Earth Observation Applications Approach to Data and Metadata Deployment on the European DataGrid Testbed

L. Fusco[1], J. Linford[1], W. Som deCerff[2], C. Boonne[3], C. Leroy[3], M. Petitdidier[3]

1 ESA-ESRIN, Frascati, Italy
2 Royal Dutch Meteorological Institute, de Bilt, The Netherlands
3 Institut Pierre Simon Laplace, Université Pierre et Marie Curie, Paris, France

I. Introduction

The European DataGrid (EDG) experience represents an important challenge for the current state-of-the-art in developing and deploying extended, large scale, Grid testbeds. The Earth Observation (EO) community has taken the EDG opportunity to examine the correspondence between EO application needs and actual and potential functionalities offered by Grids today. The application chosen for deployment is a typical one in the EO domain. It consists of elaborating low-level (e.g. raw) satellite data into higher-level products and validating them against similar products obtained from instruments using other platforms, such as ground-based instruments, balloon and other airborne observations, observations from ships, etc.

The selected EO Application usecase involves processing and validating global atmospheric ozone observations made by the GOME instrument flying on board the European ERS satellite, throughout a 7-year mission. This requires processing and maintaining a large set of data files distributed on the Grid and the timely location and retrieval of small subsets of selected data files for the validation. This requires the creation and deployment of metadata catalogues that can be updated and accessed by partners with different roles in the collaboration, while providing secure access control and restricted accesses to both data and catalogues.

The EO application partners involved are the European Space Agency (Frascati, Italy), the Netherlands Royal Institute of Meteorology (KNMI, de Bilt, Holland), and Institute Pierre Simon Laplace (IPSL, Paris France). The partners formed a virtual organization of more than fifteen scientists and engineers who collaborated on setting up the EO Grid infrastructure and deploying the application. The partners' main objectives were to (1) demonstrate how Grid can respond to the complexity and the constraints imposed by applications in EO domain and (2) identify the benefits of the technology and how it can improve the work of EO technical and scientific users.

This paper aims to point out how these objectives are actually or potentially fulfilled by EDG middleware and services. An overview of the DataGrid middleware and infrastructure is followed by a description of the application. The problem of metadata handling and the use of different metadata catalogue services is then addressed and the three different solutions experimented with to finally deploy the application on the Grid are described.

II. DataGrid Testbed overview

The DataGrid middleware [R1] extends considerably the functionalities provided by the underlying Globus (GT2) [R2] and Condor [R3] technology. The testbed itself [R4] consists of several sites distributed in France, UK, Switzerland, Italy, Germany, the Netherlands and Spain. Each site offers one or more high-capacity processing and storage resources - Computing Elements (CE) and Storage Elements (SE) in EDG terminology, that register their presence and capabilities in the Information Index (II). This information can then be searched by one or more Grid Resource Brokers (RB), which match incoming job requirements against available resources and select the 'best' resource where to execute the job. The Resource Broker optimisation techniques include the selection of Computing and Storage Elements based on the jobs' declared data requirements. Job requirements are described using Job Description Language (JDL) which allows the application to specify required attributes to be matched against the available resources' attributes. The DataGrid workload management middleware takes care of selecting the target resources (CE and SE), submitting the job (executable and auxiliary files), monitoring the execution and retrieving the results. The EDG Storage Element may be a stand-alone disk pool or a front-end cache to a Mass Storage System.

The DataGrid Replica Location Service (RLS) [R4] manages a set of distributed Local Replica Catalogues (LRC) and Replica Metadata Catalogues (RMC), that allow the registration and subsequent location of individual data files and their replicas, which are distributed on the available Storage Elements. EDG middleware also provided Spitfire, a Grid-aware, web-service interface to a back-end SQL database that can be used by Grid applications needing a database service. LRC, RMC and Spitfire share the same technology, which is java-based servlets running in a servlet container connecting to a back-end RDBMS through JNDI.

