ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering (IJATCSE), Vol.2 , No.3, Pages : 22-25 (2013)
Special Issue of ICCSIE 2013 - Held during 24 May, 2013 Bangalore, India

Adaptive Job Scheduling in Heterogeneous Multicluster Systems

Harpreet Kaur[1], Gursimranjeet Kaur[2], Amit Chhabra[3]
[1]Deptt. Of Computer Science and Engineering, Guru Nanak University Amritsar, India, harpreetaneja23@gmail.com
[2]Deptt. Of Computer Science and Engineering, Guru Nanak University Amritsar, India, rabia.gem@gmail.com
[3]Deptt. Of Computer Science and Engineering, Guru Nanak University Amritsar, India, chhabra_amit78@yahoo.com

Abstract: Job scheduling and Processor allocation are two major areas of concern for improving the performance in parallel systems. Job scheduling decides the job sequence for processor allocation where as processor allocation is concerned with assignment of processors for incoming job. Resource fragmentation and speed heterogeneity are two major factors that affect the overall performance of the system. Previous works have focused on the speed heterogeneity but only at inter-cluster level i.e. computing speed among the clusters differs but the speed of processors within the cluster is same. In this paper heterogeneity within the same cluster is also considered and an adaptive scheduling policy is proposed which deals with the above stated issues.

Key words: Job Size, Resource Allocation, Scheduling Heterogeneous, Multi-cluster Systems.

INTRODUCTION

A cluster is a group of loosely coupled computers that work closely together as a single computational facility. Clusters consist of hundreds and thousands of standard CPUs Cluster applications have become more common in recent decades due to availability of cluster resources. The trend in the recent times is combining many clusters into huge computing environments to get more computing resources, a design familiar with supercomputers abilities known as “Multi-clusters”. Multi-cluster systems are made up of multiple, geographically distributed clusters that provide large computational power. As a results lower job utilization times and higher system utilizations are achieved which makes larger job sizes (in terms of processors) possible by allowing jobs to use processors in multiple clusters simultaneously in different clusters. Scheduling algorithms in traditional and distributed systems, which run on homogenous and dedicated resources, cannot work well in new circumstances. So the earlier used techniques are needed to be improved or redesigned for effective scheduling in heterogeneous multi-cluster environment.

In an attempt to improve performance of parallel systems many scheduling strategies have been developed in recent times. Generally, there are three scheduling schemes, Static scheduling, dynamic scheduling and adaptive scheduling [1].

Static scheduling is performed at compile time. Jobs are then allocated to individual processors before execution. The allocation remains unchanged during the execution. Dynamic scheduling is performed at run time and the load between the nodes could be shared even during execution time whereas adaptive scheduling is also done at run time but it could not share the load between nodes after job starts execution.

Heterogeneity is an important issue in a computational multicluster environment but it puts a challenge in designing effective load sharing policies. A heterogeneous multicluster system is feasible only if it brings performance improvement in all participating sites. Performance can be measured in the terms of job response time or average waiting time. A scheduling algorithm is said to be feasible if average response time of all the participating jobs improves. Job scheduling and Processor allocation are two major areas of concern for improving the performance in parallel systems. Job scheduling decides the job sequence for processor allocation where as processor allocation is concerned with assignment of processors for incoming job. Most common policies try to allocate an entire parallel job onto a single participating site but sometimes a job cannot fit in any single site due to occupation of resources by other jobs [8]. How job scheduler handles this situation is an important issue which comes under processor allocation.This paper proposes a scheduling policy which for heterogeneous multi-cluster system in which cross-site selection (co-allocation) policy is used.

RELATED WORK

Substantial amount of work has been done on various platforms viz. shared memory systems, distributed memory multiprocessors, clusters, multi-clusters and grid. Much work has been done on single cluster system.

