ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Enhanced Node Utilization and Responsiveness for Parallel Workload Based On Priority Consolidation in the Cloud

R.Suganya[1]

Department of Computer Science and Engineering, Syed Ammal Engineering College, Ramanathapuram, Tamilnadu, India[1]

ABSTRACT— The cloud computing environment contains remote data centers which work for run the complex applications. Many complex applications require parallel processing capabilities effectively to run the parallel jobs using virtualization technologies. A parallel job often requires a certain number of nodes to run, to schedule parallel job in data centres. A set of nodes is likely to be fragmented by parallel jobs with different node number requirement. An important consideration in this environment is to improve response time, throughput and utilization. In Existing approach, each nodes computing capacity is partition into two tier using priority based consolidation. The foreground VM tier runs higher priority jobs and Background VM tier runs low priority jobs. This leads to low system utilization and large job wait times. To overcome this, the proposed approach is to effectively partition the data centre node into k-tiers for improved node utilization and responsiveness.

KEYWORDS— Cloud computing, parallel job scheduling, resource consolidation, CMCBF, SS

I.INTRODUCTION

Cloud computing is the delivery of computing as a service rather than a product, whereby shared resources, software, and information are provided to computers and other devices as a utility (like the electricity grid) over a network. The cloud computing paradigm promises a cost-effective solution for running business applications through the use of virtualization technologies, highly scalable distributed computing, and data management techniques as well as a pay-as-you-go pricing model. In recent years, it also offers high-performance computing capacity for applications to solve complex problems. Improving resource utilization is essential for achieving cost effectiveness. Low utilization has long been an issue in data centers. Servers in a typical data center are operated at 10 to 50 percent of their maximum utilization level. 10 to 20 percent utilization is common in data centers. For a data center, or a subset of servers in a data center that mainly handles applications with high-performance computing needs and runs parallel jobs most of the time, the problem can be significant. There are two factors that may reduce the utilization of nodes that run parallel jobs: 1. A parallel job often requires a certain number of nodes to run. A set of nodes is likely to be fragmented by parallel jobs with different node number requirement. If the number of available nodes cannot satisfy the requirement of an incoming job, these nodes may remain idle [4], [5], [6].

2. Typical parallel programming models, such as BSP [7]often involve computing, communication, and synchronization phase. A process in a parallel job may frequently wait for the data from other processes. During waiting, the utilization of the node is low. Based The most basic but popular batch scheduling algorithm for parallel jobs is first come first serve (FCFS) [8]. Each job specifies the number of nodes required and the scheduler processes jobs according to the order of their arrival. When there is a sufficient number of nodes to process the job at the head of the queue, the scheduler dispatches the job to run on these nodes; otherwise, it waits till jobs currently running finish and release enough nodes for the job. FCFS

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

may cause node fragmentation and methods such as backfilling [9] and Gang scheduling [10] were proposed to improve it. However, they do not target on the utilization degradation caused by parallelization itself.

II. LITERATURE REVIEW

There have been many efforts on scheduling mechanisms for parallel jobs in clusters. FCFS is the basic but popularly used batch scheduling. Backfilling[9],which was developed as the EASY for IBM SP1, is a technique that allows short/small jobs to use idle nodes while the job at the head of the queue does not have enough number of nodes to run. Backfilling can improve node utilization, but it requires each job to specify its maximum execution time so that only jobs that will not delay the start of the job at the head of the queue are backfilled. Furthermore, a pre-empted job is often given a reservation for a future time to run. Different methods of assigning reservations differentiate several variances of backfilling techniques [9]Backfilling techniques address the low-utilization problem caused by different node number requirements of parallel jobs. However, backfilling does not deal with low resource utilization due to parallel jobs themselves.

1) Parallel and Distributed Simulation in the Cloud

Cloud computing offers the ability to transparently provide computing services remotely to users through the Internet, freeing them of the burdens associated with managing computing resources and facilities. It offers the potential to make parallel and distributed simulation capabilities much more widely accessible to users who are not experts in this technology and do not have ready access to high performance computing platforms. However, services hosted within the "cloud" can incur significant performance degradations. In here the analysing approach is the potential benefits and technical challenges that arise in utilizing cloud platforms for parallel and distributed simulations and a potential solution approach.