The middleware has continued to evolve throughout the duration the three-year project; the Storage Element has converged towards a more standard Storage Resource Manager (SRM) service [R6]. Security was initially based on GSI and allowed 'binary' access control on a per-VO basis. Later, a Virtual Organization Management Service (VOMS) was introduced which provides fine-grained, role-based access control [R7]. Initially the Globus MDS Information System was deployed and was later replaced by the R-GMA system [R8] developed by the project.

III. EO Application

III.I. Description

The GOME Ozone profiling application was chosen as an ideal candidate for evaluating the DataGrid testbed. The large data volumes and large number of files, the processing-intensive nature of the scientific algorithms and the scattering of datasets, processing resources and participating organizations over an extended geographic area, are all factors where Grid technology can offer improvements over conventional computing solutions. Furthermore, since the application is fairly representative of the product processing, refinement and quality control procedures that routinely take place in the EO applications domain, the problems and solutions encountered can be considered representational of a wide range of Earth Observation applications.

Dataset	Number of files handled (per year)	File Size
Level 1	4,724	15 Mb
Level2 (NNO)	4,724	19.5 Mb
Level2 (OPERA)	9,448,000	12 Kb
Lidar	12	2.5 Mb

Table 1. GOME application data volumes

The entire GOME dataset, consisting of 7 years of global ozone observations by satellite, was to be re-processed by two different ozone profiling algorithms, OPERA and NNO [R9]. The results of the satellite observations would then be validated against ground-based Lidar observations. The OPERA algorithm, based on Optimal Estimation, is developed by KNMI (Holland), while the NNO method, based on Neural Networks, is developed jointly by the University of Tor Vergata and ESA-ESRIN (Italy). The LIDAR observations are extracted from the Network for the Detection of Stratospheric Change (NDSC) database by IPSL (France), who also developed the validation algorithm.

A usecase was developed to describe the complete, end-to-end processing and validation chain (Figure 1). The usecase involves four distinct datasets shared among the three EO institutes: Level1 raw data, Level2-OPERA and Level2-NNO products and LIDAR data. Each year of observations requires approximately 267 GB of data contained in several million files (Table 1). The OPERA and NNO algorithms used different approaches to storing the Level2 profiles. OPERA splits each orbit into several thousand Level2 pixel files, while NNO keeps the Level2 profiles together in a single orbit file.

