GRASP: A Grid Resource Allocation System based on OGSA

Oh-Kyoung Kwon, Jaegyoon Hahm, Sangwan Kim, Jongsuk Lee
Grid Technology Research Department, Supercomputing Center, KISTI
{okkwn, jaehahn, sangwan, jsruthlee}@kisti.re.kr

Abstract

In this paper, we describe GRASP, a Grid resource allocation system based on OGSA. In order to submit job to the Grid resources in more efficient and convenient manner, we support some features for user-friendly resource allocation such as resource brokering, scheduling, monitoring, and so forth. GRASP supports any scientific applications with the high performance computing features such as MPI and applications with high throughput computing features such as parameter studies.

1. Introduction

The Grid resource allocation system concerns about delivering the users plentiful computing power among the distributed resources. ManagedJobService in Globus Toolkit™ 3, which we refer to as GT3, is the service to be used to run a job on a remote resource [1]. However, in order to build more useful Grid, there should be added some features for user-friendly resource allocation such as resource brokering, scheduling, monitoring, and so forth. Our research goal is to develop a resource allocation system which allows users to submit their jobs in more efficient and convenient manner.

We have proposed the architecture of GRASP (Grid Resource Allocation Services Package) to meet above requirements and have also implemented the services as well as the tools such as job submission client.

2. GRASP

GRASP is the resource allocation component of the K*Grid middleware toolkit called MoreDream (see Figure 1) [2]. K*Grid is the Grid infrastructure which has been constructed over the nation under the leading of KISTI in Korea. GRASP is composed of the OGSI-compliant services [3]. We have designed and implemented the Grid services for the Grid resource allocation based on GT3 Core which is the OGSI specification implementation. To make the job submission process efficient and simple to the user, the functions of GRASP are focused on the collective layer of the Grid layered architecture. GRASP services can interact with ManagedJobService of GT3 or the allocation service of GRASP on the resources.

GRASP supports any scientific applications with the high performance computing features such as MPI and applications with high throughput computing features such as parameter studies. GRASP is composed of several useful services needed to allocate the resources in Grid. Firstly, the resource brokering is done by GridSchedulingService. This service finds out resources from the index service which are fit to a user’s job. To select proper resources it performs matchmaking between the resource specification from the user and the job/user specification preferred by the resource administrator. And then the resources are allocated to the job. Secondly, JobSubmissionService does co-allocation of resources and co-monitoring the job. Co-allocation in GRASP makes it possible the job submission to the multiple distributed resources simultaneously. A MPI job which requires multiple distributed resources cannot be supported by GT3. However, we have solved that problem in GRASP by designing a MPICH initialization process in which all
subjobs are synchronized by JobSubmissionService (see Figure 2) [4]. And co-monitoring allows the user to monitor his/her job flow. Thirdly, ResourceManagerService authenticates the user for the job execution on a local resource and submits the job to the local batch queuing system such as PBS.

Lastly, we have designed the job description language, named JRDL (Job & Resource Description Language) to overcome the limitation of the GT3. RSL 2, which is GT3 job description language, just describes job specifications rather than resource specifications such as resource preference. RSL 2 also not considering about co-allocation. Therefore, we have proposed JRDL to meet both requirements for the job and resource’s preference. The resource preference part is used in matchmaking step in GridSchedulingService. JRDL is designed on the based on XML schema.

3. Main title

We have implemented the services of GRASP based on GT3 core:

There are two persistent services on the client area as follows: JobSubmissionFactoryService and GridSchedulingFactoryService instantiates JobSubmissionService and GridSchedulingService, respectively. And there are also two persistent services on the resource area as follows: ResourceManagerFactoryService and MasterResourceManagerService instantiates ResourceManagerService and redirects to ResourceManagerFactoryService in the actual user area, respectively.

The user can submit a job by writing JRDL manually. However, it is difficult for application users to describe JRDL. Therefore, we have provided job submission client tool. Tool supports the job and resource description, job submission, file transfer server (e.g. GASS server) invocation, job monitoring, output file view and std out/std err streaming.

4. Conclusion

We have designed and implemented the services of GRASP which are required to submit job to the Grid resources in more efficient and convenient manner. The services are OGSI-compliant Grid services which are implemented by using GT3 core. The main features of GRASP are co-allocation, scheduling, job monitoring, and resource selection by matchmaking. GRASP supports any applications for both high performance computing and high throughput computing.

5. References


Figure 2. GRASP Architecture