Analysis of Multiagent Based Interactive Grid Using Formal Methods - A Reliable Approach

Madhuri Bhavsar
Associate Professor
Department of Computer Science and Engineering
Institute of Technology, Nirma University
Ahmedabad, India
Email:madhuri.bhavsar@nirmauni.ac.in

***Abstract*—This paper presents analysis, design and implementation of MultiAgnet framework for user centric interactive Job management system. A reliable approach is adopted by analyzing this system using formal methods. Formal methods allow a software engineer to create a specification that is more complete, consistent, and unambiguous.**

Most of the grid middleware supports for batch processing of jobs which is dominating paradigm leading challenges for execution of parallel and interactive jobs on the grid environment. In current situation starting from the console based job submission to monitoring, the process is cumbersome. Specifically in long running job, lack of interactive support creates difficulty in tracing the failure of job which may sometimes leads to wastage of resources. Overcoming these lacunas, efforts taken for this project have resulted into development of Globus compatible interactive user driven scheduler facilitating user an ease in managing jobs.

***Keywords*-Formal Methods in Grid, HPC,Interactive jobs in grid, User driven scheduler**

I. Introduction

A Computational grid is an emerging computing infrastructure that enables effective access to high performance computing resources wherein job management, resource management and scheduling are key grid services. To make grid environment greatly useful, user should be given an ease in handling grid. In this area of High Performance Computing (HPC), where the lifetime of application codes often exceeds that of most HPC machines, the ability to access these machines through a simple generic end-user interface is of real interest [1].It is increasingly been recognized that a large pool of HPC users requires interactive, on demand access to HPC resources[2]. Most of the grid middleware supports for Batch processing of jobs which is dominating paradigm. As HPC is becoming a valuable and emerging technology, it needs to accept the challenges for execution of parallel and interactive jobs on the grid environment facilitating the user through the common interface. It would also be of great importance if the user is permitted to handle the job in interactive manner by obeying constraints specified. This paper shares about the system developed in the grid environment which facilitates the user to handle the jobs compatible with Globus toolkit.The Globus Toolkit is an open source software toolkit used for building grids[6]. Moreover the concentrated part of this paper is Analysis carried out for the implementation of this project using formal methods helps in preventing ambiguity, inconsistency and incompleteness.

II. Need of Interactive Job Management

The goal of Grids is to virtually combine geographically distributed IT resources from many different administrative domains, into single customized resources that can perform a single or a series of specific tasks[4]. Grid should facilitate an appropriate mode to deal with these tasks. There are two main kinds of tasks/jobs in the grid system: Regular (batch) jobs and interactive/visualization tasks [3]. Middleware systems for interactive jobs invocation in the grid based systems are in the initial state. There is tremendous need for an end-user interface to a wide area Computational Grid, which will be comprised of a heterogeneous environment of machines in diverse administrative domains [7].[8]Defines brokering infrastructure for interactive applications whereas [9] explores several interactivity support techniques in Grid. Graphical framework for Grid interactive applications are defined in [10] very efficiently. Grid computing middleware system, Agent Teamwork, defined in [11] are used to dispatch agents to coordinate user jobs. Interactive access is defined to be when users can input information into the application and received timely output from the application while on demand access is defined as being allocated immediate access to the requested resources rather than having to wait for resources to be allocated by a batch queuing system. For long running jobs, interactive access helps user to handle and monitor jobs in effective manner retaining grid resource usage optimum.

III. Architecture of MultiAgnet Framework for Job Management

Realizing the need to handle interactive jobs by giving preferences to user requirements is motive of this research which helps in fulfilling user driven facilities. So the design

978-0-7695-4246-1/10 $26.00 © 2010 IEEE
DOI 10.1109/ICETET.2010.51

and the development of user driven scheduling algorithm for interactive job management in the Grid environment was challenging task. This facilitates the user an ease in interacting with grid environment for deployment of jobs with fulfilling the specified requirements. To carry out these requirements, multiagent framework is build. This multiagent framework consists of group of interacting intelligent agents working together to achieve a goal of providing flexibility of managing jobs to the user. Fig 1 shows scenario where user is allowed to handle the jobs deployed on the grid.

IV. Analysis using formal method

Analysis is concerned with devising a precise, concise, understandable and correct model of the real world. The prototype which is developed has been analyzed using formal methods.

