IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 3, March 2017

Resource Allocation Approach using Gang Scheduling: Design and Analysis

Swati Dwivedi[1], **Anand Muni Mishra**[2]

Dept. of Computer Science & Engineering, JNCT, REWA (M.P.)[1, 2]

Abstract: Cloud computing is a computational infrastructure that become popular due to it's computational abilities. In this environment the huge amount of traffic and data is deal with the computational resources to solve the computational complexities. In order to deal with such kind of traffic and data the resource scheduling is an essential component of cloud server management. Resource scheduling is a technique for maximizing the utilization of expensive resources and minimizing the consumption. In this presented work the cloud resources scheduling approaches are investigated. Additionally a gang scheduling technique based approach is presented for improving the performance of multi-processor cloud servers. In order to optimize the performance of gang scheduling approach the priority normalization is performed before developing the gangs of processes. Additionally the priority is categorized on five different classes. Here the higher priority jobs are selected and processed first. The gang scheduling is utilized with the time shared processors with parallel execution of jobs. Therefore the simulation is configured for multi-processor technique. The simulation of the proposed scheduling technique is performed on the basis of CloudSim tool kit. Additionally the performance of the scheduling technique is compared with the FCFS technique. The experimental results demonstrate the proposed approach is improved as compared to FCFS technique and consumes fewer resources as compared to traditional technique.

Keywords: VM Scheduling, Gang Scheduling, FCFS, Resource Allocation, Cloud Computing, Processes.

I. INTRODUCTION

Nowadays, the service quality requirements of users and institutions increased. Thus, computer systems that provide many services (particularly, parallel machines) need to be highly utilized and provide a short response time for users jobs. Parallel job schedulers should match both requirements and workload (jobs) with resource availability (architecture, processors etc.) in order to maximize the system's performance. To meet the increasing demand for computing resources, the size and complexity of today's data centers are growing rapidly. At the same time, cloud computing infrastructures are becoming more popular. Resources in a cloud computing infrastructure may be managed in a cost-effective manner. Static resource allocation based on peak demand is not cost-effective because of poor resource utilization during off-peak periods. In contrast, autonomic resource management could lead to efficient resource utilization and fast response in the presence changing workloads [1] [2].

A. Scheduling in Cloud Computing

Scheduling allows optimal allocation of resources among given tasks in a finite time to achieve desired quality of service. Formally, scheduling problem involves tasks that must be scheduled on resources subject to some constraints to optimize some objective function. The aim is to build a schedule that specifies when and on which resource each task will be executed [3]. In real time systems, a scheduling policy is responsible not only for ordering the use of systems resources. It should also be able to predict the worst-case behavior of the system when the scheduling algorithm is applied.

B. Virtual Machine (VM) Scheduling

Virtualization plays an important role in providing resources to the users efficiently in cloud environment. Virtualization can be done in various ways like server virtualization, memory virtualization, storage virtualization, etc. For efficiently achieving virtualization, virtual machines (VM) are designed. Although virtual machine (VM) migration has been used to avoid conflicts on traditional system resources like CPU and memory, micro-architectural resources such as shared caches, memory controllers, and non-uniform memory access (NUMA) affinity, have only relied on intra-system scheduling to reduce contentions on them.

In cloud systems based on virtualization, virtual machines (VM) share physical resources. Although resource sharing can improve the overall utilization of limited resources, contentions on the resources often lead to significant performance degradation. To mitigate the effect of such contentions, cloud systems use dynamic rescheduling of VMs with live migration technique [4], changing the placement of running VMs. However, such VM migration has been used to resolve conflicts or balance load on traditional allocatable system resources such as CPUs, memory, and I/O sub-systems. VM migration can be triggered by monitoring the usages of these resources for VMs in a cloud system [5, 6].

