An Adaptive Distributed Query Processing Grid Service

Fabio Porto[1], Vinícius F.V. da Silva[2], Márcio L. Dutra[2,3], and Bruno Schulze[2]

[1] EPFL - School of Computer and Communication Sciences,
Database Laboratory, Lausanne, Switzerland
fabio.porto@epfl.ch
[2] National Laboratory for Scientific Computation - Computer Science Dep,
RJ - Brazil
[3] Military Institute of Engineering - System Engineering Dep, RJ - Brazil

Abstract. Grid services provide an important abstract layer on top of heterogeneous components (hardware and software) that take part into a grid environment. We are developing a data grid service prototype that aims at providing transparent use of grid resources to data intensive scientific applications. Our prototype was designed having as target three main issues: (1) dynamic scheduling and allocation of query execution engine modules into grid nodes; (2)adaptability of query execution to variations on environment conditions and (3) support to special scientific operations. We propose a new node scheduling algorithm and show how it can be integrated into a simple distributed and parallel query optimization strategy. Our implementation demonstrates a speedup of 16.6 with 18 scheduled nodes and a steady throughput rate, obtained applying a dynamic adaptive strategy.

1 Introduction

The development of grid services as proposed by the Open Grid Services Architecture OGSA [1] promotes the isolation of user applications in respect to the heterogeneity inherent to the grid environment. While middleware systems like the Globus toolkit [2] offer some basic functionalities on top of a grid infrastructure, mainly regarding: authentication, remote task scheduling and file transfer, this is not enough for the deployment of complex applications involving the processing of users programs and the access to distributed data. For these more complex types of applications an extension was proposed of the web service technology towards (grid) web services, where services state and life cycle can be managed [3]. This extension was first proposed in the OGSI - Open Grid Services Infrastructure and more recently has been turned into the WSRF Web Services Resource Framework [3]. The idea is that by designing and composing grid services, one may achieve higher level functionalities specific tailored to the envisaged application and still support the fundamental characteristics offered by established distributed systems such as Common Request Broker Architecture CORBA, from the Object Management Group. Regarding data services for

J.-M. Pierson (Ed.): VLDB DMG 2005, LNCS 3836, pp. 45–57, 2005.

the grid, some projects like OGSA-DAI [4] and Data Grid [5] aim at providing a uniform service interface for data access and integration of databases in the grid, in addition to higher-level services like data replication.

Our project contributes to the grid data service layer by conceiving high-level services for data intensive grid applications. In this context, we are developing a Configurable Data Integration Middleware for the grid (CoDIMS-G) that is a distributed grid service for the evaluation of scientific queries. The design of CoDIMS-G focused on conceiving efficient and adaptable query evaluation strategies for the grid environment. Efficiency comes with adequate initial resource allocation based on grid nodes historical profiles and managing inter-node communication. Adaptability is the result of online routing decisions and transfer block size management.

We designed a query optimization strategy tailored for queries including intensive computations, like scientific programs. The strategy integrates a new node scheduling algorithm that selects grid nodes for parallel evaluation of fragments of the query execution plan. In addition, we implemented a distributed query engine, running as grid services, that adapts execution to the actual data flow in allocated grid nodes. We extended the Eddy operator [6] to deal with the adaptivity of tuple block size routing through scheduled nodes and to implement an iteration mechanism over a query execution plan fragment, as required by loop control found in some scientific applications.

We currently use CoDIMS-G to support the pre-processing stage of a scientific visualization application (SVA)that is being developed at the National Laboratory of Scientific Computing (LNCC), in Brazil, as a testbed for our prototype. The SVA computes the path of virtual particles through a fluid flow [7]. Our challenge is to minimize the elapsed-time for pre-computing the virtual particle trajectory. We modelled this scenario as a database problem and used CoDIMS-G to generate and evaluate a parallel query execution plan that transparently access grid resources and dynamically adapt to unforeseen fluctuations during query execution.

Our implementation demonstrates a speedup of 16.6 [1] with 18 scheduled nodes and a steady throughput rate.

The rest of this paper is organized as follows. In section 2, we present important work covering grid data services and scientific visualization. Next, section 3 introduces our running example based on the pre-processing stage of fluid path visualization. Section 4 introduces the CODIMS-G architecture and section 5 presents the distributed query optimization strategy. Emphasis is given on the G^2N scheduling algorithm and its integration into the query execution plan. Section 6 presents the query execution engine components and discuss the influence of our distributed adaptive strategy into query execution. Section 7 presents initial results obtained running CoDIMS-G in a controlled environment with up to 20 grid nodes. Finally, we present our conclusions and future work.

[1] We computed speedup as the ratio between the elapsed-time of a centralized execution and a parallel one.