Journal of Engineering and Applied Sciences 12 (Special Issue 10): 8944-8948, 2017
ISSN: 1816-949X
© Medwell Journals, 2017

Towards Energy Saving with Smarter Multi Queue Job Scheduling Algorithm in Cloud Computing

Jaspreet Singh and Deepali Gupta
Department of Computer Science and Engineering,
Maharishi Markandeshwar University, Sadopur, Ambala, Haryana, India

Abstract: The cloud environment provides a cluster of interconnected virtualized systems which are progressively provisioned and recognized as pooled computing resources. The primary objective of job scheduling in cloud is to accomplish a framework that achieve foremost system throughput. The adequacy of cloud environment is to great extent depends upon the proficiency of its scheduler. The cloud service provider implement a scheduler that designates the optimal job scheduling by effectively distributing and executing cloud user jobs on available resources in such a manner as to limit execution time, reduce energy consumption, overall cost and various other factors that can dominate the performance in cloud environment. So, keeping in mind these performance parameters we analyzed and implemented a smarter multi queue job scheduling algorithm for cloud computing. This algorithm accommodate to virtualization trait of cloud computing, vary from traditional job scheduling algorithms character-concentrate on energy efficiency and maintain optimal resource fairness under the cloud computing. To assess the performance of the proposed algorithm we compared it with efficient multi queue job scheduling technique which is presented by researcher in recent past. Through, the experiments, the smarter multi queue scheduling algorithm has demonstrated that it can intensify the performance by effectively reducing the energy consumption by 41.22%. Finally, the comparative simulations and numeral results of both strategies on different number of user jobs indicates that our recommended smarter MQS approach is much more oriented towards energy saving and to some extent reduce execution time in cloud environment.

Key words: Smarter multi queue job scheduling, energy efficient scheduling in cloud computing, job scheduling in cloud computing, reducing, parameters, consumption

INTRODUCTION

Cloud computing have provided a more nimble and versatile service oriented architecture for the end user to run their services. The cloud computing deals with highly dynamic environment which enable users to plug in from anywhere and make use of computing resources which own huge scale storage. In order to achieve high performance in cloud platform an excellent job scheduling strategy is required which outperforms suitable mapping of cloud user's jobs to computing resources to maximize profit and provide powerful system flexibility.

Every enterprise whether small, medium or big want to take benefit offer by this technology for its business because cloud environment allows them to access several distinct applications that actually present on multiple diverse locations from one's machine location by providing a separate virtualized platform that assist cloud users to execute their jobs with reduction in both energy consumption as well as job completion time.

So, cloud computing can be characterized as a model for empowering expedient on demand network access to a shared pool of configurable computing resources such as servers, storage, applications and services that can be quickly equipped and released with negligible management endeavor (Gurav *et al*., 2016). Cloud environment focus on commercial implementation of scientific concepts such as distributed, grid and parallel computing (Huang and Huang, 2010). The attribute of cloud can incorporate ultra large scale, virtualization, high reliability, versatility on demand service, high extendibility and extremely inexpensive (Wang *et al*., 2011). The data centers who are equipped with hundreds to thousands of servers and which are based upon virtualized computation

Corresponding Author: Jaspreet Singh, Department of Computer Science and Engineering, Maharishi Markandeshwar University, Sadopur, Ambala, Haryana, India

and storage technologies lays the foundation for processing power in cloud computing (Cao and Zhu, 2013). By virtualization the clients request to cloud service provider regarding their requirement for processing of jobs on the pool of resources present in cloud, cloud service provider after checking for feasibility concur with clients regarding the acknowledgment to fulfill the requirement (George *et al*., 2015).

The contributions of our study are summarized as follows: first, our research proposes and implemented framework model to manage cloud user jobs and resources. The framework includes four major components as queue manager, scheduler, MPI table and network. The queue manager function is classify and manage cloud user jobs. Network contain cluster of virtual systems. The MPI table is used to collect the information of these vacant systems present in the network and maintain sorted list of vacant systems accordingly. The scheduler performs optimal job scheduling strategy by making use of these components.

Second, the study implemented a smarter multi queue scheduling method. The user process classification is done by establishing two job queues. The threshold prioritized value is calculated and based upon this value the job gets allocated in these both large and small queues.