Bucur [1] used Distributed ASCI Supercomputer (DAS) (multi-cluster) system in her research and studied the performance of co-allocation. *Co-allocation* is a technique in which a parallel job is broken into components and each component can be processed in a different cluster. For example, suppose that a job is waiting in a cluster’s ready queue. This job may require more nodes than are presently available on its particular cluster, but collectively there may be enough available nodes elsewhere in the multi-cluster to

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering (IJATCSE), Vol.2 , No.3, Pages : 22-25 (2013)
Special Issue of ICCSIE 2013 - Held during 24 May, 2013 Bangalore, India

accommodate the job. Co-allocation allows jobs to be mapped across cluster boundaries. In doing so, resource fragmentation is reduced and system utilization is increased. In this paper site and cluster are used interchangeably.

Bucur and Epema [1] studied the effect of system configurations on performance of co-allocation. Their observations show that in addition to the scheduling techniques, architectural and placement considerations also improve the performance of co-allocation.

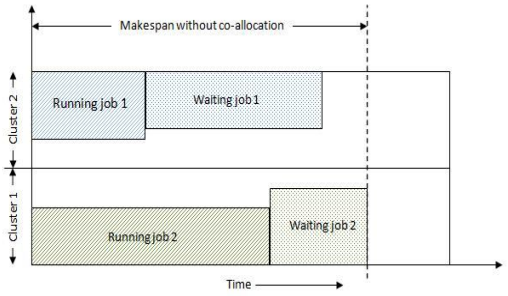

Fig 1. Scheduling without co-allocation [1]

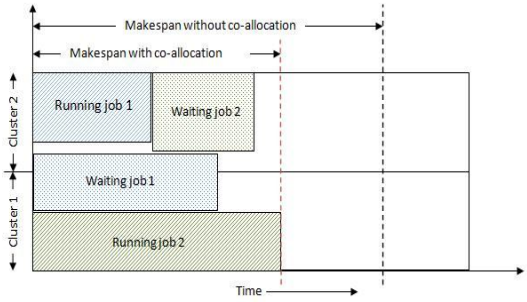

Fig 2. Scheduling with co-allocation [1]

The work by Bucur is focused in scheduling rigid jobs in a multi-cluster system. Fig1 shows the effect and Fig 2 compares the effect of co-allocation on the jobs in the waiting in the queue.

Dandamudi et al. [12] also worked on cluster system and proposed a two level space-sharing policy for heterogeneous cluster systems. The policy is based on the concept of Basic Processor Units (BPUs) to compute the partition size. The author used adaptive space sharing partitioning taking system load conditions into account and showed that it shows substantial performance improvements.

Extensive work has also been done on job scheduling in multicluster systems.

Huang [3], in his work, used moldable property of parallel jobs for scheduling in a grid. The author highlighted the concepts of resource heterogeneity in computational grid environment and challenges it posses on effective scheduling in multicluster environment.

Huang and Chang [2][10] showed that the speed heterogeneity and resource allocation are major factors which affects the overall system performance. The different allocation strategies results in leftover processors which in turn leads to the problem of *Resource Fragmentation* means there is no single site which can accommodate the incoming job whereas total number of processors in the whole system are enough or more than requirement of job. They introduced best site selection policy for such a heterogeneous grid known as *best-fit.* In this policy a particular site is chosen on which a job will leave the least number of free processors if it is allocated to that site.

England and Weissman [13] analyzed the costs and benefits of load sharing of parallel jobs in both homogeneous and heterogeneous grids.

. Later, Huang [8][9] developed adaptive processor allocation policies based on the moldable property of parallel jobs for heterogeneous computational grids. The author studied the speed heterogeneity only at inter cluster level in which nodes on different sites may have different computing speeds but the nodes on same site have same speed.

In this paper heterogeneity at both inter-cluster and intra-cluster considered at the same time number of processors at sites is also different.

PROPOSED POLICY

Multi-cluster Model

In the model, the multi-cluster system consists of several clusters (sites). Each participating site is a heterogeneous parallel computer. Each site is heterogeneous i.e. the number of processors as well as the computing speed of each processor are different. This means a parallel job can be allocated on any subset of nodes in a site. The parallel computer system uses space-sharing and run the jobs in an exclusive fashion. The system deals with an on-line scheduling problem in which jobs are submitted to the global queue. The jobs contain the information regarding the requirement of processors required by the job.

