Volume 3, No. 4, April 2012
Journal of Global Research in Computer Science
ISSN-2229-371X
REVIEW ARTICLE
Available Online at www.jgrcs.info

A SCHEDULING IN HETEROGENEOUS COMPUTING ENVIRONMENT USING ALEA SIMULATOR BASED ON GRIDSIM

Satheesh Kumar N[1], Ilayaraja AP[2] and PNV Pavan[3]
School of Computing Science Engineering,VIT University, Vellore-632014, Tamilnadu, India.
{satheesh.040[1], apilayaraja2, nvpavan273}@gmail.com

Abstract: Grid computing is well established and is used for large scale simulations. The resources available on the grid vary from supercomputers to clusters and single PC's. In this project we are using simulator of grid (GridSim) that simulates computers and installed ALEA scheduler on the top of the GridSim. This work will concentrate on the representation of parallel jobs and their execution. Our goal lies in the comparison of various scheduling approaches and evaluating the best performance algorithm for the grid environment. Grid Simulators are often used for evaluation of new scheduling methods. The job is composed of large number of independent tasks executed on separate processing units. The objective of the project is to evaluate the performance of simplest scheduling algorithms based on various parameters. Taking into consideration the resource and the job specifications, this project also calculates the total usage of the resources. Grids build on the concept of distributed computing, where there is no longer a centralized computer establishing a hierarchy. Instead, software provides a way to divide up tasks so they are processed in parallel. That makes it possible to use a range of inexpensive computers from single or different manufacturers to address a complex computational task, recreating the power of a supercomputer with off-the-shelf PCs.

INTRODUCTION

In recent years, the tremendous increase in the data storage capacities, processing facilities and information carrying capabilities of the interconnecting networks made Grid Computing a vital area of research and computing platform of the future. One of the objectives of computational grids is to offer applications the collective computational power of distributed but typically shared heterogeneous resources. Unfortunately, efficiently harnessing the performance potential of such system (i.e., how and where applications should execute on the grid) and successfully executing the applications without any failures, these failures are the challenges principally to the distributed, shared and heterogeneous nature of the resource involved [1]. Though the grid has become very popular in a short period of its emergence, static case of scheduling of jobs are only available (i.e., the processing of jobs should be known in advance) this has become major drawback in large scale computing application like Grid Computing; in our work we started a survey to find a solution for this problem.

The solution is which we will provide in this project will be able to deal with common problems of the job scheduling in Grids like heterogeneity of jobs and resources, and dynamic runtime changes such as arrivals of new jobs. In this project we are using simulator of grid (GridSim) that simulates computers and installed ALEA scheduler on the top of the GridSim. This work will concentrate on the representation of parallel jobs and their execution. Our goal lies in the comparison of various scheduling approaches and evaluating the best performance algorithm for the grid environment. There are many technologies and architecture that precede Grid Computing in terms of attempting to "merge" computing power, share storage, and facilitate program-to-program communication. Comparison of some of these architectures is described below. Grids build on the concept of distributed computing, where there is no longer a centralized computer establishing a hierarchy. Instead, software provides a way to divide up tasks so they are processed in parallel. That makes it possible to use a range of inexpensive computers from single or different manufacturers to address a complex computational task, recreating the power of a supercomputer with off-the-shelf PCs. In recent years the field has seen a burst of activities which received attention in the technical field and the popular press[3]. This computational grid called as grid computing refers to the share of coordination resources and the solution of the questions in one dynamic, with many organizations virtual organization VOs. It now become a quite active research field in today's world which is considered as Internet for next generation. And through the high-speed computer network, it concentrates all kinds of supercomputers, the data-storage system, the virtual reality system, each kind of special-purpose instrumentation equipment and so on [7].

