Optimized Resource Filling Technique for Job Scheduling in Cloud Environment

AV.Karthick
Assistant Professor / MBA
SMCET
Kalayarkoil, India

E.Ramaraj, Ph.D
Professor / Dept of CSE
Alagappa University
Karaikudi, India

R.Kannan
Assistant Professor / MCA
SMCET
Kalayarkoil, India

ABSTRACT
Job scheduling is one the complicated problem in Cloud Computing. We intend a grouping method to develop the combinational backfill algorithm based on smadium and long queue technique using random fashion. The proposed algorithm helps to improve the resource gap, reduce the system idle time and helps to attain high resource usage and provide quality system in cloud environment. To make the most efficient use of the resources, accomplish the optimization for cloud scheduling problems. It is not possible to predict the job execution time in cloud environment. Hence the scheduler must be dynamic. Previous Scheduling strategies like FCFS, SJF, Round Robin and CBA are deficient in filling the Resources gap effectively and create more fragmented space. The Proposed work, Optimized Resource Filling (ORF) properly utilize the resources and increase the unused available working space and reduce starvation, when compared to traditional and balance spiral method. Its ultimate goal is to produce high usage of available resources, balance the system and reduce system unused time and to improve throughput of the system. This paper introduce smadium concept for cloud resource management. ORF tries to fill the unused space created by the scheduler.

Keywords
optimized resource filling, smadium, random fashion.

1. INTRODUCTION
Cloud computing consist of many numbers of virtual and physical servers, administration of hundreds or thousands of system is a major task. Cloud deliver secure, speedy, convenient data storage and computing service cantered by internet. It offers three kinds of services like Infrastructure as a service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). The users are charged on the basis of Pay per use and On-Demand access [9]. Cloud computing, as a current business offering, intended to facilitate computing across widespread and varied resources, rather than on local machines or at remote server farms. Although there is no clear definition of Cloud Computing, most authors seem to agree that it consists of clusters of distributed computers Clouds providing [12] on-demand resources or services over a network with the scalable and reliable data centre.

Cloud computing focus on reliable, error-tolerant and scalable infrastructure for hosting Internet based application services. Cloud computing is a cost reduction model for provisioning services and it makes IT management easier and more responsive to the varying needs of the business [7]. Cloud Computing is defined by a large-scale distributed computing model that is motivated by economies of large scale, in which a collection of virtual and scalable, power saving, storage efficiency , platform as a services are delivered on demand to outside customers through Internet. Software industry is day to day changing. Large amount of capital invested in hardware and software of the companies need today is going to become obsolete by its updating and new techniques are adopted. Increase the performance of your hardware/ software requirement [6] can be easily with the help of cloud, without affect the current work.

The scheduling process consists of many iterative steps. The given scheduling is to give enhanced response time and throughput. The time taken by the service provider to respond the client request is called response time [3]. Cloud networking is a hybrid and combination of cloud computing and virtual networking. The solution given by the cloud computing benefits of more deeply into the network and gives a tighter integration of virtualization and networking levels. It provide online access of data services need to provide the data transfer with the use of network and path diversity and encoding techniques and dealing with ubiquitous mobility of user, content and information matter in a joined way.

2. LITERATURE REVIEW
Dynamic resource scheduling in cloud resources is one of the changing task. Cloud computing provides an on-demand service capabilities. He introduces a load-adaptive cloud resource scheduling model based on ant colony algorithm. It is one of the issues to be solved in a lot of technical fields. Xue Qiang et al [15] projected a load-adaptive time slot priority queue algorithm to solve the conflict of the real-time traffic. A variety of prediction methods do exist, but none is very exact. In this study, explained a grid system where both parallel and sequential jobs require service. The First Come First Served algorithm [11] has the limitation of insufficient space usage and cause severe fragmentation when resources are not available for large jobs. It is used to be a better solution to this problem and has shown to be very effective when gang scheduling is implemented.

