User Centric Job Management In Grid Using Multiple Agents An Unified Approach

Madhuri Bhavsar

***Abstract*—It has always been accepted that a community of HPC users requires interactive, on demand access to HPC resources. Grid middleware technologies have been aimed at the execution of the sequential batch jobs which is a dominating paradigm, while lacking support for interactive applications. The reason is that grid middleware was developed for compute intensive jobs, which may run for a long time before a result becomes available, which may sometimes leads to unattended execution of jobs. To extend this application domain by providing interactivity support to the users, Agent based technology is used. The research work which is explored and partially implemented considers various aspects of interactive job management in the grid environment.**

***a) :* This paper explores the specific nature of scheduling the grid based interactive jobs which is closely analyzed and reveals the complex approach adopted by intelligent and multiagent technology.**

***Index Terms*—Grid computing, agents, Job Scheduling, HPC.**

I. Introduction

GRID Computing can be seen as a high performance computing system facilitating the sharing of compute resources, allowing users to discover and use remote resources. Users are able to submit the jobs to remote resources and typically have no explicit control over the resources. Thus both users and resources can be viewed as autonomous entities, where this autonomy gives rise to inherent uncertainty, since an individual can not predict how another will respond to a changing situation.. There are two main kinds of jobs in the grid system; Regular (Batch) jobs and interactive jobs. The main difference between those types is that in the interactive jobs, the time slot reserved for running a job on a computational machine must be synchronized with user preferences. Interactive access is defined to be when users can input information into the application and receive timely visual output from the application. Typically the user prepares a job, submits it to the grid, waits for it to be completed and retrieves the output after the job is executed.[3] The users need the HPC system to excel at their jobs providing interactive on demand access to HPC systems. Currently Interactive, on demand access to HPC systems has only been enabled on a very limited basis which is described in section 3. We investigate the effectiveness of using agents to manage the user centric job execution and propose the multiagent framework which facilitates the graphical visualization for user to manage his jobs. The framework which is proposed is an extension to the exiting grid middleware architecture.

Madhuri Bhavsar is with the Department of Computer Science and Engineering, Institute of Technology, Nirma University, Ahmedabad, India, e-mail: madhuri.bhavsar@nirmauni.ac.in

II. Agents

The driving force behind grid computing is the need to share computing resources in a coordinated manner. Grid computing environment is inherently an open environment in which control is distributed between autonomous entities thereby increasing complexity as well as dependability. This complexity is exacerbated in open environment were independently developed autonomous agents interact with each other in order to achieve their goals.

An agent can be viewed as an encapsulated system component that is situated in some environment and is capable of flexible and autonomous actions in that environment in order to meet its design objectives and should exhibit features like Reactivity, Autonomy, and Social Ability and Proactive ness. Usage of this technology provides a flexible approach which can alleviate a number of issues presented in distributed and Grid based systems. This agent technology is emerging as one of the unifying standards for the integration of distributed application and facilitating the user for resolving the complexity arising during job handling.

III. Related Work

In recent years sufficient work has been undertaking from the deploying of various technology for exploring grid services to the use of agent technology for solving various issues in the grid. Authors of presented GAIN (Grid Agent Infrastructure), a multi agent system that supports reliable execution of grid workflow application. Supports enhancement in load balancing using agents which could include fresh node into the system along with simulation. provides agent based framework for grid resource management by solving the complexities like heterogeneous substrates, policy extensibility and many more. used agent technology for reactive power optimization for which they have designed a grid computing architecture. Proposed a grid architecture MAGDA based on mobile agents and resolved various issues in grid by using features such as migration, cloning & messaging. Explored various capabilities of existing resource managers along with interactive and on demand access potential offered and also compares on-demand features of four prominent batch queuing system. It has also come up with LLGrid project which is large scale interactive grid system handles parallel matlab processes. Provides architecture of Multi agent system consisting exception diagnosis agents. Defined the work in which an agent based grid management infrastructure is coupled with a performance driven task scheduling that has been developed for local grid load balancing. Proposed the notion of an agent based grid computing infrastructure (AGCI) which

is developed to manage healthcare processes in real world setting.Authors of Developed AgentTeamwork, which is a grid computing middleware system that dispatches a collection of mobile agents to coordinate a user job over remote computers. This AgentTeamwork has a mobile agent execution platform facilitates jobs migration and resumption of jobs.

IV. Multiagent Framework

Most of the early work on grids has been for batch jobs. In this paper we introduce and describe an architecture and runtime environment for enabling interactive grid based on multiagents. To handle both the types of job execution methods like batch execution and interactive execution, agents can provide a useful abstraction at each of the Grid layers.

