

Comparison of Data Grid Solutions

データグリッドソリューションの比較検討

Abstract

The Grid is an infrastructure that enables the sharing of geographically dispersed computing resources. It has been used for many large-scale scientific projects with new business applications on the horizon. Many of these applications utilizing the Grid are expected to generate enormous amount of data, and there is an imminent need for enhanced mechanisms for utilizing the necessary distributed storage resources. The Data Grid is a middleware solution that addresses the issues particular to operating very large, heterogeneous, distributed storage facilities. In this paper, we identify functional requirements for the Data Grid and examine and compare three Data Grid solutions: Globus Toolkit, Avaki Data Grid, and the Storage Resource Broker.

あらまし

地理的に離れた計算資源を有効に活用するグリッドは、多くの大規模科学プロジェクトで用いられてきたが、今後は新しいビジネス応用が期待される。グリッドを利用する多くのアプリケーションでは、膨大なデータを生成するため、様々なストレージ資源をうまく活用する機構がどうしても必要となる。データグリッドはこのような要求のほか、巨大で不均質で分散したストレージ設備を運用しなければならないといった要求に応えようというミドルウェアである。

本稿では、データグリッドに必要な機能要件を明らかにし、既存の三つのデータグリッドソリューションであるGlobus Toolkit、Avaki、SRBを取り上げ、比較検討する。



Kazuhiro Matsuo
SVP and GM at Fujitsu Laboratories
of America.



Sung Lee
Researcher at Fujitsu Laboratories
of America.



Jonathan Agre
Director of the Pervasive Computing
Department at Fujitsu Laboratories
of America.

1. Introduction

In recent years, tremendous amounts of data are generated daily in scientific fields such as astronomy, physics, bioinformatics, and environmental science.¹⁾ Therefore, existing and planned applications, enabled by advancements in technologies in communication networks, measurement instruments, multimedia and developments in the grid technology itself, are expected to handle those data. The Data Grid refers to a portion of a grid that is designed to solve issues related to sharing storage resources for a large amount of data. The data grid solution is also strongly expected for application areas of business and government.

Astronomy applications such as virtual observatories including the National Virtual Observatory (NVO) (500 terabytes of data per year beginning in 2004)²⁾ in the US and the AstroGrid (petabyte-scale taped storage per survey) in the United Kingdom³⁾ are expected to produce large volumes of data as they are deployed. The Visible and Infrared Survey Telescope for Astronomy (VISTA),⁴⁾ scheduled to go operational in early 2006 and expected to complete the first survey in 12 years, will produce 250 gigabytes of raw data per night and 10 terabytes of stored data per year.

In bioinformatics fields, the Protein Data Bank,⁵⁾ a database of 3D protein structures, houses 20,000 entries and adds approximately 2,000 new structures every 12 months. Swiss-Prot⁶⁾ and TrEMBL,⁷⁾ two protein databases at the European Bioinformatics Institute,⁸⁾ maintain annotated protein sequences. The sizes of these protein databases are relatively small (on the order of a few gigabytes); however, there is an ongoing effort of interconnecting them^{9),10)} enabling international and interdisciplinary collaboration.

Environmental science applications are similarly expected to generate large data sets. The Earth Resource Observation Systems Data Center,¹¹⁾ a distributed archive center for NASA¹²⁾ housing remotely sensed land data, expects its data volume to reach 3 petabytes by 2005. The Global Land

Coverage Facility (GLCF),¹³⁾ which distributes the Earth science data including satellite images of land cover and coastal marsh health data, currently holds more than 8 terabytes of raw data. NASA expects its total data holdings to rise to 15 petabytes by 2007.

Other data grid applications include business and government solutions for data archiving and document management.¹⁴⁾ By allowing business to decouple the physical management of documents from their logical view, maintenance tasks such as adding more storage, upgrading systems, and migrating data can be done without affecting end users. Transana,¹⁵⁾ a tool for the transcription and qualitative analysis of audio and video data, currently stores more than 700 hours of data in MPEG-1 format (650 MB/h) and is in beta phase for MPEG-2 format (2 GB/h). The National Archives & Records Administration (NARA)¹⁶⁾ is also in the process of setting up a distributed persistent archive, which currently houses 1.3 terabytes at one of its three data storage centers and is expected to grow significantly. Government applications can also benefit from the data grid by enabling secure sharing of data among different agencies using existing systems.

As shown in the present issue, most activities on grid at Fujitsu are in the computational field. Our activities on data grid are just starting up as the market demand emerges. In this paper, we introduce data grid technologies and compare three different core technology solutions for the data grid: Globus Toolkit,¹⁷⁾ Avaki Data Grid,¹⁸⁾ and the Storage Resource Broker.¹⁹⁾ These three core technologies are used in various projects mentioned above. The Globus Toolkit has been deployed in the European DataGrid²⁰⁾2020 and the NASA Information Power Grid²¹⁾ among many other projects. The NVO, the GLCF, the Transana, and the NARA archive projects are implemented with the SRB as their core technology. The Avaki Data Grid has been used in many life science applications. In the following section, we present functional requirements of a data grid system. In Section 3, we have a short summary of each core technology and provide a comparison in

Section 4. Lastly, we present a vision for the data grid field along with a summary.

2. Functional Requirements in Data Grid Technologies

Grid computing has, in general, four-layer architecture.²²⁾ Physical computing, storage, and network resources form the bottom layer called the Fabric layer. The Resource and Connectivity Protocols (RCP) layer above the Fabric layer enables sharing of resources and secure exchange of data among resources. Protocols and APIs (Application Program Interfaces) in this layer are considered core technologies.²³⁾ The Collective layer above the RCP layer implements interactions across collections of resources such as directory services, monitoring and diagnostic services, and access control policy services. The top layer is the user applications layer. Here we focus on the core technologies in the RCP layer.

2.1 Requirements of general grid systems

In general grid systems, requirements such as resource discovery, failure recovery, security, etc, must be satisfied. In addition to those, the data grid system must handle issues such as data organization (global naming, location transparency, and hierarchical organization), a wide variety of data types (files, database objects, URLs, etc) and storage media types {disks, tape farms, Storage Area Networks (SANs), Network Attached Storage (NAS) etc}, data access control and transfer protocols, and an efficient search capability over large volumes of heterogeneous data items.

Fault management in the data grid typically involves replicating data items and providing management functions to maintain consistency.

One main difference between data grid and distributed storage systems (such as the Distributed File System)^{24),25)} is the management and control of the distributed resources. The data grid must provide a solution that is applicable across several administrative domains. In distributed storage systems all resources are typically managed and

administered within a single domain, for instance under a single enterprise network, greatly simplifying many of the key data grid concerns such as data access control.

2.2 OGSA Requirements

Significant standards development activity has been started in the grid computing area. The Global Grid Forum (GGF)²⁶⁾ is a forum to support the development and deployment of Grid technologies and applications, and promote the standardization of such technologies. The Open Grid Services Architecture (OGSA),²⁷⁾ proposed by GGF, identifies the functional requirements for general grid systems.

