Towards Semantic Data Integration for Advanced Data Analysis of Grid Data Repositories

Alexander Wöhrer* and Martin Blöchl† and Peter Brezany* and Günther Gruber† and Andreas Langegger† and Herbert Schentz‡ and Wolfram Wöß†

Abstract. This paper is based on a joint research effort between the Johannes Kepler University Linz, University of Vienna and the Federal Environmental Agency. It points out important requirements for future semantic data integration systems and outlines a general architecture which fulfills them, called G-SDAM. Resulting artefacts of the semantic registration process are described as well as how they work together to allow semantic mediation. A preliminary cost model for data integration on the Grid, including its effects, is introduced. The paper also describes components needed for adaptive query execution on the Grid and focuses on two of them, namely the monitoring component and the knowledge base, which can be used during the initial optimization phase as well as for the continuous steering process.

1. Introduction

The first decade of the new millennium is called Data Decade, because the data growth is outpacing computational growth, and future advances in science and engineering will require the tight coupling of computation and analysis of huge data collections [2]. Grid computing [3] offers a distributed infrastructure which allows large-scale resource sharing for multi-institutional virtual organisations in order to jointly work on a common goal and its achievement. As a necessary step, data sources have to be virtualized and meaningfully made available as a seamless integrated data source for advanced data analysis. Our pilot application, represented by our partners from the Umweltbundesamt (Federal Environmental Agency), provides us with the needed complexity to drive our research. There are a lot of large data sources provided by 27 organizations spread across Europe. In the future, new partners bringing new data sets will be included into this long-term project. The data sources include data from different interest domains, as shown in Figure 1. The data sets can be dynamically included (published) into collaborations (virtual organizations) or withdrawn from them (e.g. for privacy or security reasons). A crucial issue is an effective semantic integration of these data sources. Besides the factors mentioned above, there are other crucial requirements put on the dynamics and adaptability of
the semantic integration concepts. Within the global virtual organization, data sources will be dynamically integrated into different problem-oriented virtual sub-organizations performing different data exploration processes, for example: geostatistics, flow analysis, building prediction models. There is a strong need to address performance aspects of the associated integration processes, because they significantly influence the whole turnaround time of the system and response times in individual data exploration tasks. The analyses of our pilot application lead to the following essential requirements for our integration system:

- **Use of semantic technologies.** In order to incrementally integrate new relevant data sources meaningfully into an existing integration, it has to be registered [14]. This concept is also important if ontologies change, whether the purpose is to accommodate new information, adjust the representation of a particular domain, or correct errors. For example, in our pilot application, it is assumed that the ontology will have a very static core but must be permanently extensible, since science and technology involved in measurement of different ecological parameters do not stop and will bring in new concepts.

- **Distributed integration architecture.** As the number of data sources increases, which is typical for many advanced Grid applications, the centralized architecture becomes an administrative and performance bottleneck. There is an urgent need for investigation of distributed integration concepts.

- **Adaptive Query Execution.** The Grid environment is unpredictable and volatile, consequently, traditional static query optimization can lead to poor performance, especially in the expected long running query evaluations. The system should be able to respond dynamically to newly available or unexpectedly missing information sources and determines its behavior according
to information about its environment. The concept of Adaptive Query Processing (AQP) [9] seems especially promising for Grid environments.

- **Exploitation of special mechanisms provided by the Grid.** The Grid provides mechanisms for: (a) systematic access to remote data and computational resources addressing the security, authentication and authorisation problems involved; (b) dynamic resource discovery, allocation and monitoring; (c) monitoring network connections, which is essential for a query engine to efficiently execute queries in wide-area environments; and (d) support for data replication management, which contributes to data access optimization and fault tolerance increase.

The remaining paper is organized as follows. Section 2. introduces the architecture of our Grid Seamless Data Access Middleware (G-SDAM) and discuss some aspects for semantic data integration and a preliminary cost model in more detail. The components for adaptive query processing are described in section 3. in more detail. The paper is finished with our conclusions in section 4.