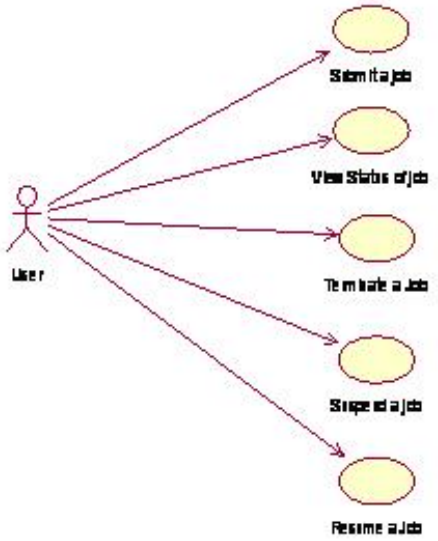

Figure 1. Use case diagram for handling jobs

Formal methods allow a software engineer to create a specification that is more complete, consistent, and unambiguous.[5] This mathematical specification can then be analyzed to prove correctness and consistency and thus helps in avoiding failure of the model leading to the reliability.
Basic components of formal methods are data invariant, state, operation with precondition and postcondition. data invariant is a condition that is true throughout the execution of the system that contains a collection of data. State is the stored data which a system accesses and alters during execution. Operations are an action that takes place in a system and reads or writes data to a state. Precondition defines the circumstances in which a particular operation

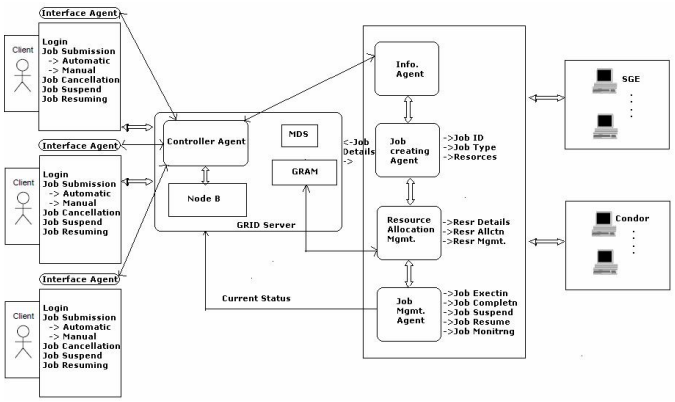

Figure 2. Architecture of MultiAgent Framework

is valid and Postcondition defines what happens when an operation has completed its action

Following Agents are identified for the framework -

- ULA - Interface Agent
- CA- Controller Agent
- InfA - info Agent
- JcA - Job Creating Agent
- RaMa - Resource Allocation and Mgmt Agent
- JmA - job managing agent

Requirement Analysis using Formal Specification language are carried out as mentioned -

- A Set named MULAG consists of every Agent
- States are modeled by 6 sets of Agents
- Six identified states UIA,CA,InfA,JcA,RaMa, JmA are Components of MultiAgent Framework provides interactivity to the grid user

States-
UIA,CA,InfA,JcA,RaMa, JmA : MULAG
AllocRes,UnAllocRes:AVAILRES
GN1,GN2,... ...GNn: NIRMAGRID
RegUser $\subseteq$ User

- CanJob,SubJob,SchJob,SusJob $\subseteq$ Job
- RegUser = $\forall$ U $\in$ RegUser $\Rightarrow$ U $\in$ User U(Certified) (DN)
- MaxJob $\rightarrow$ Maximum number of jobs
- SubJob = (N,T,Para,WD,ST,C) : T=Interact,Batch, ST = (Auto,Man);if ST = Man . C = cpu,mem,ss,si
- SchJob = (Jid,Allocres,status,Para)
- SusJob =(Jid,AllocRe,current_status)

Data Invariants

- Submitted job should be grid enabled
- #Subjob $\leq$ MaxJob
- if no resources then ST can not be manual. SubJob.T = Interact
- SchJob'=SchJob + x x SusJob SusJob'=SchJob - x

- CanJob'=CanJob + x $\Leftrightarrow$ x $\in$ SchJob SchJob'=SchJob + x
- Every Resource type is a part of NirmaGrid $\forall$ x $\in$ SubJob.c $\Rightarrow$ x NirmaGrid
- Submitted job should be mapped to scheduled job SchJob $\Leftrightarrow$ SubJob
- SchJob.St = Man $\Rightarrow$ c Nirmagrid else c = ø

Operations

- User should be able to select & submit a job
- If resources available then manual selection of resources is possible
- User should be allowed to resume the suspended job.
- User should be able to cancel the job
- If ST is manual, submitted job added to manual Sch.job list with resources = C, else added to AutoSch job list
- User subjob.T = interactive then user should be able to see the status.