Heterogeneous system components and configurations (devices, platforms, etc) must be handled to ensure proper sharing of resources across multiple administrative domains with varying security mechanisms and policies. The following functions are necessary in a data grid system.

- 1) Security
 - Authentication, authorization, privacy, and integrity
 - 2) Resource management
 - Dynamic resource provisioning, load balancing, and service level agreement (SLA)
 - 3) Failure recovery
 - Availability of resources
 - Management and sharing of data is fundamental in data grid system. The following functions are required to manage data.
 - 1) Data access
 - Discovery, access, and manipulation of different types of data (files, database, objects, multimedia data)
 - 2) Data replication and or caching
 - Meeting performance and availability objectives
 - 3) Metadata catalog and service
 - Attribute-based searching
 - 4) Schema transformation
 - Data transformation from one schema to another
- In Reference 28), the authors proposed a set of basic services that provide data transfer, storage, and

management in the context of OGSA.

3. Data Grid Solutions

Three existing technologies offering solutions for data grid systems are considered for comparison. The summary of each technology is given in this section, and the comparison is given in the following section.

3.1 Globus Toolkit

The Globus Toolkit is the open-source grid middleware solution from the Globus Alliance¹⁷⁾ which researches and develops the fundamental technologies necessary for Grid systems. The Globus Toolkit provides mechanisms for security, resource management, information access, and data management.

The Globus Toolkit uses the Grid Security Infrastructure (GSI), based on encryption, X.509 certificates, and the Secure Sockets Layer (SSL), to provide secure authentication and data transmission. The communication between two entities is first mutually authenticated using certificates and a third party certificate authority. Once authenticated, the communication can be optionally encrypted to provide data confidentiality. The delegation capability provided in GSI enables a single sign-on capability within the grid.

The Metadata Catalog Service (MCS) provides the publication service, which is the process of making data sets and their associated attributes available to a user community. The MCS also provides the discovery service, which is the process of identifying data items of interest to the user. The MCS is implemented with an Apache Web service front and a MySQL relational database backend.

The GridFTP, Reliable File Transfer (RFT), and Replica Location Service (RLS) provide data management services in the Globus Toolkit. The GridFTP protocol, based on FTP (File Transfer Protocol), is a high-performance, secure, and reliable data transfer protocol, optimized for high-bandwidth wide-area networks. The features of GridFTP

include GSI and Kerberos support, parallel transfers using multiple TCP (Transmission Control Protocol) streams in parallel, even between the same source and destination, partial file transfers, 3rd party (direct server-to-server) transfers, striped data transfer, authenticated reusable data channels, and command pipelining. The GridFTP can also support both manual and automatically negotiated TCP buffer and window sizes for optimal performance.

The RFT, an OGSA based service, provides interfaces for controlling and monitoring 3rd party file transfer using GridFTP servers. The RFT service uses PostgreSQL to maintain soft-state information about connection to provide reliable and recoverable data transfer.

The RLS maps a globally unique logical name to physical data items. Intended to be a part of a data replication management service, it by itself does not guarantee replicated data consistency or filename uniqueness. The RLS is composed of Replica Location Indices (RLIs) and Local Replica Catalog (LRCs). The LRC maintains the mapping between the logical file name and one or more physical file names of replicas at its local storage system. The selection among multiple replicas may be based on performance estimate (e.g., transfer estimates based on the network configuration and conditions), thus providing the load balancing. The RLIs, which supports user queries, receive the state information from LRCs periodically. Currently, the prototype is available as an alpha release for testing and evaluation.

In a typical data access scenario, the MCS, the RLS, and the GridFTP are used. First, a user locates data items of interest by querying the MCS with metadata attributes. The MCS returns a list of logical names matching the attributes. The user then queries the RLS to obtain physical locations of the corresponding data items and accesses the data via the GridFTP (**Figure 1**).

3.2 Avaki Data Grid

The Avaki Data Grid solution^{18),29)} is based on Legion³⁰⁾330, an object-based system designed for

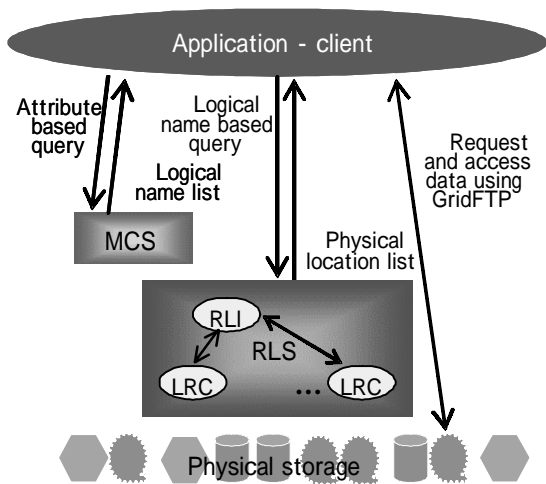


Figure 1
Data access in Globus.

providing a virtual workspace by making heterogeneous, distributed resources available to end users. The Avaki Data Grid allows users to securely access distributed data without knowing the physical location of the data. The data is made available by their owners via a single catalog on the Avaki server. Retrieved data can be integrated, aggregated, and transformed to suit the end user's need.

The Avaki Data Grid uses a three-level global naming scheme to uniquely identify all system components. Human-friendly names, most often path name (/home/user/directory/filename) are at the top-level. The top-level name is mapped to a unique location-independent identifier called object identifier, which includes an RSA (Rivest, Shamir and Adleman) public key as a part of the identifier. The inclusion of an RSA public key enables authentication and secure communication of data objects without a third party. This secure naming protocol³¹⁾ has been proposed to GGF as an open standard for naming in the grids. The location-independent object identifier is then mapped to a location specific current address, called object address (OA). Currently, two types of addresses are supported: TCP and UDP (User Datagram Protocol). The OA of a data item may change over time, which allows migration of the data even while being used.

The authentication, authorization, and data

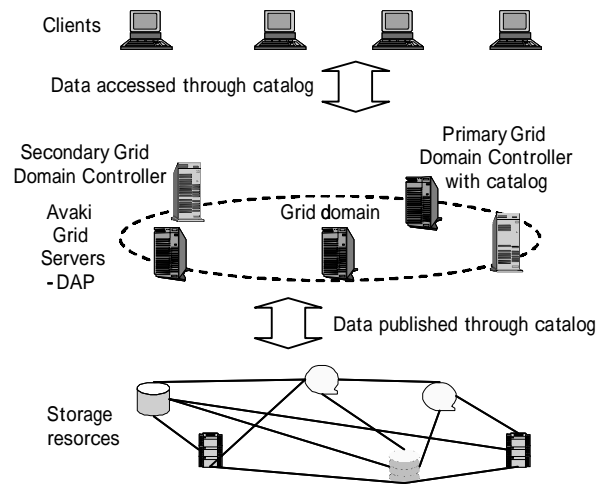


Figure 2
Avaki Grid domain.

integrity and confidentiality mechanisms in the Avaki Data Grid support a grid that is comprised of many organizations with different policies. For user authentication, a login using a user name and a password, Kerberos, Network Information Service (NIS), Lightweight Directory Access Protocol (LDAP), RSA SecurID, etc., can be used. Avaki employs an Access Control List (ACL) policy to implement protection. For each action on a data item, the "allow" list and the "deny" list can create arbitrary sharing policies. Using appropriately created ACLs, one can form various groups assigning the same privilege to members. In order to protect data during transmission, Avaki offers three modes to transmit data: private mode using OpenSSL³²⁾ encryption, protected mode with checksum, and clear text mode.

