Framework for Querying Distributed Objects Managed by a Grid Infrastructure*

Ruslan Fomkin and Tore Risch

Department of Information Technology, Uppsala University,
P.O. Box 337, SE-751 05 Uppsala, Sweden
{Ruslan.Fomkin, Tore.Risch}@it.uu.se

Abstract. Queries over scientific data often imply expensive analyses of data requiring a lot of computational resources available in Grids. We are developing a customizable query processor built on top of an established Grid infrastructure, the NorduGrid middleware, and have implemented a framework for managing long running queries in Grid environment. With the framework the user does not specify the detailed job and parallelization descriptions required by NorduGrid. Instead s/he specifies queries in terms of an application-oriented schema describing contents of files managed by the Grid and accessed through wrappers. When a query is received by the system it generates NorduGrid job descriptions submitted to NorduGrid for execution. The framework considers limitations of NorduGrid. It includes a submission mechanism, a job babysitter, and a generic data exchange mechanism. The submission mechanism generates a number of jobs for parallel execution of a user query over wrapped data files. The task of the babysitter is to submit generated jobs to NorduGrid for the execution, to monitor their execution status, and to download results from the execution. The generic exchange mechanism provides a way to exchange objects through files between Grid execution nodes and user applications.

1 Introduction

Nowadays a lot of scientific data are stored in Grids. Scientists need to access and analyze them. Their analyses often imply expensive computations that need to process a lot of data. Thus scientists need to use external computational resources to process their analyses, and storage resources to store and share huge amounts of data. For this many Grids are developed to provide computational resources and storage facilities.

For example, the ATLAS collaboration [1] motivates many Grid projects such as LCG [2], EGEE [3], and NorduGrid [4]. These projects provide storage facilities to store and share data produced by ATLAS [1] and to be produced by the Large Hadron Collider (LHC) [5], along with computational resources to analyze the data.

* This work is funded by The Swedish Research Council (VR) under contract 343-2003-955.

J.-M. Pierson (Ed.): VLDB DMG 2005, LNCS 3836, pp. 58–70, 2005.

A typical analysis of data for the LHC projects is selections of subsets of the input data. The selections, called *cuts*, consist of not only simple logical predicates but also numerical computations. We show that such analyses can be expressed in a declarative way using an extensible query language.

We are developing *POQSEC* [6] (*Parallel Object Query System for Expensive Computations*) that processes scientific analyses specified as declarative SQL-like queries over data distributed in the Grid. It utilizes computational resources of Swegrid [7] and storage resources of Nordic countries through the middleware Grid infrastructure NorduGrid [4]. The goal of the POQSEC project is to provide a transparent and scalable way to specify and execute scientific queries. A user should be able to specify his/her query transparently in a client database without respect to where it will be executed and how data will be accessed.

Currently we have implemented a framework for submitting user queries for execution in the Grid. The system then creates jobs executing the queries, submits the jobs to NorduGrid, monitors execution of the jobs by NorduGrid, downloads results of the jobs, and delivers results of the queries to the user. The user states queries to POQSEC in terms of a database schema available in the client database. The schema contains both an application-oriented part and Grid meta-data. The application schema describes data stored inside files in Grid storage resources, for example events produced by ATLAS. Wrappers are defined for accessing the contents of these files, e.g. in our application we use a wrapper of the ROOT library [8]. The Grid meta-data contains information about the files. Thus user queries can restrict data both in terms of application data contents and meta-data about files. The latter is very important since there is a huge amount of Grid data files and queries are normally over a small percentage of them. User queries are parallelized to a number of jobs for execution. The parallelization is done by partitioning data between jobs. Our preliminary results show that the parallelization gives significant performance improvements.

The rest of the paper is organized as follows. Related work is discussed in Sect. 2. Section 3 describes the POQSEC architecture. It is followed by a description of an application from High Energy Physics, which is our test case. The implementation of the framework is discussed in Sect. 5, and Sect. 6 concludes the paper.

2 Related Work

Another system that utilizes a Grid infrastructure and provides high-level declarative query language for data access and analysis is Distributed Query Processing system (DQP) [9] or its web service version OGSA-DQP [10]. The DQP is part of the Grid infrastructure myGrid [11], which fully controls resources and where resources can be allocated dynamically. The resources for the query execution are allocated and provided by a user. Any of them can be utilized by DQP dynamically. It is different from our system where NorduGrid is a middleware above autonomous local batch systems that control computational resources. Unlike the DQP, we need to consider the NorduGrid limitation that jobs are not