Management and Access of Biomedical Data in a Grid Environment

Assel Matthias[1], Krammer Bettina[1] and Loehden Aenne[1]

HLRS - High Performance Computing Center of University Stuttgart
Nobelstr. 19, 70569 Stuttgart, Germany
email: {assel,krammer,loehden}@hlrs.de
phone: (+49 711) 685 62515, *fax:* (+49 711) 685 65832

Abstract

A huge number of applications, ranging from physics, chemistry and aerospace to healthcare, require lots of computational power and, even at the same time, data at a very large scale. Although grid computing is basically used on compute-intensive problems such as fluid dynamics or structural mechanics, this focus has slightly shifted to applications whose data is distributed over various locations. Access to these data resources must be carefully controlled using a sophisticated management system that allows data-intensive applications to easily, fast and securely query large amounts of data. Meanwhile, a handful of established systems for distributed data access over a grid have been developed and mainly used within some research projects.
In this paper, three existing data management solutions - the *OGSA-DAI* middleware, the *EGEE gLite Data Management* subsystem, and the *SDSC Storage Resource Broker* - shall be briefly introduced. Afterwards, key issues and requirements in developing a sophisticated management system for biomedical data access will be identified and finally compared to the functionalities of the existing concepts. Furthermore, implications for designing virtualization services, which allow secure access to biomedical data resources, shall be roughly presented.

1 Introduction

Grid applications used to store their scientific data almost exclusively in files, which then are transferred to corresponding sites to perform any computation on the archived values. An increasing number of applications, for example many applications in the life and earth sciences, many business applications, and even many healthcare applications are heavily dependent on databases. Consequently, the interest in integrating databases into a grid environment is becoming more and more popular and the development of reusable and standardized middleware solutions is strongly demanded and more or less promoted.
Since databases offer a much richer set of operations, such as queries and transactions, and there is much greater heterogeneity between different database mangement systems (DBMS) than there is between filesystems, the available grid services for file handling cannot simply be adapted to manage databases on a grid in a similar way. The variety of functionalities and interfaces as well as

some important properties such as security, performance, and reliability of each single DBMS, all result in an integration of the systems themselves instead of developing a new grid-enabled database management system from scratch [1].
Besides the individual requirements of each DBMS, the complexity of data management on a grid furthermore arises from the scale, dynamism, autonomy, and distribution of such resources. To conceal these complexities and the diversity of the underlying infrastructure, the middleware system has to ensure that all resources appear transparent to their users. This could be achieved by hiding the different data resources and their internals behind a layer of virtualization services that guarantees data access in a consistent, data resource-independent way. The services provided should therefore implement standard interfaces to support the integration of multiple client application technologies in a common way and should allow different types of data resources - including relational, XML, and files - to be exposed onto grids [2].
Within *ViroLab*, an EU funded research project of the 6th Framework Programme for Research and Technological Development in the area of integrated biomedical information for better health, *a Virtual Laboratory for Infectious Diseases* will be developed that facilitates medical knowledge discovery and decision support for HIV drug resistance [3].
During the past few years, large, high quality, clinical and patient databases have become available, which can be used to relate genotype to drug-susceptibility phenotype, but this data is inherently distributed over various sources (virological, clinical, and drugs databases) and might change dynamically over time. Although using a grid-based service oriented architecture, the integration of biomedical information from viruses, patients, and literature constitutes one of the major challenges of the project [4].
The goal of this paper is to investigate grid services for meeting data management and access needs of biomedical resources by considering services provided by established grid data management solutions.

2 State-of-the-Art for Data Management Technologies

OGSA-DAI[1]

The *OGSA-DAI project* was conceived by the UK Database Task Force, which is working closely with the Global Grid Forum among others. The aim is to develop middleware to assist with access and integration of data from separate sources via the grid [5].
OGSA-DAI is a middleware product, which supports the exposure of data resources onto grids and allows these resources to be accessed via web services. Various interfaces are provided and many popular database management systems, such as relational databases, XML databases, and files are supported.
The software includes a collection of components for querying and updating data within each of these types of resources. The data can be transformed,