A data grid domain is composed of a set of Avaki Grid Servers (**Figure 2**), available for most UNIX-variants, as well as Microsoft Windows platforms. The Grid Domain Controller (GDC), one of servers, handles logins and directory navigation services through the catalog. Failover is handled by designating a secondary GDC, which is a replica of the primary GDC. However, the secondary GDC is read-only. Therefore, no new users or servers can be added while the primary one is out of service. The Avaki Grid Servers allow data access through an NFS (Network File System) style server called a Data

Access Point (DAP). The DAP allows standard-based, secure remote data access through an NFS-style interface.

Avaki keeps a single copy of data at the owner's local site to avoid handling replica management. It employs a distributed cache to improve performance over a wide area and make data available in the event that the original data is not available. The cache items are kept by the Avaki Grid Servers, based on various caching options for each data, specified by each data owner.

Reverse proxy caching keeps frequently-requested results near the data source, while forward proxy caching keeps data closer to the end users to reduce network congestion and improve application performance. Caches can be loaded during off hours, and data can be cached on-demand or scheduled. On-demand caching fetches data upon a user request and allows block transfer. Scheduled caching forces a data refresh at specified intervals and the entire data is cached. The owner also specifies the coherency duration (how long the cached data is considered valid) for each data item and each cache. It is critical to select proper cache options in order to maintain data coherency among cached items.

Data is made available when the owner publishes the data in the unified catalog, which maintains the tree-structured view of data items in the data grid. Both system- and application-level metadata is maintained in a DB and can be searched. Avaki also uses metadata to convert relational databases to XML (eXtensible Markup Language) files. Metadata such as column names and data types become a part of the converted data file.

The system currently handles the following types of data: relational database data (results of SQL (Structured Query Language) statements and stored procedures), URL(Universal Resource Locator)-based data sources such as servlets, CGI (Common Gateway Interface) scripts, or Web services, XML documents, XSL (eXtensible Stylesheet Language) style sheets, and files {in direct-attached storage, network-attached storage (NAS), or storage area network (SAN)

storage}.

Data access is handled via standard methods. Users access data through standard file system protocols or Web interfaces, and applications access data through standard methods such as Open Database Connectivity (ODBC), Java Database Connectivity (JDBC), Web Services/Simple Object Access Protocol (SOAP), file read, and Java Server Pages (JSP) Standard Tag Library (JSTL). Thus JDBC 2.0-compliant databases such as Oracle, DB2, Microsoft SQL Server, Sybase SQL Server, and MySQL are all supported.

Since each grid domain relies on one catalog for data access, this approach can cause a performance bottleneck at times. Although Avaki handles a failure by dedicating a secondary GDC as a replica of the primary one, the secondary controller can be outdated based on the synchronization interval. In addition, Avaki does not support replication of data items and instead uses cached items when the data item is not available anymore. This presents reduced data availability and possibly leads to inconsistent data items.

3.3 Storage Resource Broker (SRB)

The SRB¹⁹⁾ developed at the San Diego Supercomputer Center (SDSC), is one of the first middleware solutions providing uniform access to distributed, diverse storage devices in a heterogeneous computing environment. The SRB offers a transparent view of data organization by providing a global logical name space and a logical grouping regardless of physical data locations. Each data item in the grid is assigned a globally unique logical identifier. When a data item is ingested into the grid, the human-friendly name of the data is mapped to the globally unique identifier. This mapping information and the physical location along with other metadata are stored and maintained in the metadata catalog service called the MCAT (Metadata CATalog) server.

SRB supports both the GSI and authentication and encryption schemes developed by SDSC based on RSA encryption algorithms. It features a replica

management system which allows both automatic and manual data replication and provides a replica synchronization mechanism.

Another feature of the SRB is the container abstraction, which is suitable for storing a large number of small files without a performance penalty. The SRB also provides tools to monitor and log all activities on the server and audit the usage of the data sets. The SRB has a mass storage hierarchy capability, as data is staged at disk caches and migrated to tape drives as more disk cache space is needed.

The SRB architecture, based on a client/server model, includes three components: SRB clients, SRB servers, and the MCAT server (**Figure 3**). The MCAT server manages metadata associated with data objects, users, and resources in a relational database (currently DB2 or Oracle) to catalog a large number of data objects, enable searching and analysis of a large sized data collections, and control access to data sets. Both system and a limited set of user-defined attributes (2 integers and 10 strings) can be associated with each data object. Additionally, the MCAT server includes a schema converter which allows application specific metadata schema to be developed, if necessary. The MAPS (Metadata Attribute Presentation Structure) is the interface used to communicate with MCAT.

An SRB server fulfills client requests by communicating with the MCAT server and accessing storage devices. SRB server implementations are available for AIX, Sun Solaris (both 32 and 64 bits addresses), Linux (both 32 and 64 bits addresses), SunOS, DEC OSF (Open Software Foundation), SGI, Mac OS X, and the Cray C90. The SRB currently handles UNIX file systems, archival storage systems such as UniTree and HPSS, and database objects such as DB2 and Oracle. In addition to performing I/O operations on data sets, clients can manipulate and query the metadata associated with the data.

The SRB currently supports two types of drivers for I/O operations: the file-type and the DB-type. The file-type driver provides UNIX-like file I/O

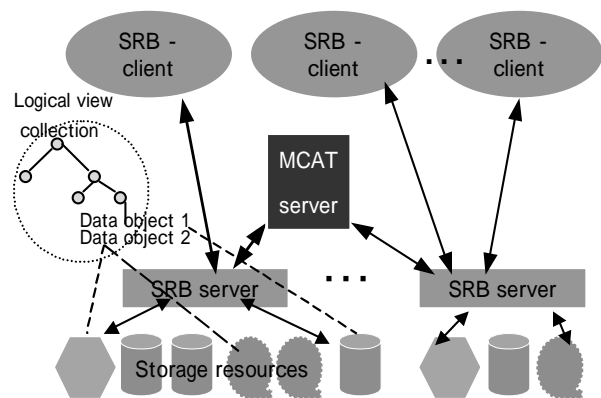


Figure 3
SRB architecture.

interface to UNIX file system, UniTree, HPSS, FTP, and HTTP (HyperText Transfer Protocol). It also supports the following operations; create, open, close, unlink, read, write, sync, seek, stat, chmod, mkdir, opendir, closedir, and readdir. The DB-type driver provides an I/O interface to DBMS, and the supported operations are create, open, close, unlink, read, write, and seek. More advanced DB operations such as select and join operations are supported through shadow objects, which are mechanisms to interact with non-file type data. The server-driven parallel I/O capability improves the performance of the system.