V. Implementation & Results

As a first step toward providing a solution to the problem defined above , a single interface, which should bind and interact with the various heterogeneous resources needs to be created. This interface becomes the part of the middleware Globus and may be compared to an operating system of a Grid, which allows the user to select the resources, specify the constraints and schedule the job on grid nodes in transparent manner with handling security and permission issues.

The developed system provides users with an easy to learn and use GUI based UI for job submission, monitoring, and control high performance computing services.

The implementation is divided among various modules. The modules are divided as:

- Graphical User Interface
- Fetching data from MDS
- Fetching data of processes running over remote node
- Maintaining Input & Output between remote node and admin node
- Implementation of scheduling algorithm
- Deployment of Job-Modules as per scheduling decision on remote node
- User centric Job management functionalities

Fig 3 & 4 shows facilities of jab handling assigned to the user. User can specify his constraint like deadline for job execution. And to act upon this, the algorithm executes to balance the load, if required. Figure 5 compares the results obtained after executing the job with and without agents and proves that use of agents does not make any delay in job management activities.

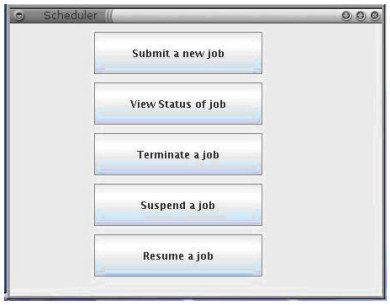

Figure 3. GUI for job handling

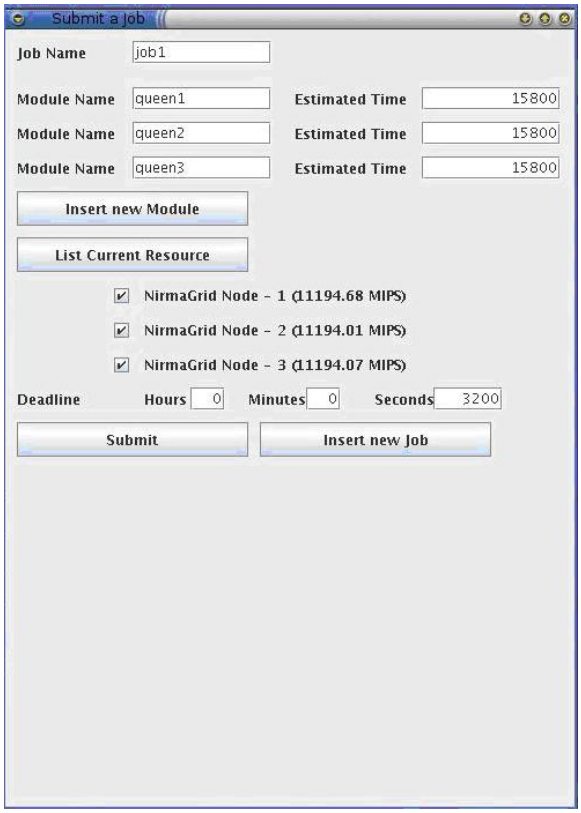

Figure 4. Job submission

In Figure 6, a comparison between FCFS algorithm and Proposed algorithm is done based upon the total number of Jobs scheduled from the pool given by, taking in to consideration the deadline of the Jobs and achieving Load balancing as per Ideal Load of each Resource