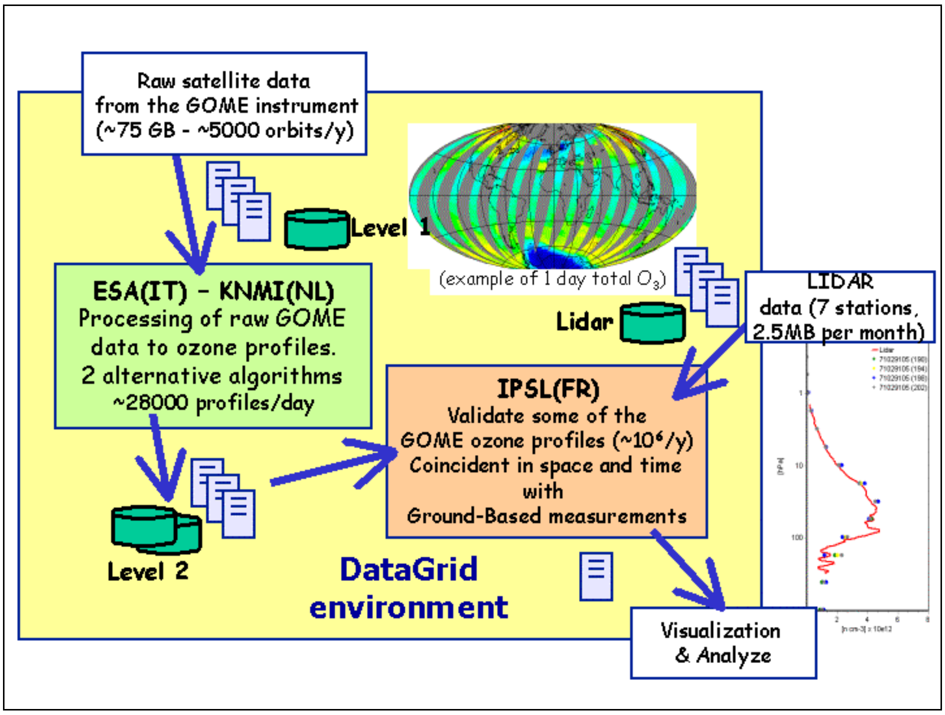

Figure 1. Scheme for GOME profile processing and validation using DataGrid

The discrete steps of the processing chain are shown in Figure 2. This requires transferring the Level1 data from EO archives to DataGrid Storage, running the two different processing algorithms to produce the Level2 data and then running the validation procedure to verify the results. Each institute in the collaboration carried out a particular role: ESA-ESRIN was responsible for transferring the GOME Level1 data to the Grid testbed, both KNMI and ESA-ESRIN were responsible for processing Level2 products, while IPSL was responsible to extract the Lidar data from the NDSC database, to transfer them to the Grid testbed and to carry out the validation.

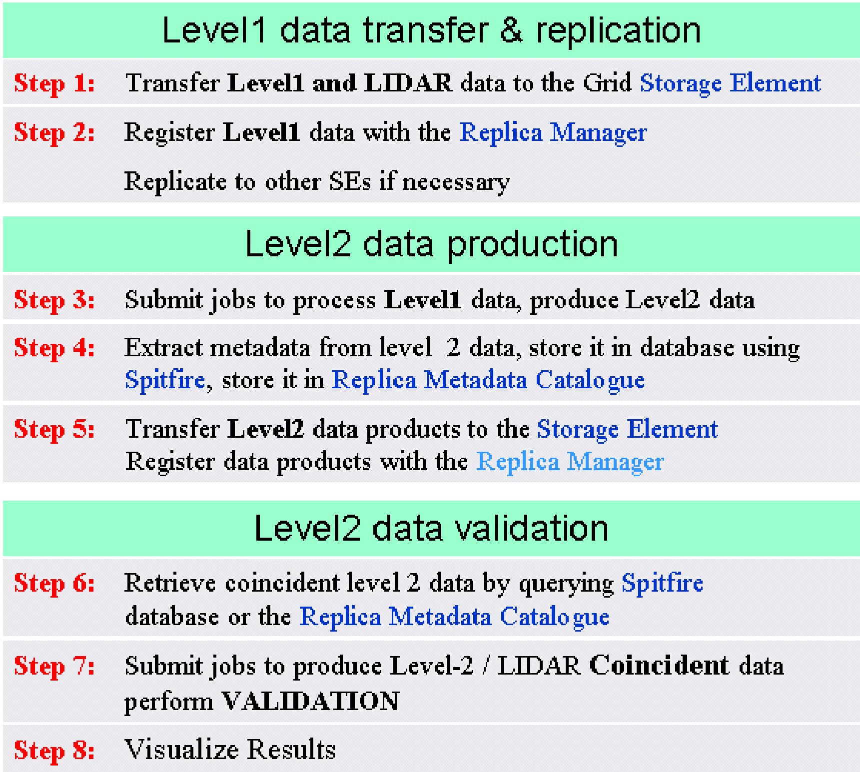

Figure 2. Discrete steps of the end-to-end GOME data processing and validation chain

