AASRI Conference on Computational Intelligence and Bioinformatics, CIB 2012

Agent-based Robust Grid Scheduling Framework for High Performance Computing

Syed Nasir Mehmood Shah∗, Nazleeni Haron, M Nordin B Zakaria, Ahmad Kamil Bin Mahmood

Department of Computer and Information Sciences
Universiti Teknologi PETRONAS, Bandar Seri Iskandar, 31750 Tronoh, Perak, Malaysia

Abstract

Grid computing is the enabling technology for high performance computing in scientific and large scale computing applications. A number of powerful parallel systems have been built with increasing number of processors and multi-core solutions in order to meet the increased computational demands. Scheduling is a fundamental issue in achieving high performance on computational Grids. A Grid scheduling system must meet the functional requirements of heterogeneous domains (e.g., user domains, application domains, and network domains). Sometimes these requirements conflict with one another. Moreover, a Grid scheduling system must also satisfy non-functional requirements, such as reliability, efficiency (in terms of time consumption), effectiveness in resource utilization, and robustness.

Many frameworks have been proposed for Grid scheduling system. Agents are able to behave autonomously, intelligently, learn from environment, and adaptability. Agent-based Grid scheduling is concerned with the design and development of software agents for the resource discovery, job scheduling, resource management, and allocation of computing resources on a computational Grid. This paper presents a new Agent based Grid scheduling framework which possesses high degree of robustness, performance, efficiency, scalability and heterogeneity.

* Corresponding author. Tel.: +60-125936195
E-mail address: nasirsyed.utp@gmail.com

1. Introduction

'Scheduling' is described by the Grid Scheduling Dictionary Project as follows: "The process of ordering tasks on compute resources and ordering communication between tasks. Also, known as the allocation of computation and communication over time" [1].

'Robustness' is one of several quality factors associated with any software. The robustness of a computational system can be defined as the degree to which a system can function correctly in the presence failures and stressful environmental conditions [2].

Grid scheduling presents several challenges that make the implementation of practical systems a very difficult problem. Our research aims to propose a Grid scheduling framework that makes efficient utilization of resources, possesses a high degree of abstraction in inter-task synchronization, and is robust.

To achieve this, we need to develop a Grid infrastructure which will be flexible, scalable, and adaptive in nature and will enable dynamic composition of loosely coupled heterogeneous resources. One way of implementing these Grid capabilities is to use multi-agent system. Agents have the ability to adapt themselves according to prevailing circumstances in a Grid environment. Agents also provide services that are robust and scalable. Because of these characteristics agents are the natural choice for achieving our goal. In our research; we propose a framework named Agent-based Robust Grid Scheduling Framework for Grid infrastructure. It will satisfy the efficiency, performance, Quality of Service (QoS) provisioning and robustness objectives.

2. Related Research

Grid computing can be viewed as a mainstream technology for large-scale resource sharing [3]. Grid computing increases a system's computing capability and the tendency has been to use them to solve complex and large-scale scientific problems using geographically dispersed computing resources. A large number of complex and large-scale scientific issues cannot be solved by using a traditional network or super computer; that is why research and development of Grid computing has been making progress gradually. A Grid computing system connects available computing resources, such as computers, applications, and storages devices, to networks for high performance computing and reduces system execution time [4, 5].

Grid scheduling is a process of ordering tasks on compute resources and ordering communication between them. It is also known as the allocation of computation and communication over time [6]. Grid scheduling can be divided into two phases. The first stage is the selection of proper resources for job allocation, and the second stage is the placement of jobs onto the selected resource queue for execution [7].

A multi-agent system is a network of agents that work collectively to solve problems that are beyond their individual capabilities [8]. Multi-agent systems are distributed and autonomous computing systems made up of autonomous agents that support reactivity, and are robust against failures locally as well as globally [9]. Due to the highly heterogeneous, distributed, dynamic, and complex Grid computing environments, multi-agent systems appear to be a suitable approach to solve the Grid scheduling problem.

A number of initiatives have been taken to improve the efficiency of computational Grids but this research is still in early development. [10] proposes a resource management system for scheduling jobs on heterogeneous and distributed resources (Nimrod-G). In Nimrod-G, resource agents manage the execution of jobs on resources, and a central resource broker performs the different functions like resource discovery, trading and discovery.

In [11, 12], authors have proposed an agent-based resource management system (ARMS) for Grid computing. In this proposed system; each agent represents a local Grid resource and acts as a service provider for high performance computing power. Agents cooperate and collaborate with each other using a technique of service advertisement and discovery.