DOI10.17148/IJARCCE.2017.6385

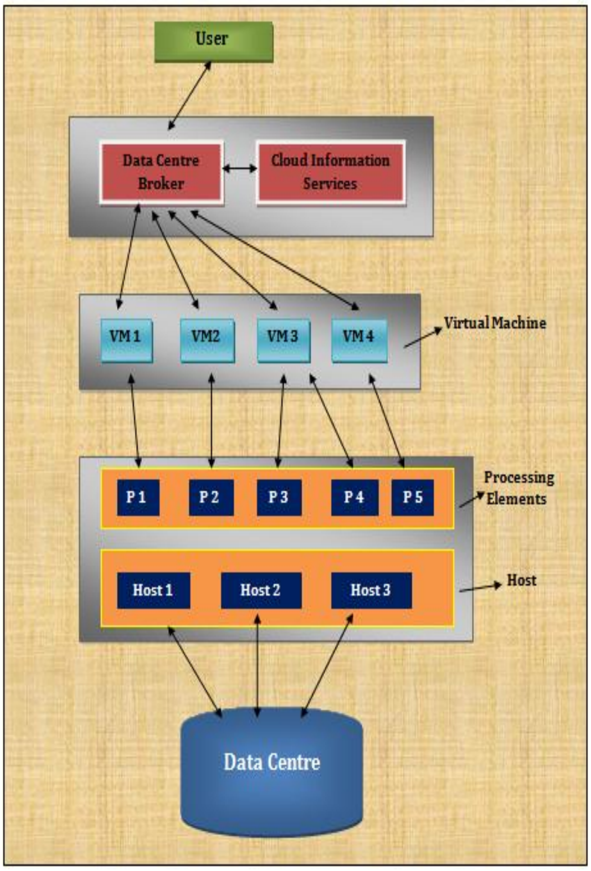

Figure 1: VM Scheduling in Cloud Computing Environment

VM Scheduling is a three step process [7], as:

- ✓ **Resource Discovering and Filtering:** Data Center Broker looks for the resources present in the system and take information about those resources.
- ✓ **Resource Selection:** It is a deciding state in which the specific resource is selected based on certain constraints.
- ✓ **Task Submission:** Task is provided to the selected resource for completion.

Virtualization techniques mainly aim at achieving scalability, availability, throughput and optimal resource utilization. During operation, sometimes VM's are transferred from one system to another without disturbing the operation of other VM's operating in parallel. This is called VM migration. Migrations are performed in order to improve the utilization of the resources and to reduce the down time.

II. LITERATURE SURVEY

The given section introduces the different techniques and methods that are recently developed for optimizing the solutions for effective scheduling algorithm and Methodology for optimization of resource allocation. These techniques are helps to develop an effective methodology for resource allocation strategies.

Cloud computing refers to the model, which is the pool of resources. Cloud makes on-demand delivery of these computational resources (data, software and infrastructure) among multiple services via a computer network with different load conditions of the cloud network. User will be charged for the resources used based upon time. Hence efficient utilization of cloud resources has become a major challenge in satisfying the user's requirement (QoS) and in gaining benefit for both the user and the service provider. In this paper, **Dr. M. Dakshayini et al. [8]** propose a priority and admission control based service scheduling policy that aims at serving the user requests satisfying the QoS, optimizing the time the service-request spends in the queue and achieving the high throughput of the cloud by making an efficient provision of cloud resources.

Nowadays cloud computing has become a popular platform for scientific applications. Cloud computing intends to share a large scale resources and equipment's of computation, storage, information and knowledge for scientific researches. Job scheduling algorithms is one of the most challenging theoretical issues in the cloud computing area. Some intensive researches have been done in the area of job scheduling of cloud computing. In this paper **Shamsollah Ghanbaria et al. [9]** have proposed a new priority based job scheduling algorithm (PJSC) in cloud computing. The proposed algorithm is based on multiple criteria decision making model.

