Astronomical Data Analysis Software and Systems XIV **P1.1.24**
ASP Conference Series, Vol. XXX, 2005
P. L. Shopbell, M. C. Britton, and R. Ebert, eds.

Japanese Virtual Observatory (JVO) prototype 2

Masahiro TANAKA, Yuji SHIRASAKI, Satoshi HONDA, Yoshihiko MIZUMOTO, Masatoshi OHISHI

National Astronomical Observatory of Japan, 2-21-1 Osawa, Mitaka, Tokyo 181-8588, Japan

Naoki YASUDA

Institute for Cosmic Ray Research, University of Tokyo, 5-1-5 Kashiwanoha, Kashiwa, Chiba, 277-8582, Japan

Yoshifumi MASUNAGA

Ochanomizu University, 2-1-1 Otsuka, Bunkyo, Tokyo, 112-8610, Japan

Yasuhide ISHIHARA, Katsumi ABE, Jumpei TSUTSUMI

Fujitsu Ltd., 1-9-3 Nakase, Mihama, Chiba, 261-8588, Japan

Hiroyuki NAKAMOTO, Yuusuke KOBAYASHI, Tokuo YOSHIDA, Yasuhiro MORITA

Systems Engineering Consultants Co. Ltd., 9-8 Sakuragaoka, Shibuya, Tokyo, 150-0031, Japan

Abstract. We describe the architecture of the Japanese Virtual Observatory (JVO) prototype system version 2. JVO aims at seamless access to astronomical data archives stored in distributed data servers as well as data analysis environment. For this purpose, it is important to establish a framework for access to remote servers, including remote procedure calls (RPCs) and data transfer. A data request for distributed database is described in the JVO Query Language. The JVO system parses the query language, decomposes it into individual remote procedures such as retrieval of catalog, image and spectrum and cross matching, and generate a work flow. Based on this work flow, remote procedures are called. For RPCs of JVO prototype system 1, we employed Globus toolkit 2 (GT2). However, latency time of GT2 RPCs was too long for successive short-time jobs. Therefore, we employed Globus toolkit 3 (GT3) for JVO prototype system 2. As a result, we find that Grid Service in GT3 improves performance of RPC. In addition to Grid Service, Reliable File Transfer (RFT) is used for efficient data transfer. Astronomical data stored in distributed servers are discovered through a registry server which provides metadata discussed in the IVOA registry working group and is built using a XML database.

1. Introduction

JVO[1] is a project of National Astronomical Observatory of Japan. The objective of JVO is to provide astronomers with seamless access to huge data archives produced with astronomical telesopes of Japan such as Subaru telescope. JVO is also participating in IVOA[2] (International Virtual Observatory Alliance) to discuss standard protocols for the purpose of accessibility to astronomical data archives from/to the world.

We have developed JVO prototypes iteratively; the JVO prototype version 1 (Proto 1) was developed in 2002, the prototype version 2 (Proto 2) in 2003, and the prototype 3 is under development. In the course of Proto 1 development, we defined JVO Query Language (JVOQL; Mizumoto et al. 2003) based on the SQL through extending cross match and image retrieval functionalities. And we constructed Proto 1 system which consists of distributed database and analysis servers federated through Grid (Ohishi et al. 2004, Shirasaki et al. 2004). As a result, we confirmed that Proto 1 accepts properly query commands described in JVOQL, and actually operates as a distributed database. However, we found several problems, including the performance of remote execution and data discovery. These issues are improved in JVO Proto 2 as described in this paper.

2. System Design

The system configuration of JVO Proto 2 is shown in Figure 1, which is basically same as Proto 1. Technical components adopted for JVO prototypes are compared in Table 1, and described in the following sections.

Table 1. Technical components adopted for JVO prototypes

	Registry	Grid middleware	remote execution	data transfer
Proto 1	UDDI	Globus v.2	globus-job-run	GridFTP
Proto 2	XML DB	Globus v.3	Grid Service	RFT & SFS

3. Execution Controller

User's query commands described by JVOQL are processed by the **Controller**. The Controller cooperates with the following subcomponents; **JVOQL Parser**, **Scheduler** and **Executer**. These components are written in Java language and installed in the portal server. The JVOQL parser analyzes the syntax of the JVOQL command and decomposes it into query elements. The Java code to parse the syntax of the JVOQL is generated with JavaCC (Java Compiler Compiler). The Scheduler receives an output of the parser, and generates a work flow which consists of job elements for individual servers. Based on this

