DATA MANAGEMENT SYSTEM

A TECHNICAL WHITE PAPER
April 2004

Poznan Supercomputing and Networking Center

Introduction

The development of data management technology caused a rise of separate, data-oriented systems. Such systems are essential components of architecture of many contemporary systems, playing a special part in the grid environments. The basic goals of data management systems are storing, delivering and describing data.

Data management is one of the core services in the Grid systems. Grid computing is an important new field focused on large-scale resource sharing. We integrated both issues within the PROGRESS [PRO] project aimed at creating an "Access Environment to Computational Services Performed by a Cluster of SUNs" . The PROGRESS grid-portal environment has been designed and implemented as a result of the "Access Environment to Computational Services Performed by a Cluster of SUNs" project. This initiative was undertaken within the PIONIER National Program [PIO] and was funded by the State Commitee for Scientific Research and Sun Microsystems Poland. The project lasted until May 2003, but we are keen to continue our research afterwards. The PROGRESS project is aimed at developing an access environment to computational resources, which would allow the creation of a comfortable workplace for grid users.

In the scope of the PROGRESS project we have designed and implemented the Data Management System (DMS). DMS enables the creation of a distributed environment capable of storing and enabling access to a large amount of data such as the data used and produced by scientific computing experiments performed in grids.

The first prototype of aggregate tools and services realized in the PROGRESS project were presented during the HPC Consortium Meeting in Glasgow in 2002. At this stage it was possible for visitors to utilize the mechanism of task submitting for test applications together with the use of basic functions of Data Management System[DMS1].

The demo version of the PROGRESS HPC Portal testbed including DMS was presented at the Supercomputing 2002 (SC2002) conference in Baltimore, at SUN Microsystems Inc stand. The testbed prototype, using three bioX grid, enabled applications for testing developed tools and the integrated system. These applications are available for running scientific experiments in the cluster of three Sun computing systems and two data servers distributed between Poznan and Krakow, which serve as the PROGRESS testbed grid. The binaries of applications were stored in the Data Management System which was also used as the source for grid job input data and a place for storing the results of scientific experiments. There was also a special item within the DMS which serves as a proxy to
scientific data banks – an example of such bank used in the PROGRESS testbed is the SRS [SRS]

The full functionality of DMS including administrative website, which made it possible to remotely manage computational resources, was introduced at the Supercomputing 2003 conference in Phoenix, 15-21.11.2003.

Application area

The Data Management System has been developed within the scope of the PROGRESS project, which can be treated as a multi-layered grid environment composed of several components including:

- grid services management system with their communication environment,
- security of hardware and software resources,
- data management and data exploring together with interfaces to the external data sources (DMS),
- services and applications management from website,
- web-based visualization of biological computations

The main task of DMS is to store and provide access to data within an open grid environment. This is not, however, the databases features duplicating – the basic concepts which lied down at the DMS creation were the development and implementation of the environment allowing flexible access to data stored inside the system, accessible by the most popular protocols and opened to standards defined by grid organizations.