[1] Open Grid Services Architecture Data Access and Integration

compressed and decompressed, and delivered in different ways, for example to clients, other OGSA-DAI web services, etc.
OGSA-DAI is compliant with two popular web service specifications: WS-I[2] and WSRF[3]. The WSRF version of OGSA-DAI is compatible with the Globus Toolkit's implementation of WSRF. Therefore, it can be deployed within a grid environment and thereby provides a means for users to grid-enable their data resources [6].
The OGSA-DAI security infrastructure is based on two mechanisms. To ensure authorization control, OGSA-DAI provides its own concept by mapping user credentials obtained from X.509 certificates to services and data resources whereas communication messages can be made secure using the security functionalities provided by the Globus Toolkit.
OGSA-DAI is based on a multilevel architecture, which consists of a number of layers each serving a different purpose. A high-level schematic representation of this architecture is depicted in figure 1.

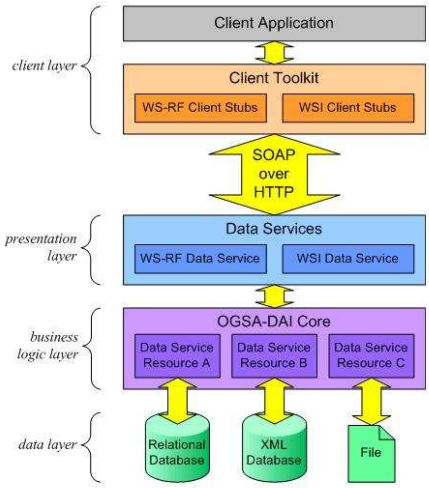

Fig. 1: OGSA-DAI's multilayered Architecture

A client typically communicates with a corresponding data service using OGSA-DAI's client toolkit, which supports developers to simply connect their applications with an OGSA-DAI data service, by providing convenient ways to construct and send SOAP requests and interpret the subsequent responses. A data service consists of components known as data service resources. As shown in figure 1, multiple data service resources can be deployed to expose multiple data

[2]Web Services Interoperability
[3]Web Services Resource Framework

resources. Each of them provide basic functionalities to interact with the corresponding resource type by using resource-dependent mechanisms, mainly standard query statements. At this point, users can implement new web services to expose their own data resources and easily provide their own application-specific functionality.
The OGSA-DAI software is used in many different grid projects around the world, some of them focusing on the medical or biomedical fields, e.g. Cancer Biomedical Informatics Grid (caBIG) [7], eDiaMoND [8], BioGrid [9], and even more.

EGEE - gLite Data Management

Originally, *EGEE* used middleware based on work from previous European projects. In parallel, EGEE has developed its own new middleware solution, *gLite*, which combines low level core middleware with a range of higher level services [10].
The gLite grid services follow a Service Oriented Architecture, meaning that it will be easy to connect the software to other grid services, and also that it will facilitate compliance with established grid standards, e.g. WSRF.
Data management in the gLite context has four main service groups related to data and file access: Catalog Services, Storage Element, File Handling and Data Transfer.
In order to provide transparency of the stored data to the users, the middleware offers the capabilities for accessing distributed data resources which are implemented along the SRM (Storage Resource Manager) interface. Accessing the data in files can be done through the Storage Element (SE). Access to the files is controlled by the Access Control Lists (ACL). The detailed semantics of file access will be different depending on what kind of storage back-end is being used beneath the SE.
Data transfer and scheduling services will expose all nontrivial interfaces to users for data placement in a distributed environment. The applications need to specify files they intend to access during their jobs either by name or by a metadata query. A client user application may look like a Unix shell which can seamlessly navigate in this virtual file system, listing files, changing directories, etc.

Storage Resource Broker (SRB)

The San Diego Supercomputer Center (SDSC) *Storage Resource Broker (SRB)* is a distributed, attribute-based file system [11].
The SRB is based on a client-server architecture that provides a uniform interface for connecting to heterogeneous data resources over a network and accessing replicated data sets. It employs a Data Grid Management System (DGMS) to store all its metadata and user-defined attributes through a single hierarchical logical namespace. The SRB software infrastructure can be used to enable Distributed Logical File Systems, Distributed Digital Libraries, Distributed

Archives etc. The most common usage of SRB is as a Distributed Logical File System - a synergy of database and file systems across multiple storage systems. The user can access the distributed data as a single file hierarchy. The SRB middleware has features to support the management, collaboration, controlled sharing, publication, replication, transfer, and preservation of distributed data. The SRB data management environment provides library functions that can be utilized by higher level software, including end-user client applications ranging from web browsers to Java class libraries to Perl and Python load libraries.
The SRB has become a default DGMS for collaborative data management in multiple academic data centers around the world. It is not an Open Source product, although the source code is freely available to academic organizations and government agencies.