Gang scheduling is currently the most popular scheduling scheme for parallel processing in a time shared environment. In this paper **D. Walsh et al. [10]** first describe the ideas of job re-packing and workload tree for efficiently allocating resources to enhance the performance of gang scheduling. Authors then present some experimental results obtained by implementing four different resource allocation schemes. These results show how the ideas, such as re-packing jobs, running jobs in multiple slots and minimizing the average number of time slots in the system, affect system and job performance when incorporated into the buddy based allocation scheme for gang scheduling.

In this paper, **Yanyong Zhang et al. [11]** show that gang scheduling delivers poor performance towards workloads with high I/O intensities (I/O ratio higher than 50%). They propose an I/O-aware extension of gang scheduling, IOGS, which co-locates jobs with their files. While IOGS performs better for high I/O intensity workloads, its performance for workloads with lower I/O intensities is rather poor because of high system fragmentation. Further, authors propose an adaptive strategy, adaptive-IOGS, which attempts to combine the advantages of both gang scheduling and GS, and we show that adaptive-IOGS is better than the other two schemes in many scenarios. Finally, we combine process migration techniques with adaptive-IOGS, and propose Migration-IOGS, which is shown to be the best among the four for a wide spectrum of workloads.

Effective scheduling strategies to improve response times, throughput, and utilization are an important consideration

DOI10.17148/IJARCCE.2017.6385

in large supercomputing environments. Parallel machines in these environments have traditionally used space-sharing strategies to accommodate multiple jobs at the same time by dedicating the nodes to a single job until it completes. This approach, however, can result in low system utilization and large job wait times. **Yanyong Zhang et al. [12]**discuss three techniques that can be used beyond simple spacesharing to improve the performance of large parallel systems. The first technique they analyze is backfilling, the second is gang scheduling, and the third is migration. The main contribution of this paper is an analysis of the effects of combining the above techniques. Using extensive simulations based on detailed models of realistic workloads, the benefits of combining the various techniques are shown over a spectrum of performance criteria.

III. PROPOSED WORK

The need of efficient computing is increases day by day, in order to handle this traffic it is required to improve the techniques of computing. Cloud computing is a technique which offers to deal with the huge computational complexities. In this chapter a technique for improving the performance of cloud server is described using the efficient scheduling.

A. Domain overview

Cloud computing is a huge computational infrastructure it includes hardware and software. In this system every resource is sharable. Additionally these resources are works on pay per use basis. Therefore it is less expensive than the traditional computing infrastructure. It becomes popular in a very short time due to its ability of scalable computing and storage services. Every resource in this environment are scalable due to this it handle a significant amount of computational traffic. But proper uses of these resources are also required to improve their computational efficiency. Therefore a significant amount of research work is carried out for improving the resource scheduling strategy. Thus the resource scheduling is an essential concept of cloud computing. In this presented work the resource scheduling technique is investigated for finding the best fit resource combination to maximize the execution of the jobs with minimal amount of resource utilization. In this context a number of different techniques for scheduling are studied and Gang scheduling approach is selected for study. The main reason behind selection of gang scheduling technique is their working with the multi-processor environment. It is a promising approach for scheduling therefore the method is optimized with some modifications. In this section the basic overview of the proposed approach is described and the detailed working of the proposed modified Gang scheduling is given in the next section.

B. Methodology

The proposed methodology is described using figure 3.1 the different component of the proposed model is given as:

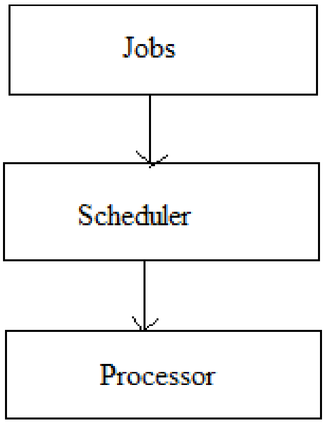

Figure 3.1 job allocation

The basic idea of the proposed model is given using figure 3.1. In this given model the jobs are appearing in a queue and the scheduler fetch the jobs from this queue and allocate to the processor for performing the execution of the jobs. If the jobs are scheduling in well manner the resources are effectively utilized. Here the gang scheduling algorithm is used with the scheduler to find maximize the process execution. The extended model of the above given model is described using figure 3.2. According to the diagram let the jobs are $j = \{j_1, j_2, \dots, j_n\}$ with some priority values such that$P = \{P_1, P_2, \dots, P_n\}$. Here the same number of priority values are associated with the jobs, thus the jobs can be denoted by some tuple such that $J_t = \{(J_1, P_1), (J_2, P_2), \dots, (J_n, P_n)\}$. The priority values are the time requirement of the jobs execution which is different from each other. The gang developer is an additional process which categorizes the appeared jobs for allocation of the processors. Here all the processors are working in time shared manner. In the time shared technique the fixed amount of time slot is assigned to the jobs and under this time the process is executed.

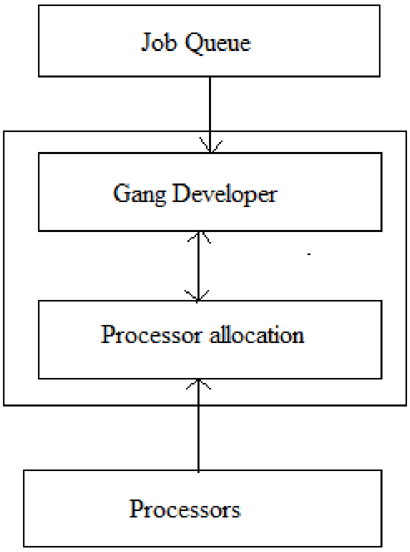

Figure 3.2 proposed model

DOI10.17148/IJARCCE.2017.6385

Before selection of the jobs to assign the processor the jobs are categorized in the following five categories. In order to compute the job priority the min-max normalization approach is used first to scale the entire job priority list in the range of (0-1). For normalizing the values the following formula is used.

$$\text{norm value} = \frac{\text{current value} - \text{minimum}}{\text{maximum} - \text{minimum}}$$

Table 3.1 job categorization

High	0.00 – 0.20
Moderate high	0.21 – 0.40
Average	0.41 – 0.60
Low medium	0.61 – 0.80
Low	0.81 – 1.00

According to the priority range the jobs are categorized, the higher priority job is selected first and assigned to the processors for execution.

C. Proposed algorithm

This section summarizes the provided details of the proposed gang scheduling in terms of the algorithm steps.

Table 3.2 Proposed Algorithm

```
Input: jobs with priority $J_t = \{(J_1, P_1), (J_2, P_2), \dots, (J_n, P_n)\}$, list of processor $M = \{M_1, M_2, .., M_p\}$
Output : Allocation of jobs
Process:
1.  $[\min, \max] = \mathrm{readJobList}(J_t)$
2.  $\mathrm{for}(i = 0; i \le n; i++)$
a.      $\mathrm{normvalue}[i] = \frac{\text{current value} - \text{minimum}}{\text{maximum} - \text{minimum}}$
3.  end for
4.  $S_{val} = \mathrm{SortValues}(\mathrm{normvalue}[i])$
5.  $\mathrm{for}(i = 0; i \le p; i++)$
a.      $\mathrm{Gang}[i] = S_{val}[i]$
6.  end for
7.  $\mathrm{Assign}(\mathrm{Gang}, M)$
```

IV. RESULT ANALYSIS

This section provides the details of the performance computation and their comparison with the similar technique.

A. CPU Cycle

The amount of CPU Cycles required to complete the jobs in queue is termed as required CPU cycles. In this work, both approaches required of CPU cycles for completing their respective jobs. In this, performance parameter we show that the graph representation of the how much cycles are consumed during the process of each job execution.

The figure 2 show the CPU Cycles of the proposed and traditional FCFS algorithms of resource allocation scheduling. In this diagram the amount of CPU Cycles required is given in Y axis and the different runs of the proposed system are reported at X axis. According to the obtained results the proposed algorithm consumes lesser resources to process each jobs as compared to the traditional technique.