Grid computing is something similar to cluster computing, it makes use of several computers connected is some way, to solve a large problem. There is often some confusion about the difference between grid vs. cluster computing. The big difference is that a cluster is homogenous while grids are heterogeneous. With the rapid expanding of gird scale, grid integrates not only cluster resources, but also more wide-area personal resources which are mostly non-dedicated. Therefore, the characteristics of node unavailability in grid environment distinguish greatly from that in cluster environment [4]. Simple allocation schemes such as First-Come-First-Serve (FCFS) or FCFS with First-Fit backfill (FCFS/FF) are used in practice [3]. P2P systems cannot guarantee any QoS. They provide their services on a "best-effort" basis [13]. The responsibility and privileges of the participating entities are not defined beforehand. Each peer is responsible for defining and maintaining the access policies for his resource [12]. On the other hand, the computers in the cluster are normally contained in a single location or complex [9]. The rationale behind the Grid approach is that more processing power, global collaboration among researchers, access to and sharing of data, instruments and expertise are required to solve many of today's large, complex scientific problems [8]. In recent years the field has seen Unfortunately, efficiently harnessing the performance potential of such system (i.e., how and where applications should execute on the grid) and successfully executing the applications without any failures, these failures are the challenges principally to the distributed, shared and heterogeneous nature of the resource involved [1]. a burst of activities which received attention in the technical field and the popular press[3]. GridSimulators are often used for evaluation of new scheduling methods.

The job is composed of large number of independent tasks executed on separate processing units. The objective of the project is to evaluate the performance of simplest scheduling algorithms based on various parameters. Taking into consideration the resource and the job specifications, this project also calculates the total usage of the resources.

RELATED WORK

In this project I am trying to create various jobs and machines with different characteristics and evaluate its performance based on various parameters, while I have started this work, I have undergone a literature survey based on various scheduling algorithm because my work is majorly related on scheduling the jobs to different machines and evaluate the performance [4]. Job scheduling in parallel systems has been extensively researched in the past. Typically this research has focused on allocating a single resource type (e.g. CPUs) to jobs in the ready queue. However, the use of many of these scheduling algorithms has been limited due to restrictions in application designs, runtime systems, or the job management system itself.

Therefore, simple allocation schemes such as First-Come-First-Serve (FCFS) or FCFS with First-Fit backfill (FCFS/FF) are used in practice [3]. Heuristics are proposed to obtain the optimal scheduling, such as min-min heuristic, max-min heuristic, suffrage heuristic. However, due to the diverse failures and error conditions in grid environments, the unavailability of resource nodes is increasingly becoming severe and poses great challenges to grid workflow scheduling. For example, grid resources are mostly non-dedicated and can enter and depart without any prior notice. In addition, the change of resource local policy, the breakdown of software and hardware and the malfunction of network fabric can result in resource inaccessibility. Hence, jobs fail frequently and QoS cannot be guaranteed [2]. Volunteer computing or public resource computing is based on the idea that willing participants with idle resources, like CPU, will happily donate these in the aid of some cause or task without any tangible remuneration for their use. It is characterized by the ease of entry and exit of resource providing participants in sharing their resources.

Typically resources are allocated to volunteers on a pull basis, with blocks of tasks being allocated on request, and collected when the next set of tasks is requested. To see the effectiveness of volunteer computing in providing tremendous amounts of computing power, one has to look no further than the SETI@home (Anderson et al. 2002) project. In 2001, the average computing throughput provided by volunteers was 23.76 Teraflops! There are a many other BOINC (Anderson 2004) based projects including Rosetta@home, TANPAKU@home, LHC@home, etc. One social change that is observable in the volunteer computing environment is that a high computing to data ratio is becoming less important, and therefore the advent of broadband has the potential for data intensive applications to benefit from this model. However, whilst some might argue that BOINC does not strictly meet the criteria established for defining a Grid due to the topology (star or similar (Sarmenta 2001)), the work is often published at Grid conferences and provides a usefully different allocation mechanism (Anderson et al. 2005).

Rational of RCGS is that it computes the reliability of node during task's running time and then makes scheduling decision based on the reliability cost of task. Finally, performance evaluation is conducted to compare RCGS with other scheduling algorithms, and performance evaluation results demonstrate that RCGS improves the dependability of grid workflow execution and the success ratio of tasks execution, and accordingly decreases the makespan of workflow execution [4]. It is known that extensive research is essential before working out on our interesting area and finding problems, solutions and to know how other person or researchers are approaching on the same issue before any design or implementation can be attempted, the analytical and problem-solving thinking is required, which altogether balance out the implementation side. A complete overview of the extensive Grid world needs to be obtained in order to find the existing problem and possible solution. Ian Fosteret.al. Grid workflow is a complex and typical grid application, distinguishing from the traditional workflow in: (1) efficiently utilizing wide-area distributed resources; (2) cooperating among heterogeneous organizations; (3) solving computing-intensive and data-intensive tasks; and (4) making workflow more like service flow to exploit OGSA idea. Grid workflow provides an approach to complex and collaborative scientific researches, like high energy physics, geophysics, astronomy and bioinformatics.

