Astronomical Data Analysis Software and Systems XIII
ASP Conference Series, Vol. 314, 2004
F. Ochsenbein, M. Allen, and D. Egret, eds.

Grid Data Distribution strategy: Design and Implementation of a Pipeline Oriented Data Management System

N. Lama, C. Vuerli, R. Smareglia, F. Gasparo, F. Pasian

INAF/Osservatorio Astronomico di Trieste, Via G.B.Tiepolo 11, I-34131, Trieste, Italy. e-mail: family-name@ts.astro.it

M. Genghini

Istituto Astrofisica Spaziale e Fisica Cosmica, Bologna

Abstract. Dynamic data distribution is a key factor in Grid computing. The DMC project, aiming at improving collaborative research by allowing data to be shared more easily across applications cooperating within a federated environment, is described. DMC is the data management system chosen by the Planck Satellite Survey Community, and specifically by the two Data Processing Centers, as a common infrastructure for the data handling applications being developed. Particular reference is here made to the design of the model, the data structures and to the portability of the Planck experience to other pipeline-oriented distributed environments, with particular reference to Grid-enabled systems.

1. Introduction

The aim of the project is to provide a pipeline-oriented data management system specialized with data products required by grid oriented data processing modules. The underlying principle of DMC is to have a service tool through which a pool of applications can store and retrieve their data products from a number of geographically distributed data repositories. These concepts make the DMC a tool particularly suited to data grid applications [1]. Originally required within the framework of the Planck IDIS (Integrated Data and Information System) Working Group, the system has been designed so as to be fully portable to other experiments, missions and data management projects. Design details are given in [Vuerli 2001a; Vuerli 2001b; Lama 2002].

2. DMC model: THE CORE

The DMC has a multi-tier software architecture which is object-oriented and is organised into independent layers: the DMCI (DMC Interface) and the physical

[1] `http://wwwas.oat.ts.astro.it/draco/DRACO-home.htm`

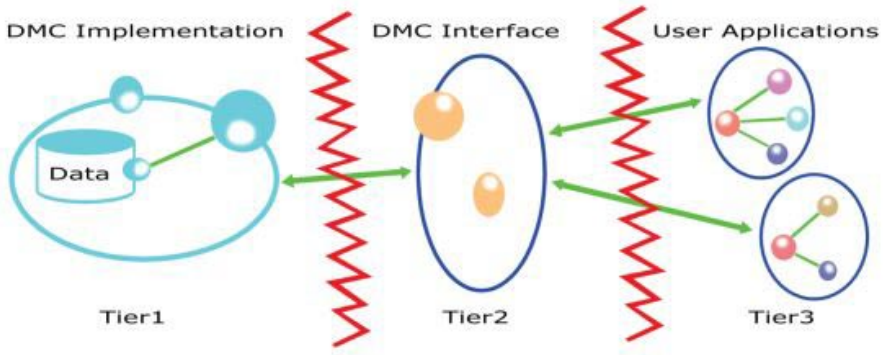

Figure 1. DMC multi-tier layout

implementation (see Figure 1). The DMCI is the User Interface (or Presentation Layer), a set of interfaces (API-like) through which scientific applications can exploit the DMC services. These interfaces hide the actual physical implementation from the user or the calling application. The DMC Physical implementation is the Data Services Layer which communicates directly with the Database. A crucial objective was to hierarchically develop the DMC; the result is that the DMC is implemented by a Business Services Layer, related to application oriented objects, plus a DMC Core implementation. The latter is the Basic Services Layer, which implements the foundation for the data handling. It provides a set of basic services portable to all those experiments that are pipeline/module oriented. The core organizes data products within the associated module or pipeline producer object, aiming at speeding up data exchange between clients.

3. DMC compatibility with Grid concepts

The DMC is a Digital Library that can be mounted on the top of a Data Grid infrastructure [Pasian 2004, Smareglia 2004] and provides services for manipulating, presenting, discovering, browsing and displaying digital objects. It is a particular implementation of the Generic Virtual Data Access and Integration Technology layer. It enhances and specializes the following core services of Grid-enabled data storage resource [Stockinger 2001].

Data Formats – Metadata management is a Virtual Organization (VO) task. According to [Segal 2001], experiment specific or more generally VO-specific metadata is managed by the VOs software infrastructure and not by DataGrid Middleware tools. DMC data model design [Vuerli 2001, Lama 2002] foresees clients to store information through metadata management common API (e.g. database schema, FITS file structure). The usage of undistinguishable Binary Large Object (BLOB) data is not encouraged since it limits data sharing, which is the aim of DMC itself. The Digital Library nature of the DMC guarantees its smooth evolution following forthcoming metadata requirements.

