A Unified Relational Approach to Grid Information Services

Peter A. Dinda
Department of Computer Science
Northwestern University

Beth Plale
Department of Computer Science
Indiana University

Abstract—**We propose an approach to Grid information services (GIS) that is based on the relational data model and that integrates static and dynamic information into one information framework. Many highly desirable queries involve constraints on compositions of information, queries that map naturally into joins within the relational model. We lay out the case for this approach, describe our plans, and discuss our current status.**

I. INTRODUCTION

A Grid Information Service (GIS) is a database (in the generic sense of the word) of information about the entities within a wide area distributed high performance computing environment. The concept is related to directory services such as LDAP-based directories and the Jini Lookup Service, service discovery protocols such as SLP, and naming services such as DNS and Overlook, but differs in the scale and nature of data maintained, and in the nature of application queries, as we describe below.

A GIS consists of a set of objects, relationships between objects, and systems needed to query and update the objects and relationships. An object describes an entity in the computing environment. It has a unique identifier, a timestamp, and a set of attributes. Updates to the database take the form of additions or deletions of objects and of changes to the attributes of existing objects. The GIS makes updates available to queries as soon as possible. It also manages access to the objects, making sure that they are updated and read only by valid users. It may present different views of the objects to different users. Examples of Grid entities include organizations, people, computational resources (hosts, clusters), communications resources (switches, routers, topologies), services, benchmarks, software, event channels, sensors, scientific instruments, and others. An extended version of parts of this discussion is available elsewhere [1].

We are implementing an approach to GIS that is based on the relational data model and integrates static and dynamic information into one information framework. Our core observation is that many highly desirable queries in future Grid environments will involve constraints on *compositions of information*. Such queries map naturally into joins within the relational model. In addition, we expect that updates will be a common feature in GIS systems, and relational databases are more likely to achieve high update rates.

Effort sponsored by the National Science Foundation under Grants ANI-0093221, ACI-0112891, ACI-0128390, and EIA-0130869. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation (NSF)

II. APPLICATION NEEDS

Parallel and distributed applications and the software systems that map them to the Grid make the most demands of the GIS. Consider mapping a data-parallel application. At startup time, the application is not interested in individual machines per se, but rather in compositions of them. For example, if it has been coded to run on four processors, then, at startup, it will want to ask questions like "find a set of four unique hosts which in total have between 0.5 and 1 GB of memory and which are connected by network paths that can provide at least 2 MB/s of bandwidth with no more than 100 milliseconds of latency." As it runs, it will want to ask things such as "alert me when the load on any one of these four hosts is at least 25% different from the average across all the machines" so that it can trigger load balancing steps.

There are three things to notice from this example. First, the application's queries compose information about multiple resources in unique, application-specific ways. Second, the data objects on which the queries depend are dynamic to different degrees. Host memory changes more slowly than upgrades to network paths, while host load fluctuates far more quickly than either. The implication is that some data objects in the GIS will be updated quite frequently. Finally, the application needs to query information streams in addition to information in a database.

III. CURRENT LIMITATIONS

As Grid resources and applications evolve, their requirements will be increasingly more difficult to meet using directory services based on a hierarchical data model, such as LDAP [2]. These requirements, and current limitations in meeting them, take the following five forms.

- Rapidly growing size: Given the rapid growth of interest in Grid computing, it is clear that the number of data objects is exploding. Traditional directory services are not designed to support such large datasets.
- Complex relationships among data objects: As the number of objects stored in the GIS grows, the relationships between them also grow in complexity, and new relationships are found to have value and therefore must be represented. Unfortunately, the hierarchical data model requires that the GIS designer must decide which relationships are important early and encode them as an explicit link in the data model. Modifying a schema after-the-fact can be broadly disruptive.
- Frequent updates to objects and their relationships: Up-

dates to Grid data objects are becoming increasingly frequent both because there are simply more data objects (and relationships) and because more and more dynamic kinds of data are being introduced. Current directory services, including LDAP v3, are well known to perform poorly in the presence of frequent updates.