SYSTEM MODEL

Gridsim environment is very important in the buffer management policies. The process of creating an experiment in Gridsim requires following [12] [13]. In this first we initialize the Gridsim package by calling GridSim.init() and GridSim.init Network Type methods (GridSimTags.NET). The proposed system which is developed using GridSim, it acts as the main part of the project over this GridSim an Visual Modeler (VM) and ALEA scheduling simulator is embedded. Here using the Visual Modeler two property parameters has been created i.e., User Property Dialog Box and Resource Property Dialog Box with help of the various characteristics involved in this property various machines and jobs has been created this will act as the input for our system and after creating this we have tried to use the scheduling simulator to schedule the various jobs to various machines based on its availability and finally its performance has been evaluated and finally the various scenarios is shown through graph.

In this services register into the grid information service fieg 1. Here maintain the services storage. We create one or more grid resource. Each grid resource contains single or many machines and every machine have one or more processing elements Grid user entities have one or more Gridlets or jobs to be processed. Gridlets have the job description or the data. We create the network topology by connecting the grid user and resource entities. Each resource has number of processes, speed of processing and internal process scheduling policies. Grid user contains one or more Gridlets or jobs to be processed. Gridlet ig 1 contains the job information or data. In this we can consider the three Gridlets. The job scheduling system has two active actors. The user actor is nothing but the job, the jobs which are created called as users and the resources are another actor which is nothing but the machines

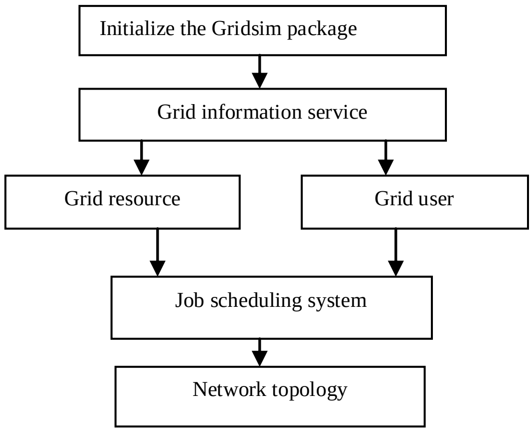

Figure 1 system model

With the various characteristics and MIPS ratings a machines are created, the jobs which are created will be sent to the job scheduling system over there the queue of several jobs will be formed with the help of ALEA scheduler all the jobs will be scheduled based on different scheduling strategies. Here in our project this ALEA scheduler is capable to handle both static and dynamic situation. In the static case, all jobs are known in advance while the dynamic situation means that jobs appear in the system during simulation. In this case generated schedule is changing through time as some jobs are already finished while the new ones are arriving. Comparison of FCFS, local search and dispatching rules is presented for both cases and we demonstrate that the new local search based algorithm provides the best schedule while keeping the running time acceptable.

FUNCTIONAL REQUIREMENT SPECIFICATION

User Case:

The user creates a job based on the user property and all the jobs which are for processing will be submitted to the job scheduling system. All the jobs which are submitted over there will be placed in a queue format. Finally all the jobs will be scheduled based on various strategies.

a. The users (Jobs) are created based on user property dialog box.
b. Based on various budget and deadline characteristics the jobs have been created.
c. The jobs created will be submitted to the jobs scheduling system.
d. All the jobs which have been created will be placed on the form of queue.

Resource Case:

In case of multiple resources, all the jobs will be issued for processing based on the characteristics of the jobs and the resource which is needed for processing a job will be submitted to the resources which are created with different characteristics will be provided with various jobs for processing.

a. The resources are with the processing elements and MIPS ratings using visual modeler.
b. The created machines with various properties will be provided with various jobs.
c. The machines will process all the jobs which are allotted to it.
d. After all the jobs have been processed it will provide the performance of the system.

