Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 2, Issue. 5, May 2013, pg.214 – 218

SURVEY ARTICLE

Survey of Resource and Grouping Based Job Scheduling Algorithm in Grid Computing

Sandeep Kaur[1], Sukhpreet Kaur[2]
[1]CSE & SGGSWU, India
[2]CSE & SGGSWU, India

[1] sandeepkandhola24@yahoo.com; [2] preetsukhpreet@gmail.com

Abstract— Grid computing is distributed computing environment, which enabling the dynamic selection, sharing resources based on availability, capability, performance or cost of these computing resource and simultaneously also based on organization specific requirements. The main purpose of Grid computing is to share the computational power, storage memory, network resource to solve a large problem. The goal of the job scheduler should be maximize the resource utilization and minimize the processing time of the job. Although Grids has been used for executing application with compute-intensive jobs but there are several applications with light weight jobs. The job grouping strategy improves the performance of application with large number of small processing requirements jobs. Motivation of this paper is to encourage the researcher in field of grid computing, so that they can understand easily the concept of job grouping strategy and can contribute in developing more efficient job grouping based scheduling algorithm.

Key Terms: - Grid computing; scheduling algorithm; Grouping strategy; Job scheduling; Load balancing

I. Introduction

User jobs might be small and of varying lengths according to their requirements. The overall processing undertaking the jobs with less processing requirement involves high overhead time in term of job transmission to and from Grid resources and job processing at the grid resources. Design an efficient scheduling strategy, in that scenario is challenge. When small jobs submitted individually to the grid resources over the networks create a communication overhead that is greater than the total computation time of each job at the resource, and this also leads to poor utilization of communication network and uneven utilization of the resources. Grouping based job scheduling algorithm, minimize the total processing time by reducing communication time of small scale and computation time, and on the other hand maximizing resource utilization than without grouping based scheduling. Grouping at the scheduling level is efficient in term of minimize overhead time and computation time, for reducing overall processing time of jobs. [5]. The Grouping algorithm in [4] integrated Greedy algorithm and FCFS algorithm to improve the processing of Fine-grained jobs. During job grouping activity computational capabilities and communication capabilities of the resources is considered in [14] to improve the performance. When fine-grained jobs are grouped into group then these jobs are represented in a coarse-grained form which helps in reduce the network latencies. After completion of grouping of all jobs, every grouped job is sent to the corresponding resources and the results of the processing are sent back to the user at their respective resources. This paper mainly focuses on small jobs scheduling in a grid, how they are grouped into coarse-grained jobs, how they are allocated and what is advantage and disadvantage of existing job grouping scheduling algorithm.