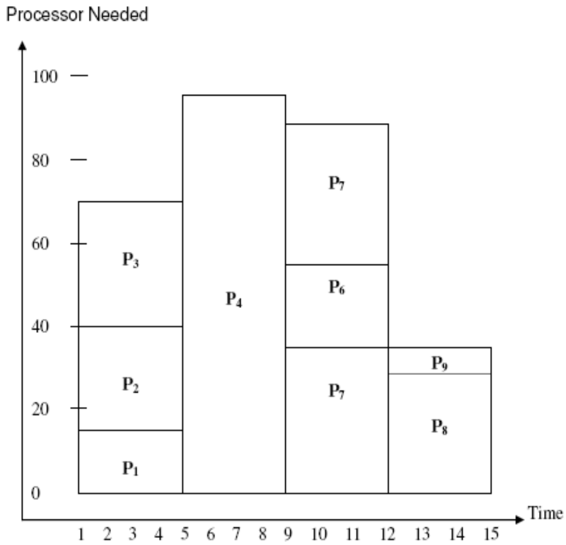
In combinational backfill algorithm (CBA) gives high priority to small jobs. He proposed improved backfill algorithm using balanced spiral (BS) method [14]. Perkovic and Keleher study Conservative Backfilling consist of random queue ordering both with and without sorting by length and random reordering as well. Reorder the backfill queue to EASY algorithm by Tsafrir et al. Talby and Feitelson hybrid the combination of three types of priorities in the order of queue jobs. Chiang et al [2] propose generalizations of the Shortest Job First scheduling algorithm to order queue jobs used static and random reservations. In random reservations, jobs are reserved and the ordered of job reservations can change with each new job arrival or if the priorities of waiting jobs modify. Priority scheduling is the scheduling algorithm which assigns a certain priority to each task. Round robin scheduling (RR) is the scheme in which a FCFS queue is managed with a fixed time quantum for each job. Multilevel queue scheduling separates the ready queue in to many queues. Multilevel feedback queue scheduling allows jobs to move from one queue to another. The central aim of the research has two approaches namely is cluster scheduling are gang scheduling and backfilling. PB-FCFS gives a combinational approach for cluster scheduling. It is a hybrid of FCFS and backfilling with a better priority for the job, in some cases of the job cannot be queues due to deficit available resources. Proposed algorithm in does not give importance to the priority of a job because limiting factor of raising the priority is the maximum priority.

Job scheduling policy gives a reasonable scheduling decision making for parallel jobs in clusters. In First-Come-First-Served parallel job scheduling method the time order of jobs submitted by users, it always selects the first job in the queue to map the suitable computing resources and complete it. The system resource deployment is low by the FCFS scheduling method and the pits case is large processing job are in the head of the waiting job queue may take up computing resources for a long time, while other computing resources are at rest not to be used by other short jobs [2]. Therefore, there are gaps in the computing resources. The scheduling algorithms such as FCFS, Short Job First, Long Job First, priority-based scheduling will show the way to resource gaps. With the help of computing resources, backfilling scheduling algorithm on the basis of FCFS is proposed in to settle the problem of resource fragment. The computing resource is available the unused free space is find by the region. Based on the backfilling identification the waiting jobs in the queue, backfilling scheduling determines based on short and narrow job can be completed. It gives better utilization rate of resources at some extent and improves the throughput of the system. Easy backfilling is a combination of two traditional backfilling scheduling methods concluding conservative backfilling and an aggressive backfilling. Conservative backfilling can backfill a small job in the back of the queue as long as it does not delay the start time of all the jobs in the front of the queue as like FCFS. EASY backfilling picks a small job to backfill if it does not delay the start time of the first job in the queue. The resource utilization is improved. The requirement of user-estimated run-time of jobs is lower. The small jobs will be able to grant more opportunities for backfilling. It is more flexible to backfill. The large job may be delayed to run more easily.

3. EXISTING BACKFILL ALGORITHM

However existing job scheduling algorithm like FCFS, SJF, EASY and conservative backfill algorithms are failed to fill the resource gap effectively and not fully utilize. In combinational backfill algorithm (CBA) small jobs are getting high priority and avoid large jobs leads to starvation. FCFS is the simplest way to schedule jobs. The arrival rates of the jobs are dispatched and low system utilization. A smaller job is waited in the queue as long as it does wait for the previously queued job completed their execution. i.e small job is allowed to leap forward as long as it does not delay the job at the head of the queue. This denotes it suffers from fragmentation. Backfilling allows smaller jobs to move forward in the schedule as long as such movement does not allow the other scheduled jobs in the queue. Space sharing algorithm result in poor usage of jobs are utilization despite a waiting queue of jobs and response time are relatively high.