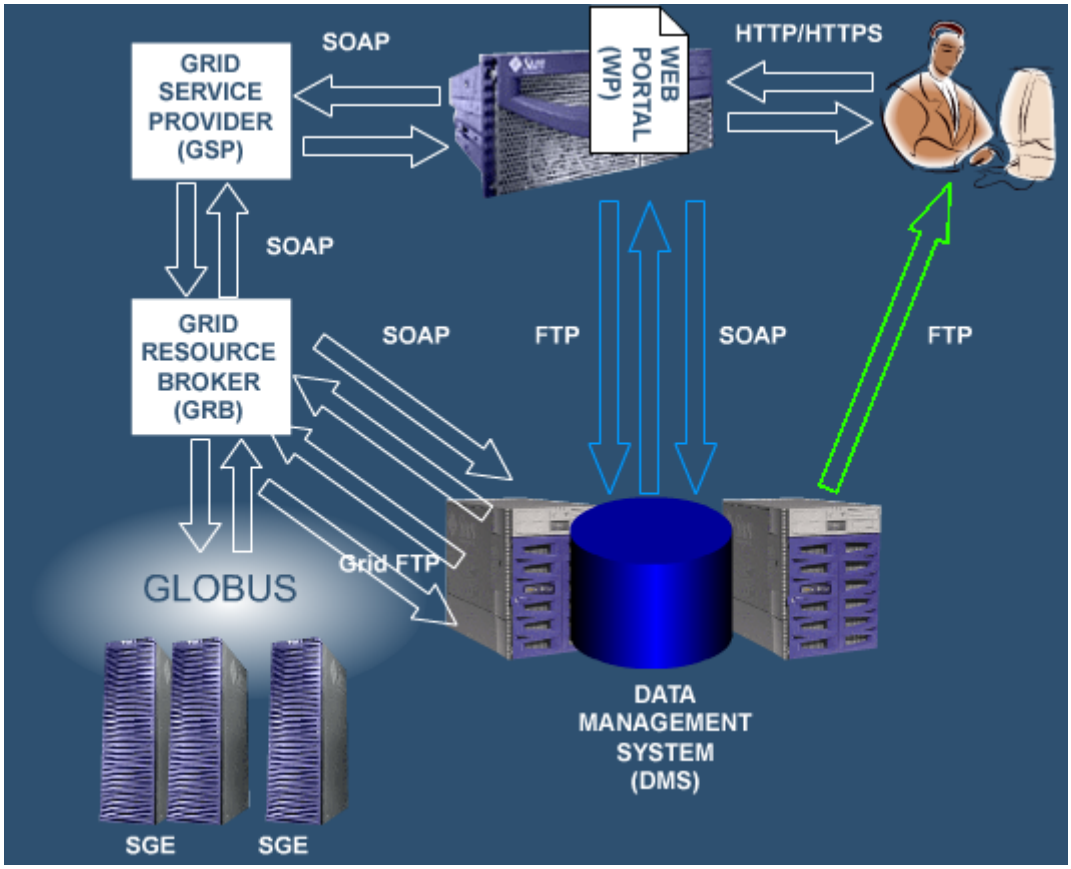


Fig. 1 DMS within the Progress project

The Data Management System is a distributed system featuring data access and delivery. It is based on the pattern of autonomic agents using the accessible network infrastructure for mutual communication. The SOAP protocol [SOAP], which is a part of the Web Services technology, was chosen as the communication technique. Web Services cover a stack of technologies enabling the creation of distributed applications implemented in any programming language and executed on any platform. According to those definitions, each

DMS module can be treated as a separate service using XML messages and passing them in a form of SOAP envelopes to external applications.

DMS is a kind of a virtual file system keeping the data organized in a tree structure. The main units of this structure are metadirectories, which make it possible to hierarchize other objects and metafiles. Metafiles represent a logical view of computational data regardless of their physical storage's location. The computational resources managed by DMS can be described by metadata which create an abstract, semantic and explorable layer of resources. The fulfillment of presuppositions of flexible access to data requires DMS to support various transmission protocols – DMS realizes data access through the HTTP, FTP GridFTP [GridFTP] and GASS [GASS] protocol.

Basic Patterns

The following chapter describes the functionality of the Data Management System from the external applications perspective as well as internal architecture of the system with regard to its modularity and logical division. This also chapter covers the ideas of security policy applied to the system.

System functionality

The basic task of DMS is to store and deliver data in an open grid environment. The main functional assumption made at the realization of this system was to hide the data management complexity from the end user as well as to create an easy and intuitive method of data access. DMS possesses a unified access interface implemented in the Web Services technology and, moreover, thanks to its modular architecture the system is ready to use new standards defined by grid organizations.