[1] `http://jvo.nao.ac.jp/`

[2] `http://www.ivoa.net/`

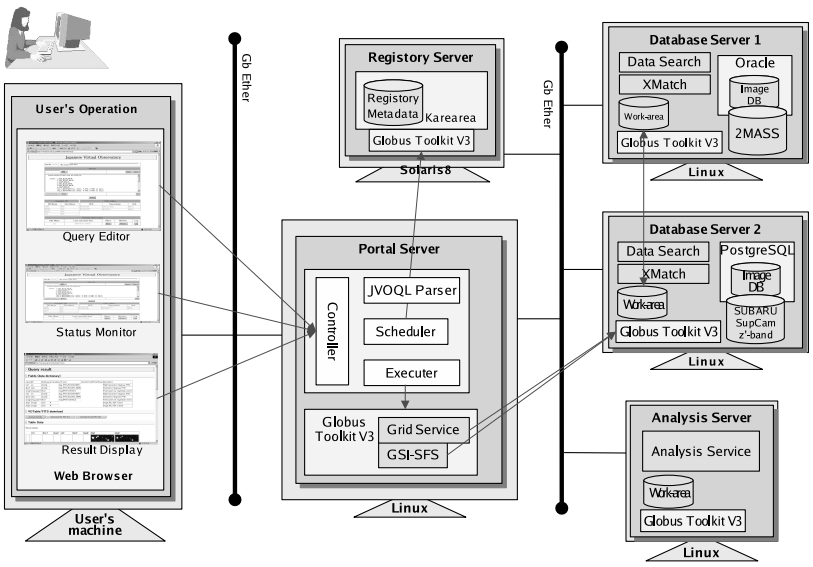

Figure 1. JVO system configuration

work flow, the Controller calls the Executer which calls procedures on remote machines.

4. Data Discovery

When the Controller performs a query for distributed data servers, it is necessary to find the location of the requested data. Such information is retrieved from a Registry server. For Proto 1, we constructed a registry server using UDDI (Universal Description, Discovery, and Integration). UDDI is designed to find services in the Web Services framework, however, UDDI is not necessarily suitable for search services of complicated astronomical metadata. Therefore, we redesigned the registry system for Proto 2. The contents of metadata are defined so as to keep compatibility with the metadata standard proposed in IVOA. We stored these metadata into an XML database product KAREAREA, and enabled metadata search by XPath. We plan to use the OAI-PMH (Open Archives Initiative Protocol for Metadata Harvesting) for exchanging metadata between data servers and a registry server.

5. Remote Execution

Standard protocols for remote procedure calls (RPCs) are necessary for federation of astronomical databases distributed all over the world. For this purpose, we considered to apply standard Grid middleware, **Globus Toolkit**, developed by Globus Alliance[3], and investigated its applicability to JVO. We constructed

[3] `http://www.globus.org/`

Proto 1 using Globus Toolkit version 2 (GT2). However, the result showed that elapsed time to execute a basic JVOQL example was more than 10 minutes (Ohishi et al. 2004). This is probably because the `globus-job-run` command used for Proto 1 is not designed for a series of light-weighted, i.e., pseudo-realtime procedure calls. To improve the performance, we reimplement RPCs of JVO Proto 2 using **Grid Service** introduced in Globus Toolkit 3 (GT3). The Grid Service is based on the Web Services. After the implementation into Proto 2, we examined two queries which take 2.3 and 13 seconds as actual processing times in remote servers, respectively. The result is that elapsed times including RPCs are 2.8 and 16 seconds, respectively. The measured overhead time is only around 30 ms. Thus the performance of Proto 2 is much improved compared with Proto 1. This result shows that Proto 2 system using Grid Service can be a basis of practical virtual observatory systems.

6. Data Transfer

We employed two data transfer protocols for JVO Proto 2; one is the Reliable File Transfer (RFT), and the other is the Self-certifying File System (SFS)[4]. The RFT is one of the GT3 services, and it provides interfaces for recoverable file transfer using the GridFTP protocol. The RFT enables a portal server to issue file copy commands between two remote hosts. This function is useful for multi-server operation like cross match (XMatch). The SFS is a secure network file system over the Internet. We find the SFS does not provide data transfer service between remote hosts, which is a benefit of the RFT, since SFS server and client processes cannot coexist on a single machine.

As mentioned above, we adopted Grid Service for remote execution, and RFT and SFS for data transfer. However, the use of different protocols brings complicated implementation. Furthermore, IVOA standard protocols like SIAP (Simple Image Access Protocol) are implemented with single protocols like HTTP or Web Services. Therefore, we are considering to utilize HTTP or Web Services for RPCs and data transfer in the next prototype and operational JVO systems.

Acknowledgments. This work was supported by the JSPS Core-to-Core Program and Grant-in-aid "Information Science" (15017289 and 16016292) carried out by the Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

Mizumoto, Y., et al. 2003, in ASP Conf. Ser., Vol. 295, ADASS XII, ed. H. E. Payne, R. I. Jedrzejewski, & R. N. Hook (San Francisco: ASP), 96

Ohishi, M., et al. 2004, in ASP Conf. Ser., Vol. 314, ADASS XII, ed. F. Ochsenbein, M. Allen, & D. Egret (San Francisco: ASP), 296

Shirasaki, Y., et al. 2004, in ASP Conf. Ser., Vol. 314, ADASS XII, ed. F. Ochsenbein, M. Allen, & D. Egret (San Francisco: ASP), 46

[4] `http://www.fs.net/`