SRB client applications are provided with a set of APIs to communicate with SRB servers. In addition, three GUI (Graphical User Interface) based clients are provided. The srbBrowser is a JAVA-based SRB client. The inQ for MS Windows platforms is implemented with the client API, the SRB C++ wrapper, and the MFC (Microsoft Foundation Class) library. MySRB is a Web browser-based client.

The SRB provides a replica synchronization mechanism. However, without a conscious synchronization effort, the replicated data items could easily become inconsistent. The SRB does not provide any intelligent mechanism such as a load balancing scheme to select data among replicated items. If the original data is available and the user does not specify which data to select, then it is always the original one that is selected. If the original data

is not available, then one of the replica items is selected. The user can also specify which one to select among many replicas.

4. Comparison

All three solutions address some of the issues and requirements for data grid systems. Depending on one's application needs, each one has its advantages (**Table 1**). For data-intensive applications, the Avaki Data Grid and the SRB offer more stable solutions. However, the Globus Alliance is a major driving force in the GGF towards standardizing Grid solutions. The Globus Toolkit, though less mature, is striving to provide infrastructure and tool support for data-intensive applications and is in the process of adding many features similar to those found in the SRB.

The SRB provides a global name space to ensure uniqueness of file names across the system.

Similarly, the Avaki Data Grid employs its object identifiers in a three-level naming scheme to guarantee filename uniqueness.

The Avaki Data Grid, advertised as an Enterprise Information Integration solution, offers better support for integrating and aggregating database data that involves a sequence of data accesses and transformations through database operations or conversions into XML, etc. The SRB supports DBMS through the DB-type driver, which provides a set of primitive operations. Unlike the Avaki Data Grid where the more advanced DB operations (select, update, etc) are supported through standard access methods, the SRB uses the less convenient shadow objects to support those operations.

The SRB, however, supports the variety of large storage system such as HPSS, which is an advantage of the SRB over the Avaki.

Grid systems often span many organizations, and

Table 1
Core technology comparison of data grid solutions.

Core technology	Globus Toolkit	Avaki Data Grid	SRB
Architecture	Client-server	Client-server	Client-server
OS	UNIX, Microsoft Windows support under development	UNIX, Microsoft Windows	UNIX, Mac OS X, Cray C90
Storage media	UNIX file systems	File systems, databases	UNIX file systems, archival storage systems (UNITREE, HPSS), Database objects (DB2, Oracle), URL
Metadata	System and application metadata	System and application metadata Convert relational database to XML	System and limited application metadata
DBMS support	No support	Support for manipulating database – results of SQL statements and stored procedures	Handles database as an object, but little support for data manipulation
Security			
Authentication	GSI	Integrated directory service for user authentication	GSI and authentication developed at SDSC
Replica Management^{note)}			
Selection	Based on transfer performance estimates	Cached data may be used	Original data unless specified otherwise, or unavailable
Load balancing	Based on transfer performance estimates	Cached data may be used	None
Consistency	No guarantees	Based on caching options and coherency duration	Manual synchronization only
Failure Recovery			
Metadata availability	Not specified	Secondary read-only available	Single metadata catalog
Data availability	Supported through replica	Supported through cache, but no guarantees	Supported through replication
Reliable transfer	Reliable and restartable data transfer	Not specified	Not specified

Note) Avaki Data Grid does not have a replica management system, but it uses caching to address similar issues.

it is desired that a single login would suffice for all security requirements among these administrative domains. All three systems offer single sign-on features; specifically both Globus Toolkit and SRB authentications are based on GSI.

The latest version of the Globus Toolkit (GT3) is based on OGSA and its implementation is OGSI-compliant with the exception of GridFTP. However, an OGSI-compliant GridFTP is expected in the future. The Avaki Data Grid supports XML, SOAP (Simple Object Access Protocol), and WSDL (Web Services Definition Language); thus a thin layer of OGSI-compliant interface can be implemented. The SRB is in the process of adding support for XML, SOAP, and WSDL.

5. Research Issues

A persistent system that allows sharing of data must keep data available regardless of the state of the resource where the original data item is stored. As shown in all three systems examined in this paper, replica management is an important issue in dealing with faulty system components. When data is replicated, one of the main issues is maintaining its consistency. Various options have been employed, but further work is needed.

When data is shared among geographically distributed clients, it may be necessary to provide mechanisms to control the quality of service. The GridFTP is optimized for high-bandwidth wide-area networks; however, more sophisticated and predictable mechanisms of delivering Quality of Service may be desirable in applications with rigid performance constraints.

The tremendous amount of data expected by the projects makes the efficient use of metadata critical. However, the generation of metadata currently requires human interaction. Although end-users play a crucial role in metadata generation, it is often expensive to create and maintain such metadata. Machine assistance for generating metadata will be highly desirable.

In Reference 23), the authors envisioned a

semantic grid as the next generation of grid computing. As evidenced in standard activities, the grid computing is moving towards a service-oriented architecture. The application of the semantic Web technology to a service-oriented model has proven quite effective and efficient in accomplishing end user's tasks.^{33),34)} The semantic Web can be applied to grid computing to achieve a similar outcome, reducing effort in creating applications on the grid.

6. Conclusion

Each of the three grid solutions examined in this paper addresses many issues in the data grid systems. However, currently no single solution is a clear winner in addressing all of its issues and requirements. With emerging standards in the Grid, it is also desirable to have standard-compliant interfaces for these data management services and this remains an ongoing process for these systems.

This paper only covered three of the emerging middleware solutions for data grids. Recently, Oracle³⁵⁾ announced Oracle 10g,³⁶⁾ Oracle's grid infrastructure software. There are other emerging technologies that can be applied in data grids such as UNICORE,³⁷⁾ United Devices,³⁸⁾ and peer-to-peer storage systems.^{39),40)} We do not include them in this comparison and it remains in our future work.

Future business opportunities may exist for solutions that can efficiently handle the large data sets that will be generated by corporations as they capture more and more multimedia information for detailed record keeping about their operations, their interactions with customers and their planning activities. Along with the storage needs, multinational corporate customers will need highly available, efficient, and accessible services to effectively utilize this global data store.

With Fujitsu's wide variety of customer base including top science research institutes, large government agencies, and global business organizations, it is anticipated that there will be growing demands for data grid solutions, inevitably generating a wide variety of new and challenging

requirements. This comparison of typical data grid technologies will aid customers and system engineers to come up with solutions that will suit each customer's needs and requirements.

