International Journal of Computer Network and Security(IJCNS)
Vol 4. No 1. Jan-Mar 2012 ISSN: 0975-8283
www.ijcns.com

Survey on Scheduling Methods in P2P Desktop Grid

T.Avudaiappan[1], B.S.Balamurugan[2], L.srinivasan[3] ,C.Pradeep[4]
[1,2,3]PG Scholar, Department of Computer Science and Engineering, Karpagam University, India
[4]PG Scholar, Department of Computer Science and Engineering, Karunya University, India
[1]avudaiappanmecse@gmail.com, [2]balabeme@gmail.com, [3]srini.jeg@gmail.com,[4]pradeeprsc1@gmail.com

Abstract

Grid computing is a next level of infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities with variety of shared resources. In this high end computing infrastructure job scheduling is a complex one due to grid infrastructures. This proposal gives a detail analysis of various scheduling algorithm based on different criteria such as dynamic environment, task orientation, adaptability etc. In this analysis different parameters are taken based on the classification of the algorithms environment.

Keywords **-** Grid computing, scheduling, **agent based, heuristic based and task independent scheduling model**

1. INTRODUCTION

Many Grid applications require the coordinated processing of complex workflows which includes scheduling of heterogeneous resources within different administrative domains. A typical scenario is the coordinated scheduling of computational resources in conjunction with data, storage, network and other available Grid resources, like software licenses, experimental devices, etc. The Grid scheduler should be able to coordinate and plan the workflow execution. It should reserve the required resources and create a complete schedule for the whole workflow in advance.

Grid schedulers usually cannot control Grid resources directly, but work like brokers or agents, or are even tightly coupled with the applications as the application-level scheduling scheme.

Computational grid a type of parallel and distributed system, that enables the sharing, selection, and aggregation of geographically distributed autonomous resources connected through internet depending on their availability, capability, performance, cost, and users quality-of-service requirements. The purpose of job scheduling in this environment is to balance the entire system load while completing all the jobs as soon as possible according to the environment status. Scheduling can occur prior to execution, assuming a fixed state of the grid engine (static scheduling), or can be done during execution (dynamic scheduling) as described in fig.1. The main drawback of dynamic scheduling is the overhead occurring due to determining the schedule while the program is running. And the major benefit of dynamic scheduling over static scheduling is that the system need not be aware of the run-time behavior of the application before execution.

The rest of the paper is structured as follows: related works in section 2 which describes the different types of scheduling model. Then in section 3 an elaborated analysis is given. And section 4 ends the paper with the conclusion.

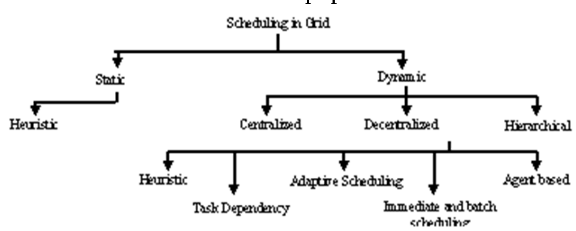

Figure. 1. Classification of grid

2. RELATED WORKS

We can also generalize a scheduling process in the Grid into three stages: resource discovering and filtering, selecting the resource and scheduling according to certain objectives, and job submission criteria. A Grid scheduler receives applications from Grid users, selects feasible resources for these applications according to the acquired information from the Grid Information Service (GIS) module and finally generates application-to-resource mappings, based on certain objective functions and predicted resource performance.

A. Prediction Based Replication Strategies [8]

In this methodology for predicting machine availability from monitored data in distributed computing environment is discussed. Volunteer-based grid computing resources are characteristically volatile and frequently become unavailable due to the autonomy that owners maintain over them. This resource volatility has significant influence on the applications the resources host. Availability predictors can forecast unavailability, and can provide schedulers with information about reliability, which helps them to make better scheduling decisions when combined with information about speed and load. The basis for this idea is that checkpointing and replication are two of the primary tools for dealing with resource unavailability. Schedulers can replicate jobs to reduce the effect of failure and ultimately reduce makespan. However, replication requires additional grid cycles, which can have direct cost within a grid economy, and an indirect cost in tying up attractive resources especially under relatively higher loads. Availability predictors can forecast resource behavior, and allow schedulers to consider resource reliability when making job replication decisions.