The Data Management System contains three logically distinguished modules: Data Broker (DB), Metadata Repository (META) and Data Container (STOR), which together create the basic layer of the data management environment, the so-called DMS core. Each system module is treated as a separate network service communicating with the external environment with the help of the SOAP protocol. Apart from these three components, additional modules should be distinguished, which complete the data managements environment and extend its overall functionality to fulfill the enterprise requirements:

- administrative portal - it makes full functionality offered by DMS available to the end users,
- RAD (Resource Access Decision) [RAD1] - the authorization system based on the CORBA standard, allowing control and management of the policy access to DMS resources. RAD states as optional modules, delivered together as additional DMS package,
- PROXY – it provides a SOAP interface to the external databases. Within the PROGRESS installation Proxy enables access to SRS (Sequence Retieval System) resources. SRS is a data environment module, which stores the biological sequences essential for the computations_executes within the grid. The system uses a pilot installation in Poznan and it is configured with the use of indexed biological data banks. In addition, a set of scripts was prepared to keep the local copy of biological data up to date.

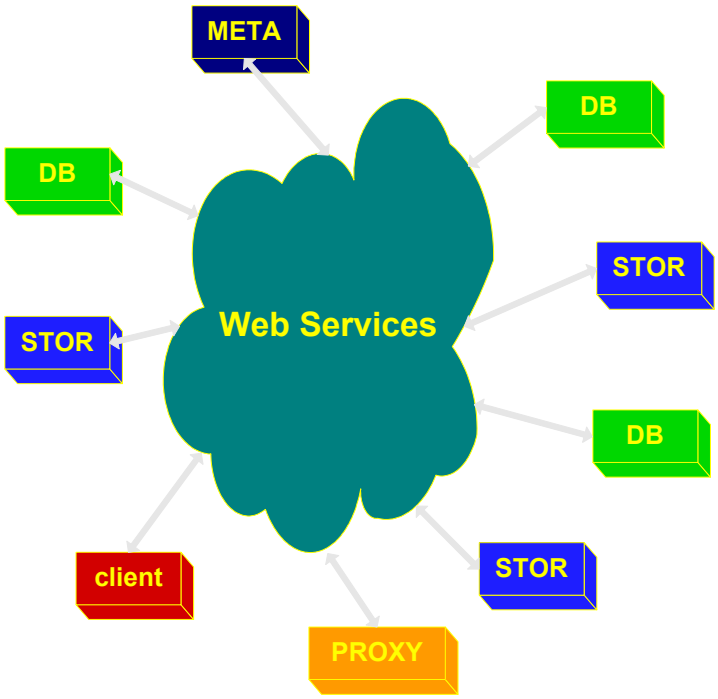


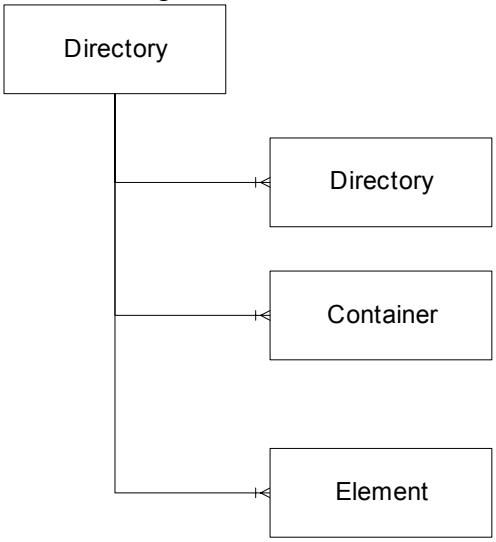
Fig. 2 DMS as a distributed file system

As it was previously mentioned, DMS is considered as a virtual file system keeping the resources in the hierarchic structure of the metadirectiores. Although DMS is a distributed system, this structure is consistent owing to the Repository module, which is a central and single point of metadata management.

DMS provides its services in a form of Web Services API to the front-end applications. It is important to mark the DMS position in an open grid environment – DMS is a middleware system, belonging to the collective layer as well as the resource layer (Data Container), according to the grid services view.