References

- 1) F. Berman et al.: Grid Computing – Making the Global Infrastructure a Reality. Chapter 36, Chichester: John Wiley & Sons, 2003.
- 2) US National Virtual Observatory.
<http://www.us-vo.org>
- 3) Astro Grid.
<http://www.astrogrid.ac.uk>
- 4) Visible and Infrared Telescope for Astronomy.
<http://www.vista.ac.uk>
- 5) Research Collaboratory for Structural Bioinformatics Protein Data Bank.
<http://www.rcsb.org/pdb>
- 6) European Bioinformatics Institute SWISS-PROT.
<http://www.ebi.ac.uk/swissprot/>
- 7) European Bioinformatics Institute TrEMBL.
<http://www.ebi.ac.uk/trembl/>
- 8) European Bioinformatics Institute.
<http://www.ebi.ac.uk/>
- 9) Worldwide Protein Data Bank.
<http://www.wwpdb.org/index.html>
- 10) Human Mitochondrial Protein Database.
<http://bioinfo.nist.gov:8080/examples/servlets/index.html>
- 11) Earth Resources Observation Systems (EROS) Data Center.
<http://edc.usgs.gov/>
- 12) National Aeronautics and Space Administration.
<http://www.nasa.gov/>
- 13) Global Land Cover Facility.
<http://glcf.umiacs.umd.edu/index.shtml>
- 14) Nirvana Storage.
http://nirvanastorage.com/solutions/solutions_main.htm
- 15) Transana.
<http://www2.wcer.wisc.edu/Transana>
- 16) US National Archives & Records Administration.
<http://www.archives.gov/index.html>
- 17) The Globus Alliance.
<http://www.globus.org>
- 18) Avaki.
<http://www.avaki.com>
- 19) San Diego Supercomputer Center Storage Resource Broker.
<http://www.sdsc.edu/DICE/SRB/index.html>
- 20) The DataGrid Project.
<http://eu-datagrid.web.cern.ch/eu-datagrid/>
- 21) NASA Advanced Supercomputing Information Power Grid.
<http://www.nas.nasa.gov/About/IPG/ipg.html>
- 22) I. Foster: The Grid: A New Infrastructure for 21st Century Science. *Physics Today*, 55, 22, p.42 (2002).
- 23) F. Berman et al.: Grid Computing – Making the Global Infrastructure a Reality. Chapter 3, Chichester: John Wiley & Sons, 2003.
- 24) Coda File System.
<http://www.coda.cs.cmu.edu/>
- 25) Microsoft Distributed File System.
<http://www.microsoft.com/windows2000/techinfo/howitworks/fileandprint/dfsnew.asp>
- 26) Global Grid Forum.
<http://www.ggf.org>
- 27) I. Foster et al.: The Open Grid Services Architecture Platform. draft-ggf-ogsa-platform-2, Feb. 2003.
- 28) F. Berman et al.: Grid Computing – Making the Global Infrastructure a Reality. Chapter 15, Chichester: John Wiley & Sons, 2003.
- 29) F. Berman et al.: Grid Computing – Making the Global Infrastructure a Reality. Chapter 10, Chichester: John Wiley & Sons, 2003.
- 30) Legion.
<http://legion.virginia.edu>
- 31) J. Apgar et al.: Secure Grid Naming Protocol: Draft Specification for Review and Comment.
<http://sourceforge.net/projects/sgnp>
- 32) OpenSSL.
<http://www.openssl.org>
- 33) Masuoka et al.: Ontology-Enabled Pervasive Computing Applications. *IEEE Intelligent Systems*, 18, 5, p.68-72 (2003).
- 34) Masuoka et al.: Task Computing – the Semantic Web meets Pervasive Computing. Proceedings

- of the 2nd International Semantic Web Conference, 2003.
- 35) Oracle.
<http://www.oracle.com/>
- 36) Oracle Grid Computing.
<http://www.oracle.com/solutions/grid/>
- 37) UNICORE Forum.
<http://www.unicore.org/>
- 38) United Devices.
<http://www.ud.com/home.htm>
- 39) Peer-to-peer Archival Storage Facility.
<http://www.research.microsoft.com/~antr/PAST/default.htm>
- 40) The OceanStore Project.
<http://oceanstore.cs.berkeley.edu/>



ま え が き

多くの科学分野では、膨大なデータが生み出されており、それらのデータを迅速に処理しなければならない機会が急速に増大している。このようなニーズに応えるためにグリッドの概念が導入された⁽¹⁾。データグリッドとは、グリッドコンピューティングの中で主にストレージの共有に重点を置いたシステムやソリューションを言う。

データの大きさの例で言えば、天文学の分野では、米国や英国の仮想天文台 (NVO⁽²⁾やAstroGid⁽³⁾) では年にペタバイト程度のデータを取り扱う。バイオインフォマティクス^{(4),(5)} 環境科学⁽⁶⁾などの分野でも同様の巨大データを扱う。NASA⁽⁷⁾やGLCF (Global Land Coverage Facility)⁽⁸⁾などもペタバイトクラスのデータの取扱いが必要となっている。

ビジネス分野や公共サービス分野でも、データアーカイブやデータ管理のために、データグリッドの必要性が高まっている⁽⁹⁾。とくに、ビデオや音声のデータなどマルチメディアデータをアーカイブするニーズが拡大し⁽¹⁰⁾ データグリッドがますます重要になっている。米国の国立公文書館 (NARA)⁽¹¹⁾ が代表例である。

グリッドコンピューティングに関する富士通の活動は本特集で見られるように、科学技術計算分野が中心であるが、データグリッドに関しても、ニーズの立ち上がりを受けて、取組みを開始している。

本稿では、データグリッド技術の機能要件の説明と、既存の3データグリッドソリューションとして Globus Toolkit⁽¹²⁾ Avakiデータグリッド⁽¹³⁾ SRB (Storage Resource Broker)⁽¹⁴⁾の現状と課題を紹介する。Globus ToolkitはヨーロッパのDataGrid⁽¹⁵⁾やNASA⁽¹⁶⁾などで使われている。SRBは、NVO, GLCF, NARAなどで用いられている。Avakiデータグリッドは、多くのライフサイエンスでのアプリケーションがある。

データグリッド技術の機能要件

グリッドのアーキテクチャは、4層構造を成している⁽¹⁷⁾。つまり、物理層であるファブリック層、コア技術であるRCP (Resource and Connectivity Protocols) 層、分散した資源を管理するコレクティブ層、そして、アプリケーション層の4層であ

る。ここでは主にRCP層のコア技術を紹介する⁽¹⁸⁾。データグリッドとしての機能要件

グリッド実現のためには、資源発見、故障回復、セキュリティなどの機能が必ず必要となる。データグリッドではそれらの機能に加えて、ストレージに特有の機能要件が必要である。それらは以下のとおりである。

- (1) データ機構 (グローバルネーミング, ロケーション透明性, 階層的組織化)
- (2) 幅広い種類のデータ型 (ファイル, データベースオブジェクト, URLなど) や異なるストレージメディア型 (ディスク, テープ格納, SAN, NASなど) への対応
- (3) データアクセスや転送プロトコル
- (4) 不均質なデータアイテムの膨大な量に対する効率的な検索能力の実現

データグリッドでは、故障管理のためにデータアイテムのレプリカを作成すること、整合性を保つための管理機能を持つことが一般的である。

単一のドメインを取り扱う分散ストレージシステム^{(19),(20)}と異なり、データグリッドでは異なる機関にまたがるストレージ資源をアクセス制御することが必要なので、そのための諸機能を備えていなければならない。