III.II. Large dataset distribution on the available Grid Storage

Before the Level1 data could be processed it needed to be transfered from EO Mass Storage Archives and distributed on Grid Storage elements, where it could be accessed by processing jobs running on the Grid worker nodes. One approach has been to access the data at runtime directly in the archive system, via GridFTP. However the archive has a relatively high access latency and this would create bottlenecks when a large numbers of jobs were submitted simultaneously to the Grid. This was resolved by doing a bulk dataset transfer to Grid storage in advance of scheduling the processing tasks. The files were transfered to Storage Elements distributed throughout the Grid, their locations were registered, along with a system-generated GUID, in the LRC. Meaningful aliases were associated to the assigned GUIDs using the RMC, allowing the files to be located by the application later on. A strategy for dataset distribution and balancing the storage load was developed and deployed which resulted in an even distribution of the dataset over the available storage resources. Later, when jobs were submitted to the Grid for processing, the EDG RB used the Replica Optimisation Service to ensure the jobs were sent to execute on nodes that were closest to the data sources.

III.III. Metadata catalogue creation and data retrieval

In the EO applications domain the use of metadata catalogues, organized in databases, is very common. There is frequently a need to select data corresponding to given geographical and temporal coordinates, or specific data-quality values, algorithm version, etc. By searching the metadata catalogues for the corresponding tuples of the metadata attributes, EO users can identify specific subsets of the data, which cover specific 'areas of interest' being investigated. Using the information returned by the metadata searches, the physical data files can be located and retrieved.

Different tools and methods were available to create and access metadata catalogues. Three different solutions were experimented, Spitfire, MUIS and RMC. Spitfire, developed by EDG, provides transparent and secure access to databases (e.g. MySQL) for Grid middleware and applications, while MUIS is the proprietary ESA EO product catalogue. RMC is the EDG Replica Metadata Catalogue, whose job is to maintain aliases and alias attributes for data files registered in EDG Replica Catalogues (both also powered by Spitfire).

Spitfire

Two Spitfire databases were set up for Lidar data and Ozone profile metadata. Programs were written to query these databases and retrieve the relevant filenames for specific time and geo-location criteria. The EDG Replica Catalogue was then used to retrieve the physical locations of the files and the physical data could then be retrieved using GridFTP.

Web Portal

The Level1 satellite data are stored in orbit files and the details are inserted in the ESA product catalogue, MUIS. A web-portal was constructed to access both the MUIS catalogue and the Lidar metadata base. Using the Portal is was possible to precisely pick the orbit(s) passing over a chosen Lidar station. The portal was also capable of transfering the orbit data from the ESA Archives to the Grid, and to submit jobs to the DataGrid to process them into Level2 products and to carry out the validation.

RMC

While both Spitfire and the Portal methods offered 'stand-alone' solutions (i.e. independent of the Grid Data Management services), the Replica Metadata Catalogue introduced in the final release of the testbed middleware offered a Grid metadata service integrated with the EDG Replica Catalogue.

Throughout the processing and validation chain, whenever new EO data was introduced to the Grid - whether Level1 data transferred from EO archives using GridFTP, or Level2 products processed by jobs running on the EDG worker nodes - it was transferred to a Storage Element. The file details (logical name and physical location) were registered in the Replica Catalogue and the metadata details were registered in the metadata catalogue (e.g. Spitfire or RMC). Selecting the particular Storage Element was either done 'manually' by the user, based on available SE information published in the II, or automatically by the Resource Broker, based on the input data requirements specified in the JDL.

Using either the filename or a combination of metadata keys, specific data products could be easily located and retrieved, regardless of their particular location. Furthermore, the Replica Optimisation capability of the Resource Broker could be fully exploited to ensure that jobs were sent for execution 'near' to the data sources.

IV. Deployment

The DataGrid testbed provides a widely distributed computing environment with sufficient storage and computing capacity to meet the needs of the application. The testbed middleware also allows transparency with regard to the various scattered locations of application users, datasets and computing resources, with basic security access-control provided by GSI. The computing task to be performed was both data- and processing-intensive and had to be broken down into several thousand Grid jobs. Level1 data was first replicated to the available Grid Storage Elements and registered in the Replica catalogue. Each job then processed one or more raw satellite observations (Level1 data) and produced the corresponding ozone profiles

(Level2 data). The ozone profiles were then checked against the LIDAR profiles using the validation algorithm.