We assume a global scheduler which handles all job scheduling and resource allocation activities. The global scheduler contains the all the information regarding the incoming jobs in the queue, the status of all the participating sites i.e. number of available processors with all the sites before and after each allocation.

In order to take heterogeneity into consideration, he processors in the system are rated in terms of Basic Processor Unit (BPU) [12]. The physical processor with the lowest processing capacity in the system is considered to represent 1 BPU. The ratings of other processors are expressed in terms of BPUs [12].

Scheduling Process

When a job is submitted to the system, the scheduling is done in two phases: Job Selection and Processor Allocation.

When a job is submitted to the global scheduler it is added to the end of the queue at global scheduler. Job Selection policy (FCFS) selects the job from the queue to be sent for scheduling.

During the second phase, the site is to be selected on which the job must be run. Processor allocation is concerned with assignment of required number of processors for incoming job from available sites. As stated earlier, the computing power is calculated in terms of average unallocated BPUs at each site According to the availability of the required number of processors in the system at particular time different policies for processor allocation from different site are

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering (IJATCSE), Vol.2 , No.3, Pages : 22-25 (2013)
Special Issue of ICCSIE 2013 - Held during 24 May, 2013 Bangalore, India

applied. Following are the different policies used in our proposed approach.

The *fastest-first* policy is assumed for site-selection if there are no more jobs in the queue than the first job i.e. the job which is submitted as in this case the system load is low and this allocation does not affect the allocation other jobs in the queue.

The *most-fit* tries to allocate the job to a cluster which produces a leftover processor distribution, leading to the most number of immediate subsequent job allocations. It requires a more complex searching process, involving simulated allocation activities, to determine the target cluster. For each cluster which has enough processors for the waiting job, the system performs a series of simulated activities, based on the best-fit policy, to measure how many immediate subsequent allocations can follow the allocation decision [9]. After each cluster has been checked, the system selects the cluster with the largest number of immediate subsequent allocations to perform current job allocation.

In *Cross-site execution (coallocation)* scheduler will try to use this policy to run a parallel job across several sites if there is no single site having enough free processors. However, a parallel job might take much longer execution time when running across site boundaries. This is because the speed and bandwidth of inter-site network is usually much slower and less than those of intra-site network [6]. All the jobs arrive at waiting queue (WQ).

The proposed scheduling process can be explained using the algorithm whose pseudo code is given below:

```
Algorithm Schedule_a_job
    For each parallel job Ji to be scheduled do
        Perform Job Selection using FCFS algorithm
        Let xi be the number of BPUs needed by Ji
        Check the status of all the sites for
            available_processors
        If
        (xi < Number of available processors in any
            site)
          Perform site selection using site_select(Ji, xi
            ) algorithm
        Else if
        (xi < Number of available processors in any
            site)
          Perform coallocation using co-allocte( xi )
            algorithm
        End if
        Reconstruct available_processors using
            build_table algorithm
    End for
```

Fig 3. Schedule_a_job Algorithm

```
Algorithm FCFS
    for each i (1<= i<=p) do
        return(Ji)
    end for
```

Fig 4. FCFS Algorithm

```
Algorithm coallocate_select (Ji, xi)
While(Ji. gets xi processors)

Arrange all sites in decreasing order of number of available
processorsin a Coallocate_queue

Pick the first sit from the Coallocate_queue. //acc.to worst-fit

End while.
```

Fig 5. Co_allocate Algorithm

```
Algorithm site_select (Ji, xi)
    Check the jobs in waiting_queue
    If WQ=0 //no other job except first job
        Choose a site according to fastest-first
criteria.
    Else if WQ>0
        Choose a site according to most-fit criteria.
        End if
```

Fig 6. site_select Algorithm

```
Algorithm Build_table
    For each job Ji leaving the system do
        For all the site on which Ji was running
        Update available number of BPUs by
subtracting the allotted BPUs to job Ji from the existing
value.
    End for
    Recalculate the average number of BPUs at Sj
    Arrange the entries in the available_processors
```