In [13] the authors proposed a middleware framework for Grids. This framework is based on an advanced reservation mechanism. It satisfies the user by providing QoS assurance for Grid applications. This is a cost

effective solution for the efficient utilization of resources. The framework is also a robust one because it handles uncertain runtimes of applications intelligently. However, there are shortcomings with this framework. Firstly, it has not been integrated into any Grid toolkit and so no real-time performance tests have been undertaken. Secondly, it does not support dynamic scheduling.

In [14], a novel multi-agent reinforcement learning method named ordinal sharing learning (OSL) method is proposed for job scheduling in Grid environment. This method has been proposed to resolve the issues of load balancing in computational Grids. The approach supports the scalability issue by using an ordinal distributed learning strategy, and realizes multi-agent coordination based on an information-sharing mechanism with limited communication. Simulation results show that the OSL method can attain the goal of load balancing effectively, and its performance is even comparable to some centralized scheduling algorithm in most cases. The convergence property and adaptability of the proposed method are also explained. But inter-task synchronization, dynamic scheduling and robustness features have not been considered in the proposed method.

[15] proposed an agent-based approach to integrate services and resources. In this proposed approach, service and resource agents contain behavioral rules and can modify these rules based on their interaction with other agents and with the environment in which they operate. In [16] authors also proposed an economic models for resource allocation and to manage supplies and demands in Grid computing environments.

Most of these agent-based Grid scheduling systems are centralized. Grid scheduler agents manage only the execution of jobs on resources. In a true Grid environment, resources are geographically distributed and owned by different parties. It is not practical that a single point in the virtual system retains entire Grid information that can be used for job scheduling.

3. Proposed Agent-based Robust Grid Scheduling framework

A preliminary overview for the framework is shown in Fig. 1. A Grid comprises of the following four elements:
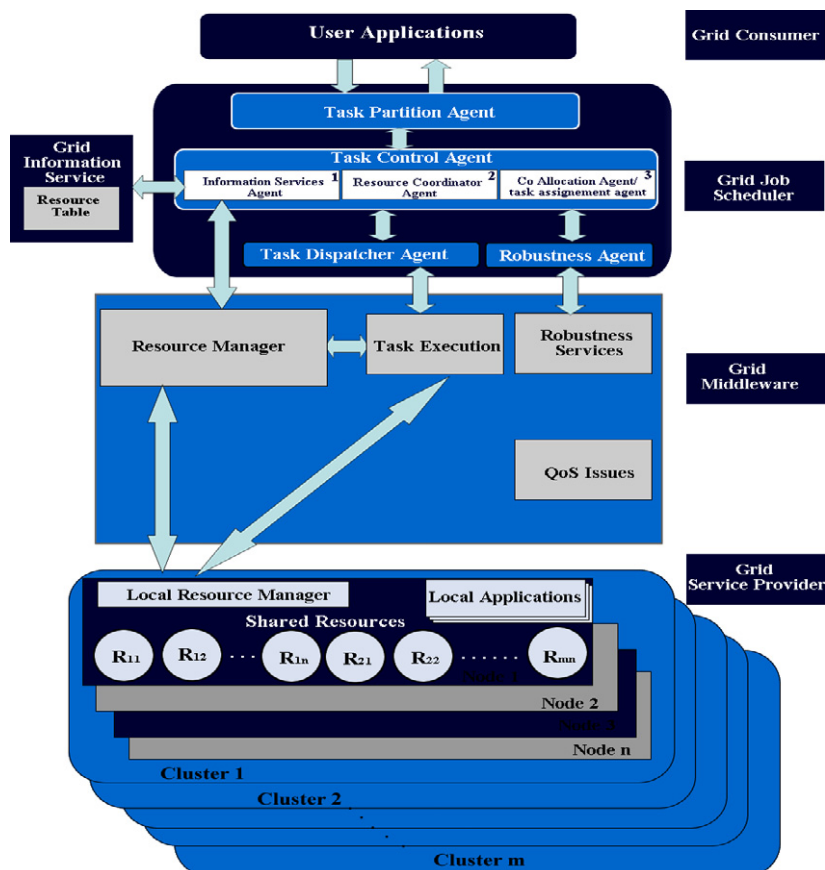
Fig. 1.   Robust Agent-based Grid Scheduling Framework

3.1. Grid consumer:

   Grid consumer is the user of the grids. Each consumer needs computational power for execution of their jobs. Consumers belong to various organizations and access the Grid resources in a decentralized way. Each consumer demands computing resources based on nature of jobs. A user job is submitted to the Grid scheduler in the form of application.