OGSAでの機能要件

グリッドコンピューティングの分野で標準化活動が積極的に進められている。Global Grid Forum (GGF)⁽²¹⁾が中心となり、Open Grid Services Architecture (OGSA)⁽²²⁾が提案され、グリッド分野に共通の機能要求を定めている。とくにデータグリッドにかかわる機能は以下のとおりである。

- (1) セキュリティ
 - 認証, 信頼付与, プライバシー, 完全性
 - (2) 資源管理
 - 動的な資源提供, 負荷分散, サービスレベルアグリーメント (SLA)
 - (3) 故障回復
 - 資源の利用可能性
- データの管理と共用はデータグリッドシステムでは基本となる機能である。データ管理に関しては、つぎのような機能が要求される。

- (1) データアクセス
 - 異なるタイプのデータ (ファイル, データベース,

オブジェクト, マルチメディアデータ)の発見, アクセス, 操作

(2) データのレプリカ作成とキャッシング

性能確保と利用可能性の両立

(3) メタデータのカatalogとサービス

属性ベースの検索

(4) スキーマ変換

あるスキーマから別のスキーマへの変換

データグリッドソリューション

データグリッドソリューションを現在提供しているGlobus Toolkit, Avakiデータグリッド, SRBの三つの技術について紹介する。

Globus Toolkit

Globus Toolkitは, グリッドに必要な基本的技術を研究開発するGlobus Alliance⁽¹²⁾から提供されるオープンソースのグリッドミドルウェアである。それは, セキュリティ, 資源管理, 情報アクセス, データ管理の諸機能を提供する。

Globus Toolkitは, 暗号化, X.509認証, SSL (Secure Socket Layer) に基づくGrid Security Infrastructure (GSI) を用い, セキュアな認証とデータ転送を実現している。GSIに含まれるデリゲーション機能によって, グリッド内でシングルサインオンが可能となる。

Metadata Catalog Service (MCS) では, データセットとそれに連関した属性を作成してユーザに提供するパブリケーションサービスと, ユーザの興味のあるデータ項目を特定するデスクバリサービスが提供される。MCSはWebフロントにApacheとバックエンドにMySQLリレーショナルデータベースとで作られている。

Globus Toolkitではデータ管理サービスを提供するのはGridFTP, RFT (Reliable File Transfer), RLS (Replica Location Service) である。FTP (File Transfer Protocol) に基づいたGridFTPプロトコルは, 高性能で信頼性のあるデータ転送プロトコルであり, ブロードバンドの広域ネットワークに対して最適化されている。GridFTPの特徴としては, GSIとカーベラス (オープンネットワークのための認証システム) などのセキュリティ機能, 複数の並列TCP (Transmission Control Protocol) ストリームを使った並列転送, 部分ファイル転送,

サードパーティ転送, 細切りのデータ転送, 認証され再利用可能なデータチャネル, 命令パイプラインなどがある。

RFTはOGSAに基づくサービスであり, GridFTPサーバを使って, サードパーティのファイル転送を制御しモニターするインタフェースを提供する。RFTサービスは, PostgreSQLを使用し, 信頼でき, 回復できるデータ転送を実現するための接続情報を保持している。

RLSは, 全体で唯一の論理名を物理的なデータ要素にマップする。データの複製管理サービスの一部ではあるが, 複製されたデータの整合性やファイル名の唯一性を保証しない。RLSは複数のRLI (Replica Location Indices) と複数のLRC (Local Replica Catalog) から成る。LRCはローカルなストレージシステムにある一つかそれ以上の複製の物理的ファイル名に論理ファイル名をマッピングする。ユーザのクエリをサポートするRLIは, LRCから定期的に状態情報を受け取る。現在, 版のプロトタイプが提供されている。

典型的なデータアクセスのシナリオは, MCS, RLSおよびGridFTPを使って構成される。まず, ユーザはメタデータ属性でMCSに問い合わせ, 求めるデータ要素を特定する。MCSは属性にマッチさせ論理名のリストを返す。そしてユーザはRLSに問い合わせ, 対応するデータ要素の物理的な位置を得て, GridFTPでデータをアクセスする (図-1)。

Avakiデータグリッド

Avakiデータグリッド^{(13),(23)}によるソリューションは, バージニア大学で開発したオブジェクトベースシステムLagion⁽²⁴⁾をベースにしている。このシステムではデータの物理的な位置を知らなくても分散されたデータにユーザが安全にアクセスできるようになっている。

Avakiデータグリッドは三層のグローバルネーミングスキーマを使い, すべてのシステムのコンポーネントを特定できる。

Avakiデータグリッドのセキュリティ機能では, 認証, データの完全性, 機密性の機能を持ち, 異なるポリシーを持つ多くの組織体で構成されるグリッドをサポートしている。

Avakiデータグリッドのドメインは, 一群のAvaki グリッドサーバで構成されており (図-2),

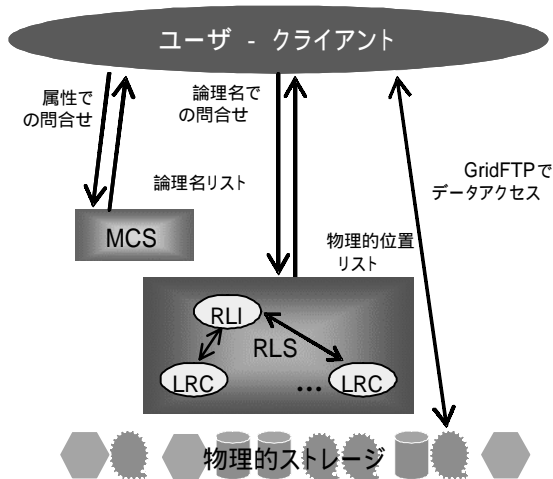


図-1 Globusにおけるデータアクセス
Fig.1-Data access in Globus.

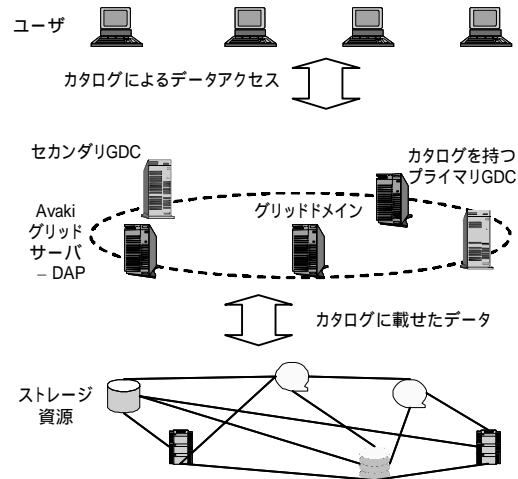


図-2 Avakiデータグリッドのドメイン
Fig.2-Avaki Data Grid domain.

