Jestr Journal of Engineering Science and Technology Review 7 (4) (2014) 109- 113

JOURNAL OF
Engineering Science and Technology Review

www.jestr.org

Efficient Method for Parallel Process and Matching of Large Data set in Grid Computing Environment

E. Sankar* and K. Ashokkumar

Computer Science and Engineering,Sathyabama University,Chennai,India.

Received 11 March 2014; Accepted 25 September 2014

Abstract

Data management is one of the challenging issues in grid computing and its environments. Because grid computing systems and its applications deals with huge amount of data sets, due to the heterogeneous grid resources that belongs to different organizations and various locations with many access policies. Here To achieve the promising potentials of tremendous distributed resources, useful and capable Scheduling Algorithms are important. Task Scheduling is the mapping of tasks to a selected group of resources which may be distributed in different administrative domains. In this the Parallel Processing of the distributed systems will works using the grid scheduling algorithms. Genetic Algorithm which is a type of scheduling algorithm used for task scheduling to the various resources are working as parallel in the distributed systems.Basically, a Grid scheduler receives applications from Grid users, selects sufficient resources for these applications according to acquired information from the Grid Information Service module, and in conclusion generates application to resource mappings based on assured objective functions and predicted resource performance. Information about the status of available resources is very important for a Grid scheduler to make a proper scheduling, particularly when the heterogeneous and self motivated nature of the Grid is taken into account .The function of the Grid information service is to provide such information to Grid schedulers.

Keywords: grid computing, parallel processing, task scheduling, heterogeneous resources.

1. Introduction

1.1 Grid Computing:

It is a collection of computer resources from various and multiple location to reach a common destination .The grid can be thinking of as a distributed system's with non-interactive workloads that involve a large number of files. What distinguish grid computing from usual high performance computing systems like cluster computing is that grids tend to be more loosely coupled, various, and in nature dispersed. Even a single grid can be committed to a exacting application, usually a grid is used for a selection of purpose. Grids are often constructed with general-purpose grid middleware software databases.

Grid size varies a substantial amount. Grids are the form of distributed computing where a Super Virtual Computer is composed of different networked loosely coupled computers acting together to perform many tasks. For certain applications, Distributed or grid computing, which will be seen as a particular type of parallel computing that relies on complete computers connected to a network by a conventional network interface, such as Ethernet .This an difference to the usual concept of a supercomputers, which has many processors connected by a local high-speed computer bus.

Grid computing allows the use and alignment of computer and data resources to solve compound arithmetical problems. This is the latest development in an evolution that earlier brought onward such advances as distributed computing, the worldwide web, and mutual computing.

We need one supervised system with enough memory and well computation power, but in this world no single system has been such kind of things as like our expectation. So we need to combine together few systems to make one cluster system. This cluster system can run the application with more size. But our problem is we are having all data set in different place. Cluster formation can happen with few systems only. And also we need more processing power and computation power. We can't expect such things from cluster system.

To avoid this problem we go for grid system. We may call this as group of cluster can form a grid system. In this grid system we have multiple clusters in each and every data set. So, implementation in the form of searching, matching everything is possible in data set. Because each dataset has their cluster system, so execution is effortless.

Grid Architecture

Efficient access to and movement of huge quantities of data is required in more and more fields of science and skill. With this data sharing is vital, for example enable access to information stored in databases that are managed and administered in parallel. In trade areas, archiving of data and data management are essential requirements.

* E-mail address: Sankar.cse28@gmail.com
ISSN: 1791-2377

Data grid -- provides the data management features to enable data access, synchronization, and distribution of a grid.

