International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 4 Issue 7, July 2015

Implementation of Computational Grid with Improvement in Performance and Throughput

Kajal Patel, Hiral Darji

***Abstract*— In Grid computing that have developed the networked system that acts as a Grid. One of the grid participants acts as the submission host from where we are going to submit the host. The main goal of computational grid is to reduce the processing time by distributing the long task among multiple machines which have their computation cycles free. The advantage of high-performance networks in conjunction with low-cost, powerful computational machines have made possible the development of a new set of technologies termed "computational grid". These technologies are making possible the creation of very large scale distributed computing systems by interconnecting geo graphically distributed computational resources via very high-performance networks. This provides high computational power that can be brought to bear on large-scale problems in several domains. Results for the various performance parameters and compare the existing result of various performance parameters like Execution time, Throughput. The main aim of this research is decrease the execution time of jobs from the available resources. We developed proposed algorithm and take results. Results are decreasing the execution time with compare of general min-min algorithm.**

***Index Terms*—Grid Computing, Fault Tolerance**

I. Introduction

Grid computing is an interesting research area that integrates geographically-distributed computing resources into a single powerful system. Many applications can benefit from such integration. Examples are collaborative applications, remote visualization and the remote use of scientific instruments. Grid software supports such applications by addressing issues like resource allocation, fault tolerance, security, and heterogeneity. Parallel computing on geographically distributed resources, often called distributed supercomputing, is one important class of grid computing applications. Projects such as SETI@home, Intel's Philanthropic Peer-to-Peer Program for curing cancer and companies such as Entropies show that distributed supercomputing is both useful and feasible [1]. Grid computing is a kind of parallel computing that enables the sharing, selection, and aggregation of geographically distributed "autonomous" resources, at runtime, as a function of availability, capability, performance, cost, and users' quality-of-service requirements. One of the services that a Grid can provide is a computational service. Computational services execute jobs in a distributed manner. A Grid providing computational service is often called Computational Grid.

Kajal Patel *M.E Student of Computer Engineering,SCET,Kalol,Gujarat,India Mobile No:9898428988*
Hiral Darji *Assistant Professor of Computer Engineering,SCET,Kalol,Gujarat,India Mobile No:9904317117*

II. LITERATURE SURVEY

1. Min-Min Scheduling Algorithm:

The Min-Min algorithm is a simple algorithm which runs fast and delivers the satisfactory performance. Min-Min begins with the set meta-task (MT) of all unassigned tasks. It has two phases. In the first phase, the set of minimum expected completion time for each task in MT is found. In the second phase, the task with the overall minimum expected completion time from MT is chosen and assigned to the corresponding resource. Then this task is removed from MT and the process is repeated until all tasks in the MT are mapped. In most situations, it maps as many tasks as possible to their first choice of service resources. However, the Min-Min algorithm is unable to balance the load well since it usually schedules small tasks first. This algorithm takes $O(n^2m)$ time. Here n is number of tasks in MT and m is number of resources. [2]

2. Research on Fine-grained Job Scheduling Algorithm:

This paper focuses on lightweight jobs scheduling in Grid computing. In 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 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 AFJS (Adaptive Fine-grained Job Scheduling) [3] 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 algorithm.[4]

3. Earliest Deadline First With Shortest Remaining Time:

Earliest deadline first With Shortest Remaining Time (EDSRTF) is a scheduling algorithm in which scheduling is done according to the deadline of jobs and remaining time. The job with the earliest deadline will get resource first while the job with large value of deadline will have to wait irrespective of their execution time. In EDSRTF scheduling algorithm first execution time along with the deadline of the job is entered. Then according to this scheduling algorithm, first six jobs are directly assigned to the processor P1, P2, P3, P4, P5 and P6 respectively. For the next jobs; processors are assigned according to the deadline of the jobs. If the current job on the processor has large value of deadline than the arriving job, then processor will stop executing the job and will execute the new coming job as its value of deadline is small. But if the current job has small deadline value then processor will keep executing the current job. Then in this case the next processor will be checked for the job and this process will continue until the job gets the processor for its execution. This process is repeated for each job. In the case if no processor satisfied this deadline condition then job will get the processor with shortest remaining time.[5]

4. The Grouping Based Job Scheduling Algorithm:

In the first phase, the scheduler receives information about the resource status from the Grid Information Service (GIS), and sorts the jobs in descending order. In the second phase, the system selects jobs in First Come First Served (FCFS) order and forms different job groups. The scheduler will select resource in FCFS order after sorting them in descending order of their MIPS. The jobs are put into the job groups until the sum of resource requirement of the jobs in that group is less than or equal to amount often source available at selected site. As soon as the job group is formed, the jobs are assigned to the corresponding resource. After execution the job groups, the result is sent to the corresponding user and the resources are available to the Grid System.[6]

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. 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.