• Sophisticated queries: Existing hierarchical directory services have a limited and restrictive query language: the language is procedural, lacks sophisticated processing semantics, and requires that the user know the tree structure. A declarative query language such as SQL, on the other hand, states *what* is to be retrieved but leaves the *how* up to the query engine. A key need is to be able to integrate data on a user specified condition, and perform an aggregate computation over the results.

• Querying data streams: Existing directory services do not provide support for data streams, a type of data that will become increasingly common as the Grid evolves.

IV. Approach and current results

At the highest level of abstraction, we are investigating and building a GIS based on the relational model that meets the application needs and avoid the limitations that we laid out earlier. Our current work uses SQL as the data modeling, data manipulation and query language, MySQL and Oracle database servers for static data, and the dQUOB system for streaming data, although we remain implementation agnostic. The following are important elements of our work.

• Workload characterization: We have worked extensively with users to develop use cases and models for updates and query stream. We have developed GridG, a synthetic Grid generator, which allows us to populate a GIS with arbitrary quantities of host and network level data.

• Schema development: An early focus of our work has been on modeling the core objects and relationships of a GIS. We have completed the first version of the schema design and made a MySQL implementation publicly available. We are currently finishing our second schema design, which provides more sophisticated network modeling.

• Performance evaluation: RDBMSes are known to provide high performance for updates (an important aspect of the TPC-C benchmark) which we can exploit within a GIS. We have found update rates on the order of 1000s per second to be easily achievable even with our initial MySQL implementation.

• Updates: We are studying the update mechanisms of relational GIS systems, testing the following suppositions. Updates should be pushed into the GIS and not pulled by it. In some cases, the GIS should push updates directly to consumers or at least provide trigger condidtions. The GIS must be able to filter updates and squelch updaters when needed

• Time-bounded non-deterministic queries: The nature of GIS queries is similar to that of decision support queries (TPC-H) in relational database systems. GIS queries will require joins over numerous tables at potentially distributed locations. This has the potential for introducing serious performance problems, both for particular queries and also for the GIS as a whole. However, the nature of the answers that GIS users require is also different from that of traditional decision support queries. In particular a typical GIS user is less interested in getting *all* the answers from a query than in getting *some* answers within a limited time.

We exploit this different nature by extending the SQL query model and select statement. First, we allow the user to state that a query is non-deterministic, allowing the database to return any subset of the full query results. The second extension is to introduce a time bound, an upper limit on how much time the database should work in constructing its non-deterministic answer. The GIS itself can modify the time bound when it is busy.

We have implemented time-bounded non-deterministic queries on top of existing database systems. To do so, we shadow each object in the database with a random number. We rewrite the query (generating SQL) so that it includes a range constraint on the random numbers associated with the objects in the input tables, creating a random sample. The range is set according to the desired selection probability of the sample (not the selectivity of the query), which is in turn determined by the time bound.

• Data stream support: This work leverages off the dQUOB system [3], wherein an SQL query is evaluated purely over data streams instead of tables in a database. Our plan is to be able to seamlessly switch from the RDBMS to dQUOB depending on the dynamicity of the underlying data, thus allowing the user to use the same query language for a wide range of purposes.

• Decentralization and scaling: An important challenge in a GIS system is decentralizing authority and scaling to large databases and high update and query rates. These are important considerations within the database community in general, and we plan to exploit many of the ideas there, such as federated database systems and parallel database engines running on clusters. We are also studying the use of content delivery overlay networks as a mechanism for achieving weak consistency among distributed GIS systems.

References

[1] Peter Dinda and Beth Plale, “A unified relational approach to grid information services,” Tech. Rep. GWD-GIS-012-1, Global Grid Forum, February, Informational Draft.

[2] IBM International Technical Support Organization, *Understanding LDAP*, IBM Corporation, 1998.

[3] Beth Plale and Karsten Schwan, “dQUOB: Managing large data flows using dynamic embedded queries,” in *IEEE International High Performance Distributed Computing (HPDC)*, August 2000.