Finally, we analyzed and executed distribution strategy for cloud user submitted jobs. The user processes present in both queues are picked and form series of merge task sets, a set contain two merge tasks (one task from small queue and other from large queue). Further in a sequence scheduler allocate these unmapped merge processes to available resources in optimal manner.

Literature review: Li (2009) worked on building non pre-emptive priority M/G/1 queuing model for task scheduling to meet the QOS requirements for cloud computing users. When tested technique yielded maximum profits to the cloud service provider. Rezaee *et al*. (2011) presented multi-level queue management technique. Weighted Fair Queue (WFQ) combined with fuzzy inference system to form Fuzzy Dynamic Weighted Fair Queue (FDWFQ) that schedules the incoming requests into different queues and specify a weight to each queue that determines the number of jobs to be scheduled next from that particular queue. The algorithm tends depicts improvement in system's loss ratio reduction. Yadav and Upadhayay (2012) put forwards a MLFQ (Multi-Level Feedback Queue) scheduling which divide the queue in three parts. All the user process gets executed in all three queues for a specific period of time. The algorithm achieve the success in minimizing the waiting and turnaround time but the concern related to CPU is that it have to wait to build a queue of all the processes which directly affect the utilization of resources. The research carried out by Xiao and Wang (2012) designed a technique which ranks all client request according to profit they give to service provider, the major goal of the algorithm is to boost the benefits for cloud service provider in case current resource are not sufficient to process all request in time. Vaithiyanathan *et al*. (2013), the efficient scheduling algorithm is proposed which produces scheduling order in which task selection is done on the basics of priority and deadline of the task. Bossche *et al*. (2013), the researchers proposed a cost effective online hybrid scheduling algorithm that consider computational cost, data transfer cost and network bandwidth constraints. The result shows that large numbers of application are scheduled within specified framework yielding additional gain in cost efficiency.

MATERIALS AND METHODS

Scheduling strategy in cloud computing

Description: In cloud computing job scheduling remains most demanding and crux issue because job scheduling simply can be viewed as set of user jobs versus set of available virtual system. All the constraints such as deadlines, energy efficiency, makespan, etc., defined by user input are considered by job scheduling framework.

Scheduling problem deals with ideal mapping of user tasks to virtual systems satisfying given constraints to improve some objective function (Shishira *et al*., 2016). As randomly huge numbers of jobs are submitted in cloud environment, so, scheduling become arduous task. The large number of scientific application in cloud environment categorized as dependent jobs, independent jobs and parallel job. Among these jobs, a task is defined as the interdependence of these independent job (Komarasamy and Muthuswamy, 2015) or a set of tasks frame a job in cloud computing (Santhosh *et al*., 2016). So, after submitting application to cloud it breaks up into several task and then scheduler make several decision on allocation of resources to tasks, deciding the order of execution and managing the overhead related to virtual machines (Li *et al*., 2010).

Problem definition: As the efficient multi queue scheduling technique devised by Karthick *et al*. (2014) which works by dividing the user jobs in small, medium and large queue then carry dynamic selection of jobs from these queues to allocate them for processing. No doubt this strategy removes the starvation problem associated with traditional algorithm FCFS, SJF, EASY, etc. but to handle multiple queues (three queues), scheduler have to switch from one queue to another to process user job, therefore, this switching can affect some ready jobs to

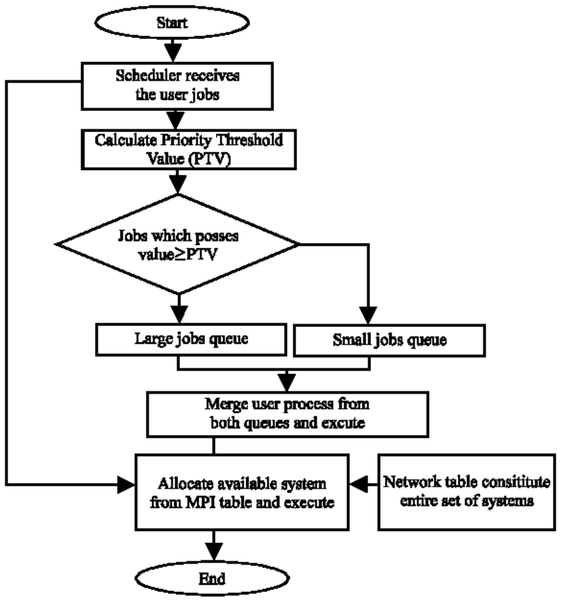
Start
Scheduler receives the user jobs
Calculate Priority Threshold Value (PTV)
Jobs which posses value≥PTV
Large jobs queue
Small jobs queue
Merge user process from both queues and excute
Allocate available system from MPI table and execute
Network table consititute entire set of systems
End