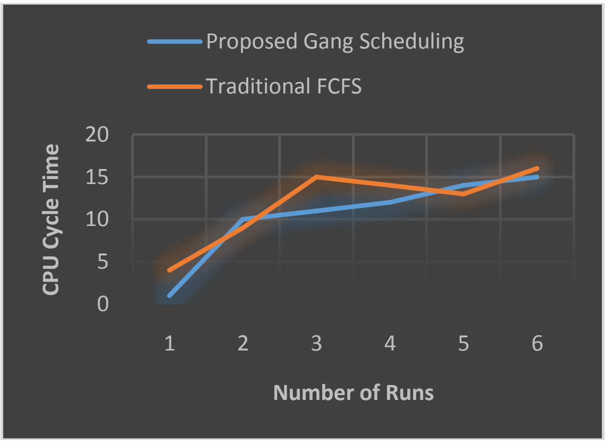

Figure 2 Number of CPU Cycles

B. Optimization Time

In different optimization algorithm number of iterations is required to optimize the best fit solution. Here the given time is the amount of time by which a bets fit solution is obtained. The optimization time shows when the available jobs are capturing the resources of they are free. If the resources are busy to execute other job which is currently holds then time to wait until it's been freed. The optimization time can be measure by following program.

$$\mathit{Optimization\ Time} = \mathit{Total\ resource\ holding\ time} - \mathit{Total\ Jobs\ free\ time}$$

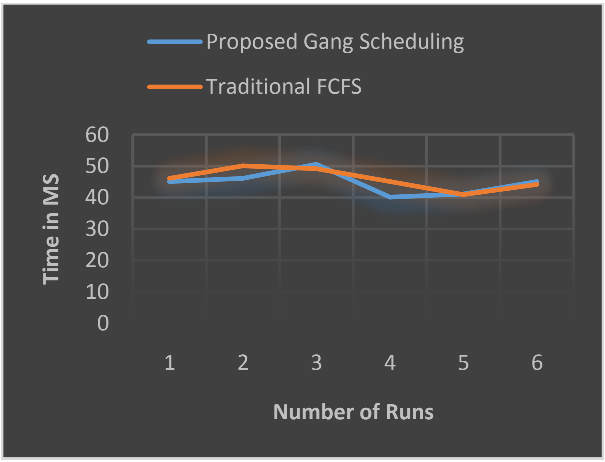

Figure 3 Optimization Time

The figure 3 shows the amount of time consumed during optimization of problem. In this diagram the blue line shows the performance of proposed gang scheduling algorithm and red line provides the performance of traditional FCFS algorithm. The X axis shows the different observations made during experimentation and the Y axis shows the amount of time consumed.

C. CPU Utilization

In order to execute an individual job an amount of time is consumed. This time of execution is termed as CPU Utilization.

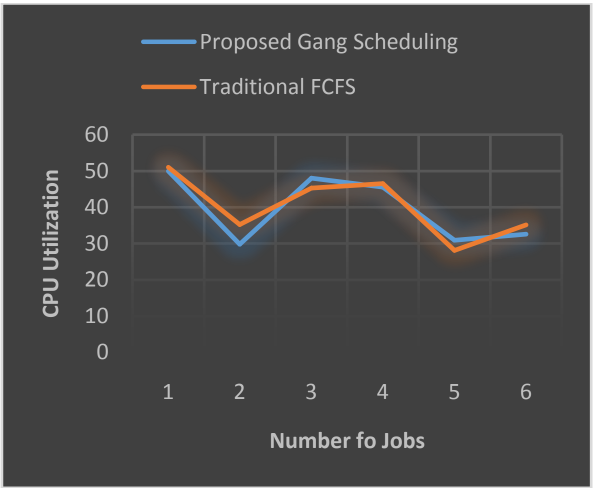

Figure 4: Comparative CPU utilization

The figure 4 shows the CPU utilization of the traditional scheme of resource scheduling and allocation. In given figure X axis contains the processes arrived for scheduling and the Y axis contains the amount of CPU Time consumed. In each of the modeled technique the results analysis for the CPU utilization the time utilization is given in terms of milliseconds. According to the comparative performance analysis the proposed technique needs less CPU cycles to execute the task as compared to the traditional technique. Therefore the proposed technique is much efficient as compared to the traditional technique

