Dynamic Query Optimization Approach for Semantic Database Grid

Xiao-Qing Zheng (郑晓庆), Hua-Jun Chen (陈华俊), Zhao-Hui Wu (吴朝晖), and Yu-Xin Mao (毛郁欣)
Grid Computing Lab, College of Computer Science, Zhejiang University, Hangzhou 310027, P. R. China
E-mail: zhxqingcn@zju.edu.cn, huajunsir@zju.edu.cn, wzh@zju.edu.cn, maoyx@zju.edu.cn
Revised May 6, 2006.

Abstract Fundamentally, semantic grid database is about bringing globally distributed databases together in order to coordinate resource sharing and problem solving in which information is given well-defined meaning, and DartGrid II is the implemented database grid system whose goal is to provide a semantic solution for integrating database resources on the Web. Although many algorithms have been proposed for optimizing query-processing in order to minimize costs and/or response time, associated with obtaining the answer to query in a distributed database system, database grid query optimization problem is fundamentally different from traditional distributed query optimization. These differences are shown to be the consequences of autonomy and heterogeneity of database nodes in database grid. Therefore, more challenges have arisen for query optimization in database grid than traditional distributed database. Following this observation, the design of a query optimizer in DartGrid II is presented, and a heuristic, dynamic and parallel query optimization approach to processing query in database grid is proposed. A set of semantic tools supporting relational database integration and semantic-based information browsing has also been implemented to realize the above vision.

Keywords database integration, query optimization, semantic database grid

1 Introduction

The Web has changed the way people communicate with each other and the way business is conducted by means of Hypertext Markup Language, Hypertext Transfer Protocol, search engines and browsers. Most of today's web content is suitable for human to read. Typical use of the web today involves people's seeking and making use of information, searching for and getting in touch with other people, reviewing catalogs of online stores and ordering products by filling out forms. But the main obstacle to providing better support to web users is that, at present, the meaning of web content is not machine-accessible. The user must either wade through thousands of irrelevant texts or make intelligent guesses of keywords to narrow down the selection. An optional approach is to represent web content in a form that is more easily machine-processable and to use intelligent techniques to take advantage of these representations[1]. This is the main intention of the semantic web that enables the web resources to be understood by the resource usage and service mechanisms such as the search engines and browsers. At the same time, grid computing is the technology that enables a large-scale distributed computing system to carry out the controlled sharing of computing resources. The semantic web and the grid focus on different aspects of application demands. They should converge on the same final target in the long run, and each area will benefit from the progress of other areas. For example, the extension of the grid in semantics and the extension of the semantic web in computation ability converge at the semantic grid[2]. However, most of existing data are stored in relational databases. Therefore, for semantic grid to achieve its full potential, one critical challenge is how to globally publish, seamlessly integrate and transparently locate geographically distributed database resources with such open settings. Web-scale database integration has been one of the main driving forces for both semantic web research and grid development. DartGrid II proposes a semantic-based approach and provides a set of tools and middlewares to support the global sharing of database resources using grid as platform and dynamically integrate information from autonomous local databases managed by heterogeneous database management systems in dynamic, open and multi-institutional environment. All services in DartGrid II are implemented by employing Globus Grid Toolkit 4 and conformed to the standard of open grid services architecture (OGSA) and the specification of open grid services infrastructure (OGSI).

The value of applying semantic web technologies to the information and knowledge in grid application was immediately apparent. It enables user, agent and program to maximize reuse software, services, information and knowledge. In particular, the semantic web resource description framework (RDF) and the web ontology language (OWL) became W3C recommendations. As grid developers have found a need for interoperable metadata they too are turning to RDF, and grid application developers in domains, e.g., life sciences are already working with ontologies—shared vocabularies.
which can be expressed in OWL. We embody and exploit these standards and use ontology to define conceptual model or standard terminology and the relations between them in certain domain. Databases in DartGrid II are semantically registered to the grid service called semantic registry service by mapping the schemata of relational databases to the semantics of ontologies. End-user browses the ontologies to generate a visual conceptual query by semantic browser developed for DartGrid II, and then semantic query service translates a semantically enriched query into a distributed query plan by converting the shared ontologies to the schemata of local databases. Query results will be returned to user as semantically wrapped format (RDF) and presented in semantic browser. Consequently, DartGrid II provides a uniform access to diverse database resources for supporting global query, data mining, e-learning and other high-level applications in which every database node can join or quit the system dynamically.