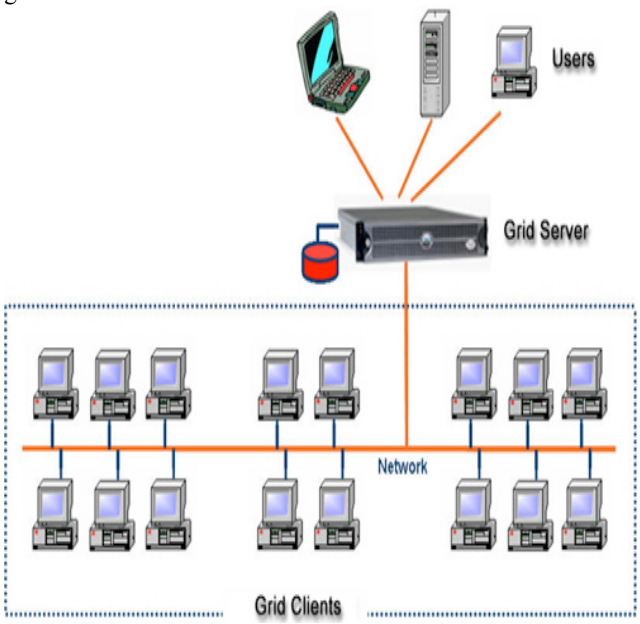

Fig. 1. Grid Computing

1.2 Data Management:

Data Services. A grid necessarily consists of two distinct parts, compute and data.

Compute grid provides the core resource and task management services for grid computing: sharing, management, and distribution of tasks based on configurable service-level policies.

2. Existing information recovery methods in Grid Computing:

Information Retrieval & Search Algorithms
This will provide the following characteristics:

Accuracy: The total amount of information now stored in electronic media shows deficiency in recall .the percentage of significant information retrieved. Giving the user reasonable-sized response with high accuracy can mean lost hundreds of significant texts.

Speed: As the amount of text that must be searched increases, the speed of searching can become a store blockage. In practical terms, the need for fast search means that more computational-intensive processing such as NLP techniques must either apply very selectively or run as batch indexing tool prior to retrieval.

Consistency: Many information retrieval environments require indexing of the text by the groups of indexers or by the authors. This leads to a decrease in accurateness from the inevitable inconsistencies, which automatic processing could help to avoid.

Ease of use: The growth of individual computer has made outdated the usual model of information rescue with a trained human agent, giving systems receptiveness of high speed.

3. Existing System

In the existing system a Grid will act as a set of sites where each comprising a number of processors and a restricted amount of storage and a set of users which associated with a site and a set of files, each of a specified size, initially mapped to sites according to some distribution. It assumes that all processors have the same performance and that all processors at a site can access any storage at that site. Each user generates jobs according to some allocation. Each job requires that a specified set of files be available before it can execute. It then executes for a specific amount of time on a single processor, and finally generates a specific set of files which may leads huge processing and parallel searching cannot be done.

4. Proposed System

The proposed system provides a secured and optical replication of task scheduling in grid computing process is shown. Basically, a scheduler receives applications from Grid users, selects reasonable resources for these applications according to acquired information from the Grid Information Service module, and finally generates application-to-resource mappings, based on certain objective functions and predicted resource performance. Our system is developed for efficient and optimal replication of task scheduling in grid computing.