2) Backfilling Using Runtime Predictions Rather Than User Estimates

The most commonly used scheduling algorithm for parallel supercomputers is FCFS with backfilling, as originally introduced in the EASY scheduler. Backfilling means that short jobs are allowed to run ahead of their time provided they do not delay previously queued jobs (or at least the first queued job). In here the run time job predictions are too short so the concept name divorcing kill-time from the runtime prediction is used. To make this work, predictions need to be corrected adaptively if proved wrong. The end result is a surprisingly simple scheduler we call EASY++, which requires minimal deviations from current practices (e.g. using FCFS as the basis), and behaves exactly like EASY as far as users are concerned. Nevertheless it achieves significant improvements in performance, predictability, and accuracy, and we argue it can (and in our opinion should) replace the default currently in use on production systems. In addition, our techniques can be used to enhance any backfilling algorithm previously suggested.

3) An integrated approach to parallel scheduling using gang-scheduling, backfilling, and migration

In this work Effective scheduling strategies to improve response times, throughput, and utilization are an important consideration 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. This paper discusses three techniques that can be used beyond simple space sharing to improve the performance of large parallel systems. The first technique we 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.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

III. EXISTING SYSTEM

In previous versions cpu takes responsiveness as the top priority and need non-trivial effort to make them work for data centres in the cloud era. In this work, we propose a priority-based method to consolidate parallel workloads in the cloud. We leverage virtualization technologies to partition the computing capacity of each node into two tiers, the foreground VM tier (with high CPU priority) and the background VM tier (with low CPU priority). We provide scheduling algorithms for parallel jobs to make efficient use of the two tier VMs to improve the responsiveness of these jobs. By improving resource utilization for datacentres that run parallel jobs, particularly we intend to make use of the remaining computing capacity of datacentre nodes that run parallel processes with low resource utilization to improve the performance of parallel job scheduling. The parallel jobs we deal with have the following characteristics:

1) The job execution time is unknown.
2) Saving and restoring the state of a job is relatively cheap with checkpoint support.
3) The CPU usage of the processes of a job can be estimated either during design phase or through the historical data

CMBF Algorithm:

CMBF schedules jobs to run according to their arrival time when there is enough number of nodes. When the number of idle nodes is not sufficient for a job, another job with a later arrival time but smaller node number requirement may be scheduled to run via backfilling. To avoid starving a preempted job, CMBFuses the following policy: a preempted job is scheduled to run whenever it sees the total number of nodes that are either idle or occupied by jobs with a later arrival time is equal or greater than the number of nodes it needs. The job may preempt jobs arriving later but being scheduled on some nodes. The scheduler instructs these jobs to save states, suspends their execution and moves them back to the job queue.

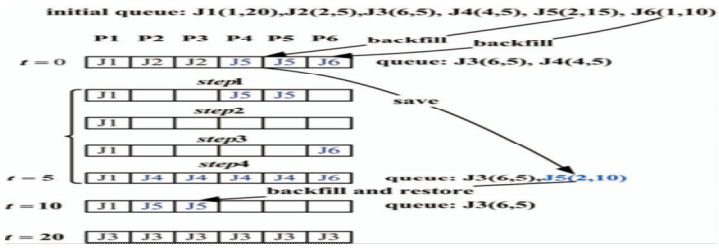

Fig:1.1 CMBF algorithm

AMBF Algorithm:

AMBF only tracks backfilling jobs for the job at the head of the queue and allows the head of-queue job to preempt other jobs. The rest of jobs in the queue are not allowed to preempt jobs, in another word, they can only be dispatched to idle nodes. The algorithm pseudocode of AMBF is similar to that of CMBF except that only the head-of-queue job.