PERFORMANCE EVALUATION

After creating various jobs and machines with various characteristics, scheduling simulator will start performing its operation. In our project an FCFS scheduling algorithm is used to schedule the jobs to various machines. Based on the output received a graph has been generated to estimate its flow to evaluate their performances. Finally the comparison is made with various scenarios and the result is shown graphically. The design of our system consists of simple GridSim based operation. In our analysis a GridSim acts as the test bed that is this simulator is placed on the top of the operating system which acts as the heart of our processing. Moreover this GridSim is given with an GUI called Visual modeler and another scheduler simulator is used called ALEA simulator. The detailed description is given in the design methodology.

RESULTS AND DISCUSSION

When we started a work on our project, our main aim was to evaluate performance of various resources with various scenarios, the concept lies when evaluating performance is to have various scheduling algorithm and various resources with different characteristics. As the grid has become very popular in its short period of its emergence, the usage of performing scheduling operation is not up to the mark because the scheduling process exists as of now is only minimal i.e., when anyone trying to perform a operation he should make the system or scheduler to know the jobs which it has to be scheduled in advance. So this is not a permanent solution in this kind of environment. After having done a extreme literature survey we have started working on evaluating performance of the system with various scheduling algorithms such as FCFS, Earliest Deadline First, Easy Back filling and so on., our work is not only performing scheduling process though evaluating the performance is the major issue in our project but before that we have found a way to create a Jobs and Resource, after creating Jobs and resource with various characteristics, these has been used as the input for our processing. Finally each and every jobs has been given to the resources in order of not keeping any of machines idle i.e., jobs will be allocated to the machines based on their availability. But final result we obtained is to evaluate the performance only for the FCFS scheduling algorithm.

This basic scheduling algorithm has taken lot of time to make the ALEA to coordinate with the GridSim. The task manager takes the jobs and an appropriate node list from scheduler and allocates the jobs on to the nodes. Fig 2 shows resource utilization base on machines, each machine contains processing elements. In this user side utilized resource 250, sever side utilized resource 200. Fig3 The scheduling decision made resource predication on runtime of task.