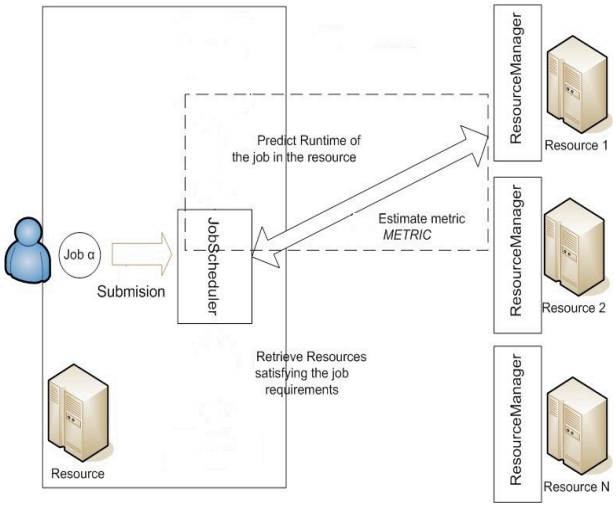

Fig. 2. Grid Computing Parallel task Schedule

The system is built upon the related work on Grid security, trust management, and job scheduling. The system matches trust requirements by user jobs with a judicious security index at Grid sites, which extends security-aware Grid job scheduling in the direction of delay tolerance and job replications. Parallel communication of SD and TL values, trusted execution of replicated jobs, trusted propagation of job results after various delayed executions. Execution of replicated jobs and propagation of delayed job results can be implemented by having the Grid sites digitally sign the data and code so that the recipients can verify the data integrity and authenticate all parties involved.

Parallel Computing:

Parallel computing is the concurrent use of multiple processors to do computational works.

In traditional programming, a single processor executes program information in a step-by-step behavior. Some operations, may have multiple steps that do not have time dependency and therefore it can be separated into multiple tasks which to be executed concurrently. For example, adding a number to all the elements of a matrix does not require the result obtained from summing one element be acquire before summing the next element. Elements in the matrix can be made available to numerous processors, and the sums performed concurrently, with the results available faster than if all operations had been performed consecutively.

Parallel computations can be performed on shared-memory systems with multiple processors, distributed memory clusters made up of smaller shared memory systems, or single processor systems. Coordinating the simultaneous work of the multiple processors and synchronizing the results are taken care by program calls to parallel libraries.

Grid Scheduling

Schedulers are responsible for the management of jobs, such as allocating Resources required for any particular job, partitioning of jobs to schedule parallel execution of tasks ,data management ,event association, and service-level management capabilities. These schedulers then form a hierarchical structure, with meta-schedulers that form the root and other lower level schedulers, while providing specific scheduling capabilities that form the leaves. The objective of scheduling is to minimize the completion time of a parallel application by properly allocating the tasks to the processors. In a wide sense, the scheduling problem exists in two forms: *static* and *dynamic*. In static scheduling, the characteristics of a parallel program (task processing times, communication, data dependencies, etc.) are known prior to program execution. In dynamic scheduling algorithms, the goal includes not only the minimization of the program completion time but also the minimization of the scheduling overhead which constitutes a significant portion of the cost paid for running the scheduler .We have experimented with four algorithms – Hill climbing algorithm, simulated annealing, Taboo search and Genetic algorithm. The criteria used for evaluating the performance were the time for schedule preparing and execution time.

Working and Proposed Algorithm:

Genetic algorithm is a search technique used in computing to find exact or approximate solutions to optimization and search problems. GA algorithms are categorized as global search heuristics.

1. Initialization: convert random generated schedule S to vector V
2. (of binary values), add it to the input population Pin

- Set V best=V, best Cost=F(V best)
- Generate next X binary vectors (schedules) and add them to Pin

3. Copy Pin to the new population P new
4. Make Y mutations on random members of Pin and new members add to P new
5. Make Z crossovers using 2 members of the population Pin; add new offspring to P new
6. Select X+1 members of P new with min. Cost and create new population Pin with them.
7. Find member of P new with min. objective function; if Cost < best Cost, update V best
8. If stop condition is fulfilled - terminate; else go to step 2.

Typical Structure of Genetic Algorithm is follows

GeneticAlgorithm for Parallel Process and Match:

```
Evolutionary algorithms
 {
                        initialize the resources;
                  calculate the initial resources;
          while (termination criterion not reached)
                                                  {
                      select solutions for next;
   resources;
                perform crossover and alteration;
                                evaluate resources;
                                                  }
 }
```

Genetic Algorithm is the particular class of evolutionary algorithms also known as evolutionary

Computation that use techniques inspired by evolutionary biology such as inheritance, mutation selection, and crossover. The most common type of Genetic Algorithm works like this a population is created with a group of individuals created at random. The individuals in the population are then evaluated. The evaluation function is provided by the programmer and gives the individuals a score based on how well they perform at the given task. Two individuals are then chosen based on their fitness, the higher the fitness, higher the chance of being selected. These individuals then replicate to create one or more offspring, after which the offspring are mutated at random. This continues until a suitable solution has been found or a certain number of generations have passed, depending on the requirements of the solution.