Data Broker is designed to be the main access point to the DMS resources and services. It provides the Web Services interface with storing, access, describing and delivery of data. These services can be divided into the following categories:

- catalogue-based services –for creating a metacatalogue, removing it or moving to another place in the structure within all its contents;



- **Fig. 3 Metadirectory content**

- file-based services – for adding, deleting, renaming file and additionally determining the physical localization of the file and accessing the file. DMS provides separate services to files' replication.
- security services – to determine if the final user is entitled to a given operation on specific resource according to the metadata information.
- metadata services – each object stored in the DMS can be described by metadata. DMS delivers the predefined DublinCore [DC] standard and makes it possible to create user-defined metadata. Metadata services are Multilanguage-oriented.

Internal architecture

The logical structure of the system consists of three layers and a administrative website which is presented presents in the following drawing:

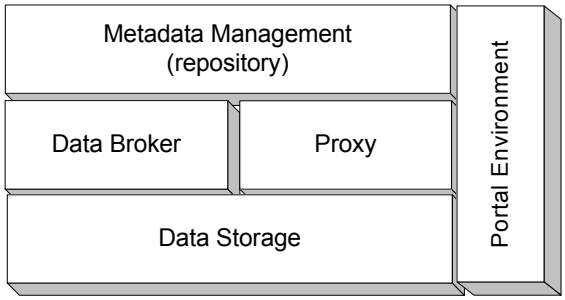


Fig. 4 Logical structure of DMS

The highest layer of the system is the Metadata Repository. This layer is responsible for all metadata operations as well as their storage and maintenance.
The Metadata Repository is the main element of DMS. Here it stores the following sorts of information:

- metadata about resources: data files, its physical localization and possible way to access them,
- metadata about rights: all information related to the rights – users, their groups, access rights.
- metadata describing the standards of file description, e.g. Dublin Core (DC)

Access to the repository resources is services by the Metadata Management (MM) module. The main task that it fulfils is collecting data broker requests. This is realized according to the current knowledge about the data (metadata) and also basing on the state of the Data Containers and information from them.
There is one instance of the MM module in the DMS. It is motivated by the fact that it stores and manages the critical information about the metacatalogue structure, user data and security policy for the whole DMS
The next logical layer is created by the Data Broker and Proxy. Data Broker is designed to be the main:

- access point to the DMS resources and services. The basic tasks that it delivers are as follows:
- asynchronous responding to the clients requests, without blocking access to those services for others clients,
- fulfilling the security policy on the repository elements level (access to data files, directories),
- passing on the clients requests to the metadata repository
- collecting and sending back the results to clients

DB is a module that mediates in the flow of all requests directed to the DMS. There is no possibility for external application to pass direct requests to the repository or even data Containers.
Within the confines of work over DMS the technology of communication between DB and others DMS modules was designed and initiated. The idea of this mechanism can be illustrated basing on a single client request description:

- at first DB authorizes the client that submitted the request. To this end DB uses remote authorization module to verify that the given user is authorized to access specific resource.
- after a successful verification process DB prepares an appropriate query (basing on the clients request) and sends it to the central metadata repository.
- the received results are transferred to the client.
- All the above-mentioned operations are executed in a distributed agent environment basing on the SOAP protocol for exchanging communicates (Web Service technology).

