

Multiagent Framework based Interactive Job Management for Grid

Sneha Viral Mehta
International Institute of Information Technology,
Pune-57, India.

ABSTRACT

Grid , a hardware and software infrastructure that provides dependable, consistent, pervasive and inexpensive access to high end computational capabilities, with intelligent cooperative agents, enable the system to well-suited for many types of services and autonomously adapt to users computation needs as well as dynamically changing computing resource environments. Grid technologies need to be extended which includes graphical, interactive sessions known as Interactive Grids. Interactive Grids permit end-users to access and control a remote resource. The motive of this paper is to intricate the effectiveness of Grid computing by Interactive Agent based Job management. An agent based interactive job management system is developed to incorporate the concept of agent in the grid.

Keywords

Grid, Interactive Grid, MultiAgent, JMS

1. INTRODUCTION

Grid Computing [1] is an active research area which promises to provide a flexible infrastructure for complex, dynamic and distributed resource sharing. Recent research on Grid has largely focused on issues of performance, scalability and standardization. Managing access to computing and data resources is a complex and time consuming task. As Grid computing matures, deciding which systems to use, where the data resides for a particular application domain, how to migrate the data to the point of computation (or vice versa), and data rates required to maintain a particular application “behavior” become significant. Traditional use of Grid Computing Systems has been for batch jobs in the scientific and academic computing. It is envisioned that the next generation Grid computing systems should support graphical interactive sessions [2].

Autonomy is used to describe the character of an agent. An agent can fulfill high-level tasks directly or through cooperation with other agents [3]. Agent and multi-agent technologies provide a promising approach to make Grid technologies and solutions based on Grid and Cluster technologies smarter, more flexible, and adaptable. Agents could play an important role in Grid Computing. Intelligent MultiAgent approaches are well-suited for many types of services. Intelligent cooperative agents enable the system to autonomously adapt to users computation needs as well as dynamically changing computing resource environments.

Main objective of this paper is to elaborate the effectiveness of Grid computing by Interactive Agent based Job management. This special issue brings together fields of grid computing and multiagent technology together with providing interactivity to present their work on the applications of agent-based techniques and approaches in managing and allocating resources in grid computing environment.

2. RESEARCH METHODOLOGY

Globus is a grid middleware, considered to be the de facto standard for grid computing. So, for practical experience of the grid computing it is best to start with Globus.

2.1 Designing MultiAgent Framework for Job Management using Globus

Globus [4] is required to setup grid environment. In which one node has been chosen as a submission node, and the other two are container nodes. These container nodes are actually having the schedulers of OpenPBS, SGE and also condor can be used which creates heterogeneous environment for clusters. When the submission node submits jobs to the containers, then these jobs are scheduled to run on other nodes.

All components and sub-systems in a grid environment can be represented as agents. Different agents communicate using Inter-agent Communication (ICL). Each agent collects relevant identification information, resource information, state information, etc. The multi-agent system-based grid architecture model highly abstracts the substance application. It can be divided into layers in logic. Communication between agents [5] and Grid nodes is shown in Figure 1.

- Through Interface Agent user submits a job.
- Each and every Agent communicates with controller agent which is the master agent.
- Controller Agent through GRAM calls Info and Resource Allocation Agent to assign a suitable node and resources.
- Resource database returns nodes itinerary to the Info Agent.
- Job creating Agent assigns job-id and resources.
- Each Agent begins a user process on an available node.
- After the available node execution, the result is back to the sub-agents.
- Sub-agents pass the result to their main master Agent or Controller Agent.
- Controller Agent handles with all the back results and submits the ultimate result to the user through Interface Agent.