Modules Brief Description:

User Interface:

The client will get authorized and authenticated in order to communicate with the Trust layer Agent. The user window is designed in which the user feeds inputs in order to communicate to the Grid Server. The inputs that the user specifies are the Server Id, the user name and password. The entered data is given for the registration process. Once the user is registered, the user can choose a dataset that can be given as the input data which will be submitted to the server for processing.

Resource Allocation:

This process is initiated by a Trust Layer Agent. The user requests and submits the job to the Trust Layer Agent. Allocation of resources is based on Computational capacity, Time constraints and Cost limits. The tasks are allotted for the resources based on their availability. The tasks are created for the input data that is submitted by the user i.e., the data set. After the user selects the input data and clicks on the submit button, the data will be forwarded to this window for the resource allocation process.

Parallel Resources:

The trusted resources are the resources available in the TLA in order to process the job requested by the user into number of tasks. The Grid Information System has multiple Grids that each consists multiple resources. The processing of the job will be according to these resources.

Job Submission:
When the resource matches the parameters of a particular resource, then that resource is accepted. If the resource does not match the user's satisfaction, then that resource is rejected. The status of the resources will be identified and the task will be allocated to that particular resource. In case the grid is overloaded, the task will be shared by the neighbor who is idle to receive and process that task.

Performance Evaluation:
The response time and the success rate of execution jobs is calculated in this module.

Working model and SIMULATION:
Once we click on the user node it takes to the login and authentication screen and gets the login name and the password from the user press submit button which takes to the user node where the user gets registered.

Results and Efficiency:
After the user got registered provide the input job which takes to the location of our input which is a biological data. After getting the input data press submit which takes to the grid information system where grid scheduler with the registered user and selected input data will be present.

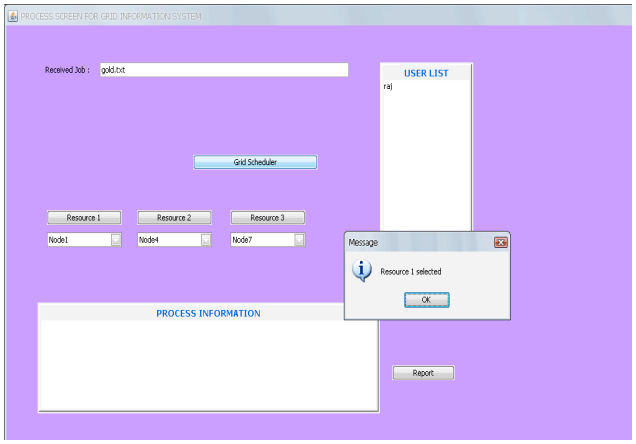

Fig. 3. Grid infrastructure with selected sample data.

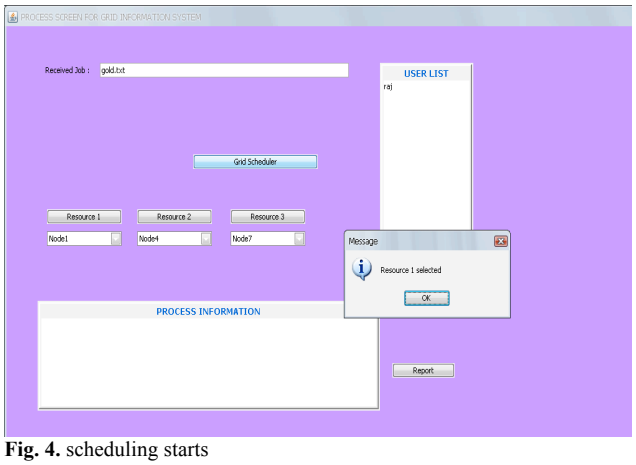

Fig. 4. scheduling starts

In this the grid scheduler will have the distributed resources each has a different nodes for which the grid scheduler allocates the task parallel. After the task got allocated simultaneously the resource got selected and the processing starts for that particular resource.