Fig. 1: Flowchart

wait in queue for some specific time recognized as job idle time till scheduler approaches them to map resources for them which overall leads to increase in energy consumption and job completion time. Further, dynamic selection has high probability for indefinitely postponement of the user process, so, the processes miss their deadlines. Smarter MQS technique is an alternate method in which user jobs is categorized in shorter number of queue, i.e., two queue small and large queues followed job distribution by formation of merge pattern (consisting of set of merge user task from both queues), so, it decreases the energy consumption in significant manner and also limits the job completion time.

Scheduling algorithm process and flowchart: The flow chart depicting the whole scheduling process is given in Fig. 1. The summarized working process for proposed smarter MQS approach described as:

Step1: After receiving the cloud user jobs the cloud scheduler calculates the priority threshold value. Now, on basis of this value the categorization of user job is done in small job and large job queues.

Step 2: Further, the processes are merged from both queues forming a merge pattern containing number of merge user processes set for execution.

Step 3: Scheduler by referring to MPI table maps the merged user processes for the execution to the available systems in the cloud environment.

Pseudo code of smarter MQS

```
• Let the input jobs = {$T_1$, $T_2$, $T_3$, $T_4$, ..., $T_n$}
• Let the network nodes = {$Sys_1$, $Sys_2$, $Sys_3$, ..., $Sys_n$}
• Create jobs matrix based on various processing factors
• For each job processing do
• Create the threshold point for prioritized group of jobs. Multi_queues = Break.Prioritized (Based on energy(threshold))
• End job_processing
• Initialize system network with MPI network message
• MPI response table (acknowledgement nodes.add);
• If job ready (Exists in execution matrix's)
• Load network configurations
• Distribute jobs through divide and merge policy
• Eliminate executed jobs from network processing end_if
• End repeat till all jobs will not get executed
• Evaluate system performance
• Stop
```

Calculation of user jobs: When the cloud user submits the jobs for processing the energy consumption, execution time for jobs and overall can be calculated below mentioned equations:

Calculative equations:

$$\text{Energy consumption } \rho = \sum_{p}^{s} (\text{Energy consumption}_{pi} + (\text{Wating time}_{pi} \times \text{Ideal energy}_{pi}))_s$$

$$\text{Time consumption } \rho = \sum_{p}^{s} (\text{Time consumption}_{pi} + (\text{Wating time}_{pi})_s$$

$$\text{Overall cost} = \frac{\text{Budget}}{\sum_{i=1}^{n} \text{Energy}(J)_i}$$

RESULTS AND DISCUSSION

Experimental scenario and performance metrics: To evaluate the performance of our smarter multi queue model (smarter MQS), it has been simulated in the cloud environment to find best schedule to process multiple user jobs on available virtual systems. A number of jobs with different attributes are created randomly and also a number of processing unit having random attributes are created randomly. We have examined the overall performance of our proposed smarter MQS technique by

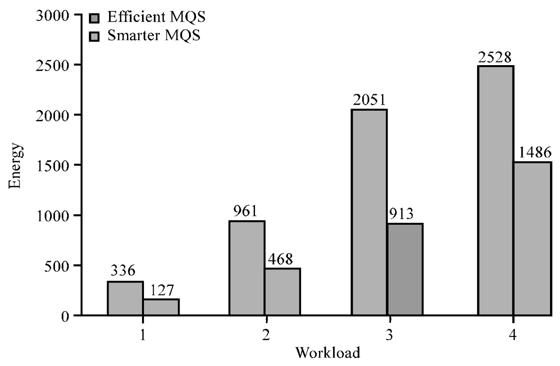

Fig. 2: Energy consuption comparison for both alghorithms (No. of jobs: 5, 10, 15, 20)