III. PROBLEM DESCRIPTION

Based on Survey in Min-Min Algorithm, whenever one acts better than the other based on the standard deviation of minimum completion time of all unassigned tasks in a meta-task. Evaluation of our new heuristic was done through a simulation environment called Alchemi. The experimental results show that the Selective algorithm outperforms the traditional Min-Min heuristic. Many issues remain open like deadline of each task, cost of execution on each resource, cost of communication and many other cases that can be a topic of research. Finally, we intend to reuse our new scheduling heuristic in an actual environment for practical evaluation.

IV. SIMULATION ENVIRONMENT

Alchemi Framework[7]

The aim of Alchemi grid computing framework is not to develop the grid software as easy as possible but flexible, scalable, reliable, and extensible. The key features of the Alchemi are,

- Internet based cluster computing for desktop computer without a shared file system.
- Dedicated execution by cluster and individual nodes.
- Object-oriented grid thread programming model.

V. METHODOLOGY AND IMPLEMENTATION

A. Related Definitions of Proposed Algorithm

The related definitions of proposed algorithm are as follows:

- ETkj (Execution Time)
  The amount of time taken by machine(mj) to execute task tk given that mj is idle when task tk is assigned.
- CT j (Completion Time)
  The expected completion time of mj.
  Amount of time required for any particular task to be completed.
- Mat(mj) (Machines availability time)
  The time at which machine (mj) completes any previously assigned tasks.
- Group (CTk, tk, Machines)
  The function "f" is used to group all the tasks and machines that has minimum completion time.
  The best minimum task/machine pair (m,n) is selected from the Group.
- MeanCT
  It is used to find the mean completion of all the machines.

B. Proposed Algorithm:

Input: Task tk of machines (mj)
Output: Reschedule task tk on machine (mj)

1) There are tasks

ISSN: 2278 – 1323

2) For all the task (tk) to be schedule
3) For all the machine (mj)
4) Compute CTk,j as CTk,j = Mat (mj) + ETk,j
// Where CT = Completion Time, ET = Execution Time and Mat(mj) = Machines availability time
5) End for
6) Create Group as (CTk, tk, machines) = f (CTk, tk, 1, 2 …)
7) End for
8) Now, select the best minimum pair (task, machine) from the Group
9) And Compute minimum CTm,n and reserve the task m on machine n
10) End while
11) Calculate MeanCT = ($\sum CTj$) / No. of machines
12) For all the machine mj
13) if (CTj >= MeanCT)
then select the task reserved on the host
14) End for
15) For all the task tk reselected with for all machine mj (CTj <= MeanCT)
16) Compute New CTk,j = CT (task k, host j)
17) if (New CTk,j <= MeanCT)
then select that task and machine which have best minimum time.
18) Now, Reschedule task tk on machine mj
19) Compute New CTm,n
20) End