3 Requirements for Biomedical Data

Unlike normal data-intensive and compute-intensive grid applications, there are some specific requirements for services dealing with biomedical data which differ from other grid services.
Besides standard mechanisms and interfaces for discovering and accessing data resources, for instance using web service interfaces, the transformation of heterogeneous biomedical data - view figure 3 that illustrates a protein data set differently stored in three databases - is, due to different formats of each resource, one of the most challenging issues. Therefore, querying and handling of metadata plays a decisive role to obtain interoperability between data sets at different centers across different countries. Furthermore, standardization of data format, data structure, and data model should be achieved in the future in order to facilitate the exchange of data [12].

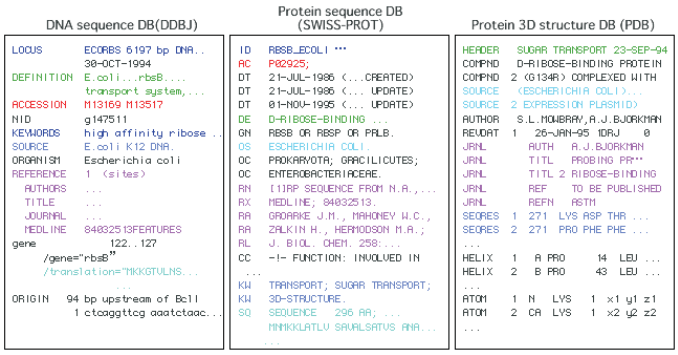

Fig. 2: Example of the RbsB protein, a ribose binding protein, which is differently represented in the DDBJ [13], SWISS-PROT [14] and PDB [15] databases

When dealing with confidential data, security mechanisms are of the utmost

significance and are seen as the most critical part. Authentication and authorization are a precondition for transferring data over a grid, but the high sensitivity of biomedical information and personal data also demands secure transmission and secure storage. In order to keep the privacy and protect the confidentiality of patients, medical data need to be encrypted, anonymized, and even deidentified before sharing it within a grid environment.
The use of anonymized information also needs to satisfy the requirements under the Data Protection Act as well as explicit consent from ethics committees for doing research, and, of course, from the patients [12].
Further requirements, such as the permanent availability of resources or the guarantee of QoS[4] parameters heavily depend on the actual user scenario and should also be considered during the design and development phase if demanded by applications and / or users.

4 Implications for Designing Virtualization Services

The design of own virtualization services that can be used to securely access biomedical data resources should be principially based on existing technologies, otherwise the effort for developing a suitable solution from scratch will be "endless" and almost impossible. In order to determine the most useful existing concepts, a comparison of their basic functionalities with respect to the requirements for accessing biomedical data has been performed and an overview is presented in table 1.

Requirement	OGSA-DAI	gLite	SRB
Interfaces for Access & Discovery	Web services (SOAP over HTTP)	Library functions (Perl, Java)	Library functions for Higher Level Software
Transformation of Data	Data can be transformed but needs to be adopted	Not provided	Not provided
Authentication & Authorization	X.509 certificates	User certificates	Password authentication
Secure Transmission/Encryption	Message/Transport-Level Security	Transport-Level Security	Mechanisms for data encryption
Anonymization of Data	Not provided	Not provided	Not provided
Monitoring	Simple logging	Logging subsystem	Simple logging
Extensibility	Source available - designed to be extendable	Source not available	Source available for academic research