Fig 7. Build_table Algorithm

CONCLUSIONS AND FUTURE DIRECTION

This paper proposes an adaptive scheduling technique which makes allocation decisions based on the policy which further accommodate more jobs for immediate execution for the jobs in a heterogeneous multi-cluster system. The scheduling technique allocates processors to the incoming job in order to reduce the waiting time by co-allocating the jobs if no site is able to provide the required number of processors to the incoming job instead of waiting for the processors to be available. The heterogeneity of multi-cluster in terms of computing speed of processors within a cluster and among the clusters in the system is considered which poses new challenges. In future a test bed will be built to evaluate the performance of the proposed scheduling technique.

International Journal of Advanced Trends in Computer Science and Engineering (IJATCSE), Vol.2 , No.3, Pages : 22-25 (2013)
Special Issue of ICCSIE 2013 - Held during 24 May, 2013 Bangalore, India

REFERENCES

[1] *Anca I.D. Bucur, and Dick H.J. Epema, "Scheduling Policies for Processor Coallocation in Multicluster Systems," IEEE Transaction on Parallel and Distributed Systems, VOL 18, No. 7, July 2007*

[2] *Huang, K.-C. and Chang, H.-Y. 2006. An Integrated Processor Allocation and Job Scheduling Approach to Workload Management on Computing Grid. In the Proceedings of the 2006 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'06), Las Vegas, USA, 703-709.*

[3] *Huang, K.C., Shih P.C. and Chung, Y.C. 2007. Towards feasible and effective load sharing in a heterogeneous computational grid. In the Proceedings of the second international conference on Advances in grid and pervasive computing, ed, 2007, 229-240.*

[4] *Huang, K.C., Shih P.C. and Chung, Y.C. 2009. Adaptive Processor Allocation for Moldable Jobs in Computational Grid. At 10th International Journal of Grid and High Performance Computing, 1(1), (March 2009), 10-21.*

[5] *Bucur, A.I.D. 2004. Performance analysis of processor co-allocation in multicluster systems. PhD Thesis, Delft University of Technology, Delft, The Netherlands.*

[6] *Huang, K.C., Shih P.C. and Chung, Y.C. 2008. Adaptive Processor Allocation with estimated job execution time in Heterogeneous Computing Grid*

[7] *Jones, W.L. 2005. Improving Parallel Job Scheduling Performance In Multi-Clusters Through Selective Job Co-Allocation. A P.hd Dissertation presented to the Graduate School of Clemson University, December 2005*

[8] *Huang, K.C., Shih P.C. and Chung, Y.C. 2008. Using Moldability to improve Scheduling Performance of Parallel jobs on Computational Grid. Springer-Verlag Heidelberg 2008*

[9] *Huang K.C, Kuan-PO Lai 2010. Processor Allocation Policies for Reducing Resource Fragmentation in MultiCluster and Cloud Environment. 2010 IEEE.*

[10] *Po-Chi Shih, Huang, K.C., Yeh-Ching Chung. Improving Processor allocation in Heterogeneous Computing Grid through Considering Both Speed Heterogeneity and Resource Fragmentation.2009 IEEE.*

[11] *Feitelson,D.G., Mu'alen A.W. Utilization, predictability, workloads, and User Runtime Estimate in Scheduling the IBM SP2 with Backfilling,IEEE Transaction on Parallel and Distributed Systems.*

[12] Dandamudi, S.P. and Zhou, Z. 2004. Performance of adaptive space-sharing policies in dedicated heterogeneous cluster systems. *Centre for Parallel and Distributed Computing, School of Computer Science, Carleton University, Ottawa, Ont., Canada K1S 5B6. Future Generation Computer Systems*, 895-906. DOI = 10.1016/j.future.2004.02.001

[13] England, D. and Weissman, J. B. 2005. Costs and Benefits of Load Sharing in the Computational Grid. *In Job Scheduling Strategies for Parallel Processing*, 160-175.