VI. RESULTS

A. *Alchemi Performance Graph*

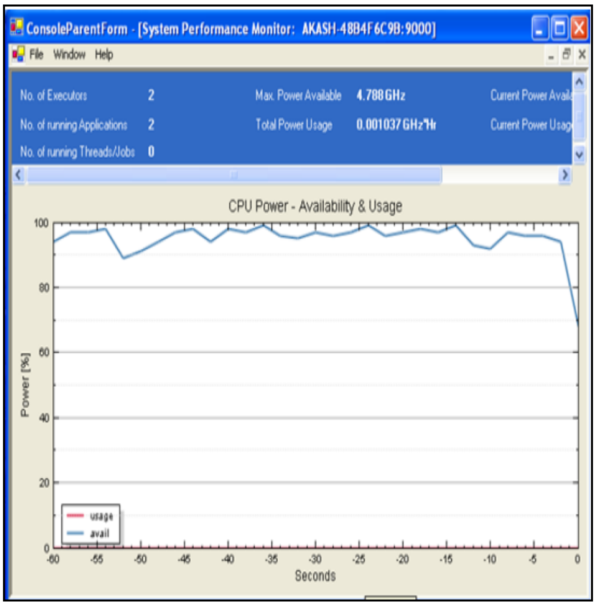

Figure 5.1: Alchemi Performance graph with two executors at idle time

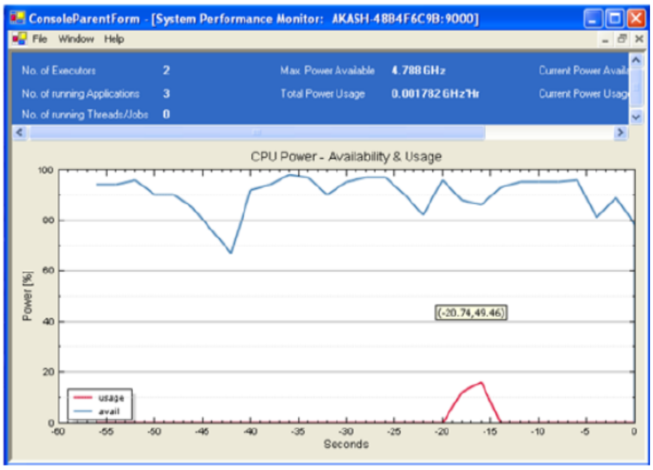

Figure 5.2: Alchemi Performance graph with two Executors at application runtime

B. *Various Performance Parameters after Applying Proposed Algorithm*

No. of Executor	Execution Time (Second)	Max Power Per Time	Max Power Per App	Running Thread	Current Power Usages
01	14.12	19.534	01	01	34%
02	13.11	17.632	01	02	40%
03	11.54	16.577	01	03	47%
04	10.86	14.579	01	04	58%
05	08.69	12.638	01	05	75%

Table : Various Performance Parameters after Applying Proposed Algorithm

VII. CONCLUSION

In scheduling strategy for high throughput computing in a Grid Environment is proposed. Min-Min algorithm considers overall performance of the resources for deciding the assigning sequence of the tasks. To decide which task will be submitted for execution in Min-Min Algorithm, whenever one acts better than the other based on the standard deviation of minimum completion time of all unassigned tasks in a meta-task.

REFERENCES

[1] Akshay Luther, Rajkumar Buyya, Rajiv Ranjan, and Srikumar Venugopal "Peer-to-Peer Grid Computing and a .NET-based Alchemi Framework.
[2] Kobra Etminani , Prof. M. Naghibzadeh "A Min-Min Max-Min Selective Algorithm for Grid Task Scheduling"@2007
[3] Shweta Jaiswal "A Survey Host Load Prediction In Computational Grid Environment" www.ijera.com ISSN : 2248-9622, Vol. 3, Issue 5, Sep-Oct 2013, pp.1430-1434

[4] Yeqing Liao, Quan Liu "Research on Fine-grained Job scheduling in Grid Computing" I.J. Information Engineering and Electronic Business, 2009.
[5] Mr. Pradeep Mittal, Dipti Sharma "Job Scheduling Algorithm for Computational Grid in Grid Computing Environment". Volume 3, Issue 5, May 2013.
[6] Sandeep Kaur, Sukhpreet Kaur "Survey of Resource and Grouping Based Job Scheduling Algorithm in Grid Computing" IJCSMC, Vol. 2, Issue. 5, May 2013, pg.214 – 218
[7] CH SOMESWARA RAO , V MNSSVKR GUPT " IMPLEMENTATION OF OBJECT ORIENTED APPROACH TO MODIFIED ANT ALGORITHM FOR TASK SCHEDULING IN GRID COMPUTING" Vol.1, Issue1, pp-134-139. September 4th, 2011.

Kajal Patel is doing M.E from Gujarat Technological University (GTU) India in Computer Engineering. She received B.E degree in year 2013 in Computer Engineering from Gujarat Technological University (GTU) India. Her research work is in Grid Computing.

Hiral Darji has done her M.Tech degree from Singhania University in Computer Engineering in year 2012. She has done her B.E degree from H.N.G.U. in Computer Engineering in year 2008. Today she has 4.2 years of experience in teaching field.

ISSN: 2278 – 1323