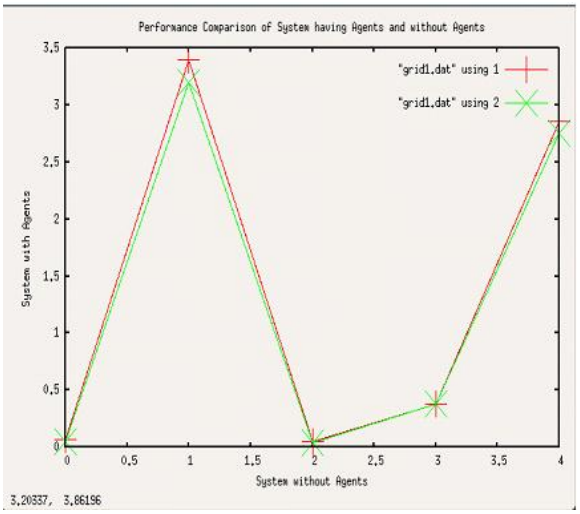

Figure 5. Performance measurement with and without agents

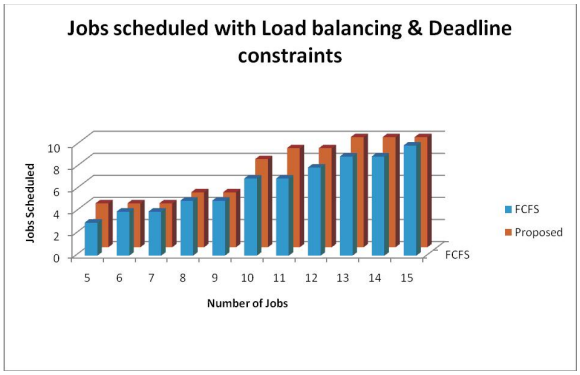

Figure 6. Total jobs scheduled

In Figure 7, analysis of the relation between the numbers of Jobs submitted with the given deadline is shown. The total number of Jobs is considered to be the same for the analysis i.e. 15 Jobs. The deadline of each Job is increased as shown in the figure and then the total number of Jobs submitted within that deadline is analyzed. As the deadline of the Jobs increases, the total number of Jobs submitted also increases.

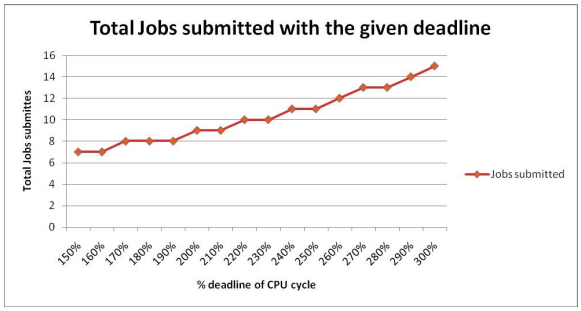

Figure 7. Jobs submitted in given deadline

VI. Conclusion

During analysis , use of formal methods helps to build the system in reliable manner. States, data invariants and operations with precondition and post condition helps in defining the requirements very clearly with consistency.

For execution of interactive Jobs, traditional scheduling algorithms would be of no use as they are designed for batch Jobs. So, for interactive Jobs, the response time is more important. And the Jobs should be deployed over the Grid such that Load balancing is achieved for maximum utilization of Grid

Actually experimental work done for development of agent based scheduling algorithm for handling interactive jobs are compressed in this paper. It is observed that Agent based system work efficiently to carry out the functionality specified by the user.

References

[1] Ali Anjomshoaa Lorna Smith, Motivation and Requirements for a User Interface to a Computational Grid in the UK,Technical reports,UKEC publications 2003

[2] Albert I Reuther, Tim Currie, Jeremy Kepner, Hahn G. Kim, Andrew McCabe, Peter Michaleas, Nadya Travinin, Technology Requirements for Supporting On Demand Interactive Grid Computing, Proceeding of the Users Group Conference (DOD-UGC'05), IEEE, 2005

[3] Marcin Lawenda, Marcin Okon, Ariel Oleksiak, Bogdan Ludwiczak, Tomasz Pio- ntek, Juliusz Pukacki, Norbert Meyer, Jaroslaw Nabrzyski, Macicj Stroinski, Run- ning Interactive Jobs in the Grid Environment, Springer LNCS 3911, pp. 758-765, 2006

[4] I. Foster and C Kesselman, " The Grid2:Blueprint for a New Computing Infrastructure." Morgan Kaufmann publisher, 2004

[5] Roger Pressman, Software Engineering, A Practitioners Approach, Fifth Edition, McGrawhill International edition, PP 682-686

[6] http://globus.org/toolkit

[7] J.Gomes, A Grid Infrastructure for Parallel and Interactive Applications, Comput-ing and Informatics, Vol. 27, 2008, No. 2:173-185

[8] E.Fernandez, A.Cencerrado, E.Heymann, M.A.Senar, Cross-Broker:A Grid Metaschedular for Interactive and Parallel Jobs, Computing and Informatics, Vol. 27, 2008, No. 2:187-197

[9] H.Rosmanith, J.Volkert, Interactive Techniques in Grid Computing:A Survey, Com- puting and Informatics, Vol. 27, 2008, No. 2:199-211

[10] M.Owsiak, B.Palak, M.Plociennik, Graphical Framework for Grid Interactive and Parallel Apllications, Computing and Informatics, Vol. 27, 2008, No. 2:223-232

[11] Munehiro Fukuda, Kochi Kashiwagi,Shinya Kobayashi, The Design Concepts and Initial Implementation of Agent Teamwork Grid Computing Middleware, Communications, Computers and signal Processing, 2005. PACRIM. 2005 IEEE

[12] J.Macro, I.Campos, I.Cotterillo, A.Monteoliva, C.Oldani, Modeling of a Watershed:A Distributed parellel Apllication in a Grid Framework, Computing and Infor- matics, Vol. 27, 2008, No. 2:285-296