The Proxy layer is responsible for access to external data sources. It provides a unified interface to different resources with a diversified data structure. DMS treats the Proxy as a separate, independent Container, that manages read-only data.
The third and the lowest DMS layer is the Data Container. It is responsible for delivering space on the storage resources, which stays under its control and manages the operation of placing data files inside its resources and accessing them from those resources on demand. Operations executed by this module are connected with the making of reservation for the data planned to be placed on the Container resources, blocking data files for access (and allowing access to the data file) and accessing information about the state of the whole Container module and separate data file staying under its control.
Keeping in mind the fact that the Container was designed to collaborate with other elements of the DMS, it implements the internal functionality using interfaces, which allows to communicate with the other DMS modules. Implemented elements are made available (similarly to other modules of the DMS) as web services that can be called with the use of the SOAP protocol for exchanging communicates (the data Container is controlled in that way). Direct data access is ensured by standard data transfer protocols used in the grid environments: the FTP protocol, GASS protocol (Grid Access to Secondary Storage) as a standard and secure version, GridFTP protocol (gsiftp – the enhance of the standard FTP protocol with the Globus Security Infrastructure).
The container is able to store data on a different type of medium: file systems, databases and tape systems. In order to realize this feature three different instances of Data Containers have been designed and implemented, each with the same interface.

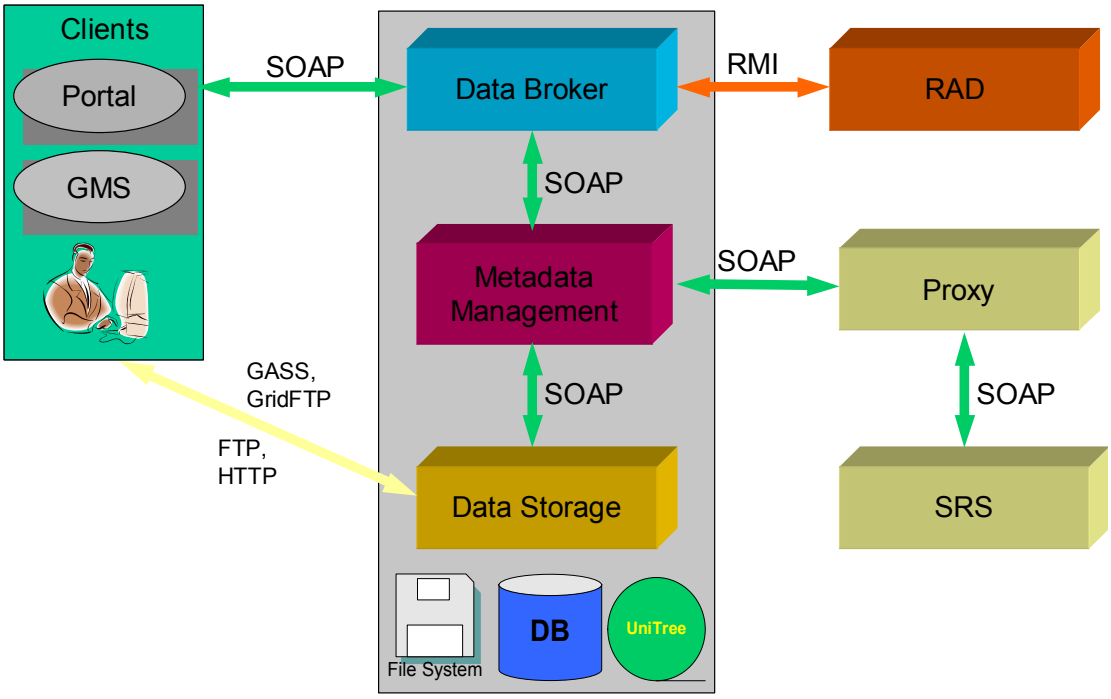


Fig. 5 Modular architecture of DMS

Security solutions

DMS is a distributed system built on the Web Services technology. It uses SSL as a basic mechanism of protection from unauthorized access to its resources. SSL ensure:

- identification and authentication of external applications communicate with DMS (i.e. computational portal, migrating desktop[MD], resource broker [RB]),
- data encryption during network communication.

DMS authorizes the end users on the level of access to the system resources (files, catalogues, metadatas). It means that DMS ensures that a user is able to remove, find or change the name of any object, only if he has suitable rights to perform this operation.
DMS can be run in two types of the authorization policy: basic (standard), also called internal and advanced (enterprise). Basic authorization is a default type of policy available after DMS installation. In this option object right management is very limited. Advanced security is activated automatically after integration with RAD. The DMS advanced security provides a comprehensive suite of security features to protect enterprise environments and securely extend corporate grids to the Internet. It makes it possible to realize a complex security policy based on users, groups, rights to objects and rights to executing particular operations.