## 2. Grid Seamless Data Access Middleware (G-SDAM)

![Figure 2. Package view of G-SDAM](image)

The G-SDAM, as shown in Figure 2, is responsible for the execution of queries that are composed according to the Global Data Structure (GDS). The query composer composes a global query that is processed in the Query Processing Toolkit (QPT). The QPT schedules and localizes the global query for the targeted Local Data Nodes (LDN) with the help of the information available after a semantic registration phase of the various sources [14]. Especially the artefacts Local Data Node Content Information, Local Data Structure, Global Data Structure, and the Mapping and Transformation Rules are the Knowledge Base for the assessment of the queries. This query is localized with the aid of the package Data Matching that uses the Mapping Rules to localize the query. The Mapping and Transformation Rules are able to indicate which Local Data Nodes contain data to answer the query. The actual data access is handled by the package Seamless Data Access. This package also returns the localized query results to the QPT. The Local Data Node Content Information is the most detailed and therefore also the most interesting artefact during the query processing. It includes detailed information about the data of the Local Data Node and may be updated after the query.
execution to include the most recent information regarding the LDN. The QPT then transforms the local query results with the help of the package Data Matching and the Transformation Rules to a data structure that adheres to the Global Data Structure (GDS). These query results according to the GDS are combined and presented to the query composer. The cost model, explained in the next section, is also a factor that determines the query processing; the query composer may impose a maximum number of connections to restrict the connection fee and therefore confine the resulting bill for this data interchange process. The bill for a data interchange process can be estimated at the end of the assessment; there it is possible to evaluate the participating LDNs and to assess the amount of data.

In this document the Grid Data Mapping Toolkit is explained in more detail, especially regarding the requirements of ecological data repositories. A more detailed description of the general architecture and functionality of the components can be found in [4].

2.1. Grid Data Mapping Toolkit (G-DMT)

The G-DMT is responsible for the registration, monitoring and basic evaluation of participating LDNs. During the initial registration of a LDN the Local Data Structure (LDS) is mapped to the GDS and the corresponding Mapping Rules and Transformation Rules are created. The GDS, introduced in section 2.2., is the common denominator for all participating LDN. In section 2.3. these artefacts are introduced in greater detail. The G-DMT also monitors the participating LDNs concerning their availability and for basic data source measurements (e.g. data throughput, response times, and so on). The GDS and the Mapping and Transformation Rules form the semantic connection between two or more data sources. The GDS represents the knowledge and semantic connections between data and the corresponding data structures. The Mapping and Transformation Rules are the connection between the LDS and the GDS.

Especially ecological data is accumulated through various methods and by different organizations. In some cases data is already accessible for further usage through services, but most of the time the future potential data sources are not directly accessible and use different data structures and file formats. Additionally ecological events can have impact on multiple ecological domains, subsequently the resulting data is interdependent. Therefore semantic knowledge about the accumulated data is needed to ensure more and better detailed research and scientific work. The GDS and the Mapping and Transformation Rules provide the ability to combine specialized data into a more global view without losing information or relevance. The G-DMT does not only register new data sources, but also semantically integrates the contained data into a uniform data structure, and provides the ability to combine the contained data with various other data sources.

2.2. Global Data Structure (GDS)

The GDS can contain the combined knowledge of all participating data sources and represents a uniform data structure. The query composer builds the query according to the GDS and obtains the result also in a data structure that accords with the GDS. Hence the GDS is the core feature to enable data interchange between various distributed heterogeneous data sources. The GDS is most likely represented as an ontology, to be able to contain knowledge as well as provide structural information. Thus the GDS is also able to accommodate the semantic information about data stored at the data sources.
2.3. Mapping and Transformation Rules

The *Mapping and Transformation Rules* are the link between the LDS and the GDS. The *Mapping Rules* are used to localize the query and therefore provides the mapping between the GDS and LDS and in principle is a correlation of the items of the GDS and LDS. The *Transformation Rules* are responsible for the transformation of the local query results into a data structure adhering to the GDS. Thus the *Transformation Rules* most certainly also have to provide functionality to instruct basic computations (e.g. value conversions, string operations) as well as more specialized operations, for example the conversions of different value ranges with the usage of statistical functions.

2.4. Cost Model

Semantic Data Integration [17] and data interchange is closely related to Data Provenance [5]. It also needs a possibility to bill the recipient of the data interchange process; therefore a *cost model* should be included. Our preliminary cost model is similar to the cost model of phone companies and pictured in Figure 3.