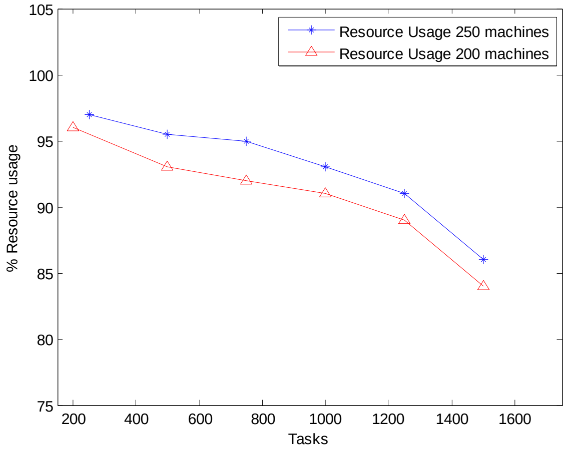

Figure 2 resource usage

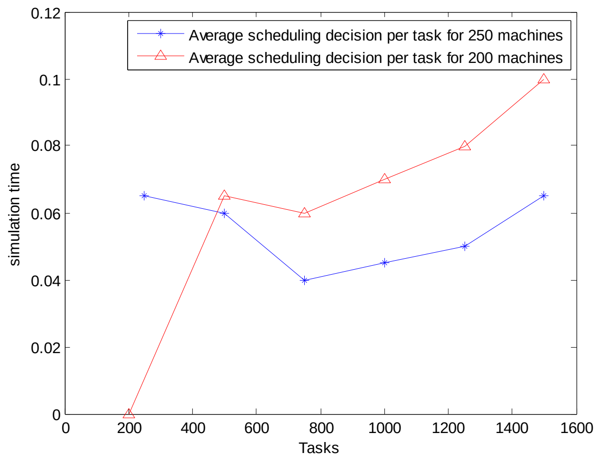

Figure 3scheduling decision

CONCLUSIONS AND FUTURE ENHANCEMENTS

We started a project by having an idea of performing scheduling operation in grids, thus the heterogeneity of jobs and resources has been created with various characteristics. To demonstrate this feature we used Grid Sim simulation and ALEA scheduling simulation. At the beginning of our work we thought of showing the performance variation using various scheduling algorithm but we were able to provide the solution only for the FCFS scheduling algorithm. Our future work will on one hand focus on extensions of the simulation environment, e.g., addition of network topology, support for different job types including preemptive and priority jobs, workflows, etc. We will also be able to model job migration or an estimated running time of the job. The environment will be further extended to simulate different failures (resource or job crash, network disconnection, etc.) to test robustness of scheduling algorithms. On the other hand, we plan to use this environment to implement and evaluate differing scheduling algorithms and technologies, such as Backfilling or Convergent Scheduling. The students who are all interested on this area can try to find the solution for using different scheduling algorithm except the one we have used (FCFS).

REFERENCES

[1]. Foster, I. and Kesselman, C. (Editors): "The Grid: Blueprint for a New Computing Infrastructure" (2nd ed.), Morgan Kaufmann, 2004.

[2]. Ding, C.H., Nutanong s., and Buyya, R.: "P2P Networks for Content Sharing", Technical Report, GRIDS-TR-2003-7, University of Melbourne, Dec, 2003.

[3]. Anthony Sulistio, Chee Shin Yeo, and Rajkumar Buyya "Visual Modeler for Grid Modeling and Simulation (GridSim) Toolkit" Berlin Heidelberg 2008.

[4]. Dalibor Klus´aˇcek, Ludˇek Matyska, and Hana Rudov´a "Alea – Grid Scheduling Simulation Environment" Morgan Kaufmann, 2007.

[5]. Yongcai Tao, Hai Jin, Xuanhua Shi "Grid Workflow Scheduling based on Reliability Cost"Infoscale'07, June 6–8, 2007, Suzhou, China. Copyright 2007 ACM 1-58113-000-0/00/0004.

[6]. Stefan Krawczyk and Kris Bubendorfer "Grid Resource Allocation: Allocation Mechanisms and Utilization Patterns" Sixth Australasian Symposium on Grid Computing and e-Research (AusGrid2008), Wollongong, Australia, January 2008.

[7]. William Leinberger, George Karypis, Vipin Kumar " Job Scheduling in the presence of Multiple Resource Requirement " 1999 ACM 1-58113-091-8/99/0011.

[8]. M. Li, Y. Cui, Y. Ren and Yuan Tian "Intelligent Security Multi-agent Framework in Grid Computing" Dalian University of Technology, 116621, China. September 28, 2007.

[9]. Junyan Wang, Yuebin Xu, Guanfeng Liu, Zhenkuan Pan, and Yongsheng Hao Computer Software Lab, Information and Engineering Faculty, Qingdao University, P.R.China "A New Resource Discovery Mechanism with Negotiate Solution Based on Agent in Grid Environments" The 3rd International Conference on Grid and Pervasive Computing – Workshops 978-0-7695-3177-9/08 $25.00 © 2008 IEEE.

[10]. Miguel L. Bote-Lorenzo, Yannis A. Dimitriadis, and Eduardo G´omez-S´anchez ?School of Telecommunications Engineering, University of Valladolid Camino Viejo del Cementerio s/n, 47011 Valladolid, Spain fmigbot, yannis, edugomg@tel.uva.es

[11]. A Security Architecture for Computational Grids* Ian Foster* Carl Kessekan2 Gene Tsudik2 Steven Tueckel jth Conference on Computer& Communications Security SanFranciscoCA USA such as those discussed in Chapters 2-5 of [4]. Cop fight ACM 199S.

[12]. A. Sulistio, U. Cibej, S. Venugopal, B. Robic and R. Buyya, "A toolkit for modelling and simulating data grids:

An extension to Gridsim," Concurrency and computation: practice and experience, accepted for publication on Dec.3, 2007.

[13]. A. Sulistio, G. Poduval, R. Buyya and C.-K.Tham, "On incorporating differentiated levels of network service into Gridsim," Future generation computer systems, 23(4):606-615, May2007

[14]. http://www.princeton.edu/~jdwoskin/grid/gsioverview.html

[15]. http://www.geocities.com/echelongrid/

[16]. http://searchdatacenter.techtarget.com/sDefinition/0,,sid80_gci773157,00.html

[17]. http://en.wikipedia.org/wiki/Mobile_agent

