# Efficient Data Indexing System Based on OpenLDAP in Data Grid*

Hongseok Lee[1], Sung-Gon Mun[1], Eui-Nam Huh[2], and Hyunseung Choo[1,**]

[1] School of Information and Communication Engineering,
Sungkyunkwan University, Korea
choo@ece.skku.ac.kr

[2] Dept. of Computer Engineering, KyungHee University, Korea
johnhuh@khu.ac.kr

**Abstract.** Grid technologies enable sharing various types of large-scale data resources generated by many unknown users for day by day jobs done at work. Finding the required data there in an easy manner is really necessary but laborious. Monitoring and information service(MIS) is very important in huge distributed systems such as grid and effective in data probing. Here we develop data indexing system(DIS) to provide the efficient data access to users based on OpenLDAP for the distributed environment. According to the comprehensive evaluation, DIS shows the better performance in terms of response time and scalability for the large number of users. It is also expected that the proposed system can be applied to globus system.

## 1 Introduction

Proportional to the increasing number of scientific disciplines, large data collections are emerging as important community resources. In domains as diverse as global climate change, high energy physics, and computational genomics, the volume of interesting data is currently measured in terabytes and will soon total petabytes. This combination of dataset size, geographic distribution of users and resources, and computationally intensive analysis results in complex and stringent performance demands that cannot be satisfied by existing data management infrastructure. A large scientific collaboration may generate many queries, each involving access to gigabytes or terabytes of data. The efficient and reliable execution of these queries may require careful management of terabyte caches, gigabit data transfer over wide area networks, scheduling of data transfers and supercomputer computation[1].

At present, few studies have been published that quantitatively evaluate the performance of the current monitoring and information services in distributed

* This research was supported by the Ministry of Information and Communication, Korea under the Information Technology Research Center support program supervised by the Institute of Information Technology Assessment, IITA-2005-(C1090-0501-0019).

** Corresponding author.

V.N. Alexandrov et al. (Eds.): ICCS 2006, Part I, LNCS 3991, pp. 960–964, 2006.

systems. The Relational Grid Monitoring Architecture(R-GMA)[2] monitoring system is an implementation of the Grid Monitoring Architecture(GMA). It is based on the relational data model and Java Servlet technologies. Its main use is the notification of events that is, a user can subscribe to a flow of data with specific properties directly from a data source. Hawkeye[3] is a tool developed by the Condor group and designed to automate problem detection, for example to identify high CPU load, high network traffic, or resource failure within a distributed system.

In this paper, DIS based on Open source implementation of the Lightweight Directory Access Protocol(OpenLDAP) which is included in Monitoring and Discovery Service(MDS2)[4], is developed in Globus environment. The Globus Toolkit(GT)[5] has been developed since the late 1990s to support the development of distributed computing applications and infrastructures. MDS2 is the grid information service used in GT. It uses an extensible framework for managing static and dynamic information regarding the status of a computational grid and all its components: networks, computing nodes, storage systems, instruments, and so on. MDS2 is built on top of OpenLDAP[6]. That is the predominant Internet directory access protocol and hence is also used in Public Key Infrastructure. It stores certificates and provides efficient access methods by harnessing storage means with communication mechanisms.

## 2 Proposed DIS

### 2.1 DIS Architecture

The topology consists of four Linux machines with hostnames monet{156, 157, 158, and 164}.skku.ac.kr set up on a 100Mbps LAN. Fig. 1 presents DIS architecture with MDS2. Host monet164 serves Grid Index Information Service(GIIS) and remaining monet{156, 157, and 158} nodes only serving Grid Resource Information Service(GRIS) include DIS. Each GRIS reads the information regarding the science data, and generates the Lightweight Directory Interchange Format(LDIF) data objects. Then, it registers LDIF to GIIS. Therefore, GIIS maintains entire information such as the index of data names and sizes of each host that operates on the system. Hence, the authorized user accesses to GIIS and searches for information that the user wants. DIS interoperates with MDS2 on the Globus Toolkit, therefore the architecture that supports many GIISs can be expanded to prevent the single point of failure.

### 2.2 Development Procedure for DIS

In order to develop DIS, the first step is to define schema, which will be represented in the Directory Information Tree(DIT). We must follow the OpenLDAP policy schema, Object Identifier(OID), and its naming convention, because a directory service based on OpenLDAP is used. One objectclass is