Tab. 1: Requirement analysis on existing solutions

[4]Quality of Service

To support a wide range of end-user applications, the interfaces describing and providing the interactions between clients and services should be as variable and general as possible. This variability mainly constitutes the interoperability between different application technologies. Therefore, the concept of web services provides standardized interfaces that can be used independently of the underlying infrastructure and that can be easily extended and adjusted due to the application developer's needs.
Although dealing with heterogeneous resources and different data formats, most existing systems do not support data transformation out of the box. Since clients communicate with web services via SOAP messages, the format - the structure of the message - can be adjusted dependent on each application requirements. That transformation simply involves an XSL-Transformation that converts the results into an application-dependent reponse message format. The style schema required for an XSL-Transformation should be dynamically loaded to support multiple clients with their specific data schema.
The high sensitivity of biomedical data demands strong security mechansims, which should be developed and organized in a multilayered way. On the one hand, the mechanisms provided by the grid middleware can be used to guarantee a certain level of secure transmission - user authentication and data encryption are typically included by default. Unfortunately, these features are not sufficient enough for managing a biomedical environment, especially controlling access rights. Additional authorization mechanisms need to be developed in order to limit, deny, or permit access to different resources in a dynamic but sophisticated way. Attributes of users including their organization, department, role, etc. should be used to administer access rights to resources but also to services.
Taken the presented data management systems into account, gLite can be confidently put aside because of its limited functionality with respect to the requirements and the restrictions it imposes on extending the existing framework. The SRB offers some interesting possibilities, but OGSA-DAI is seen as the most appropriate middleware solution due to its great extensibility based on the web service technology in combination with the Globus environment.
OGSA-DAI could be used as the core that combines functionalities for all virtualization services, and with some adjustments to its security mechanisms and an implementation of new transformation services for biomedical datasets, it might provide a good basis for developing a secure biomedical data access infrastructure.

5 Conclusions

Data management in general, but especially on a grid, forms a very complicated task and demands a sophisticated system, which allows secure access to heterogeneous and distributed data resources. Besides all standard security regulations, the privacy and confidentiality of patients data must be kept and protected before sharing their sensible information over various institutions. Fur-

thermore, their anonymized personal data must satisfy the requirements under the Data Protection Act regarding legal and ethical issues.
Unfortunately, common solutions do not strictly fulfill the specific prerequisites for accessing biomedical data but the design of own virtualization services should rely on their core functionalities. OGSA-DAI particularly provides a good starting point but different services need to be adapted and some need to be completely newly developed.
This paper should give the readers an overview of the widely spread data management solutions used within a grid environment and analyzes the main requirements for sharing confidential biomedical data. Implications on designing own and more applicable services should encourage people to focus their research on healthcare infrastructure and appropriate applications.

Acknowledgements. Research in this paper has been made possible through the support of the European Commission *ViroLab* Project Grant 027446 and all members of the *ViroLab* consortium. We are very grateful to all who contributed to this paper.

References

1. Paul Watson. *Databases and the Grid*. UK e-Science Technical Report Series. University of Newcastle, UK
2. Vijayshankar et al. *Data Access and Management Services on Grid*. IBM Research Center San Jose, USA. 2002
3. The ViroLab Project. *http://www.virolab.org*. EU IST Project. 2006
4. Sloot P., Boucher C., Bubak M., Hoekstra A., Plaszczak P., Posthumus A., van de Vijver D., Wesner S. and Tirado-Ramos A. VIROLAB - A Virtual Laboratory for Decision Support in Viral Diseases Treatment. Cracow Grid Workshop 2005, Cracow, Poland, November 2005.
5. The OGSA-DAI Project. *http://www.ogsadai.org.uk*. University of Edinburgh, 2005-2006
6. The Globus Toolkit Homepage. *http://www.globus.org/toolkit*
7. The Cancer Biomedical Informatics Grid (CaBIG) Project Website. *https://cabig.nci.nih.gov*
8. The Diagnostic Mammography National Database (eDiaMoND) Project Website. *http://www.ediamond.ox.ac.uk*
9. The BioGrid Project. *http://www.biogrid.jp*
10. EGEE gLite Data Management- Lightweight Middleware for Grid Computing. *http://glite.web.cern.ch/glite*. EU EGEE Project, 2006
11. The SDSC Storage Resource Broker (SRB). *http://www.sdsc.edu/srb*. San Diego Super Computing Center, 2006
12. Lingfen Sun, Emmanuel C. Ifeachor. *The Impact on Healthcare*. University of Plymouth, UK
13. The DNA Data Bank of Japan (DDBJ). *http://www.ddbj.nig.ac.jp*
14. The UniProtKB/Swiss-Prot Protein Knowledgebase (SWISS-PROT). *http://www.ebi.ac.uk/swissprot*. 1986
15. The RCSB Protein Data Base (PDB). *http://www.rcsb.org/pdb*