ほとんどのUNIXとMicrosoftのWindowsのプラットフォームで動く。サーバの一つであるGrid Domain Controller (GDC) がカタログによりログインとディレクトリナビゲーションサービスを取り扱う。不都合が生じた場合は、プライマリのGDCのレプリカであるセカンダリのGDCが指定されて対応する。このセカンダリGDCは読み出し専用なので、新しいユーザやサーバの追加などのサービスには対応できない。

Avakiデータグリッドはデータの単一コピーをユーザのローカルサイトに保持しておくことで、レプリカ管理を取り扱わずに済むようにしている。また、分散キャッシュを取り入れて、広いエリアでの性能の改善やオリジナルデータが利用不能の場合でのデータの利用可能性の確保を行っている。フォワードプロキシキャッシング、リバースプロキシキャッシング、オンディマンドキャッシング、スケジュールドキャッシングなどのキャッシュやキャッシュの有効期間を定める一貫性保持機能を導入している。キャッシュされたアイテム間のデータの一貫性を保つためには適切なキャッシュオプションを選ぶことが大切である。

データが利用可能になるのは、ユーザがデータをユニファイドカタログに載せたときである。このカタログはデータグリッドの中でデータアイテムを木構造のビューを保つためのものである。メタデータはシステムレベルもアプリケーションレベルもデータベースに持っており、検索できる。また、リレー

ショナルデータをXML (eXtensible Markup Language) ファイルに変換するのにもメタデータを使う。

現在取り扱えるデータ型は以下のタイプがある。RDB (Relational Data Base) のデータ, URL (Universal Resource Locator) ベースのデータソース { サブレット, CGI (Common Gateway Interface) スクリプト, Webサービスなど }, XMLドキュメント, XSL (eXtensible Stylesheet Language) スタイルシート, { 直接接続されたストレージやNAS (Network Attached Storage) やSAN (Storage Area Network) の } ファイルである。

データのアクセスは通常の方法を採っている。Avakiデータグリッドの問題点としては、各グリッドドメインはデータアクセスに単一カタログに頼っているため、時に性能のボトルネックになる可能性が高い。また、不都合が起きた場合のセカンダリGDCの指名でも、同期の仕方であまり機能しない可能性もある。さらに、データアイテムのレプリカを作らず、キャッシングを使っているため、データの利用可能性が低くなったり、場合によっては、データの不整合性を引き起こしたりする可能性がある。

Storage Resource Broker (SRB)

SRB⁽¹⁴⁾ は San Diego Supercomputing Center (SDSC) で早くから開発されたもので、均一でない計算環境で分散した様々なストレージ装置を統一的にアクセスできるミドルウェアソリューションで

ある。SRBでは、物理的なデータの格納場所にかかわらず、グローバル論理名空間と論理的なグルーピングを与えることにより、データ機構を透明に見せている。

データアイテムが取り込まれると、データの名前がグローバルな論理名空間の固有の識別名にマップされる。このマッピング情報や物理的な格納場所とほかのメタデータとともにMCAT (Metadata Catalog) サーバと呼ばれるメタデータカタログサービスにストアされ、維持される。

SRBのセキュリティ機能では、GSIのサポートに加えて、RSA (Rivest, Sharmir, and Adelman) に基づいてSDSCが独自に開発した認証・暗号スキームもサポートしており、レプリカ管理システムに特徴がある。

SRBの別の特徴としては、コンテナアブストラクションというものがあり、大量の小さなファイルを性能を落とさないでストアできる。SRBは、サーバでのすべての活動をモニターし、ログを取るツールやデータセットの利用を監査できるツールを持っている。さらに、SRBは大量ストレージ階層構造を構築できる。

SRBのアーキテクチャは、クライアント/サーバモデルに基づいており、SRBクライアント、SRBサーバ、MCATサーバの三つのコンポーネントから成っている (図-3)。

SRBは、レプリカ同期化機構を持っている。しかし、しっかりした同期化を行っていないので、レプリカされたデータアイテムが不整合になりやすい。SRBでは、レプリカされたアイテムからデータを選んで負荷バランスを行うスキームなど高度なメカニズムは取り入れられていない。

データグリッドソリューションの比較

前章で述べた三つのデータグリッドソリューションの比較を表-1に示す。この表を見ると、データグリッドシステムに対する問題点や今後の課題がよく分かる。各ソリューションに向いている用途はそれぞれ異なる。データの多い場合には、AvakiデータグリッドやSRBがより安定なソリューションを提供する。Globus Toolkitは現在のところまだ機能が十分そろっていない。推進母体であるGlobus AllianceがGGFで強力に標準化活動を展開してい

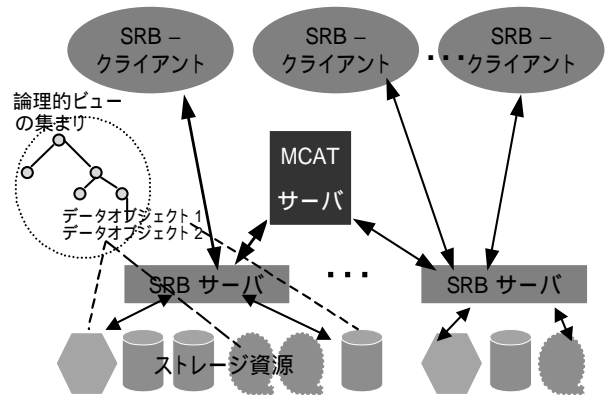


図-3 SRBアーキテクチャ
Fig.3-SRB architecture.

るので、現在の遅れを取り戻すものと期待できる。

SRBはグローバル論理名空間を提供し、システムに横断的にファイル名の唯一性を保証している。Avakiデータグリッドの場合は、三層のネーミングスキームにおいてオブジェクト同定機能を採用している。

Avakiデータグリッドでは、企業情報統合ソリューションとして知られているように、データベースのデータを統合したり、まとめたりする機能をサポートしている。SRBでは、DBMS (Data Base Management System) をDBタイプドライバによってサポートしている。また、HPSS (High Performance Storage System) などの非常に大きなストレージシステムを多種類サポートするという点では、SRBの方がAvakiデータグリッドよりも優れている。

セキュリティ関連では、どのシステムもシングルサインオンの機能を持っている。とくに、Globus ToolkitとSRBがGSIに基づいた認証を行っている。

Globus Toolkitの最新版 (GT3) では、GridFTP以外はOGSAに基づいている。AvakiデータグリッドはXML, SOAP (Simple Object Access Protocol), WSDL (Web Service Definition Language) をサポートするので、OGSA準拠のインタフェースを実装できる。SRBでは、これらのサポートを追加する作業が現在行われている。