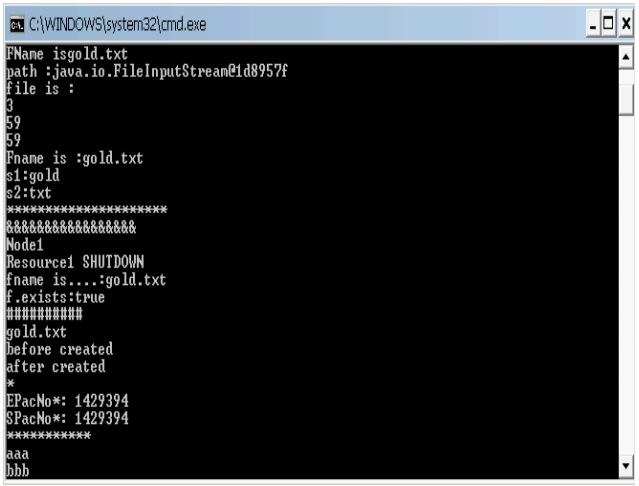

Fig. 5. Result after the task allocation

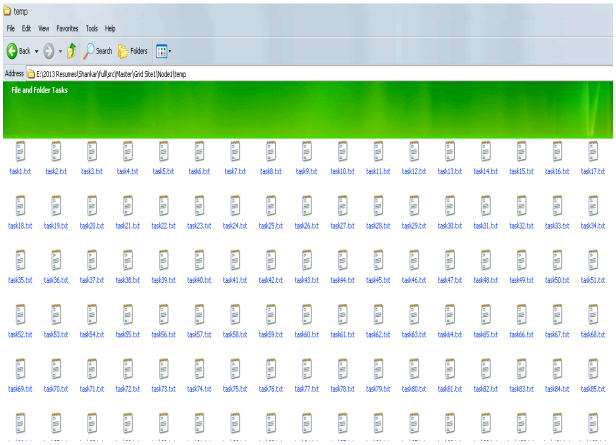

Fig. 6. The selected tasks will be added to the selected location.

Conclusion

In this proposed method we have executed the large data set from multiple resources which produced the result within the estimated time. In this integrating data sets of the data sets are effective and efficient and those different data sets are computed by the grid infrastructure in parallel manner.
Here we assumed to design and develop the framework of this model for grid computing resources as the tool. This framework will included with grid computing infrastructure to increase heterogeneous data manipulation of grid databases and this will do fast and parallel computation effectively and efficiently in the grid resources.

References

[1]. Ian Foster, Jens Vo¨ckler, Michael Wilde, and Yong Zhao, "The Virtual Data Grid: A New Model and Architecture for Data-Intensive Collaboration," Proceedings of the 2003 CIDR Conference.

[2]. http://lbsitbytes2010.wordpress.com/2013/03/20/cluster-computing-2/

[3]. http://www.wolfram.com/gridmathematica/ integrated extension system for increasing the power of your *Mathematica*

[4]. K. Ashok Kumar, M.E, C. Chandra Sekar, PH.D Proceedings of the 2013 CCIIS IEEE Conference.

[5]. F. Guim, I. Rodero, M. Garcia, J. Corbalán Barcelona The XtreemOS JScheduler: Using Self-Scheduling Techniques in Large Computing Architectures Supercomputing Center {francesc.guim, irodero, marta.garcia, julita.corbalan@bsc.es./

[6]. ALiCE Grid Computing Project Department of Computer ScienceNational University of Singapore 3 Science Drive 2, Singapore117543 teoym@comp.nus.edu.sg www.comp.nus.edu.sg/~teoym/alice.htm