Conclusion and Future Work

In this paper we introduced the concepts of Data Management System supporting storage management on the grid. Similar to computing and network resources, DMS provides services to manage and retrieve data files in order to support grid jobs. In addition, DMS exposes uniform interfaces for different types of storage resources, including disk systems, robotic tape systems, mass storage systems and databases. The Proxy module, which can be treated as a gateway between DMS and external data sources (like scientific databases) allows using external data as read-only files.
DMS functionality includes features known from digital libraries - all objects managed by the system can be described by metadata which enables processing, searching and browsing files in a form of digital documents.

DMS has already been implemented in various systems, including mass storage systems, in cooperation with the different kinds of applications – portals, migrating desktops and resource brokers. DMS has been load and stress tested using TestMaker [TS], specialized software for Web Services testing, achieving satisfactory performance in multi-users environment.
DMS is currently being deployed into Virtual Laboratory environment [VL], where it will act as a component that facilitates grid access to data for scientific experiments. It will also be used for storing electronic publications from a given knowledge domain and sharing task results between scientists and laboratories.

In the near future it is planned to accommodate DMS to work with a PostgreSQL database [PSQL]. Another goal is to make DMS a more independent, reliable and available system by implementing procedures of system recovery, consistency restoration and containers registration as well as mechanisms for automatic file replication. This will allow simplifying DMS administration process.

DMS is available now at http://progress.psnc.pl/English/opensource.html under PROGRESS SOFTWARE LICENSE - http://progress.psnc.pl/English/progress_license.txt

Contact:
Marcin Wolski – Marcin.Wolski@man.poznan.pl
DMS mail group – szd@man.poznan.pl

References

[PIO] – Polish Scientific Broadband Network, accessed from http://www.man.poznan.pl/resources/national_network/index.html
[PRO] "PROGRESS HPC Portal", accessed from http://progress.psnc.pl/portal/
[DMS1] P. Grzybowski, C. Mazurek, P. Spychala, and M. Wolski, "Data Management System for grid and portal services", submitted to Grid Computing: Infrastructure and Applications special issue of The International Journal of High Performance Computing Applications (IJHPCA), Cardiff University, UK., accessed from http://progress.psnc.pl/English/DMS.pdf
[SRS] SRS System, accessed from http://srs.man.poznan.pl
[GASS] Global Access and Secondary Storage (GASS), accessed from http://wwwfp.globus.org/gass/
[GridFTP] The GridFTP Protocol and Software, accessed from http://wwwfp.globus.org/datagrid/gridftp.html
[RAD1] Resource Access Decision, version 1.0, accessed from http://www.omg.org/technology/documents/formal/resource_access_decision.htm
[MD] M. Kupczyk, R. Lichwala, N. Meyer, B. Palak, M. Plociennik, and P. Wolniewicz, "Roaming Access and Migrating Desktop", Proceedings of The 2nd Cracow Grid Workshop, Cracow, Poland, December 2002, accessed from http://progress.psnc.pl/English/MigratingDesktop.ppt
[RB] Pukacki J.: Resource Brokering in the PROGRESS Project
Presented at Grid Resource Management Workshop, New Network Technologies, Grids and Portals Multiconference, October 20th-22nd 2003, Poznań, Poland, accessed from http://progress.psnc.pl/English/progress-broker.ppt
[DC] Dublin Core standard, accessed from http://www.dublincore.org
[SOAP] Apache SOAP, accessed from http://ws.apache.org/soap

[TS] TestMaker, test automation solutions, accessed from http://www.pushtotest.com/
[VL] Virtual Laboratory home page, accessed from http://vlab.psnc.pl
[PSQL] PostgreSQL Database Server, accessed from http://www.postgresql.org