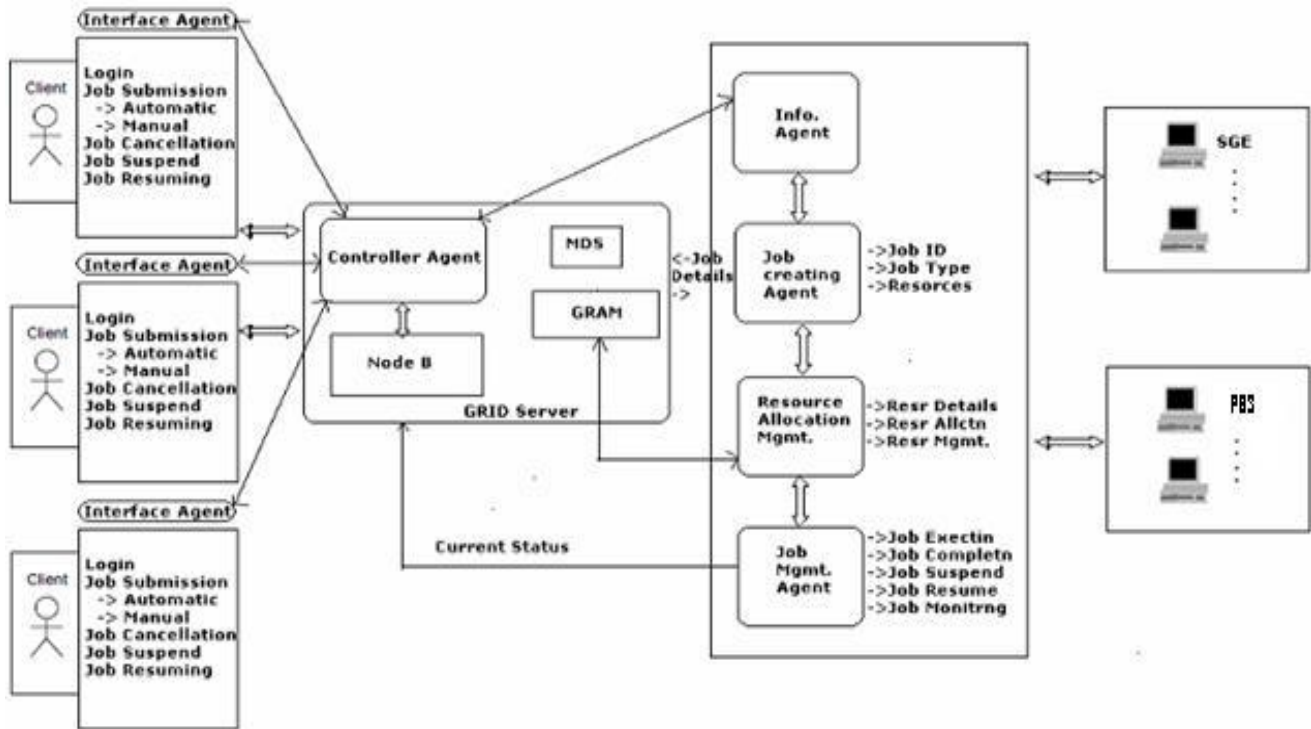


Fig 1: MultiAgent Design

2.2 Job Processing

Four main entities of architecture are: clients, middleware, schedulers and resources. A client is a user who submits a job to the system. A job refers to a collection of Java classes that the client wants to execute. The job is submitted by the client to the web portal through a graphical user interface (GUI). Also agent is an entity which helps to achieve this. The agent delegates the management of jobs to schedulers. A scheduler divides a job into smaller tasks (in the case of an independent job, a task refers to the subset of parameters that can be executed independently) and sends the tasks to the resources for execution. Job Execution takes following steps:

- Event tells the server to start scheduling cycle.
- Server sends scheduling command to scheduler.
- Scheduler requests resource info.
- Server returns the requested info.
- Scheduler request job info from server.
- Server sends job status info to scheduler and scheduler makes the policy decision to run the job.
- Scheduler sends run request to server.
- Server sends job to run.

2.3 Agent Based Job Management

Job management is again an indispensable task for the grid computing environment. Execution of job requires the resources but before that it is necessary to identify the requirement of the resources by a particular job. Then after availability and allocation of the resources is next step. monitoring the execution is also an important task and finally resources have to be made free for supplementary jobs. All these tasks have to be taken care by the job management system or JMS. Few existing tools functioning JMS are condor, Torque and SGE. JMS usually consists of three components: Queue Manager, Scheduler, and Resource Manager.

The term interactive job management refers to availability and accessibility of various tasks of job while it is executing. Following are the identified ones: Job Submission [6], Job Status, Job Suspending, Job Resuming, and Job Destroy. Representation of Agent Based System is shown in Figure 2.

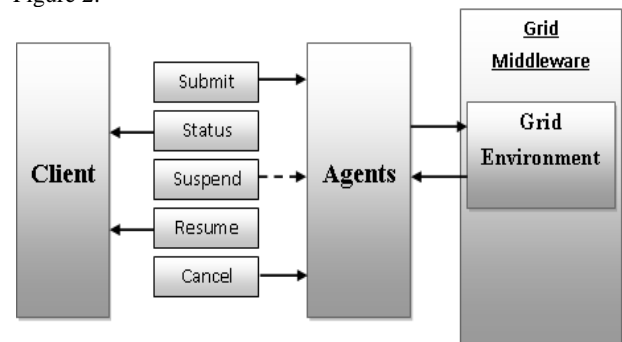


Fig 2: Representation of Agent Based System

2.4 MultiAgent Programming

In grids, performance based application applications need to simultaneously tap the computational power of multiple, dynamically. Grid programming environments (a) need to be portable to run on as many sites as possible, (b) they need to be flexible to cope with different network protocols and dynamically changing groups of compute nodes. It is believed that Java will be a useful technology to reduce the complexity of grid application programming. Hence in this system java has been used to provide both the functionalities and platform independency [7].