The optimal deployment of the GOME application on the DataGrid testbed aimed to maximize use of the available processing and storage resources, while keeping to a minimum the volumes of data that needed to be transferred between Grid nodes. The approach taken relied on the capability of the Resource Broker, using the Replica Optimisation Service (ROS), to select the processing resources by evaluating a data access cost for each positive match between the job requirements and a resource's capabilities.

V. Conclusion

Our approach towards Grid data handling aims for transparency with regard to the actual location of Grid data - the data is simply "somewhere on the Grid". This is a very powerful feature in a Grid architecture, such as DataGrid, where the location and retrieval of data and replicas are handled transparently by the Grid middleware. The data management mechanisms include the transparent optimisation of data access cost. Such a scheme leads to better load balancing and increased exploitation of available resources (including network bandwidth). In such a scheme, the user can be generally unconcerned about the location of the data and simply refers to it using a logical name (a user-assigned alias) or a Globally Unique Id (GUID) that is assigned by the middleware. EDG Data management middleware initially provided the "logical file name", or alias, which can be associated with one or more copies of data files on the Grid. In response to the applications' requirements, this was later extended with the ability to associate user-defined metadata attributes with the aliases. Using a combination of aliases and metadata attributes, the application can quickly identify specific data file subsets out of the millions of files distributed on the Grid storage.

Furthermore, datasets can be treated as 'virtual', by triggering the submission of jobs to the Grid to process products that are not yet available. As long as all 'ingredients' needed to generate a product are available, the end product can be regenerated 'on the fly'. This approach is particularly useful in Earth Observation, where information is often only needed in the vicinity of 'areas of interest', that may be defined for example, in terms of geographical coordinates and time of observation. In such cases processing an entire global historical dataset is unnecessary.

Grid is often billed as the next-generation World Wide Web. We often hear about topics associated with Grid such as Semantic Web [R10], Semantic Grid. This points to a vision of Grids becoming as universal and easy to use as the World Wide Web is today. However, this will include a large amount of added, autonomous functionality, for example, the ability of the middleware to make decisions about which services and data should be used. The approach to data we have described presents the Application end-users' point of view, which is keenly oriented towards this vision.

The unprecedented scale of EDG - tens of sites, hundreds of CPUs, hundreds of users and thousands of concurrent jobs - has tested the performance of the deployed Grid Information Systems (both MDS and R-GMA solutions were tried). The Applications' Grid data requirements similarly impacted the data management middleware solutions being developed.

The results lead towards the development of new and improved models and allow us to take on board many lessons learned. One important lesson is the need for fine-grained access control as an integrated part of data management, both for accessing and updating replica and metadata catalogues and for storing and retrieving data.

V.I. Lessons Learned

We believe the emerging standards at the present state of their development are 'low-level', i.e. geared towards basic system-level solutions. However, it is very important that the current standards at least address fundamental application requirements and indeed, our experience with DataGrid indicates these are largely satisfied, or are at least on the way to soon being satisfied.

Among the basic application requirements for Grid data management are to be able to store, locate and retrieve application data distributed throughout the Grid, which can be largely satisfied by the combined use of Replica Catalogue (RC) and Storage Resource Manager (SRM) interfaces. The RC allows us to remember where on the Grid we stored a particular replica and the SRM allows us to store and retrieve replicas without having to be concerned with the internals of the particular type of storage device being used at the different sites. However, we have demonstrated that is very useful if the RC allows datasets to be identified (referred to) by the application not just by assigned logical filenames, but also by metadata content. For example, a dataset on the Grid consisting of all GOME Level1 data files for observations made during October 1997 may be defined as all files whose logical filenames begin with the identifier 'GOM_9710'. However the other way to define the same dataset, i.e. all files with metadata keys DATE.month = 10 and DATE.year = 1997 and SENSOR = 'GOME', is often more useful and easy to use, since we do not have to know a priori the logical file names that were assigned.