Shortest job scheduling algorithm gives only importance to shortest jobs only, it ignores importance medium and long jobs. Parallel job scheduling strategies has been widely studied in the past. Existing backfilling scheduling algorithms are only available for one queued job backfilled to schedule and create waste of resource gaps.



Job	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9
Needed Processor	15	25	30	90	35	20	32	30	10

Fig 1: Job Scheduling in Existing System

In the above said example see figure 1 shows the resources of 9 different jobs are submitted in the various scheduling that help to solve the problems in cloud. Measured in terms of FCFS the jobs id P_1, P_2 and P_3, the job P_4 have not enough space to occupy in the remaining job. As like the other jobs are also not efficiently used and create starvation.

4. PROPOSED ORF TECHNIQUE

Best job scheduling is to give better system metrics like utilization of space and user metrics like waiting time and turnaround time. In the figure 2 the user submitted jobs are entered in to the scheduler. It arranges the jobs in ascending order. Then grouping the small and medium jobs grouped as smadium in one queue and remaining jobs in long queue. The arrived jobs are updated in the queue and scheduler. Queue manager is responsible for handling the scheduled jobs and dispatching the request. They send the job id matched by the queue service. It invokes the job scheduler to prioritize the jobs. The jobs are updated in the resource manager.

The queue allocates the resources to the users that are required. The scheduler dispatches the user request to the computer nodes. The resource manager allocates the job and scheduled them in round robin fashion. If the new jobs are arrived the queue manager dynamically changed them.

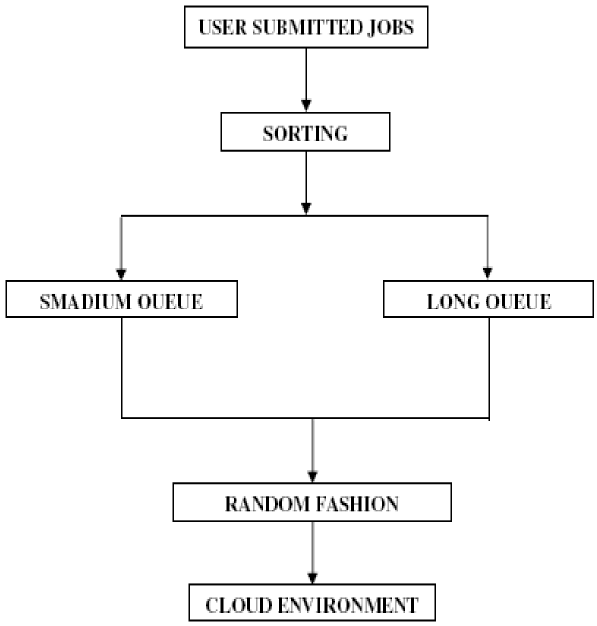


Fig 2: Architecture of ORF in Cloud Environment

The queued jobs are entered in to the resource allocation by the resource broker in random fashion based on burst time. Based on the finishing time of job they allocate the jobs in order. The Cloudsim resource simulator uses internal events to simulate the execution and allocation of PEs' share to Cloudlet jobs. Cloudsim schedules a new internal event to be delivered at the forecasted time of the remaining Cloudlets. The simulation of the multi queue scheduling algorithm and the Cloudlets' execution in three different queues are queued them