Table 1: Energy consuption (J)

No. of jobs	Efficient MQS	Smarter MQS
5	336	127
10	961	468
15	2051	913
20	2528	1486

Table2: Time (msec)

No. of jobs	Efficient MQS	Smarter MQS
5	31	29
10	68	66
15	76	75
20	109	92

running it under randomly generated cloud configurations. The simulation is carried on real time environment and for instance, N different user jobs (5, 10, 15 and 20) possessing different characteristics are created randomly. The simulation environment is carried out in. Net framework 4.5 as frontend while SQL server 2005 at backend. In this real time simulation environment the user's job creation and execution order may vary on each iterations conducted, so as the parameter (energy, time) values used for results calculation. But our simulation results concluded that our suggested smarter MQS technique on every iteration conducted will achieve better results showing sharp drop in energy consumption and more often in most of iteration limits execution time when compared with the efficient MQS technique.

Experimental results and discussion: We analysed the results by demonstrating rise in performance, sharp reduction in energy consumption, limiting execution time to some degree, minimizing cost and improvement of overall end user jobs experience. The scheduler achieved its objective to reproduce the cloud service performance by handling the workload generated by cloud user jobs efficiently. Table 1 and 2 shows the desired result values our two approaches and further the plot in Fig. 2 and 3 based upon energy consumption and execution time

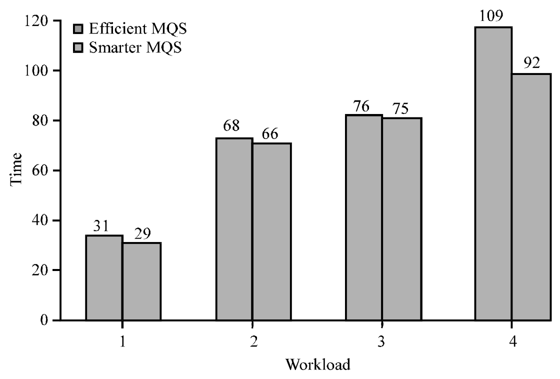

Fig. 3: Time consumption comparison for both algorithms (No. of jobs: 5, 10, 15, 20)

compares our proposed smarter MQS technique with efficient MQS technique in cloud environment, clearly experimental results demonstrate the effectiveness of our smarter MQS algorithm over efficient MQS algorithm.

CONCLUSION

In recent years job scheduling in cloud environment has attracted a lot of interest. As job scheduling is considered as major activity in the cloud because it makes direct impact on the performance of system. So, the key idea of our work is to gain the maximum profit by reducing energy consumption and limit execution time to some extent for cloud user's jobs by implementing effective job scheduling algorithm. To achieve this, in this study we have successfully employed the smarter multi queue scheduling algorithm to handle job scheduling problem in cloud computing environment. Our proposed strategy smarter MQS has consolidated the jobs from multiple queues to form merge pattern and then distributes the merge pattern of these cloud user jobs for execution to available virtual systems in an optimized manner. Experimental results demonstrated that our implemented smarter MQS technique outperforms the existing effective MQS technique in cloud environment. With such improvements, the proposed smarter MQS algorithm should be merged in the existing cloud infrastructures in order to improve their performance in terms of energy and time.

RECOMMENDATIONS

Future developments can be done on improving aspects of the cloud computing simulation by adding new parameters and also an strategy can be devise which can work on optimize the formation merge pattern of user tasks for allocation to virtual system in order to get more delightful results.

REFERENCES

Bossche, R.V.D., K. Vanmechelen and J. Broeckhove, 2013. Online cost-efficient scheduling of deadline-constrained workloads on hybrid clouds. Future Generation Comput. Syst., 29: 973-985.

Cao, F. and M.M. Zhu, 2013. Energy efficient workflow job scheduling for green cloud. Proceedings of the 2013 IEEE 27th International Processing Symposium Workshops on Parallel and Distributed (IPDPSW'13), May 20-24, 2013, IEEE, Cambridge, Massachusetts, USA., ISBN:978-0-7695-4979-8, pp: 2218-2221.

George, N., K. Chandrasekaran and A. Binu, 2015. Determination of task scheduling mechanism using computational intelligence in cloud computing. Proceedings of the 2015 International Conference on Computing and Network Communications (CoCoNet'15), December 16-19, 2015, IEEE, Trivandrum, India, ISBN:978-1-4673-7209-1, pp: 401-407.