By their ability to adapt to the prevailing circumstances, agents will provide services that are very dynamic and robust, and suitable for a Grid environment. Agents can be used to extend existing computational infrastructures. Few research groups have focused on offering an environment to combine the concept of computational grid and agents. An agent can be depicted as an autonomous entity packaged of a set of internal modules, three of which, i.e., for internal scheduling, problem solving and social communication routing, are normative and others are optional. A multi-agent system which is composed of heterogeneous, autonomous entities, have their own problem-solving capabilities and interact with one another in order to achieve an overall goal. Multi-agent systems emphasize both the autonomy of individual agents and the cooperation between agents. In general, coordination is about using a mechanism to manage the interdependencies between the collaborative activities of agents. For this paper the proposed agent will combine stated functionalities described above and work as an independent module for the client application. An agent will accept job parameters from the client and accordingly provide them to grid middleware for further measures and also keep the track of the job. The detail working is described in further discussion of this paper

V. Multiagent Framework

Most of the early work on grids has been for batch jobs. In this paper we introduce and describe an architecture and runtime environment for enabling interactive grid based on multiagents. To handle both the types of job execution methods like batch execution and interactive execution, agents can provide a useful abstraction at each of the Grid layers. By their ability to adapt to the prevailing circumstances, agents will provide services that are very dynamic and robust, and suitable for a Grid environment. Agents can be used to extend existing computational infrastructures. Few research groups have focused on offering an environment to combine the concept of computational grid and agents. An agent can be depicted as an autonomous entity packaged of a set of internal modules, three of which, i.e., for internal scheduling, problem solving and social communication routing, are normative and others are optional. A multi-agent system which is composed of heterogeneous, autonomous entities, have their own problem-solving capabilities and interact with one another in order to

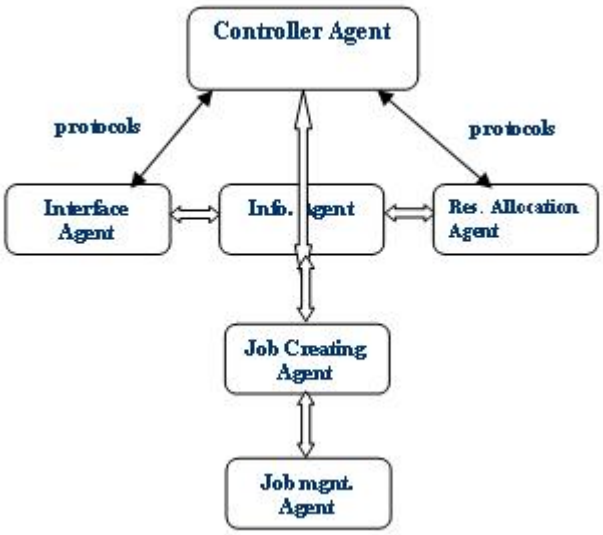

Fig. 1. Multiple Agents in the framework.

achieve an overall goal. Multi-agent systems emphasize both the autonomy of individual agents and the cooperation between agents. In general, coordination is about using a mechanism to manage the interdependencies between the collaborative activities of agents. For this paper the proposed agent will combine stated functionalities described above and work as an independent module for the client application. An agent will accept job parameters from the client and accordingly provide them to grid middleware for further measures and also keep the track of the job. The detail working is described in further discussion of this paper.

A. Functionality of a System

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 jobs, allocation of the resources and takes care the processing of the jobs . All these tasks have to be taken care by the job management system. Usually this JMS consists of three components i.e. Queue Manager, Scheduler & Resource Manager. The term interactive job management refers to availability and accessibility and accessibility of various tasks of job while it is executing. Various functionalities which should be addressed and implemented via agents are -

1) Job Submission
2) Job status
3) Job Monitoring
4) Job Cancellation
5) Job Destroy

Projected MultiAgent framework is composed of various agents who offer demanded functionalities to the user are shown in the figure 1 & 2, whereas figure 3 shows the lifecycle of a job execution. These functionalities are distributed to following agents.

- **Interface-Agent** :- Currently user needs to submit the compute intensive job at the command prompt which

takes long time for execution. Once submitted, manual intervention is difficult. This Interface-agent facilitates a user to interact with the grid system and allows a user to submit the job. User is also given a privilege to specify the resources if known in advance.