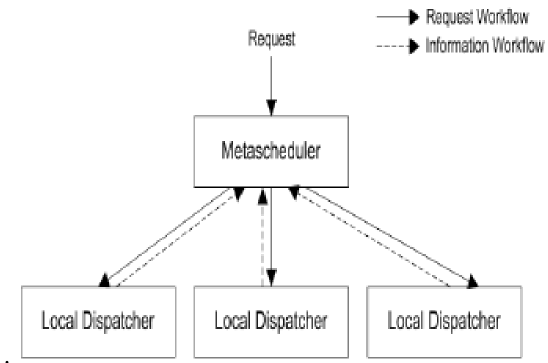


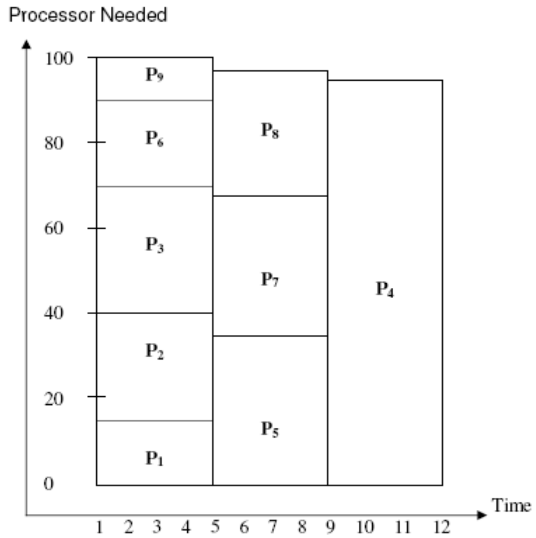
Fig 3: Structure of Meta Scheduler

The figure 3 shows the structure of meta scheduler. It is a global scheduler that changes random not like a local scheduler. It is dynamic in nature based the arrival rate of jobs. Based on the processing time the scheduler allocates the resources to the cloud environment. An optimized algorithm based ORF strategy, which can select multiple jobs from smadium and large queue combined in job queue to backfill to maximize the use of idle resources. This algorithm can attain the lower average waiting time of jobs and higher utilization of resources than existing backfilling algorithms.

4.1 Algorithm of ORF

1. Collect all jobs and its corresponding burst_time
2. If sum ((burst_time) > 0) follow next step else step 11
3. Burst_time of all jobs are added and calculated maximum burst_time needed
4. Create queue [Smadium_queue, Large_queue]
5. If size(Smadium_queue) < Maximum_size then store the jobs in Smadium_queue else
6. Store the job in Large_queue
7. If P_1=ready allocate the processor check whether any space remaining
8. If space(P_n) = = remaining space allocated the processor else
9. Allocate new space for processor
10. Follow the procedure for all jobs by repeat step 7 to 9
11. Complete the process.

A Scheduling model used by the proposed algorithm is presented in figure 2 the resource manager, scheduler and pool of resources are components of the model. It has a queue of jobs. Algorithm rearranges the job queue according to the increasing order of the remaining execution time of jobs. Each job declares its required number of resources and their types at the time of arrival in the queue. Status of resources is monitored by the resource manager. It dynamically interacts with the scheduler and resource pool. When a job arrives at time t in the cluster, it gets placed at a level according to its expected runtime.



Job	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9
Needed Processor	15	25	30	90	35	20	32	30	10

Fig 4: Job Scheduling in ORF Technique

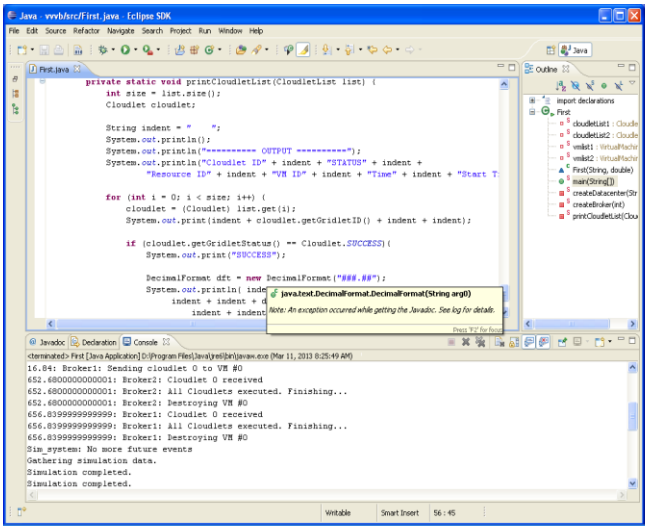
When job gets its turn of execution, it tries to have access of desired resources with the specifications. Scheduler interacts with the resource manager and gets the notification of free resources. It decides whether the job can be placed into execution or not. If job cannot be scheduled, it gets into a waiting state. Scheduler tries to backfill a lower priority job. Resource manager continuously update the pool to get the recent information about resource status. The proposed study orders the job arrival sequence in ascending order. The jobs are placed in two different queues. Job sequence with respect to the processor needed as {P1, P2, P3, P6 and P9} is filled the remaining space is used in the smadium first queue. Job id {P5, P7 and P8} in the next queue and take the largest job P4 in the fourth position of the job sequence. It clearly shows the reduction of starvation when compared to the previous existing algorithm.