Data access operations – The DMC data model is composed of an inventory of objects representing the variety of data products created along the pipeline processing path. Objects are aggregated into containers (namespaces)

and connected into data flows, expressing an invocation sequence of scientific solvers and visualization tools. DMC provides primitives for uniform access to metadata and storage structure through data model browsing (virtual directories) and advanced lookup mechanisms (queries, see below).

Local transparency and global name space – Through DMCI, users can access data in a federation of data repositories transparently. DMC-enabled applications deal with a set of virtual data repositories and access data independently of their physical location. Currently, an LDAP based IDIS Federation Layer component is in charge of dynamically resolving this link at runtime. Plans are to move towards a DataGrid-like Storage Resource Broker approach.

Persistence and Replication – DMC emphasizes the scientific computing ability to access large amount of data, stored in blocks. Data objects can be used as temporary containers (non-persistent objects) as regards local processing or particular high performance applications. Replication can be wrapped on the top of the physical COTS (e.g. Versant replication API)

Privilege and security issues – Aiming at encouraging resource/data sharing and the collaborative approach of DMC users, read/write privileges are handled at the data repository granularity level (authentication level). This well fits security requirements of those projects that, like Planck, are working group oriented. Plans are to enforce security through EDG Java Security package [Bosio 2003], data cryptography and digital certificates

Error and exception handling – DMC manages data handling errors and exceptions generated when accessing a data repository and throws exceptions on failure of consistency checks that enforce data model integrity and pre-processing data quality checks.

Check-pointing and state management – DMC services are transaction oriented; it is possible to re-build state and re-start operations, on failure. The DMC provides multiple database connection within sophisticated locking models (optimistic locking, transaction shared among different data repositories).

4. Implementation issues: TECHNOLOGY

COTS adopted The programming language is JAVA (to ensure high portability) and JNI for ad hoc integration with non-java client modules. Versant is the OODBMS choice supported Planck-wide. Java Data Object [2] (JDO) technology is being evaluated: DMC JDO-compliant implementation would provide access to relational databases, object databases, flat files, or any other compatible persistent storage device. A Java Servlet Web-based visualization tool is being developed, exploiting Starlink software experience on VO data viewing and modeling [Gray 2004, Taylor 2003].

Core implementation The data model has been designed to reflect data usage and so as to be pipeline oriented. Data are organized within a graph structure modeling pipeline path. This has been done aiming at exploiting fast data browsing by link and preventing time expensive internal queries traversing the databases to find and evaluate starting point objects. The history of the

[2] `http://access1.sun.com/jdo/`

processing path of data products is logged so to let clients browse data products following their processing path.

Queries Data retrieval features include object lookup by mnemonic alias, by version and attribute values. Modules can retrieve products owned by a specified user or produced from a module or pipeline with certain parameter values. Advanced lookup services under construction: lookup of time ordered data by sky position through scanning strategy information.

huge sized data management Maps and time series are internally managed as segmented arrays. Data are buffered within data chunks forming a segmented array structure that allows the DMC to manage huge-sized data. Data can also be stored in compressed form. This DMC architecture issue is being reviewed according to forthcoming data distribution services optimized for parallel computing on Beowulf workstation cluster using MPI [Gropp 2000a, 2000b].

5. Conclusions

In the future, a FITS file implementation of the DMCI will be developed. Modules that rely on DMCI will be able to store data within database structures or FITS files transparently. JDO technology will let DMC deal Java objects to any transactional data store transparently. The DMC was released in late October 2003, after the completion of alpha testing campaign. DMC is being currently tuned while undergoing beta tests at the Max Planck Institute for Astrophysics and at the LFI DPC pipeline integration site [Zacchei, 2004].

Acknowledgments. We wish to thank the Research and Science Support Department of ESA ESTEC for their alpha testing activity, Max Planck Institute for Astrophysics for their beta testing activity and the Planck IDIS community and the LFI DPC Consortium Institutes for comments and suggestions.

References

Bosio et al. 2003,Computing in High Energy Physics (CHEP 2003).

Gray et al. 2004, this volume535

Gropp W. et al, 2000a, Using MPI, MIT Press.

Gropp W. et al, 2000b, Using MPI-2: Advanced Features, MIT Press

Lama et al. 2002,Planck Int. Doc., IDIS DMC Architectural Design Doc.

Pasian et al. 2004, this volume257

Segal 2001, DataGrid Data Management (WP2) Architecture Report.

Smareglia et al. 2004, this volume674

Stockinger et al. 2001, European High Performance Computing Conference.

Taylor et al. 2003, in ASP Conf. Ser., Vol. 295, ADASS XII, ed. H. E. Payne, R. I. Jedrzejewski, & R. N. Hook (San Francisco: ASP), 325

Vuerli et al. 2001a, Planck Int. Doc. IDIS DMC Users Requirements Doc.

Vuerli et al. 2001b, Planck Int. Doc. IDIS DMC Data Model Specification

Zacchei et al. 2004, this volume396