Development of a Personal Data Service and an Evaluation of its Performance

Petros Pistofidis

Master of Science

School of Informatics
University of Edinburgh

2006

Abstract

This master thesis aims to address the need, originating from the community of e-scientists and professionals, for a grid data management tool. The primary goal is to create a prototype for a personal grid catalog, which offers flexible mechanisms to handle the diverse and distributed nature of grid data. The basic concepts that support this catalog are namespace collections and dynamic compilation of stored views. The namespaces enable the grid user to organise and manage the catalog items in collections. The dynamic views store query statements, which can be executed on-demand by the grid user. The proposed catalog system is supported by an XML data model, which aims to serve the description of the catalog's structure as a composition of brief grid data resource descriptions. The efficiency of the proposed grid catalog is evaluated based on performance measurements. These measurements capture the catalogs abilities to interact with remote data resources and present its internal collection structure.

Chapter 1

Introduction

The Internet and recently the Grid have provided a fertile ground for the development of IT infrastructures that enable the advanced handling of distributed data. An increasing number of individuals, both researchers and professionals, face the challenge of organizing and managing large sets of unsorted data. These data either compose their current working environment, or serve as data banks for on demand retrievals.

The principal goal of this project is to develop a prototype for a personal data service that will address this challenge. Its aim is to deliver an application environment, called Grid Data Catalog (GDC), which will provide versatile access and management over diverse grid data. The targeted user group is the constantly expanding community of e-scientist and business professionals who utilise and interact with grid infrastructures. The primary concept of this application's implementation is the management and grouping of data resources in user-defined collections (also referred as namespaces).

The structure of this thesis is composed so that it can provide a solid background, which later on will serve as the knowledge basis for the arguments supporting the project's design decisions. The first chapter is an introduction that states the problem space, the aim and the requirements of this project. In chapter two, we provide a short description of the models, architectures and frameworks which coordinate the design and support the implementation of this project. The third chapter describes the general architecture of GDC with an overview of its basic components and their role in it. This chapter also pinpoints the elements of GDC that differentiates it from the existing platforms and implementations in the same problem space.

Each of the three chapters following the general architecture focus on one of the three main components of the project's implementation. For each component we define the characteristics that imprint its role and state the requirements it has to fulfil. Start-

ing from the server side and moving to the client side and the data model we list and describe the available options that could serve our design and development. A small analysis supports each decision made on whether to facilitate or reject an approach, based on the projects specifications. At the end of each chapter implementation-specific details are presented to map the design decision to GDC's developed modules and functionality.

After the thorough description and analysis of the projects developing process, the seventh chapter addresses the evaluation of the delivered result. Application responsiveness and time measurements act as the basis of our empirical evaluation. Furthermore, design coverage, level of scalability and extensibility potentials act as the criteria to evaluate the decisions made for each component respectively; which constitutes our analytical evaluation.

Finally, the last chapter summarizes with the conclusions drawn from this project. We also propose future extensions and changes that can scale and upgrade the Grid Data Catalog.

1.1 Project's Problem Space and User Group

Inspecting the data related needs of industry and science is the best way to measure the importance of a service/application such as the one this projects aims to implement. Its practical value is great, especially when considering activities that involve handling large volumes of diverse data. The following advancements and noted facts state an immediate need for advanced Data Services:

- Heterogeneous distributed data are produced in an increasing number of research and professional environments. Scientists need to share, manage and conduct research on large collections of scientific data. At the same time, they need to keep track of parameter values and store configuration files that are essential for their everyday research activity. Focusing on another area, business professionals need to periodically store and update complex legacy data. They also need easy and fast access to remote sets of effectively organised corporate documents, template files and multimedia content.

- The adoption rate of grid technologies, as a tool to manage distributed data is rising in all sectors. Industrial, Government and Corporate Grids are emerging to provide monitored and secure access over specialized data. Large hierarchies

of working professionals become certified grid users. User communities and academic groups deploy, test and utilize the potentials of grid middleware. Many universities provide CAs(Certificate Authority) and controlled grid data access to its students and staff.