5. RESULT AND DISCUSSION

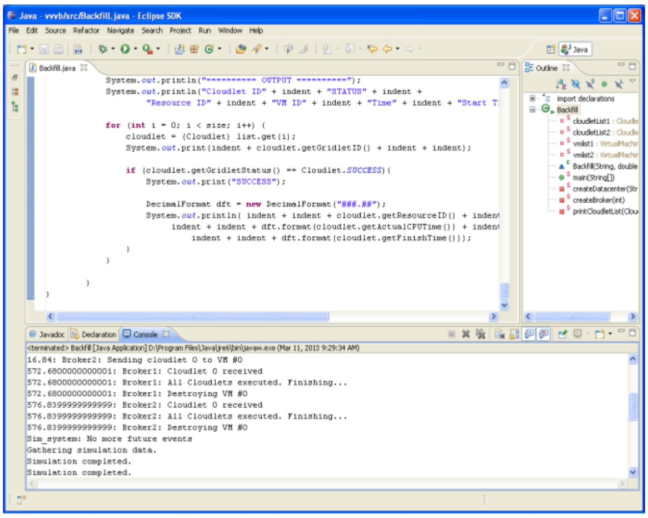
The job completion ratio of different scheduling algorithms based on average waiting time, through put and average turnaround time in the figure 5. The proposed hybrid algorithm achieve higher result and clearly shown the comparison table 1, ORF maximizes the throughput by about 15% and minimizes the average waiting time by 5% and decrease the average turnaround time by 8%. The screen shots of First Come First Serve, CBA and ORF Algorithms. This satisfies the requirement of an efficient scheduling algorithm.

Table 1. Comparison of various scheduling algorithms

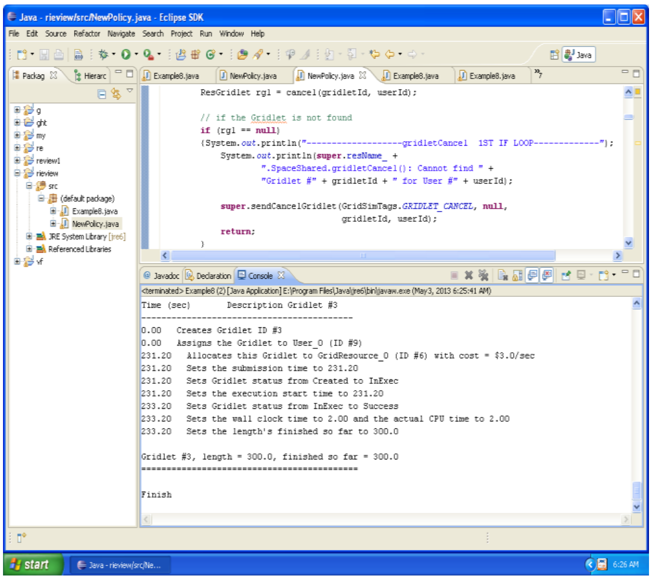
Algorithm	Average Waiting Time (ms)	Throughput	Average turnaround Time (ms)
FCFS	8	25%	15
SJF	5.3	29%	12.3
CBA	4.7	30%	9.8
ORF	3.3	35%	7