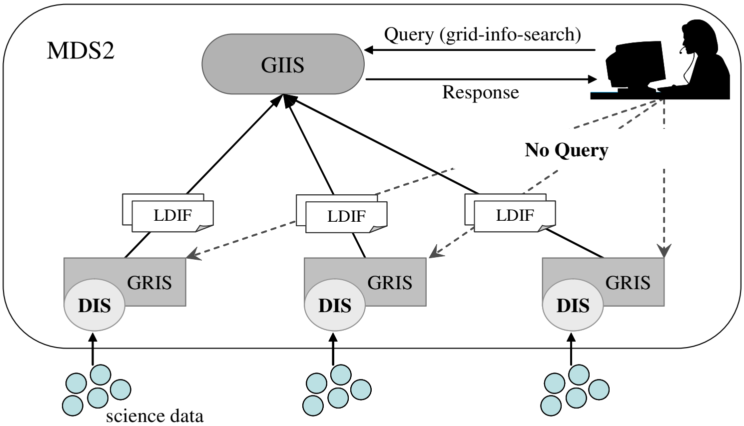
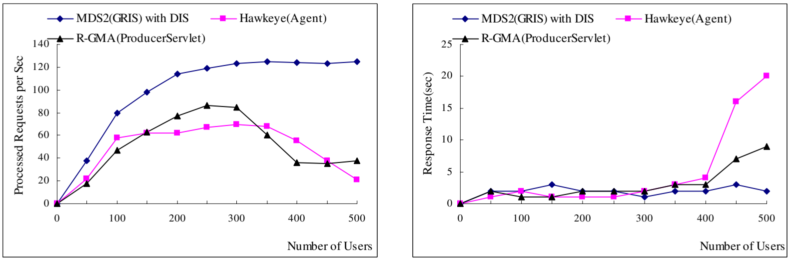



**Fig. 1.** DIS architecture with MDS2

defined with six attributes, in order for metadata of data to represent the information in DIS.

The second step is to create Data Information Provider Module(DIPM) which extracts the specific information from the data. It is a program that represents the data of every host maintained in Virtual Organization(VO). In other words, DIPM reads and represents information of the specific directory in the host. Now only the name and size of data for the information are defined. In this research the aim is to expand DIPM according to the characteristic of science data stored in the host. DIPM operates by reading the name and size of each data, also the count and size of entire data. It is the expansion module of the slapd server using the OpenLDAP generic modules API, operating on the cache memory in the slapd server, and GRIS backend. DIPM is called by the function of *fork()* and *exec()* in GRIS backend and returns LDIF data objects based on the previously defined schema. In other words, it receives single data that consist of *add* and *delete* commands, such as the configuration file by an input. It creates the LDIF data objects, the specific information according to the LDAP schema, and then transmits this to the GRIS backend as an output. As a result, DIPM is a core module in DIS.

The last step is that enables DIS by the modification of the environment configuration. In order to interoperate with MDS2, DIS is identified by MDS2. Some information must be modified to the *grid-info-resource-ldif.conf* file that is a configuration file of MDS2. For interoperation of the proposed system with MDS2, the required information includes a Distinguish Name(DN), objectclass, information of DIPM, arguments, cache time, time limit, size limit, and so on. The relatively short interval time of the cache memory, which is 1,800 seconds, is defined because of the characteristics of the science data that has considerable information movement.




(a) Processed requests (b) Response time

**Fig. 2.** Performance comparison with information servers

### 2.3 Performance Evaluation

The goal of the experiment is to compare the scalability of the existing information systems with the proposed DIS in the grid environment. The experiment is conducted on three different systems. The R-GMA and Hawkeye are selected for the comparison with DIS included in MDS2. The components on these systems are classified because of the heterogeneity in terms of system architecture. At this point, GRIS with DIS in MDS2 is compared to ProducerServlet in R-GMA and Agent in Hawkeye. In this environment, the number of requests processed per second and response time on each server are measured. The number of concurrent users is considered up to 500 users by running individual user processes. The request is the average number of requests processed by a service component per second and the response time is the average amount of time required for a service component to handle a request from a user. The values reported in each experiment are the average over all values recorded over 10 minutes time span. In the Figs. 2(a) and 2(b), the request on GRIS with DIS has a linearly increasing trend and stabilized for the requests processed per second, while both Hawkeye and R-GMA show quite unstable trends and rather slightly fluctuate. In addition, the response time on GRIS is superior to other systems. Especially, the proposed system demonstrates 4 times and 10 times better than R-GMA and Hawkeye, respectively, in terms of response time.

## 3 Conclusion

In this paper, DIS for a data management in distributed environment is developed. It consists of Data Information Provider Module, and creates metadata that indices information for the data. Then it registers metadata regarding the specific information to the previous MDS2. Through the interoperation to MDS2, it keeps compatibility for the user of Globus Toolkit. In our future work, the DIS will be expanded to include the search engine for meteorological data. The system will also be included in the gridsphere for web service. This will allow users to access to the system using a web browser, and search for the required data. Therefore, it will provide the better convenience for users.

## References

1. A. Chervenak, I. Foster and C. Kesselman, “the Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets,” Journal of Network and Computer Applications, vol.23, 187-200, 2001.
2. “DataGrid Information and Monitoring Services Architecture: Design, Requirements and Evaluation Criteria,” Technical Report, DataGrid, 2002.
3. Hawkeye: http://www.cs.wisc.edu/condor/hawkeye
4. K. Czajkowski, S. Fitzgerald, I. Foster and C. Kesselman, “Grid Information Services for Distributed Resource Sharing,” In Proc. 10th IEEE International Symposium on High Performance Distributed Computing(HPDC-10), IEEE Press, 2001.
5. Globus Forum: http://www.globus.org/
6. W. Yeong, T. Howes and S. Kille, “Lightweight Directory Access Protocol,” IETF RFC 1777, March 1995.