3.2. Grid Job Scheduler:

   Grid job scheduler receives applications from the Grid consumers. Grid job scheduler is an agent oriented scheduler, which is responsible for resource information, feasible resource selection, application to resource mapping and robustness.
   Grid job scheduler consists of four main agents namely Computational task partition agent, Job control agent, Dispatcher agent and Robustness agent.

A.   Computational task partition agent

   This agent partitions user application into several computational tasks depending upon the nature of application, resource demands and resource availabilities. Tasks will be dependent in nature. This agent will be responsible for producing tasks of variable sizes while considering the reliability, robustness and availability of computing resources.

B.   Job control agent

   This agent is responsible for keeping the up-to-date information about the jobs, tasks and computing resources. It shows the current status of each user job and computational resource in the Grid computing environment. Job control agent is a multi-level agent which includes the following agents:

- *Information services agent*

   Information service agent is responsible to coordinate with Grid Information Service (GIS) module. It finds
the resources that can satisfy computational task's requirements by looking at the resource table.

- *Resource coordinator agent*

   Resource coordinator agent is respondible to choose the most suitable resources in such a way that can satisfy the job constraints as well as to optimize the system performance. Grid resource allocation is an NP complete problem. Resource coordinator agent would play the vital role in achieving the optimal performance in the Grid computing environment. Various resource allocation strategies will be considered and implemented in this framework. Different strategies work optimally under specific jobs, workloads and resource constraints. Resource coorinator agent will analyze the various scheduling constraints, and then selects one strategy out of many strategies to attain the optimal performance under dynamic scheduling environment.

- *Co-allocation agent*

Co-allocation agent is responsible for mapping of computational task to selected resource(s). It keeps information of user jobs status at any given instant. It also helps in monitoring the deadlock in the system.

C. Robustness Agent

Robustness agent ensures the reliability of job execution. Robustness agent will coordinate with our developed robustness services to ensure the correctness, reliability and completeness of computational task execution. Few robustness parameters such as tasks duration, communication delays will be gathered by the robustness agents to ensure any uncertainties will not affect the task executin.

D. Dispatcher Agent

This agent coordinates with Job control agent and sends the user jobs to Grid resource manager. This agent is responsible for synchrinized, secured and reliable communications between job control agent and Grid resource manager.

3.3. Core Grid middleware:

This layer comprises the core Grid management related services. We will implement our agent oriented robustness services in this layer. These services will be comprised of the agents. Agents will monitor the running of the individual tasks and status of Grid resources. Agents will predict any faults during task execution on a Grid. These services will also influence scheduling the computational tasks in Grid by directly interacting with Robustness agent.

3.4. Grid service providers:

These are the basic service provider (nodes/clusters) in the Grid where computational tasks are to be allocated for execution. A service provider can be an individual, company or university etc.

4. Discussion

This framework is proposed with the main aim to provide a robust Grid scheduler that can deal with uncertainties in a Grid computing environment. The uncertainties are normally caused by fluctuations in availability of resources, network bandwidth and job demands. Reducing and minimizing uncertainties can contribute to more efficient and effective job computation since jobs are able to be computed and delivered despite the constant changes of Grid environment (due to heterogeneity of resources, jobs and networks).

Agent-based paradigm has been chosen as the suitable approach since agents are autonomous that they are capable of making decisions based on their knowledge about the environment and/or other agents. This feature is crucial and appropriate for a highly fluctuated Grid environment. Furthermore, this agent-based framework is applying both hierarchical multi-layer and hierarchical architecture [17]. Hierarchical multi-layer architecture represents a master-slave relationship among the agents and has the advantages of limited computational complexity and very scalable since the agents need to perform the functionality within the scope of their work only. The hierarchical architecture is depicted in Grid Job Scheduler fig. 1 where there are three main types of agents interrelated in a hierarchical manner. The master of the agents are Task Partition Agent who receives input data from Task Control Agent while both Task Dispatcher Agent and Robustness Agent will need to present data to Task Control Agent to be passed and evaluated by the Task Partition Agent. As depicted, the scope of work of each agent is very limited and thus this contributes to a scalable Grid scheduling environment. The hierarchical architecture is deployed specifically in the Task Control Agent in

which three agents working as peers coordinating among them to provide input for Task Control Agent. The advantage of such architecture is that if any of the agents fail to deliver specific results, other agents can still continue to provide input. This is one of the essential features for a robust scheduler.