II. RELATED WORK

In [1] the author has proposed an algorithm that combines the advantages of three algorithms first shortest processing time, second, longest processing time, and third, earliest deadline first. The author realize that virtualization technique is required to address the problem of resource unavailability and software mismatch, With the help of virtualization a dynamic user defined environment is created, so there is less possibility of rejecting a job due to unavailability of resource. Since this is the hybrid of virtual and physical machine scheduling, overhead in creating virtual machine in the presence of physical resource is reduced. The ORC [2] scheduling in Grid environment includes the best fit followed by round robin scheduling which distribute the jobs among the available processors. The remaining un-allotted jobs are queued for next execution. Thus the order of jobs execution increases the performance of the processor and distribute load effectively across the network. This will reduce the waiting time of jobs in the queue and avoid the starvation .Many developed algorithms either considered the resource characteristics or application characteristics. But the adaptive fine grained job scheduling algorithm described in [3] considered both characteristics. AFJS monitoring the resources and starts with obtaining the information about the resources. In this algorithm a constraint is specified, which says that processing time of coarse grained job should not exceed the expected time. In [4] Grid computing, the author feel, there is a need to reduce the communication time, processing time and enhance resource utilization in case of scheduling the light-weight or small jobs. There are many applications in which consist a large number of lightweight or less processing requirement jobs. Scheduling with light weight gives low performance in terms of processing time and communication time. So to achieve high Performance, less processing requirement jobs are grouped before allocation of resource. This grouping algorithm integrated Greedy algorithm and FCFS algorithm to improve the processing undertake of fine-grained jobs. The algorithm considers the dynamic characteristic of the grid environment. The time complexity scheduling algorithm is very high. It does not pay any attention to memory size constraint and pre-processing time of job grouping is high. A Dynamic Job Grouping-Based scheduling in [5] maximizes the utilization of resource, and reduces the overhead time. This strategy dynamically assembles the individual Fine-grained jobs of an application into a group, and sends these coarse-grained jobs to the Grid resources. This dynamic grouping strategy based on the processing requirements of each application, Grid resources' availability and their processing capability and granularity size. Granularity size is the job processing time at the resources is used to measure the total amount of jobs that can be completed within a specified time in a particular resource. Relationship between the total number of jobs, processing requirements of those jobs, total number of available Grid resources, processing capabilities of those resources and the granularity size should be determined in order to achieve the minimum processing time. The bandwidth aware algorithm in [6] improves the performance by reducing the delaying factors in network environment and maximizing the utilization of grid resources. Bandwidth-awareness strategy combined with the dynamic job grouping strategy is to improve transmission delay and overhead. After knowing the processing capability and the bottleneck bandwidth, the scheduler selects the resource with the largest bottleneck bandwidth and started grouping independent fine-grained jobs together based on the chosen resource`s processing capability. Job groups are sent based on Largest Job First (LJF) manner to the corresponding resource. Longer execution time jobs executed concurrently with shorter execution time jobs. This strategy improves the overall execution time of the jobs. The grouping depends on a granularity size. It is the time within which a job is processed at the resources. This value is used to measure the total amount of jobs that can be completed within a specified time in a certain resource. It is one of the main factor in job grouping strategy that influences the way job grouping is performed to achieve the minimum job execution time and maximum utilization of the Grid resources. The framework described in [7] comparatively better than non-bandwidth-aware job grouping scheduling framework. After creation a list of jobs, sent to the job scheduler for scheduling arrangement. The job scheduler obtains information on available resources and their network information from the Grid Information Service (GIS), and routing information from routers. Based on the information, the job scheduling algorithm is used to determine the job grouping and resource selection for grouped jobs. Once all the jobs are put into groups with selected resources, the grouped jobs are dispatched to their corresponding resources for computation. There are some defects in the above scheduling algorithms. First, the algorithms don't take the dynamic resource characteristics into account. Second, the grouping strategy can't utilize resource sufficiently. And finally, it doesn't pay attention to the network bandwidth and memory size. To solve the problems mentioned above, an adaptive fine grained job scheduling in [8] is proposed. This algorithm is divided into two parts. In the first part, the scheduler receives resource status using GIS. And, it sorts job list in descending order, and assigns a new ID for each job. In the second part after gathering the details of user jobs and the available resources, the system selects Jobs in FCFS order to form different job groups. The scheduler selects resources in FCFS order after sorting them in descending order of their MIPS. Jobs are put into a job group one after another until sum of the resource requirements of the jobs in that group is less than or equal to amount of resource available at the selected resource site. In [9] described model, scheduler obtain the information from GIS. This information is used for job grouping and for resource selection. When the jobs are put into a group according to the selected resources, the grouped job is dispatched to resources for computation.

GBJS gives better performance than AFJS and DJGBSDA in terms of processing time. However, in [10] the author says there is need to balance the processing load among the resource in job grouping based scheduling algorithm. The proposed TMDGJS in [11] terms of processing time and overhead time, gives better performance, by minimizing overhead time and computation time. This strategy play important role in reducing overall processing time of jobs. Jobs are put alternatively from shortest list and added into job group according to the processing capability of the selected resource.

The above study evaluates an extension to dynamic job grouping based scheduling to reduce overall processing time of applications by minimizing the job allocation overhead, computation time and by balance the processing load among the resources.

III. GROUPING STRATEGY

Although Grids have been used extensively for executing intensive jobs, but there are also exist several applications with a large number of lightweight jobs which needs small processing requirements. Sending/receiving each small job individually to/from the resources will increase the total communication time and cost. These applications involve high overhead time and cost in terms of job transmission to and from Grid resources and, job processing at the Grid resources. So there is a need for an efficient job grouping-based scheduling system to dynamically assemble the individual fine-grained jobs of an application into a group of jobs, and send these coarse-grained jobs to the Grid resources. This dynamic grouping should be done based on the processing requirements of each application, Grid resources' availability and their processing capability. The job grouping strategy results in increased performance in terms of low processing time and cost. Job grouping based scheduling also useful for maximum resource utilization and reducing communication overhead time.