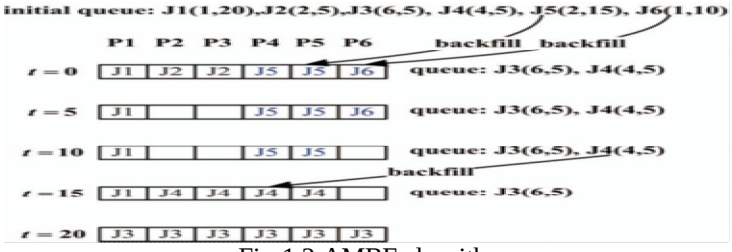

Fig:1.2 AMBF algorithm

Fig.3.3 to illustrate AMBF. After J2 departs at time 5 and J6 departs at time 10, J3 is at the head of the queue but the number of nodes it requests cannot be satisfied. There is no backfilling job for it to preempt either. As AMBF does not

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

allow none-head-of-queue jobs to preempt, J4 cannot preempt J5 and is only dispatched to run when there is enough idle nodes at time 15. As a job is less likely to preempt other jobs, AMBF also incurs job suspension and resuming less frequently than CMBF.

IV. PROPOSED SYSTEM

In this Proposed System, Load balancing is a computer networking method for distributing workloads across multiple computing resources, such as computers, a computer cluster, network links, central processing units or disk drives. It improve the utilization and responsiveness for parallel workload in the cloud. In a large data center, processes of a job may need to be allocate each other to minimize the process execution. Load balancing is usually provided by dedicated software or hardware, such as a multilayer switch or a Domain Name System server process. Load balancing aims to optimize resource use, maximize throughput, minimize response time, and avoid overload of any one of the resources. Using multiple components with load balancing instead of a single component may increase reliability through redundancy. Load balancing is especially important for networks where it's difficult to predict the number of requests that will be issued to a server. The proposed approach is to effectively partition the data centre node into k-tiers .

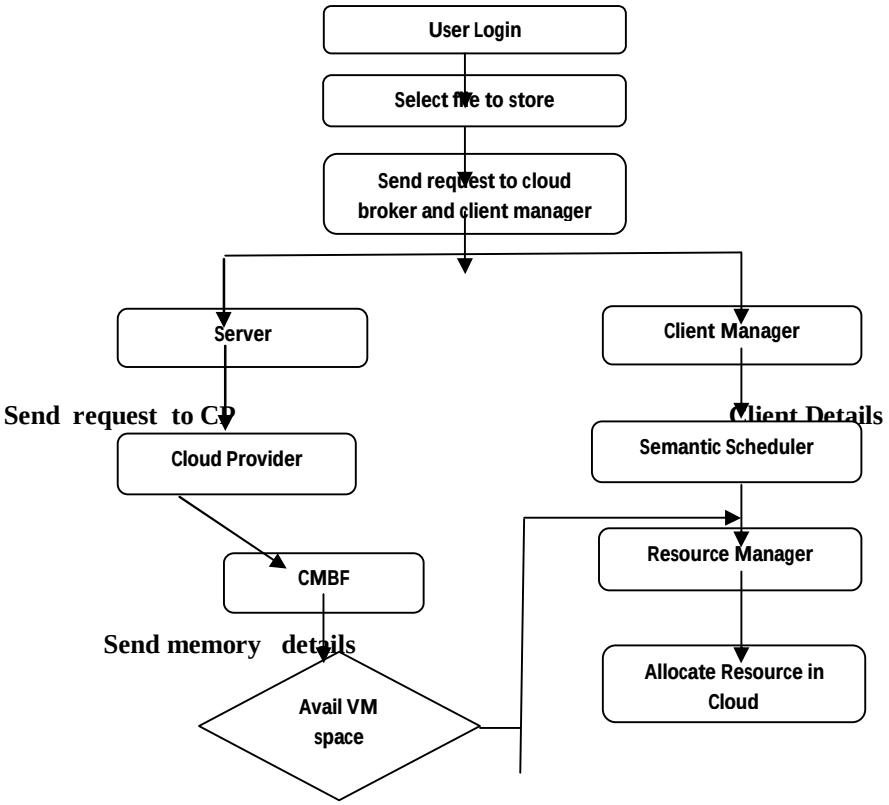

Fig.2 Architecture

Advantages of Proposed System

To improve the node utilization and responsive ness for parallel work loads

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Reduce the time consmptions while using the parallel work loads

To achieve the K-tier implementation in job scheduling following components are required