![Cost Model](image)

**Figure 3. Preliminary cost model for data access on the Grid**

This model typically consists of three parts: a basic fee, a connection fee and a data fee. The *basic fee* can be payable monthly or annually and permits the payer to compose and execute queries. The *connection fee* is paid for every connection to a Local Data Node (LDN) and should encourage the query composer to formulate precise queries. The *data fee* is paid for the amount of data a LDN supplies to a query result. This fee encourages the Virtual Organizations [7] hosting the LDNs to continuously update and extend their data and to maintain the corresponding Local Data Structures, Mapping and Transformation Rules. Typically this fee is paid depending on a measure of quantity, but it should also be possible to add measures for actuality, scientific importance, commonness or accessibility of the result data.

3. Adaptive Query Processing

The Steering Subsystem, its components are pictured in Figure 4, deals with the optimal and adaptive execution of a generated query plan. It takes into account the actual progress of the query execution plan based on monitoring of the query execution engine, issues arising from the uncertain and volatile Grid network and computing resources as well as issues arising from the heterogeneous and autonomous data sources.

The architecture of the Steering Subsystem is based on [11]. The Monitoring component collects various types of information: from the execution engine about the state, quality and progress of the evaluation of a query plan; from the Information Services up-to-date information (e.g. memory usage, workload) about the registered data/computational resources and predictions for the network environment. Based on the monitored information, events are generated and evaluated by the assessment component in order to verify whether they caused changes in the values of interesting properties and whether such changes are an issue for the current execution (effecting its optimum). Once an issue has been identified, the system tries to identify potential solutions. If a solution is found, the Query Execution Engine is notified accordingly and its behavior changes as a result. The proposed
framework in [11] addresses monitoring, assessment and response as standalone applications. This decoupling allows reusable techniques, e.g. a single monitoring mechanism for dynamically gathered execution information and a response form based on the impact of adaptation on the current execution (e.g. operator replacement). An extension to the framework represents the learning component, which is targeted towards both areas in the knowledge base described in Section 3.2. The semantic knowledge about the data for query optimization as well as the assessment rules for the execution optimization. They are initiated by different causes and need not lead to changes in the knowledge base, as illustrated in Figure 4 by the dotted line.

3.1. Monitoring

To achieve high performance query execution and the required adaptability to changes in the execution environment at query runtime, as well as support for the initial optimization phase, monitoring information about the involved areas are needed, as pictured in Figure 5. Important information for the initial query optimization in a distributed and/or parallel environment includes access related information about a data source like the available indices and the connection time as well as data related information like the sizes of relations. In order to distribute the work efficiently detailed information about the available Grid resources (computation and network) have to be available. To the best of our knowledge, no work has been reported on monitoring data source related information on the Grid as proposed in this work.

3.1.1. Grid status

A good overview about available systems to monitor Grid resources can be found in [19], including a detailed discussion of advantages and disadvantages of the various systems. A promising candidate, especially because it is working over firewalls and the underlying Java JMX technology could be used
to make the self-monitoring query execution adaptive (e.g. in terms of frequency and other fine-tuning opportunities), is JIMS [1].

3.1.2. Data source related

As already mentioned, we believe that the availability of monitoring information about data sources can further improve the possibilities and performance of AQP in Grid environments. That is especially necessary where such data access related measurements and indicators have to be used to choose among available data sources with similar content. We distinguish between two separate groups of monitoring information, namely the

- access related information, like connection time and available indexes
- data related information, like data statistics and histograms

[16] describes the importance of available data statistics towards query optimization. We believe that the initial query plans, especially the number of sources needed to answer a query, can be improved by providing similar information about involved data sources for distributed optimizers as centralized ones have, as histograms and data statistics. Most relational RDBMS provide some kind of data statistics for their query optimizers in the system tables, providing various update frequencies and automatic tuning possibilities for them.