5. Conclusion and Future Work

In this paper, the architecture of an agent-based scheduling framework of the Grid computing is proposed. This framework is based on a hierarchical multi-layer and hierarchical agent-based architecture and is designed to offer a robustness feature to a Grid scheduler. A robust Grid scheduler will compute and deliver jobs effectively and efficiently despite the constant changeability of Grid computational environment such as heterogeneity of jobs, resources and network conditions.

The success of this framework will depend on how well these agents can cope/work in a high level uncertain Grid environment. As such our future work includes finding the best type of agents that are suitable for each functionality such as finding resource capacities and resource failures. In addition, prototype will be developed and simulated to experiment with scheduling algorithms on a Grid environment comprised of one SGI super computer and five other clusters. Consequently, the performance of the proposed Grid scheduling framework in terms of flowtime, computational cost, average waiting time, average response time, average turnaround time and efficient Grid resource utilizations will be measured through extensive simulations using real workload traces which are available on [18], taken from leading computational centers. The purpose of the simulator would be to evaluate the efficiency, performance, scalability of proposed Grid scheduling framework from various perspectives.

References

[1] Grid Scheduling Dictionary Project, Retrieved Nov. 15, 2010, from http://www.mcs.anl.gov/~schopf/ggf-sched/GGF5/sched-Dict.1.pdf
[2] Ali S, Maciejewski AA, Siegel HJ, Kim J-K , "Measuring the robustness of a resource allocation". Trans Parallel Distrib System, 2004.
[3] Foster, I., Kesselman, C. and Tuecke, S. "The anatomy of the Grid: Enabling scalable virtual organizations", International J. Supercomputer Applications, Vol.15, No. 3., 2001.
[4] Foster, I. and Kesselman, C. (2003) "The Grid 2: Blueprint for a New Computing Infrastructure", 2nd edition, Morgan-Kaufmann Publishers, San Francisco, CA, 2003.
[5] Jang, S. H. and Lee, J. S., "Predictive Grid Process Scheduling Model in Computational Grid", APWeb Workshops2006, LNCS 3842, pp. 525 – 533, 2006.
[6] Yahyapour, R., Wieder, P. (2006) "Grid Scheduling Use Cases", Retrieved Nov. 15, 2010, from http://www.ogf.org/documents/GFD.64.pdf, 2006.
[7] Dong, F. and Akl, S. G. (2006) "Scheduling Algorithms for Grid Computing: State of the Art and Open Problems", Technical Report No. 2006-504, Queen's University, Canada., 2006.
[8] O' Hare, G., Jennings, N. (Eds.), "Foundations of Distributed Artificial Intelligence", Wiley, New York, 1996.
[9] Shen, W., Norrie, D., Barthes, J. (eds.), "Multi-agent systems for concurrent intelligent design and manufacturing", Taylor & Francis, London , 2001.
[10] Abramson, D., Buyya, R., Giddy, J, "A computational economy for Grid computing and its implementation in the Nimrod-G resource broker", Future generation Computer Systems, 18 1061-1074, 2002.
[11] Cao, J., Kerbyson, D., Nudd, G, "Performance evaluation of an agent-based resource management infrastructure for Grid computing", Proceedings of the First IEEE/ACM International Symposium on Cluster Computing and the Grid 311-318, 2001.

[12] Cao, J., Jarvis, S, "ARMS: An agent-based resource management system for Grid computing", Scientific Programming, 135-148, 2002.

[13] Umar Farooq, Shikharesh Majumdar, Eric W. Parsons, "Achieving efficiency, quality of service and robustness in multi-organizational Grids", The Journal of Systems and Software, Vol. 82, No.1 , January 2009.

[14] Wu, Jun and Xu, Xin and Zhang, Pengcheng and Liu, Chunming, "A novel multi-agent reinforcement learning approach for job scheduling in Grid computing", Future Generation Computer System, May 2011.

[15] Rana, O. and Walker, D, "The Agent Grid: Agent-based resource integration in PSEs", Proceedings of the 16th IMACS World Congress on Scientific Computing, Applied Mathematics and Simulation, Lausanne, Switzerland, 2000.

[16] Buyya, R., Abramson, D., Giddy, J., Stocking, H, "Economic models for resource management and scheduling in Grid computing", Concurrency and Computation: Practice and Experience, 14, 1507-1542, 2002.

[17] Baker Albert D., "A Survey of Factory Control Algorithms that can be

[18] Implemented in a Multi-Agent Hierarchy: Dispatching, Scheduling and Pull", Journal of Manufacturing Systems, Vol. 17 (4), 1998.

[19] The Grid Workloads Archive. Retrieved January 13, 2012, from http://gwa.ewi.tudelft.nl/pmwiki/