Cloud Creation

In this framework need to analyze the performance and Trusted User. For that first creating a cloud environment with the help of VM -Ware. In the cloud environment by implementing the datacenter, Data broker, VM. VM denotes the number of available virtual machines, and Cloud broker is the mediator to perform the job in the resources. The IP suite can be viewed as a set of layers, each layer having the property that it only uses the functions of the layer below, and only exports functionality to the layer above. A system that implements protocol behavior consisting of layers is known as a protocol stack. Protocol stacks can be implemented either in hardware or software, or a mixture of both. Typically, only the lower layers are implemented in hardware, with the higher layers being implemented in software. Server forward the available request to cloud service broker for their respective resource available in the VM. CSP act as a intermediate between server and the VM. CSP has number of request with their attributes. The Cloud Provider send the request to VM for the specific resource. Services made available to users on demand via the Internet from a cloud computing provider's servers as opposed to being provided from a company's own on-premises servers.

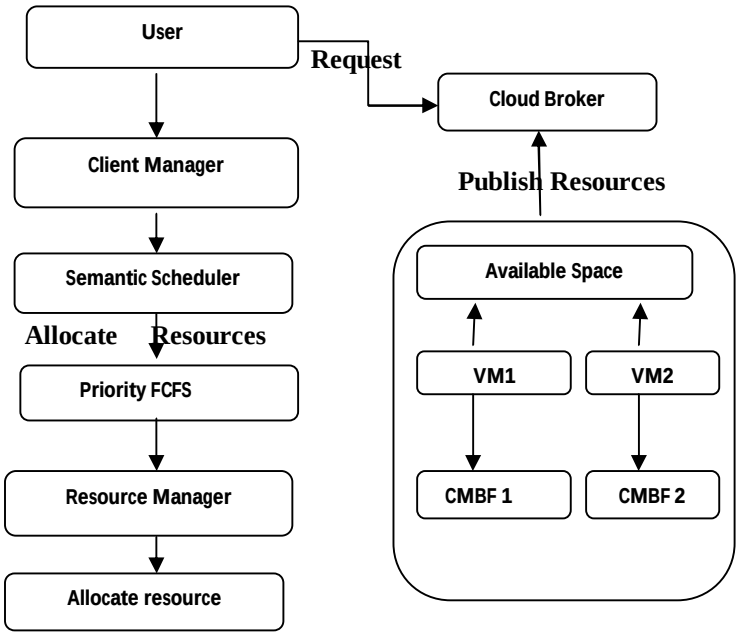

Fig:3 Dataflow Digram

Resource Selection

In this component user authorization can be performed for resource selection. Authorized user login through their own id and password, new user register their details id will be provided for successful registration. Only authorized user can select

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

the resources for processing.User can select their file or resource then their attributes can be computed for further processing. Finally request can be forwarded to server. Client details can be forwarded to both client manager and the server.Cloud computing is a highly scalable distributed computing platform in which computing resources are offered as a service leveraging virtualization.Cloud Computing distributes the computational tasks on the resource pool which consists of massive computers so that the service consumer can gain maximum computation strength.

Checking Space Availability

In this component analyze space availability to allocate the resources for processing.Identify the space for individual VM and forward the memory details to Consolidation supported Back Filling (CMCBF).CMCBF receives and stores the available space from specific virtual machine to allocate the resources.Set up the guest VM's address space. The host must also supply a firmware image with which the guest can bootstrap into its main OS.

Resource Allocation

semantic scheduler (SS) analyze the available apace to allocate the resources for processing specific task.Semantic scheduler send the request to CMCBF to identify the available in the VM. CMCBF response the request received from SS with available VM.SS schedule the request using FCFS to allocate the resources in VM. While allocating resources if any fault occur it can be automatically redirect to next available VM for resource allocation. Resource allocation is used to assign the available resources in an economic way.It is part of resource management. In project management, resource allocation is the scheduling of activities and the resources required by those activities while taking into consideration both the resource availability.

Load Balancing In K-Tire Computation

Load balancing is a computer networking method for distributing workloads across multiple computing resources, such as computers, a computer cluster, network links, central processing units or disk drives. It improves the utilization and responsiveness for parallel workload in the cloud. In a large data center, processes of a job may need to be allocating each other to minimize the process execution. Load balancing is usually provided by dedicated software or hardware, such as a multilayer switch or a Domain Name System server process. Load balancing aims to optimize resource use, maximize throughput, minimize response time, and avoid overload of any one of the resources. Using multiple components with load balancing instead of a single component may increase reliability through redundancy. Load balancing is especially important for networks where it's difficult to predict the number of requests that will be issued to a server.

