A Fault-Tolerant Distributed Scheme for Grid Information Services

Ming-Jeng Yang\textsuperscript{1}, Chin-Lin Kuo\textsuperscript{2}, Shih-Hsiang Lin\textsuperscript{2}, and Yao-Ming Yeh\textsuperscript{2}

\textsuperscript{1} Department of Information Technology, Takming College, Taipei, 114 Taiwan
mjyang@mail.takming.edu.tw
\textsuperscript{2} Department of Information \& Computer Education, National Taiwan Normal University, Taipei, 106 Taiwan
\{genemlsh178, virtualbow\}@yahoo.com.tw, ymyeh@ice.ntnu.edu.tw

Abstract. The Grid Information Service (GIS), mainly used for resource discovery and monitoring, is a key component of grid system. The resource description and specification should be mediated for efficient search and access. In this paper, we propose a distributed system for grid information services, which deploys a number of registry servers at different regions of the world. A new scheme for registering, updating, querying, and deregistering a resource in registry servers is devised. For the purpose of fault-tolerance and load-balance, the meta-data, including description and specification, of each resource can be replicated and disseminated at some registry servers instead of reproducing resource itself for service discovery. In our scheme, the workload on each registry server is balanced and the faults of registry servers can be tolerated. Also, the user could obtain all resource information satisfied with the query conditions even some of registry servers crashed.

1 Introduction

Grid technology has increasing played an important role in scientific computing field. The resources of grid are scattered on numerous places or organizations and with distinct type, function or/and owner. The issues of resource information discovering, registering and securing become more and more important. In order to discover necessary resources as soon as possible, effective mechanism of information service, such as accessing interface, meaning of parameters and resource function, is strongly demanded. Hence, resource description and specification must be clear and efficient enough. In [1] the authors propose an XML-based grid resource specification language and its usage in Resource Registry Meta-Service.

Grid is considered as a service oriented architecture (SOA) system and could be accessed through services. So we still take grid as a different type of web service. But there are still many problems required to be conquered. For example, the centralized model of UDDI [2] which is the information center of web service architecture providing XML based service specification standardization is not appropriate for the grid environment. Web service inspection language (WSIL) [3] does not provide good implementation at distribution of services and is difficult to be used in grid. There are several related works about grid information service, which include GMA [4],
Hawkeye [5] and MCS [6]. However these works are not open grid service infrastructure (OGSI) [7] compliant and therefore their ability of manipulating dynamic, heterogeneous, distributed information is profoundly limited.

The open grid service architecture (OGSA) [8] was developed to solve the challenges in such dynamic, heterogeneous and geographical grid environment. OGSA builds on the web service technology mechanisms to create, name and discover transient grid service instances with a uniform manner and it is a popular and a widely accepted architecture. Web services provide important machinery, but are lack of some important topics relevant to basic service semantics: how services are created, how long they live, how to manage faults, and how to handle long-lived state. These issues are addressed by the design of OGSI which defines essential building blocks for distributed systems. The OGSI is a formal and technical specification from OGSA. Using a combination of WSDL interface descriptions and human readable specifications, OGSI defines mechanisms for creating, naming, managing lifetime, monitoring, grouping, and exchanging information among grid services. OGSI also introduces standard factory and group registration interfaces for creating and discovering grid services. While developing a GIS system, the requirement and features of OGSA should be taken into consideration. The globus toolkit 3 [9], a reference implementation of the OGSI, provides a grid service-oriented information service.

In this paper, we present a quorum based grid fault-tolerant scheme for information service registering, updating and querying. In our scheme, based on the Leg-Ring system [10], the workload disseminated on each service node is balanced and the faults of service nodes can be tolerated. Meanwhile, by using the meta-data which describes and explains resource characteristics detailed, the service could provide efficient and accurate interface for searching, understanding and further processing.

The rest part of this paper is organized as follows: section 2 describes the architecture of the grid information service. Section 3 explicates Legion structure, quorum system and its application on this GIS. Section 4 describes these protocols that include registry, query, update and deregistry. Finally, in section 5, we draw conclusions based on our research.

2 Architecture Overview

In our design, the information service is based on the distributed architecture and the meta-data information of resources is stored at a number of service nodes. For the purpose of fault tolerance and load-balance, the meta-data, including description and specification, of each resource can be replicated and disseminated at some service nodes instead of reproducing resource itself for service discovery. Similar to meta-data based model in [11], each user in client terminal could issue a query with appropriate keywords to a server. Upon receiving the query, the local server forwards the message to some servers for discovering registered resources satisfied with the inquired conditions.

We call this information service “Quorum-based fault-tolerant information service (QFIS)”. The architecture of QFIS consists of three layers: resource layer, control layer, and service layer, which are depicted in figure 1.