3. RESULTS AND DISCUSSIONS

The basic object that grid manages is the job. Job management has several properties: 1) Batch or Interactive, 2) Defines a list of required resources, 3) Defines a priority, 4) Defines the time of execution etc..

Each submitted job can specify a list of resources required for its execution: the number and the type of the specifiable resources is platform dependent. For this paper host, nodes and number of CPUs is considered as resources. For each resource it's possible to specify min-max limits and default values in queues and server attributes to filter different classes of jobs. If a running job exceeds the amount of the resource requested it will be aborted by the server.

Also performance parameters [8] are an extremely important criterion. Some of the basic performance parameters include Resource Discovery, Selection of best resource, Execution Time, Size of the input data, Average RTT, Data Transferred and Message Overhead. Considering execution time as a parameter, agent based Grid system provides following output.

Table1. Performance Comparison of System having Agents and without Agents

Time without Agents	Time with Agents
0.0616	0.0475
3.397	3.20
0.0458	0.0408
0.383	0.375
2.856	2.756

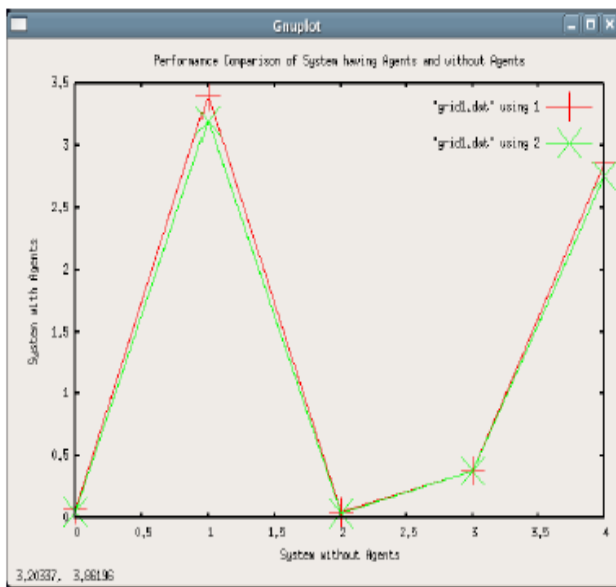


Fig 3: Performance Comparison of System having agents and without Agents (“+” System with Agents, “*” System without Agents)

Table 1 shows execution time taken by jobs with and without agents. In Figure 3 graph shows the comparison between job completion deployed in the grid environment with and without agents. It shows that the use of agents does not delay interactive job submission.

Grid technologies have been used for executing batch jobs only, there was no interactivity provided. A need is there to provide all this capabilities, and this can be easily provided by Grid middleware and Multiagent Framework, also user can interact with the system easily.

4. CONCLUSIONS

It is envisaged that the grid infrastructure will be a large-scale distributed system that will provide high-end computational and storage capabilities to differentiated users. Developed system provides an interactive job management in grid environment. It works on the concept of the agent. Complete working is carried out by multiple agents. The object oriented methodology adopted provides platform independency for the jobs. It also increases the performance as compared to normal execution of the job as jobs are already assigned their respective agents and they will be taking care of their assigned jobs. From the performance results it can be conclude that interactive, Agent based job management system provides user, flexibility while obtaining correct results.

5. ACKNOWLEDGEMENTS

I express my gratitude to Prof. Madhuri Bhavsar, Nirma University, Ahmedabad for sharing her valuable knowledge and Dr. Bharat Chaudhari, I2IT, Pune for his constant encouragement and motivation throughout this work.

6. REFERENCES

- [1] D. Minoli, “A networking approach to grid computing,” Wiley Publications, 2006.
- [2] R. Kumar, V. Talwar, and S. Basu, “A resource management framework for interactive grids,” Accepted in HPDC.
- [3] J. Chen, Y. Wu, M. Li, and B. Hui, “Multi-agent system-based hierarchy grid middleware,” University of California.
- [4] L. Ferreira, V. Berstis, J. Armstrong, M. Kendzierski, and A. Neukoetter in Introduction to Grid with Globus, IBM redbooks.
- [5] J. Cao, D. P. Spooner, J. D. Turner, S. A. and Jarvis, “Agent-based resource management for grid computing,” IEEE/ACM International Symposium, 2002.
- [6] H. XIAO, H. WU, X. CHI, S. DENG, and H. ZHANG, “An implementation of interactive jobs submission for grid computing portals,” Supercomputing Center, Computer Network Information Center.
- [7] R. V. van Nieuwpoort, J. Maassen, R. Hofman, T. Kielmann, and H. E. Bal, “An efficient java-based grid programming environment,” November2002.
- [8] K. Jurasovic, G. Jezic, and M. Kusek, “A performance analysis of multi-agent systems,” 2006.