1. Basic Job Grouping

The grouping strategy groups the small scaled user jobs into few job groups according to the processing capabilities of available Grid resources. This strategy is based on processing capability (in MIPS), bandwidth (in Mb/s), and memory-size (in Mb) of the available resources. The existing Group Based Job scheduling algorithm basically follows the below given steps:-

- The job scheduler obtains information about the available resources from the Grid Information Service (GIS
- Based on the information, the job scheduling algorithm is used to determine the job Grouping and resource selection for grouped jobs.
- Jobs are started to put into a group until job processing requirements not equal to resource Processing capability. So During job grouping the following conditions must be satisfied: Groupedjob_ MI<=Resource_ MIPS*Chunk_ size.

Where MI (Million Instruction) is job's required computational power, MIPS (Million Instruction per Second) is processing capability of the resources and Chunk_ Size is User defined time

- This grouping process continues until all the jobs are put in groups.
- After completion of grouping, the scheduler sends the grouped Jobs to the corresponding resources for computation.
- The Grid resources compute the received grouped job and send result back to the user.

2. Advantage

Above explained Grouping strategy improve the performance over the ungrouped strategy following Way:-

- Eliminate the communication time of individual small scaled jobs.
- Gives better performance in terms of processing time of jobs.
- Maximum resource utilization.
- Reduce the network latency.

3. Disadvantage

- Some resources fully occupy and other may be unutilized or idle.
- Allocation large number of jobs to one resource will increase the processing time.
- Does not take dynamic characteristics of the resources into account.
- Processing load among the available resource is not balanced.

IV. DISCUSSION

With the help of [1] virtualization a dynamic user defined environment is created, so there is less Possibility of rejecting a job due to unavailability of resource. The reliability of the system is improved by proactive failure detection and RPC fault tolerance mechanism in this system. Load balancing is also achieved. This algorithm combines the advantages of standard algorithms such as shortest processing time, longest processing time, and earliest deadline first. In [2] Optimum Resource Constraint improved the efficiency of load balancing and dynamicity capability of the Grid resources. The scheduling reduces the waiting time of job and allocates the jobs to processor based on its capability and increase the processing time of job. Grouping strategy [3]-[4] efficiently reduces the processing time of jobs in comparison to others. The total processing capabilities of each resource may not be fully utilized as; each time the resource receives a small scaled job. Job grouping strategy aims to fully utilize the resource reduce the drawbacks on the total processing time and cost. In [6] a dynamic job grouping-based scheduling algorithm, jobs are grouped according to MIPS of the resource. This model reduces the processing time and communication time of job, but the algorithm doesn’t take the dynamic resource characteristics into account and the grouping strategy can’t utilize resource sufficiently. And it doesn’t pay attention to the network bandwidth and file size. To solve the problems mentioned above, an adaptive fine-grained jobs scheduling algorithm is produced. AFJS [4] an adaptive fine-grain job scheduling algorithm integrated Greedy algorithm and FCFS algorithm to improve the processing of fine-grained jobs. This algorithm considers the dynamic characteristic of the grid environment. This algorithm can reduce the executing time, but time complexity is higher than DFJS. A [6]-[7] Bandwidth-Aware Job Grouping-Based scheduling strategy that schedules the jobs according to MIPS and bandwidth of the resource and the model sends grouped jobs to the resource whose network bandwidth has highest communication or transmission rate. So, The Bandwidth -aware job grouping scheduling framework produce good results compared to a non-bandwidth-aware job grouping scheduling framework. The bandwidth-aware scheduling approach uses to determine the priority of each resource according the processing capability. The proposed Scheduling Model in [8] Grid Computing is a grouping based job scheduling strategy that has taken memory constraint of individual jobs together with expected execution time at the job level into account rather than at the group level. It reduces the total processing time of jobs, maximizes the utilization of the resource and minimizing the wastage of CPU power, comparative to other algorithm. GBJS gives better performance than AFJS and DJGBSDA in terms of processing time. The proposed model in [10] provides a real time grid computing environment and reduces the waiting time of the grouped jobs. Approaches of Grid scheduling don’t give much emphasis on the performance of a Grid scheduler in processing time parameter in Grid environment. . In [11] A Time-Minimization Dynamic Grouping-Based Job Scheduling reduces the communication overhead time and processing overhead time of each user and gives better performance than DJGBS in terms of processing time and overhead time.