B. Peer-to-peer systems for discovering resources in a dynamic grid [12]

This approach has two P2P systems which support highly dynamic data. Both are based on Routing Indexes, which are used to efficiently route queries and update messages in the presence of highly variable data. The first system uses a tree-shaped overlay network. The second one is an evolution of the first, and is based on a two-level hierarchical network topology, where tree topologies must only be maintained at the lower level of the hierarchy, i.e., within the various node groups making up the network. The main goal of the second organization is to achieve a simpler maintenance of the overall P2P graph topology, by preserving the good properties of the tree-shaped topology.

C. Grid scheduling use cases [11]

The goal of the Grid Scheduling Architecture Research Group (GSA-RG [1]) is to define a scheduling architecture that supports the cooperation between different scheduling instances for arbitrary Grid resources, including network, software, data, storage, and processing units. The research group will particularly address the interaction between resource management and data management. Co-allocation and the reservation of resources are key aspects of the new scheduling architecture, which will also integrate user or provider-defined scheduling policies. The group started with identifying a set of relevant use cases based on experiences obtained by existing Grid projects. This information will be used to identify components, services and protocols for a Grid scheduling architecture and outline their interaction. Services and protocols from other GGF groups are considered as potential basic building blocks of such architecture and will be used wherever possible.

D. Matching algorithms [6]

The goal of matching algorithms is to design and build a scalable infrastructure for executing grid applications on a widely distributed set of resources. Such infrastructure must be *decentralized, robust, highly available*, and *scalable*, while efficiently *mapping* application instances to available resources throughout the system (called *matchmaking*). Fortunately, these are precisely the characteristics promised by new techniques and approaches in Peer-to-Peer (P2P) systems. Using P2P services can provide a robust, reliable, and scalable job submission and execution system that is able to efficiently utilize widely distributed available computational resources. Such a confluence of P2P and distributed computing is a natural step in the progression of grid computing, and has indeed been described as inevitable.[14, 15].

Applications of our proposed system have both high end Computational requirements and relatively low I/O requirements.

E. Group based adaptive scheduling mechanism [10]

This mechanism provides comprehensive taxonomy and survey of scheduling for Desktop Grid in order to better design and develop a new scheduling mechanism.

Particularly, they did not consider volunteer's properties such as volatility, availability, and credibility that strongly affect reliability and performance. Moreover, they did not provide scheduling mechanisms on a per group basis. In other words, they did not apply different scheduling algorithms to each group according to volunteer's properties.

As a result, they deteriorate the reliability of computation as well as performance. To solve these problems, this group scheduling mechanism proposes a new group-based adaptive scheduling mechanism, which adapts to a dynamic Desktop Grid computing environment.

Table 1
Comparison of grid scheduling algorithms

Parameter / Approach	Makespan	Manageable	Scalable	Adaptable	Load balancing
Peer-to-peer[12]	Good	Difficult	Yes	Better	Good
Matching algorithm[6]	High	Easy	Limited	Yes	Good
Resource availability[9]	Average	Limited	Yes	Can provide	Better
Prediction based[8]	Reduced	Easy	Better	Yes	Good
Scheduling use cases[11]	Low	Good	Yes	Difficult	Low
Group based prediction[10]	Very low	Better	Yes	Easy	Very good

The group-based adaptive scheduling mechanism classifies and constructs groups according to volunteer's properties such as dedication, volatility, availability, and credibility. Then it applies different scheduling, replication, result certification, and fault tolerance algorithms to each group. Consequently, it improves reliability and performance. The simulation results show that how it can outperform existing scheduling mechanisms.