- **Controller-Agent** :- This is core agent responsible for all the activities needs to be carried out by the MultiAgent system.
- **Info-Agent** :- Normally Grid system which is composed with Globus, provides directory service called MDS. Due to anonymous problem, nirmagrid does not have MDS. So this Info-Agent created is responsible for providing repository of existing resources and jobs. This agent endows with current status of available resources along with their configuration details which can accept jobs for execution.
- **Job-Creating Agent** :- This framework is aimed at user centric level providing flexibility to the user for usage of the grid. This agent is self running agent helps the system in creation and maintaining the job along with creation of identification mechanism.
- **Resource-Allocation-Agent** :- Acts as a resource handler in the system allocates the resources to the job according to the demand.
- **Job-Managing-Agent** :- If Submitted job is a batch job, avails the computational power and gets executed provided every action in the system is working appropriately. Interactive job needs to be monitored and if needed certain actions are invoked by the user. Job-Managing-Agent helps user to pact with the system for submitting, suspending, resuming and destroying the job.

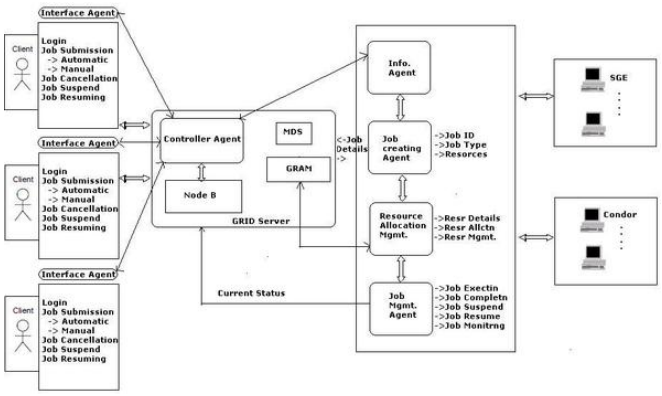

Fig. 2. MultiAgent system providing interactivity for job handling.

b) : During the implementation, various responsive issues needed to be handles are

- **Unauthorized Access Control** :- Interactivity with the Grid system may prone to the unauthorized job & malicious user. Though Grid security infrastructure provides security features, access control to the grid system should be handled at the fine grain level.
- **Performance issues** :- Compute intensive jobs submitted by the user are sensitive to QoS parameters like execution time, response time, latency between Main agent and grid system, Number of jobs executed etc. In addition to providing flexibility to the user our multiagent system should support these QoS parameters.
- **Agent based scheduling** :- Normally Grid system which is composed with Globus, provides directory service called MDS. Due to anonymous problem, nirmagrid does not have MDS. So this Info-Agent created is responsible for providing repository of existing resources and jobs. This agent endows with current status of available resources along with their configuration details which can accept jobs for execution.

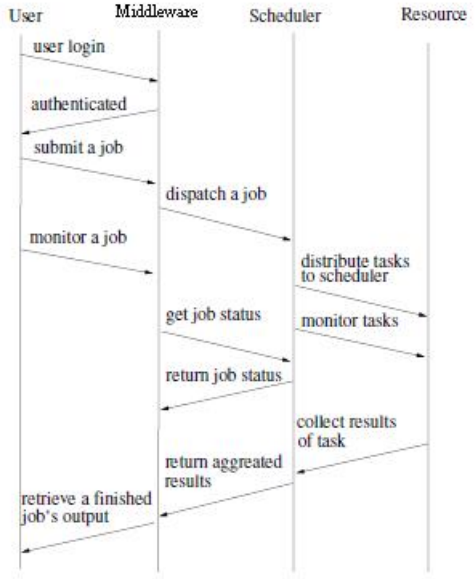

Fig. 3. Job Execution life cycle.

VI. Partial Implementation

Implementation of this multiagent framework is partially completed which is deployed on the grid environment configured with Globus 4.0 on Fedora platform. The end user can request for interactive use from any machine which is also configured in the grid. This activity is an extension to the Globus functionality, so that it can also be used for submitting request through graphical session. Figures 4 & 5 shows functionality supported by an Interface agent and Job-Managing agents.

VII. Conclusion and Future Work

Interactive grids extend the application domain for grid computing system from traditional batch jobs to graphical interactive sessions. In this paper we introduced agent based interactive grid job management. Agents identified presents demanded functionality to the grid user. Remaining agents, after implementation needs to be deployed on the heterogeneous grid platform which we have configured with 80 machines and hoping to cover almost all computers available on the campus.