1.2 Project's General Specifications

This project aims to address the needs of the user groups described above. Based on these stated needs, we can produce and formulate a set of general specifications for an application tool that can satisfy them sufficiently and effectively. We list these specications in priority order, according to their importance in the context of such an grid tool:

1. Index, store and list user connected data, through a mechanism that maps distributed heterogeneous data to names.

2. Define and compose collections of data. Populate and manage collections' content through a naming and referencing model.

3. Creat transparent access mechanisms that create a virtual personal data space for each user. Support efficient use of a wide scope of data repositories (Relational Databases, XML Databases, File systems).

4. Provide a uniform interface, with minimum requirements on client's side to enable catalog access form whatever site the user is connected at.

5. Utilize grid accessing and security mechanisms. Develop an application which corressponds to the needs of data centric Grid environments.

6. Base implementation on widely accepted frameworks that promise interoperability with existing platforms.

1.3 Project's Functional Requirements

The specifications, outlined in the previous section, can be mapped and translated into an essential set and functionalities and mechanisms which the Grid Data Catalog system must support. The user will interact with the GDC application in order to manage a personal catalog of data and metadata. In the layered architecture of grid software

engineering, GDC serves as a module located in the client application layer. Its primary role is to expose management actions on scaled sets of name-value pairs, which constitute direct or indirect references to grid data. To elaborate, the value in the catalog entries should be in the form of individual data resources, collections of data or dynamic views like, for example, a query for calculating the result value on the fly, or from data that has already been materialized and stored.

Since the proposed environment aims to participate in the context of Grids, references to open-ended heterogeneous data are expected to populate it. Metadata and Naming are currently very efficient techniques to assure flexibility in data aggregation and manipulation. Sharing collections and individual data can be easily and securely provided. Each collection name should identify a time-varying set of values currently associated with the name. The user and the client application are not responsible for organizing storage or data movement of the referenced data in this named collection; this is automated and handled by the underlying middleware.

In the proposed system there will be mechanisms for:

- Creating a named space. A locally named collection connected to a set of data.
- Initiating a new session of that named space possibly in a different computational environment each time.
- Storing name-value pairs in that space where names are relative to that named space. value may be of any type, e.g. string, real number, matrix, relation, XML doc, etc. It may be by copy or by reference. Reference could be a query or any other value forming expression.
- Retrieving name-value pairs for local use or to dispatch to a third party.
- Removing name-value pairs from the named space.
- Inspecting and managing the contents of that named space.
- Ending a session of use of that specific named space.
- Terminating the life of a named space.

Essentially the above set of actions will be mapped to handling operations of the catalog entries. The catalog will hold both actual data and metadata for data description and named-spaced description. The schema of the data model supporting the catalog should be capable of effectively describing:

- User data resources. A flexible schema that can support the description of a wide variety of data types. From direct static data, to dynamic views or references of data (e.g queries, remote file locations). From primitive data types to composite data structures. From bytes of configuration parameter values to terabytes of raw experimental data files.

- User collections information. A powerful schema which efficiently stores the contents and structure of each namepace. The performance of the managing actions will largely depend on the quality of this schema.

1.4 Project's Contribution

The benefits of a Grid Data Catalog could well serve the future work and research initiatives in web and grid problem space. A successful and efficient implementation of this service could lead to a uniform client tool for cross platform and cross middleware data management. Disjoint from science specific data types and limitations, this tool could provide a standard solution on top of data services. A solid base which can be extended and customized to serve science or business focused needs.

Furthermore, this work aims to contribute in the composition or refinement of a data services design pattern. Built on recognised and widely accepted frameworks, the Grid Data Catalog could serve as a testing model. Measuring and tuning its implementation performance could provide valuable feedback to Grid Architectures and Grid Engineers.

Finally, this data management client could fuel the advance of other data oriented research areas. Semantic Grid, Data Mining, Data Federation and Data Visualisation could benefit from efficient exposing of grid data.