F. Characterizing resource availability in enterprise desktop grid [9]

Despite the popularity and success of many desktop grid projects, the volatility of the hosts within desktop grids has been poorly understood. Yes, this characterization is essential for accurate simulation and modeling of such platforms. In an effort to characterize these types of systems, availability traces were gathered from several real enterprise desktop grids, and these traces are used for conducting a study of their characterization.

In particular, the three main contributions of this method are as follows. First, the application-level traces of four real enterprise desktop grids were obtained. Second, the aggregate and per-host statistics of hosts found in the desktop grid traces were determined. Third, the characterization results were applied to develop a performance model for desktop grid applications with various task granularities, showing that there is an optimal task size, and then cluster equivalence metric is used to quantify the utility of the desktop grid relative to that of a dedicated cluster for a parallel application.

3. ANALYSIS

Among the agent based scheduling, by using the prediction scheduling mechanism, which achieves a comparable good performance with much lower computational costs and limited communication. In a desktop grid model, the job (computational task) is submitted for execution in the resource only when the resource is idle. There is no guarantee that the job which has started to execute in a resource will complete its execution without any disruption from user activity (such as a keyboard stroke or mouse move) if the desktop machines are used for other purposes.

The selection of resources for job execution is dependent on the reliability rankings. Resource availability prediction using mathematical models were focused on predicting the availability of the client with certain confidence based on its past availability. Resource Availability patterns in various environments have been exhaustively studied by Nurmi et al. [16] and Rood and Lewis [17].

We evaluate the group-based adaptive scheduling mechanism with existing scheduling mechanisms with various parameters as given in Table 1. The group-based adaptive scheduling algorithm is based on virtual formation of idle machines. The evaluation focuses on how much performance improvement can be achieved, depending on whether volunteer groups are considered in a scheduling procedure. In other scheduling [3, 8, 9, 11], a volunteer asks its volunteer server of a new task as soon as it finishes its current task. There are a lot of scheduling heuristics in grid computing environments, for examples, MCT, MET, SA, KPB, min-min and max-min [5, 8].

This survey focuses on characterizing and categorizing the aspects of Desktop Grid systems. To this end, we proposed the Desktop Grid focusing on group based scheduling. From this survey, we extracted the challenging issues for Desktop Grid scheduling such as volatility, dynamic environment, lack of trust, failure, heterogeneity, scalability, and voluntary participation. To overcome these challenges, we proposed a new direction for Desktop Grid scheduling such as

- Resource grouping
- Reputation or incentive-based scheduling
- Scheduling for result certification
- Dynamic, adaptive or fault tolerant scheduling
- Distributed scheduling

Among these directions, this thesis focused on the resource grouping, result certification, replication, adaptive and fault tolerant scheduling.

For the purpose, a group-based adaptive scheduling mechanism in a Desktop Grid computing considers the volatility, credibility, and heterogeneous properties of volunteers in a scheduling procedure in the sense that they are tightly related with the computation completion, reliability, and performance. The scheduling mechanism constructs volunteer groups according to the properties of volunteers such as volunteer autonomy failures, volunteer availability, volunteering service time, and volunteer credibility. It dynamically applies different scheduling, replication, result certification, fault tolerance algorithms to each volunteer group. Particularly, the group based predictive scheduling mechanism completes more tasks than eager scheduling. In the case of replication, this scheduling mechanism completes more tasks than the eager scheduling because it dynamically adjusts the number of redundancy. It proposes a resource grouping method, which classifies and constructs groups according to volunteer's properties such as dedication, volatility, availability, and credibility.

This problem becomes more challenging in a Peer-to-Peer (P2P) model for a desktop grid where there is no central server that decides to allocate a job to a particular resource. This paper describes a P2P desktop grid framework that utilizes resource availability prediction, using group availability data. We improve the functionality of the system by submitting the jobs on machines that have a higher probability of being available at a given time.