技術課題

データを共有できるシステムでは、オリジナルなデータがストアされているストレージ資源の状態が

表-1 データグリッドソリューションの比較

コア技術	Globus Toolkit	Avakiデータグリッド	SRB
アーキテクチャ	クライアント/サーバ	クライアント/サーバ	クライアント/サーバ
OS	UNIX MS Windows (開発中)	UNIX, MS Windows	UNIX, Mac OS X, Cray C90
ストレージメディア	UNIX ファイルシステム	ファイルシステム, データベース	UNIX ファイルシステム, アーカイブストレージシステム (UNITREE, HPSS), DBオブジェクト (DB2, Oracle), URL
メタデータ	システムとアプリケーションのメタデータ	システムとアプリケーションのメタデータ, RDBのXMLDBへの変換	システムと限られたアプリケーションのメタデータ
DBMSサポート	サポートなし	DB操作のサポート - SQLステートメント, ストアドプロシジャ	DBをオブジェクトとして扱う データ操作のサポートは少ない
セキュリティ			
認証	GSI	ユーザ認証の統一されたディレクトリサービス	GSIとSDSC開発の認証
レプリカ管理			
セレクション	転送性能の推定をベースにする	キャッシュされたデータを利用	とくに指定しないときは、オリジナルのデータ、または使えない場合もあり
負荷分散	転送性能の推定をベースにする	キャッシュされたデータを利用	なし
整合性	保証なし	キャッシュオプションと一貫性保持機能による	マニュアルでの同期化のみ
故障回復			
メタデータ利用可能性	記述なし	セカンダリに読出し専用で利用可能	単一メタデータカタログ
データ利用可能性	レプリカによるサポート	キャッシュによるサポート、ただ保証はない	レプリカを使ってサポート
信頼できる転送	信頼でき、再起動できるデータ転送	記述なし	記述なし

: Avakiデータグリッドには、レプリカ管理システムがないが、キャッシングによって対応している。

どうであれ、データが使えるようになっていなければならない。本稿で比べたデータグリッドソリューションでは、故障したシステム部品に対処するため、レプリカ管理が重要な問題である。データを複製するときに、その整合性を取ることが重要な課題の一つであり、まだ改善の余地が大きい。

ストレージ資源が地理的に分散している場合には、QoS (Quality of Service) を確保する問題も重要である。GridFTPでは限定された環境でその機能を備えているが、もっと一般的な環境で厳しい制限の場合にも高いQoSを確保するのは課題である。

膨大なデータを取り扱う事例では、メタデータを有効に使うことは非常に重要である。このメタデータをうまく生成し、利用し、管理する環境を整備することも重要な課題である。

つぎの世代のグリッド技術はセマンティックグリッドと言われており⁽⁴⁸⁾ 標準化活動でもサービス指向アーキテクチャの方向に進んでいる。セマンティックWebの技術を使い、ユーザのやりたいタスクを簡単に実行できるような環境^{(25),(26)}が提供され始めたので、その技術を利用してユーザのアプリケーションを簡

単に作り上げるようにするのが今後の課題である。

む す び

本稿では、既存のデータグリッドソリューションの分析を行った。その結果多くの解決すべき課題が明らかになった。取り上げたソリューションについてははっきりと優劣を付けられるわけではなく、それぞれ特徴を持っている。グリッドの標準化の進行に合わせて、これらのソリューションのデータ管理サービスが標準に準拠したインタフェースを持つように進歩することが望ましい。

ここでは、データグリッドのミドルウェアとして代表的なソリューションを取り上げたが、最近オラクルがデータグリッド対応のデータベース製品 Oracle 10gを発表した⁽²⁷⁾ また、データグリッドに利用できるほかの技術的な試みとして、UNICORE⁽²⁸⁾ やUnited Devices⁽²⁹⁾ ピアツーピアストレージシステム^{(30),(31)}などもある。これらの新システムや新しい試みを含めて全体を比較することは、将来の課題である。

このような課題を持つデータグリッドであるが、

将来のビジネスの機会については、企業で生み出す巨大なデータ群を有効に取り扱う必要性が増大しているため、一般的に認識されるようになってきた。これまでのビジネスデータだけでなく、多くのマルチメディアデータを扱う必要が出てくるからである。世界でビジネスを展開する多国籍企業でのデータ管理には、データグリッド技術が必須となる。

富士通は幅広いお客様にソリューションを提供しており、データグリッドソリューションがその中でも中心的な役割を果たすことになると考えられる。

参考文献

- (1) F. Berman et al. : Grid Computing – Making the Global Infrastructure a Reality . Chapter 36 , Chichester : John Wiley & Sons , 2003 .
- (2) US National Virtual Observatory .
<http://www.us-vo.org>
- (3) Astro Grid .
<http://www.astrogrid.ac.uk>
- (4) Worldwide Protein Data Bank .
<http://www wwpdb.org/index.html>
- (5) Human Mitochondrial Protein Database .
<http://bioinfo.nist.gov:8080/examples/servlets/index.html>
- (6) Earth Resources Observation Systems (EROS) Data Center .
<http://edc.usgs.gov/>
- (7) National Aeronautics and Space Administration .
<http://www.nasa.gov/>
- (8) Global Land Cover Facility .
<http://glcf.umiacs.umd.edu/index.shtml>
- (9) Nirvana Storage .
http://nirvanastorage.com/solutions/solutions_main.htm
- (10) Transana .
<http://www2.wcer.wisc.edu/Transana>
- (11) US National Archives & Records Administration .
<http://www.archives.gov/index.html>
- (12) The Globus Alliance .
<http://www.globus.org>
- (13) Avaki .
<http://www.avaki.com>
- (14) San Diego Supercomputer Center Storage Resource Broker .
<http://www.sdsc.edu/DICE/SRB/index.html>
- (15) The DataGrid Project .
<http://eu-datagrid.web.cern.ch/eu-datagrid/>
- (16) NASA Advanced Supercomputing Information Power Grid .
<http://www.nas.nasa.gov/About/IPG/ipg.html>
- (17) I. Foster : The Grid: A New Infrastructure for 21st Century Science . *Physics Today* , Vol.55 , No.22 , p.42 (2002) .
- (18) F. Berman et al . : Grid Computing – Making the Global Infrastructure a Reality . Chapter 3 , Chichester : John Wiley & Sons , 2003 .
- (19) Coda File System .
<http://www.coda.cs.cmu.edu/>
- (20) Microsoft Distributed File System .
<http://www.microsoft.com/windows2000/techinfo/howitworks/fileandprint/dfsnew.asp>
- (21) Global Grid Forum .
<http://www.ggf.org>
- (22) I. Foster et al . : The Open Grid Services Architecture Platform . draft-ggf-ogsa-platform-2 , Feb. 2003 .
- (23) F. Berman et al . : Grid Computing – Making the Global Infrastructure a Reality . Chapter 10 , Chichester : John Wiley & Sons , 2003 .
- (24) Legion .
<http://legion.virginia.edu>
- (25) Masuoka et al . : Ontology-Enabled Pervasive Computing Applications . *IEEE Intelligent Systems* , Vol.18 , No.5 , p.68-72 (2003) .
- (26) Masuoka et al . : Task Computing – the Semantic Web meets Pervasive Computing . Proceedings of the 2nd International Semantic Web Conference , 2003 .
- (27) Oracle Grid Computing .
<http://www.oracle.com/solutions/grid/>
- (28) UNICORE Forum .
<http://www.unicore.org/>
- (29) United Devices .
<http://www.ud.com/home.htm>
- (30) Peer-to-peer Archival Storage Facility .
<http://www.research.microsoft.com/~antr/PAST/default.htm>
- (31) The OceanStore Project .
<http://oceanstore.cs.berkeley.edu/>