For a query involving more than one database, global query optimization should be performed to achieve good overall system performance. Because there are some fundamental differences between traditional distributed database management system (DBMS) and database grid system (DBGS), which stem from autonomy and heterogeneity of the database nodes participating in DBGS, query optimization techniques in distributed DBMS cannot trivially and directly be applied to DBGS. Site autonomy in DBGS refers to the situation whereby each database node retains control over local data and processing. This has a number of implications for query optimization in DBGS. Veijalainen classifies site autonomy into three types: design, communication, and execution in [4].

Design autonomy implies that the database nodes are responsible for optimizing local access paths and query processing methods. Consequently, reliable statistical information that is needed for effective global query optimization is not readily available and may not remain accurate as the database nodes change over time. Communication autonomy in DBGS means that a database node independently determines what information it will share with the global system, when it participates in the database grid, and also when it will stop participating. This adds to the complexity of query processing and optimization since any database node system may terminate its services without any advance notice. Execution autonomy results in the situation whereby the global system interfaces with database nodes at their external user interfaces, and it is not able to influence how query processing is being carried out in the database nodes. This means that there is no opportunity for low-level cooperation across systems and hence primitive query processing techniques proposed for distributed database system may no longer be applicable. For example, the semijoin and pipeline operation may be hard to implement in efficient way for lack of facilities provided by low-level and underlying system environment.

In query optimization for distributed DBMS it is assumed that component sites are equal in terms of their processing capability. This assumption no longer seems reasonable in the context of DBGS since database nodes may vary drastically in terms of their availability and processing costs. Furthermore, the same real world objects may be represented in more than one database nodes, but these representatives are not always structurally compatible in DBGS. By the way, database grid system is also different from multi-database system since DBGS supports and allows the database nodes to dynamically participate in or quit the system and sets target for facing more open settings of web environment. However, the query processing problem is much more difficult in database grid environment than in centralized and distributed multi-database. But it is very important for the success of system. We present the design of a query optimizer in DartGrid II, and a heuristic, dynamic and parallel optimization approach for processing query in database grid. In the following discussion, the global data model is assumed to be relational for convenient discussion.

The remainder of this paper is organized as follows. Section 2 presents a brief overview of query optimization in traditional distributed database. In Section 3, the architecture of a database grid query optimizer is proposed. Section 4 describes the algorithms for query optimization in database grid, and some experimental results have been discussed. In Section 5, the implementation of some visual semantic tools and the application in traditional Chinese medicine are introduced. The conclusions and future work are summarized in Section 6.

2 Related Work

Because it is a critical performance issue, query processing has received (and is still receiving) considerable attention in both centralized and distributed DBMSs. A large number of different algorithms have already been developed for query optimization in database systems. The numerous algorithms employed in various applications have already been proposed for query optimization which can roughly be divided into three categories or are combination of such basic algorithms: exhaustive search, heuristics and randomized algorithms.

Typical exhaustive search algorithm is dynamic programming proposed by Selinger et al. in [6], and [7] is its improvement. All proposed algorithms of this class have exponential time and space complexity and are sure to find the best plan according to the specific cost model. Other transformation-based techniques with top-down dynamic programming are EXODUS and Volcano. Kossmann and Storch present a new class of query optimization algorithms that are based on iterative dynamic programming (IDP) and declare that IDP algorithm can produce the best plan of all known algorithms in the situation in which dynamic programming is not