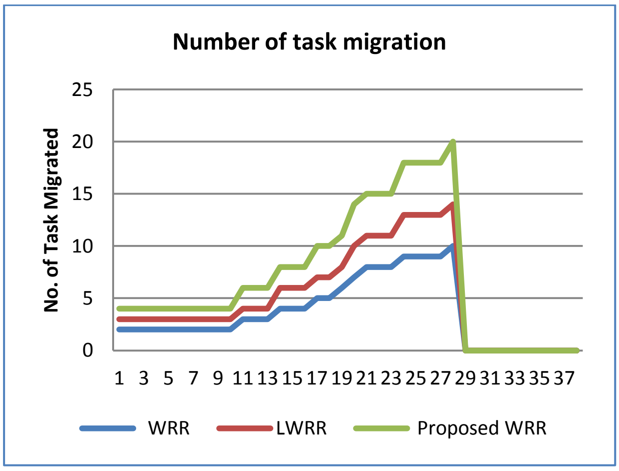

Fig.4: Number of task migration

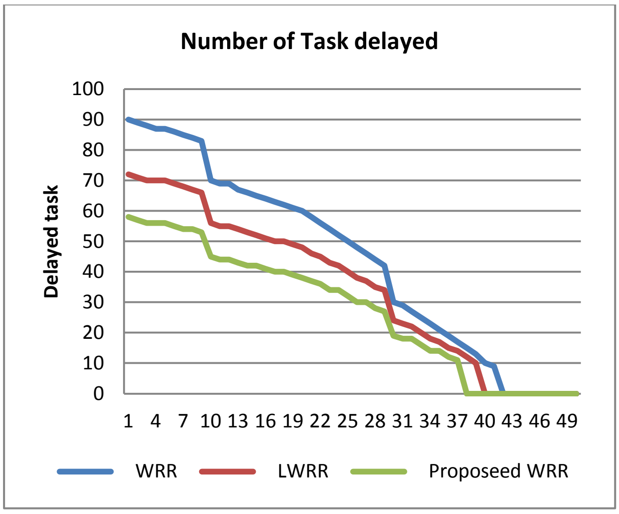

Fig.5: Number of task delayed

7. CONCLUSION

In this work, among various scheduling algorithm the weighted round robin is considered and improved by considering the capacities of each VM so as to assign job to the most suitable VM and minimize the completion time of the task running. The performance analysis and the experiment result of this proposed algorithm is compared with the existing WRR and the LWRR and the results given by the proposed system are more than 99% therefore, we can say that the proposed system enhances the performance by giving the least possible completion time as compared to the others. In future by increasing the capacities of VM or either by increasing the data center or VM more enhanced result can be obtained.

8. REFERENCES

[1] B. Mondal, K. Dasgupta, and P. Dutta, “Load balancing in cloud computing using stochastic hill climbing-a soft computing approach,” Procedia Technology, vol. 4, pp. 783–789, 2012.

[2] F. Xhafa and A. Abraham, Mata-Heuristics for Grid Scheduling Problems, Springer, Berlin, Germany, 2008.

[3] G. Gharooni-fard, F. Moein-darbari, H. Deldari, and A. Morvaridi, “Scheduling of scientific workflows using a chaos genetic algorithm,” Procedia Computer Science, vol. 1, no. 1, pp. 1445–1454, 2010, International Conference on Computational Science, ICCS 2010.

[4] H. M. Fard and H. Deldari, “An economic approach for scheduling dependent tasks in grid computing,” in Proceedings of the 11th IEEE International Conference on Computational Science and Engineering (CSE Workshops ’08), pp. 71–76, IEEE, San Paulo, Brazil, July 2008.

[5] J. Cao, K. Li, and I. Stojmenovic, “Optimal power allocation and load distribution for multiple heterogeneous multicore server processors across clouds and data centers,” IEEE Transactionson Computers, vol. 63, no. 1, pp. 45–58, 2014.

[6] L. D. Dhinesh Babu and P. Venkata Krishna, “Honey bee behavior inspired load balancing of tasks in cloud computing environments,” Applied Soft Computing Journal, vol. 13, no. 5, pp. 2292–2303, 2013.

[7] L.-T. Lee, C.-W. Chen, H.-Y. Chang, C.-C. Tang, and K.-C. Pan, “A non-critical path earliest-finish algorithm for interdependent tasks in heterogeneous computing environments,” in Proceedings of the 11th IEEE International Conference on High Performance Computing and Communications (HPCC ’09), pp. 603–608, Seoul, Republic of Korea, June 2009.

[8] M. Rahman, R. Hassan, R. Ranjan, and R. Buyya, “Adaptive workflow scheduling for dynamic grid and cloud computing environment,” Concurrency and Computation: Practice and Experience, vol. 25, no. 13, pp. 1816–1842, 2013.

[9] Mohammadreza Mesbahi, Amir Masoud Rahmani, “Load Balancing in Cloud Computing: A State of the Art Survey ”I.J. Modern Education and Computer Science, 2016, 3, 64-78.

[10] M. Xu, L. Cui, H. Wang, and Y. Bi, "A multiple QoS constrained scheduling strategy of multiple workflows for cloud computing," in Proceedings of the IEEE International Symposium onParallel and Distributed Processing with Applications (ISPA '09), pp. 629–634, IEEE, Chengdu, China, August 2009.

[11] R. Basker, V. Rhymend Uthariaraj, and D. Chitra Devi, "An enhanced scheduling in weighted round robin for the cloud infrastructure services," International Journal of Recent Advance in Engineering & Technology, vol. 2, no. 3, pp. 81–86, 2014.

[12] R. N. Calheiros and R. Buyya, "Meeting deadlines of scientific workflows in public clouds with tasks replication," IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 7, pp. 1787–1796, 2014.

[13] S. Ghanbari and M. Othman, "A priority based job n scheduling algorithm in cloud computing," in Proceedings of the International Conference on Advances Science and Contemporary Engineering, pp. 778–785, October 2012.

[14] Z. Xiao, W. Song, and Q. Chen, "Dynamic resource allocation using virtual machines for cloud computing environment," IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 6, pp. 1107–1117, 2013.

[15] Vijindra and S. Shenai, "Survey on scheduling issues in cloud computing," Procedia Engineering, vol. 38, pp. 2881–2888, 2012,Proceedings of the International Conference on Modelling Optimization and Computing.

[16] Z. Yu, F. Menng, and H. Chen, "An efficient list scheduling algorithm of dependent task in grid," in Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT'10), IEEE, Chengdu, China, July2010.