A Grid Service-Based Engineering Computation Architecture

Changqin Huang, Deren Chen, and Hualiang Hu
College of Computer Science, Zhejiang University, Hangzhou, 310027, P. R. China
{cqhuang,drchen}@zju.edu.cn

1. Introduction

Grid computing is becoming a new computing infrastructure for large-scale scientific computation and cooperative work. The Globus toolkit is the most popular grid environment and de facto grid standard, and it has adopted OGSA [1] architecture to provide applications with grid service level at present. Until now, there exist many applications to be developed or deployed based on the Globus. As a special type of large-scale computation, engineering computation is very hardly-solved because of the requirement for many valuable computing resources or specialist instrumentations, grid computing is a suited computing infrastructure for the high capability of distributed, heterogeneous and dynamic resources sharing. However, grid-based engineering computation needs to solve some important issues.

2. Overview of Architecture

This study focuses on the requirement for scientists in engineering computation field and narrows the gap between grid service and the engineering community. The overview of architecture is shown in Fig.1, This architecture consists of four components as follows:

The interface component is a portal of the system, It not only provides a visual interface for development and steering of grid-based applications, but also gives these system administrator a friendly management portal. It comprises a task submission window and two types of grid scheduling interfaces, a QoS presence and visual management interface and an advanced administrative entry.

Task-level logic component is responsible for task management, community policy and user administration. It includes virtual organization register, user manager, task definition, task import and export, task submission, task scheduling, task monitoring, task rescheduling, file transfer, and result processing. By the above-mentioned interfaces, a grid user can implement corresponding interaction with the system in a visual manner. Certainly, in this part, all task-level algorithms and user-level policies are applied in the aid of the next diverse grid services, such as resource discovering services.

Grid service logic part consists of all services and service management mechanisms. All services are divided into three types: advanced general service, engineering computation modeling and post-processing service, basic grid service. The former two types are constructed on the basis of the latter. Service management contains: service deployment, service location, service composition, service scheduling and service-level performance steering.

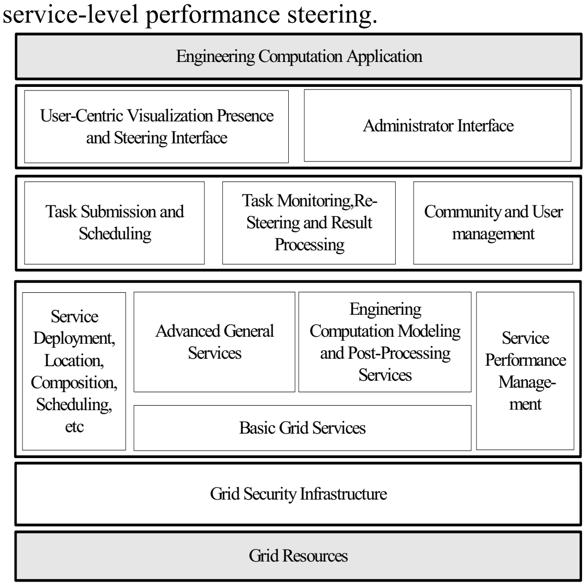

Fig.1. Engineering Computation Architecture

Grid Security Infrastructure is the last hierarchy in our architecture. It is based on the Globus Toolkit's security mechanism, main characteristics exist: context-aware resource usage authorization, community policy and task delegation management. This component is the foundation of other components, and many operations are collaboratively implemented altogether with operations of other components. Its authorization management keeps contact with grid

Proceedings of the 28th Annual International Computer Software and Applications Conference (COMPSAC'04)
0730-3157/04 $20.00 © 2004 **IEEE**

resource management. Some valued computer resources and visualization devices are cast with great security focus.

3. Visual Scheduling

Under the guidance of user-centric thought，this architecture adopts aggregated QoS driven visual scheduling, the scheduling could be performed through two types of visual windows, scheduling and QoS sessions. These visual methods provide users with a direct awareness and a friendly interaction. A visual scheduling framework is shown in Fig. 2.

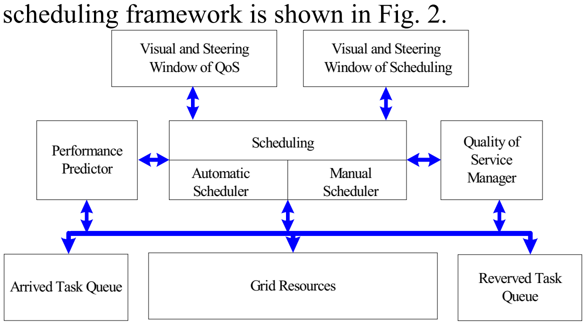

Fig.2. A visual scheduling framework.

In the engineering computation environment, aggregated QoS is modeled based on four performance metrics. The details are presented in the literature [2].

According to different types of users, the visual scheduling is performed in both manual and automatic manner with a simple monitoring mechanism. In addition to the arrived task queue, a reserved task queue is used for arranging pre-scheduled tasks. Hence reservation-based scheduling can perform well. The scheduling algorithms during the automatic scheduling can be selected by users, as various conventional algorithms can be integrated into the system. The idea is based on the fact that different algorithms are suitable for respective environments and objectives.

In our architecture, Performance predictor serves for the scheduler by predicting and computing the previously mentioned performance metrics. It analyzes the performances of the tasks to be scheduled in advance; furthermore, it evaluates QoS parameters of scheduled tasks. QoS manager is responsible for accepting these required performance values from the input, for setting performance threshold values that are conditions of triggering adjustment mechanisms and warning user, and for managing the events of post-scheduling and the interactions for a better performance.

4.Parallelized Subtask-Level Authorization Service（PSAS）

The grid-based engineering computation architecture accomplishes PSAS Framework based on common security mechanism of Globus. PSAS Framework is concerned with the different privilege requirements of subtasks, user privilege of resource usage and resource policy in virtual community, task management.

The framework includes three functional modules, the parallelizable task is decomposed and the least privileges required for each subtask is re-allotted after analyzing the source codes of the task. This function is performed in subtask-level authorization module. A delegation mechanism collaboratively performs the authorization delegation for task management together with a relevant management policy, which is responsibility of task management authorization module. Community authorization module addresses community authorization mechanism for community member. PSAS imposes similar CAS [3] mechanism, as well as the task management policy is added into the community policy server.

A context-aware authorization approach is adopted based on PSAS framework. The main idea is that actual privileges of grid users are able to dynamically adapt to their current context, and it is exercised by constructing a multi-value community policy.

All details are described in the literature [4].

5. Conclusions

The engineering computation architecture supplies the users with a visual application development and deployment environment for engineering computation. The whole is characterized by grid service. At the same time, QoS driven user-centric scheduling, community-based efficient authorization and flexible task management are designed and implemented.

6. References

[1] J. Nick, I. Foster, et al.: The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, Open Grid Service Infrastructure WG, Global Grid Forum, 2002

[2] C. Huang, G. Song, et al.: Quality-of-Service Driven Visual Scheduling in Grid Computing, Proc. of GCC 2004, 2004.

[3] L. Pearlman, V. Welch, et al.: A Community Authorization Service for Group Collaboration, Proc. of the 3rd IEEE International Workshop on Policies for Distributed Systems and Networks, 2002.

[4] C. Huang, G. Song, et al.: An Authorization Architecture Oriented to Engineering and Scientific Computation in Grid Environments, Proc. of ACSAC 2004, 2004.