V. CONCLUSIONS AND FUTURE SCOPE

In this paper, various job grouping based scheduling algorithms in grid computing have been surveyed. Grouping strategy plays a vital role to improve the overall performance of grid computing environment. Grouping of small scale job into group efficiently reduces the processing time of jobs. Grouping strategy based on the processing capability of selected resource. After resource selection, Jobs are started to put into the group. This process continues until the resource capability is less to the sum of grouped job requirement. This technique is also useful in maximum resource utilization, as jobs are grouped based on resource capability. In that scenario some system are fully occupied whereas some may be remains unutilized or idle. Allocation of large number of jobs to one resource will increase the processing time and leads unbalancing processing load among the resources in grid computing environment. So, there is an efficient job grouping-based scheduling strategy is required during the job grouping activity, so that the processing loads among the selected resource are balanced. Moreover resource can be managed by considering some more factors like current load of resource, network delay, QoS (Quality of Service) requirements’ will be taken into account ,Which will be helpful in increasing the performance of Grid computing. In future our focus on developing an efficient scheduling algorithm which not only reduce the communication time of applications with large number of less requirement jobs but also balance the load among the selected resources in Grid computing environment.

REFERENCES

[1] S.Thamarai Selvi, Ponsy R.K.Sathia Bhama, S.Architha, T.Kaarunya, K. Vinothini, Scheduling In Virtualized Grid Environment Using Hybrid Approach, International Journal of Grid Computing & Applications (IJGCA) Vol.1, No.1, September 2010.

[2] K.Somasundaram, S.Radhakrishnan,” Node Allocation In Grid Computing Using Optimal Resource Constraint (ORC) Scheduling”, IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.6, June 2008

[3] Quan Liu Yeqing Liao, “Grouping-Based Fine- grained Job Scheduling in Grid Computing”, Vol.1, pp.556- 559,
IEEE First International Workshop on Education Technology and Computer Science, 2009

[4] Yeqing Liao, Quan Liu,” Research on Fine-grained I.J. Information Engineering and Electronic Business, 2009, 1, 9-16.

[5] Nithiapidary Muthuvelu, Junyang Liu, Nay Lin Soe, Srikumar Venugopal, Anthony Sulistio and Rajkumar Buyya,” A Dynamic Job Grouping- Based Scheduling for Deploying Applications with Fine-Grained Tasks on Global Grids”

[6] T.F. Ang, W.K. Ng, T.C. Ling, L.Y. Por and C.S. Liew, 2009. A Bandwidth -Aware Job Grouping -Based Scheduling on Grid Environment. Information Technology Journal, 8: 372-377.

[7] Ng Wai Keat, Ang Tan Fong, Ling Teck Chaw, Liew Chee Sun, Scheduling Framework For Bandwidth-Aware Job Grouping-Based Scheduling In Grid Computing.

[8] S. Gomathi, Dr.D.Manimegalai, An Analysis of MIPS Group Based Job Scheduling Algorithm with Other Algorithms in Grid Computing

[9] Vishnu Kant Soni, Raksha Sharma, Manoj Kumar Mishra,” Grouping-Based Job Scheduling Model in Grid Computing.

[10] Pinky Rosemarry, Ravinder Singh, Payal Singhal And Dilip Sisodia, Grouping Based Job Scheduling Algorithm Using Priority Queue And Hybrid Algorithm in Grid Computing.

[11] Manoj Kumar Mishra, Prithviraj Mohanty, G. B. Mund,” A Time-minimization Dynamic Job Grouping-based Scheduling in Grid Computing”, International Journal of Computer Applications (0975 – 8887) Volume 40– No.16, February 2012