Gurav, P.D., M.R. Hans and N.A. Kulkarni, 2016. Role of cloud computing in grid empowerment. Proceedings of the 2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT'16), September 9-10, 2016, IEEE, Pune, India, ISBN:978-1-5090-2081-2, pp: 258-263.

Huang, Q.Y. and T.L. Huang, 2010. An optimistic job scheduling strategy based on QoS for cloud computing. Proceedings of the 2010 International Conference on Intelligent Computing and Integrated Systems (ICISS'10), October 22-24, 2010, IEEE, Guilin, China, ISBN:978-1-4244-6834-8, pp: 673-675.

Karthick, A.V., E. Ramaraj and R.G. Subramanian, 2014. An efficient multi queue job scheduling for cloud computing. Proceedings of the 2014 World Congress on Computing and Communication Technologies (WCCCT'14), February 27-March 1, 2014, IEEE, Trichirappalli, India, ISBN:978-1-4799- 2877-4, pp: 164-166.

Komarasamy, D. and V. Muthuswamy, 2015. Adaptive deadline based dependent job scheduling algorithm in cloud computing. Proceedings of the 7th International Conference on Advanced Computing (ICoAC'15), December 15-17, 2015, IEEE, Chennai, India, ISBN:978-1-5090-1934-2, pp: 1-5.

Li, J., M. Qiu, J. Niu, W. Gao and Z. Zong *et al*., 2010. Feedback dynamic algorithms for preemptable job scheduling in cloud systems. Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT'10) Vol. 1, August 31-September 3, 2010, IEEE, Toronto, Ontario, ISBN:978-1-4244-8482-9, pp: 561-564.

Li, L., 2009. An optimistic differentiated service job scheduling system for cloud computing service users and providers. Proceedings of the 3rd International Conference on Multimedia and Ubiquitous Engineering (MUE'09), June 4-6, 2009, IEEE, Qingdao, China, ISBN:978-0-7695-3658-3, pp: 295-299.

Rezaee, A., S. Adabi, A.M. Rahmani and S. Adabi, 2011. A fuzzy algorithm for adaptive multilevel queue management with QoS feedback. Proceedings of the 2011 International Conference on High Performance Computing and Simulation (HPCS'11), July 4-8, 2011, IEEE, Istanbul, Turkey, ISBN:978-1-61284-380-3, pp: 121-127.

Santhosh, B., D.H. Manjaiah and L.P. Suresh, 2016. A survey of various scheduling algorithms in cloud environment. Proceedings of the International Conference on Emerging Technological Trends (ICETT'16), October 21-22, 2016, IEEE, Kollam, India, ISBN:978-1-5090-3752-0, pp: 1-5.

Shishira, S.R., A. Kandasamy and K. Chandrasekaran, 2016. Survey on meta heuristic optimization techniques in cloud computing. Proceedings of the 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI'16), September 21-24, 2016, IEEE, Jaipur, India, ISBN: 978-1-5090-2030-0, pp: 1434-1440.

Vaithiyanathan, V., R.A. Kumar, S. Vignesh, B. Thamotharanb and B. Karthikeyan, 2013. An efficient TPD scheduling algorithm for cloud environment. Int. J. Eng. Technol., 5: 2030-2035.

Wang, X., B. Wang and J. Huang, 2011. Cloud computing and its key techniques. Proceedings of the 2011 IEEE International Conference on Computer Science and Automation Engineering (CSAE'11) Vol. 2, June 10-12, 2011, IEEE, Shanghai, China, ISBN:978-1-4244-8727-1, pp: 404-410.

Xiao, J. and Z. Wang, 2012. A priority based scheduling strategy for virtual machine allocations in cloud computing environment. Proceedings of the 2012 International Conference on Cloud and Service Computing (CSC'12), November 22-24, 2012, IEEE, Shanghai, China, ISBN:978-1-4673-4724-2, pp: 50-55.

Yadav, R.K. and A. Upadhayay, 2012. A fresh loom for multilevel feedback queue scheduling algorithm. Intl. J. Adv. Eng. Sci., 2: 21-23.