V. CONCLUSION

Scheduling is one of the most important process in the field of cloud computing. In this paper, the various existing scheduling algorithms are analyzed. Computing of complex problems requires computational recourses such as memory, CPU and others. Therefore in this generation of computing the cloud computing is invented to support the large scale computation. In this environment the cloud service providers offers scalable resources for computing as well as the storage resources. Due to this that becomes more adoptable for different purpose of applications and their demand is also increases rapidly. To increase or scale the processing capability resources sharing, virtualization, scheduling like techniques are used. These techniques help to allocate the resources according to their accurate usage. In this proposed work a new algorithm with the hybrid concept of gang scheduling are proposed and implemented for resource optimization with respect to the available set of resources. In this work gang scheduling is performing smooth and effective resource allocation to their jobs arrivals. This technique is related to no. of threads and no. of process in which processes are communicated to each other simultaneously at the same time. Therefore we need to create a matrix for resource allocation between resources and processes.

REFERENCES

[1] Hu, Ye, et al. "Resource provisioning for cloud computing", Proceedings of the 2009 Conference of the Center for Advanced Studies on Collaborative Research, IBM Corp., 2009.

[2] Karatza, Helen D. "Performance analysis of gang scheduling in a partitionable parallel system", Proc 20th Eurconf model simul, Bonn, Germany. 2006.

[3] Clark, C., Fraser and K., Hand, “Live migration of virtual machines”, In Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation (2005).

[4] Ajay Gulati, and Irfan Ahmad, “Cloud-scale resource management: challenges and techniques, In Proceedings of the 3rd USENIX conference on Hot topics in cloud computing, Hot Cloud’11, pages 3–3, Berkeley, CA, USA, 2011, USENIX Association.

[5] Gulati, A., Shanmuganathan, G., Holler, A., and Ahmad, I, Cloud-scale resource management: challenges and techniques. In Proceedings of the 3rd USENIX conference on Hot topics in cloud computing, (2011).

[6] Wood, T., Shenoy, P., and Venkataramani, “Black-box and gray-box strategies for virtual machine migration”, In Proceedings of the 4th USENIX conference on Networked systems design and implementation (2007).

[7] Choudhary&Peddoju “A Dynamic Optimization Algorithm for Task Scheduling in Cloud Environment”, International Journal of Engineering Research & Applications, PP. 2564 – 2568, Issue 3, Vol. 2, May – June 2012.

[8] Dakshayini, Dr M., and Dr HS Guruprasad, "An optimal model for priority based service scheduling policy for cloud computing environment." International Journal of Computer Applications 32.9 (2011): 23-29.

[9] Ghanbari,Shamsollah, and Mohamed Othman, "A priority based job scheduling algorithm in cloud computing." Procedia Engineering 50 (2012): 778-785.

[10] Zhou, Bing Bing, David Walsh, and Richard P. Brent, "Resource allocation schemes for gang scheduling", Workshop on Job Scheduling Strategies for Parallel Processing, Springer Berlin Heidelberg, 2000.

[11] Zhang, Yanyong, et al. "Gang scheduling extensions for I/O intensive workloads", Workshop on Job Scheduling Strategies for Parallel Processing, Springer Berlin Heidelberg, 2003.

[12] Zhang, Yanyong, et al, "An integrated approach to parallel scheduling using gang-scheduling, backfilling, and migration", IEEE Transactions on Parallel and Distributed Systems 14.3 (2003): 236-247.

DOI10.17148/IJARCCE.2017.6385