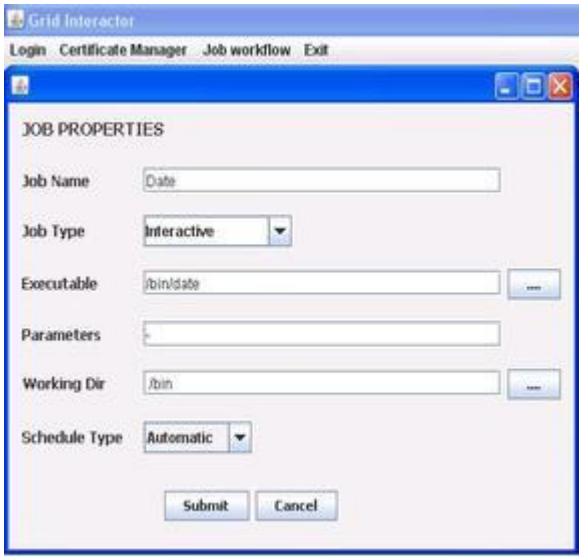

Fig. 4. Implementation of an Interface agent.

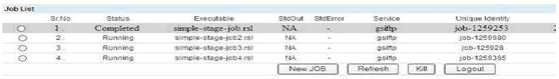

Fig. 5. Status monitoring of a job.

ACKNOWLEDGMENT

The authors would like to thank postgraduate students of M. Tech programme in computer science & Engineering for their contribution in implementing the grid for testing agents.

REFERENCES

[1] A Nagri, A Poggi, M Tomaiuolo, 'Intelligent Task Composition and Allocation through Agents', *14th IEEE International Workshops on Enabling Technologies:Infratrsucture for Collaborative Enterprise (WETICE '05).*

[2] Albert Reuther, Tim Currie, Jeremy Kepner, Hahn Kim, Andrew McCabe, Peter Michaleas and Nadya Travinin,'Technology Requirement for Supporting On-Damand Interatcive Grid Computing', *Proceedings of the User Group Conference (DOD-UGC '05), IEEE.*

[3] Enol Fernandez, Elisa Heymann, Miquel Senar, 'Resource Management for Interactive Jobs in a Grid Environment', *IEEE, International Conference on Cluster Computing Issue* ,25-28 Sept. 2006.

[4] Huaglory Tianfield, 'Towards Agent Based Grid Resource Management', *IEEE International Symposium on cluster computing and the Grid.*

[5] Huaglory Tianfield, 'Towards Agent Based Grid Resource Management', *IEEE International Symposium on cluster computing and the Grid.*

[6] Junwei Cao, Daniel P Spooner, Stephen A Jarvis, Subhash Saini and Graham Nudd, 'Agent Based Grid Load balancing Using Performance-Driven Task Scheduling', *IEEE Proceedings of the international Parallel and Distributed Processing Symposium, IPDPS'0.*

[7] N Shah, K-M Chao, N. Godwin, M Younas, Christopher Laing,'Exception Diagnosis in Agent-Based Grid Computing', *2004 IEEE International Conference on systems, Man and cybernetics, III.*

[8] Rocco Aversa, Beniamino Di Martino,Nicola Mazzocca, 'MAGDA: A Mobile Agent based Grid Architecture', *Architecture'Journal of Grid Computing,2006 Pages 395-412.*

[9] Sasa Desic, Darko Huljenic,"Agents Based Load Balancing with Component Distribution Capability', *Proceedings of the 2nd IEEE/ACM International symposium on Cluster Computing and the Grid (CCGRID '02).*

[10] Society.Munehiro, Koichi Kashiwagi,Shinya Kobayashi,'The Design concept and initial implementation of agent Teamwork Grid Computing Middleware', *IEEE*

[11] Syed Zahid, Fazilah Haron, Gian Chand Sodhy, Shailendra Singh, Chah Huah Yong,"Intelligent Agent-Based Grid Computing: Towards Effective Healthcare Delivery', *Proceedings. 2004 International Conference on Information and Communication Technologies: From Theory to Applications.*

[12] Zhongxu Li, and Yutian Liu,'Reactive Power Optimization using Agent based grid computing' ,*IEEE, Power Engineering Conference, 2005. IPEC 2005. The 7th International Volume , Issue,* Nov. 29 2005-Dec. 2 2005.

Madhuri Bavsar Madhri Bhavsar is Associate Professor at Nirma University, She has obtained B.E. From North Maharashtra University in 1993 and M.E. in Microprocessor Applications from M.S. University, Baroda. She is currently pursuing Ph.D. at Nirma University. Her current reserch interst is in Grid infrastructure for eScience and High Performance Computing. She is member of IEEE and ISTE.