Screen Shot 1. First Come First Serve Algorithm



Screen Shot 2. CBA Algorithm



Screen Shot 3. ORF Algorithm

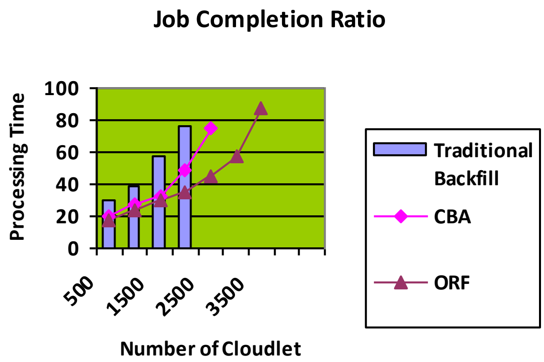


Fig 5: Result of ORF with existing algorithms

6. CONCLUSION AND FUTURE WORK

Combinational backfilling algorithm for parallel job scheduling is proposed to resolve the problem of the parallel jobs being scheduled inefficiently and ineffectively by the existing backfilling methods when there are multiple jobs that can be combined to backfill. The results of experiments show that the ORF algorithm reduces the average waiting time of jobs and improves the utilization of resources in with FCFS, SJF and CBA algorithm. The algorithm uses the resource gap abundantly. In the future, ORF algorithm combined with resource prediction will be researched.

7. REFERENCES

[1] Alexander. M., Lindsay., Maxwell. Galloway Carson., Christopher. R. Johnson., David. P. Bunde., Vitus. J. Leung. Backfilling with guarantees granted upon job Submission. Work done while Alex was a student at Knox College.

[2] Chiang. S. H., Dusseau. A. A., Vernon. M. K. 2002 The Impact of More Accurate Requested Runtimes on Production Job Scheduling Performance. In Job Scheduling Strategies for Parallel Processing, 103-127.

[3] Daniel. D., Jeno Lovesum. S. P. 2011 A Novel Approach for Scheduling Service Request in Cloud with Trust Monitor. Proceedings of 2011 International Conference on Signal Processing, Communication, Computing and Networking Technologies. ICSCCN.

[4] Dejan. Perković., Peter. J. Keleher.2000 Randomization Speculation and Adaptation in Batch Schedulers. IEEE.

[5] Feitelson. D. G., Rudolph. L., Schwiegelshohn .U. 2005 Parallel Job Scheduling-A Status Report. Lecture Notes in Computer Science. Job Scheduling Strategies for Parallel Processing. Springer Berlin, 1-16.

[6] Loganayagi. B., Sujatha. S. 2010 Creating Virtual Platform for Cloud Computing. IEEE.

[7] Martin. Randles., David. Lamb., Taleb-Bendiab. A. 2010 A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing. IEEE 24th International Conference on Advanced Information Networking and Applications Workshops.

[8] Mu'alem. A. W., Feitelson. D. G. 2001 Utilization Predictability Workloads and User Runtime Estimates in Scheduling the IBM SP2 with Backfilling. IEEE Transactions on Parallel and Distributed Systems, 529-543.

[9] Selvarani. S., Sudha Sadhasivam. G. 2010 Improved cost based algorithm for task scheduling in cloud computing. IEEE.

[10] Shengwei, YI1.,†., Zhichao, WANG1., Shilong, MA1., Zhanbin, CHE1., Yonggang HUANG2., Xin, CHEN1. 2010. An Effective Algorithm of Jobs Scheduling in Clusters, Journal of Computational Information Systems, 3163-3171.

[11] Sofia. K., Dimitriadou. 2010 Job Scheduling in a Distributed System Using Backfilling with Inaccurate Runtime Computations. International Conference on Complex, Intelligent and Software Intensive Systems.

[12] Srividya. Srinivasan., Rajkumar. Kettimuthu., Vijay. Subrarnani., Sadayappan. P. 2002 Characterization of Backfilling Strategies for Parallel Job Scheduling. Proceedings of the International Conference on Parallel Processing Workshops. IEEE.

[13] Sudha. S.V., Thanushkodi. K. 2008 An Approach for Parallel Job Scheduling using Nimble Algorithm. IEEE.

[14] Suresh.A#1., Vijayakarthick.P#2. 2011 Improving scheduling of backfill algorithms using balanced spiral method for cloud metascheduler. IEEE-International Conference on Recent Trends in Information Technology.

[15] Xin. Lu., Zilong. Gu. 2011 A Load-Adapative Cloud Resource Scheduling model based on Ant Colony Algorithm. Proceedings of IEEE CCIS