Concerning the SRM, there is currently a lack of support for storage space management, something that only becomes critically evident when maximum use of storage resources is reached by the application. Standards are needed to control how the available space of single storage resource (e.g. a disk pool or cache front-end to an archive system) can be distributed among the different VOs and users. Perhaps this may be achieved by the use of some policy-based method. Data entrusted to remote sites for storage will usually be of some value to the user, therefore some QoS guarantees are needed to protect inadvertent 'misuse', loss or damage (again, possibly using policy-based agreements). Before transfering data - indeed, before submitting any Grid jobs which produce new data, users need some way to ensure that enough storage space is available. Here both GridFTP tools and job-submission tools are involved and advance storage space reservation may be required. However, any proposed solution should not impose the obligation of advance space reservation on every user - there should always be some default controls in place that perform some optimal action if the user does not issue any specific directives. Failure of these tools/services to correctly verify available space currently produces the result 'undefined', should a problem occur. The 'out of space' error condition is not dealt with or trapped specifically by the Grid middleware layer running on top of the local system; in the event, the user gets a cryptic error message 'end-of-file was reached'. There are many cases where users receive error message that are difficult to interpret or misleading, due to insufficient handling of low-level system conditions by the middleware. Standards for Grid error handling however are as yet non existent. Perhaps there will someday exist a standard set of Grid error conditions comparable with the http style '404 page not found' error classifications.

Finally, applications need to be able to share data in a secure way, yet the imposed security mechanisms need to be sufficiently flexible to allow fine-grained control, to allow different users to produce data that other VO members are able to read but not modify. Such security standards need to be an integral part of the data management.

References

[R1] a) German Cancio, Steve M. Fisher, Tim Folkes, Francesco Giacomini, Wolfgang Hoschek, Dave Kelsey. The DataGrid Architecture Version 2 (on the Web: **http://grid-atf.web.cern.ch/grid-atf/doc/architecture-2001-07-02.pdf**)

b) The European DataGrid project: **http://www.eu-datagrid.org**

[R2] Ian Foster, Carl Kesselman, Steven Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. 2001, Lecture Notes in Computer Science vol. 2150

[R3] Douglas Thain, Todd Tannenbaum, and Miron Livny, "Condor and the Grid", in Fran Berman, Anthony J.G. Hey, Geoffrey Fox, editors, *Grid Computing: Making The Global Infrastructure a Reality*, John Wiley, 2003. ISBN: 0-470-85319-0

[R4] DataGrid Testbed User Guide (web page documentation) **http://marianne.in2p3.fr/datagrid/documentation/EDG-Users-Guide/index.html**

[R5] Peter Kunszt, Erwin Laure, Heinz Stockinger, Kurt Stockinger; REPLICA MANAGEMENT WITH REPTOR, EU-DataGrid Technical Document **https://edms.cern.ch/file/380482/1/ReptorEDG-TR-2003-04-08.pdf**

[R6] Junmin Gu, Alex Sim, Arie Shoshani, Lawrence Berkeley National Laboratory; The Storage Resource Manager Interface Specification (available from SRM working group home page **http://sdm.lbl.gov/srm-wg/index.html**)

[R7] The Virtual Organisation Membership Service (web page documentation)

http://hep-project-grid-scg.web.cern.ch/hep-project-grid-scg/voms.html

[R8] R-GMA Relational Grid Monitoring Architecture (web pages documentation) **http://www.r-gma.org**

[R9] F. Del Frate, M. Iapaolo, S. Casadio, S. Godin-Beekmann and M. Petitdidier. On the retrieval of ozone profiles using satellite data and neural algorithms. Paper submitted to Remote Sensing of the Environment journal

[R10] Tim Berners Lee, James Hendler, Ora Lassila. The Semantic Web. Scientific American, May 2001.