4. CONCLUSION

In this paper, issues of grid scheduling have been discussed. Major three classifications of scheduling models like availability-based, prediction-based and group-based scheduling are taken for analysis and parameters also identified. These parameters are tabulated based on performance of the scheduling model. In table 1 there are different scheduling algorithms are examined and results are classified based on different parameters like Makespan, load-balancing and adaptable. Here we have identified performance issues related to each classification of scheduling model.

The group based scheduling approaches show better performance in these scheduling models.

REFERENCES

[1] M. Litzkow, M. Livny, M. Mutka, Condor-a hunter of idle workstations, in: "*8th International Conference on Distribute Computing Systems*", 1988, pp. 104–111.

[2] G. Fedak, C. Germain, V. Neri, F. Cappello, Xtremweb: a generic global computing system, in: "*Proceedings of the IEEE International Symposium on Cluster Computing and the Grid*", 2001, pp. 582–587.

[3] A. Chien, B. Calder, S. Elbert, K. Bhatia, Entropia: architecture and performance of an enterprise desktop grid system, "*Journal of Parallel and Distributed Computing*" 63 [5] (2003) 597–610.

[4] D. Anderson, Boinc: a system for public-resource computing and storage, in: "*5th IEEE/ACM International Workshop on Grid Computing*", 2004, pp. 365–372.

[5] J. Kim, B. Nam, M. Marsh, P. Keleher, B. Bhattacharjee, D. Richardson, D. Wellnitz, A. Sussman, Creating a robust desktop grid using peer-topeer services, in: "*IEEE International on Parallel and Distributed Processing*" Symposium. IPDPS 2007, 2007, pp. 1–7.

[6] D. Kondo, B. Javadi, F. Cappello, A.D. Anderson, Cost-benefit analysis of cloud computing versus desktop grids, in: "*18th International Heterogeneity in Computing Workshop*".

[7] J. Smith, R. Nair, The architecture of virtualbel machines, Computer (2005),pp. 32–38.

[8] B. Rood, M. Lewis, Resource availability prediction for improved grid scheduling, in: "*Fourth IEEE International Conference on eScienc*", 2008.

[9] D. Kondo, G. Fedak, F. Cappello, A. Chien, H. Casanova, Characterizing resource availability in enterprise desktop grids, Future Generation Computer Systems 23 (7) (2007) 888–903.----7

[10] SungJin Choi, "Group-based Adaptive Scheduling Mechanism in Desktop Grid, The Degree of Doctor of Philosophy, June 2007--- 6

[11] Philipp Wieder, "Grid Scheduling Use Cases",*"Grid Scheduling Architecture Research Group (GSA-RG)*", , March 26, 2006 ---**4**

[12] Moreno Marzolla b, Matteo Mordacchini a, Salvatore Orlando a, Peer-to-peer systems for discovering resources in a dynamic grid.

[13] A. S. Cheema, M. Muhammad, and I. Gupta. Peer-to-peer Discovery of Computational Resources for Grid Applications. In *Proceedings of the 6th IEEE/ACM InternationalConference on Grid Computing - GRID 2005*, Nov. 2005.

[14] I. Foster and A. Iamnitchi. On Death, Taxes, and the Convergence of Peer-to-Peer and Grid Computing. In *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems*
(IPTPS '03). Springer-Verlag, Feb. 2003

[15] M. O. Neary, B. O. Christiansen, P. Cappello, and K. Schauser, "Javelin: Parallel computing on the internet," *Future Generation Computer Systems, Special Issue on Metacomputing*, vol. 15, issue 5-6, pp. 659-674, Oct. 1999.

[16] M. O. Neary, S. P. Brydon, P. Kmiec, S. Rollins, and P. Cappello, "Javelin++: Scalability Issues in Global Computing," *Concurrency: Parctice and Experience*, vol. 12, issue 8, pp. 727-753, Aug. 2000.