V.CONCLUSION AND FUTURE WORKS

Workload consolidation supported by virtualization technologies is commonly used for improving utilization in data centers. the difficulty in realizing parallelism, many parallel applications show a pattern of decreasing resource utilization along with the increase of parallelism. Scheduling parallel jobs for both efficient resource use and job responsiveness is challenging. a priority-based workload consolidation method to schedule parallel jobs in data centers to make use of underutilized node computing capacity to improve responsiveness. Our extensive simulation showed that our consolidation-based algorithm (AMCBF), even without knowing the job execution time, significantly outperforms the commonly used.In future work, Effectively partition the computing capacity of a data centre node into k-tiers, which may further improve the node utilization and responsiveness for parallel workload in the cloud done by sematic scheduler that to load balance a system effectively.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

REFERENCES

[1]HighPerformanceComputing(HPC) on AWS,"Amazon Inc.,http://aws.amazon.com/hpc-applications/, 2011.
[2] L. Barroso and U. Holzle, "The Case for Energy-ProportionalComputing," Computer, vol. 40, no. 12, pp. 33-37, Dec. 2007.
[3] J. Hamilton, "Cloud Computing Economies of Scale," Proc. AWSGenomics Cloud Computing Workshop, http://www.mvdirona.com/jrh/TalksAndPapers/JamesHamilton_GenomicsCloud20100608.pdf, 2010.
[4] D. Feitelson, A Survey of Scheduling in Multiprogrammed ParallelSystems. IBM TJ Watson Research Center, 1994.
[5] D. Feitelson and B. Nitzberg, "Job Characteristics of a ProductionParallel Scientific Workload on the Nasa Ames ipsc/860," Proc.Workshop Job Scheduling Strategies for Parallel Processing, pp. 337-360, 1995.
[6] J. Jones and B. Nitzberg, "Scheduling for Parallel Supercomputing:A Historical Perspective of Achievable Utilization," Proc.Workshop Job Scheduling Strategies for Parallel Processing, pp. 1-16,1999.
[7] L.G. Valiant, "A Bridging Model for Parallel Computation,"Comm. ACM, vol. 33, no. 8, pp. 103-111, 1990.
[8] U. Schwiegelshohn and R. Yahyapour, "Analysis of First-Come-First-Serve Parallel Job Scheduling," Proc. Ninth Ann. ACM-SIAMSymp. Discrete Algorithms, pp. 629-638, 1998.
[9] D. Lifka, "The Anl/Ibm SP Scheduling System," Proc. Workshop JobScheduling Strategies for Parallel Processing, pp. 295-303, 1995.
[10] D. Feitelson and M. Jettee, "Improved Utilization and Responsivenesswith Gang Scheduling," Proc. Workshop Job Scheduling Strategies for Parallel Processing, pp. 238-261, 1997.
[11] Y. Lin, "Parallelism Analyzers for Parallel Discrete EventSimulation," ACM Trans. Modeling and Computer Simulation, vol.2, no. 3, pp. 239-264, 1992.
[12] Z. Juhasz, S. Turner, K. Kuntner, and M. Gerzson, "A PerformanceAnalyser and Prediction Tool for Parallel Discrete Event Simulation,"J. Simulation, vol. 4, no. 1, pp. 7-22, 2003.
[13] R. Fujimoto, "Parallel and Distributed Simulation," Proc. 31st Conf.Winter Simulation: Simulation—A Bridge to the Future, vol. 1,pp. 122-131, 1999.
[14] A. Malik, A. Park, and R. Fujimoto, "Optimistic Synchronizationof Parallel Simulations in Cloud Computing Environments," Proc.IEEE Int'l Conf. Cloud Computing (CLOUD '09), pp. 49-56, 2009.
[15] R. Fujimoto, A. Malik, and A. Park, "Parallel and Distributed Simulation in the Cloud," Int'l Simulation Magazine, Soc. forModeling and Simulation, vol. 1, no. 3, 2010.