![Histogram Buckets - Evenly Distributed Data](image1)

![Histogram Buckets - Skewed Data](image2)

**Figure 6. Evenly distributed data vs. skewed data distribution [6]**

In D³G [18], we follow the approach to provide exact continuous statistics for certain parts of the database (likely to be the one used frequently on the Grid), making it possible to restrict the number of required data sources early. Imagine the following example: A query like `SELECT id, name FROM tab WHERE id > 50,000 AND id < 100,000` against a virtual data source combined of three different physical data sources would need to scan all three if no data statistics are available. If continuous exact data statistics are available, e.g. via D³G we know that the range of ID in data source A is between 1 and 40,000 and between 120,000 and 150,000 in data source C, the optimizer can use them to restrict required data sources – in our example only one scan of data source B would be required (given a horizontal fragmentation). Histograms are especially useful where the data is not evenly distributed, so called skewed, as shown in Figure 6. The data distribution information is then stored by the RDBMS in its system tables. This allows the cost based optimizer to use this information and compute/estimate more precise the selectivity based on the data distribution. [15] show how propagation of errors affects the quality of the estimates. A first step is to provide/expose available histograms of the various involved data sources in a uniform way.
3.1.3. Query execution

For the monitoring of the query execution we are following the approach of [10], which is based on self monitoring operators that capture metrics in the form of counter, timings and computations of tuple sizes. [8] provide evidence that this approach is appropriate for multi-node environments like the Grid without requiring modifications neither to the query engine architecture nor in the optimiser and being therefore less disruptive to the query compilation and evaluation system. Through the propagation of more accurate estimates of lower operators in the query plan to operators that lie above the estimates of the latter become more accurate. All operators can continuously update their expected output cardinalities and selectivities if monitoring is in place [10]. Allowing the exchange of monitoring information across the network has additional benefits and can be implemented by extending the exchange operator [13] of the query plan to act as a notification source, as done in [12].

3.2. Knowledge base

The knowledge base includes two components: one dedicated to semantic query optimization and the other one dedicated to Adaptive Query Processing. The first part, semantic knowledge about the data sources and their contained data, of the knowledge base is comprised by globally (valid for all data sources) and locally (valid for one particular data source) applicable semantic rules and facts about the data. They are partially given by domain experts, others are learned over time. It is used during the semantic query optimization phase to increase constraint or join elimination or refute a query at all (e.g. asserting that it will not be satisfy-able by the data and return an empty set).

<table>
<thead>
<tr>
<th>Global rule or fact</th>
<th>Local rule or fact</th>
<th>Data independent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predefined or Learned</td>
<td>Predefined or Learned</td>
<td>Fixed set</td>
</tr>
</tbody>
</table>

Table 1. Knowledge Base component dedicated to query optimization

The following examples are given for the semantic knowledge to be stored:

- Global knowledge predefined: The lowest temperature is -273.15 degrees
- Local knowledge predefined: Each employee must be at least 21 years old
- Local knowledge learned: If an employee is still working, then he was born after 1950
- Data independent: availability of a data source

The quantifier of the knowledge is given by:

Global predefined > Local predefined > Global learned > Local learned

This means that global predefined knowledge overrules local predefined knowledge, and so on. The second purpose of the knowledge base is to provide rules for the used adaptive query processing strategy. Also this part of the knowledge base is comprised by a predefined (basic) set of Event-Condition-Action (ECA) rules and learned ones over time. The quantifier of this knowledge is given by:
This means that the basic predefined ECA rules are overruled by the more precise learned ones. The data independent set of the knowledge overrules both.

| Predefined ECA-rules | Learned ECA-rules | Data independent set |

Table 2. Knowledge Base component dedicated to Adaptive Query Processing

4. Conclusion

This paper – based on a joint research effort between the Johannes Kepler University Linz, University of Vienna and the Federal Environmental Agency – points out important requirements of future semantic data integration systems and introduces a general architecture which fulfills them, called G-SDAM. The preliminary cost model pushes the data providers to keep their data source registrations up-to-date and the data requestors to formulate exact queries. In order to reduce unnecessary access to data sources as well as bandwidth usage, special metadata tailored for data integration is needed. On the one hand this is reflected in our architecture by the knowledge base, e.g. containing rules for further semantic query optimization, and on the other hand by the data (access) related monitoring information, e.g. containing basic data statistics and exposing available histograms for efficient initial query optimization. The same information sources are also used to react to the volatile Grid environment during the adaptive execution of long-running queries by our Steering Subsystem. Next steps include further implementation and integration of the monitoring components, building on D³G [18], to cover all needed aspects for Adaptive Query Processing on the Grid.

Acknowledgements

The work described in this paper is being supported by the Austrian Grid, a project funded by the Federal Ministry for Education, Science and Culture after recommendation by the Austrian Council for Research and Technology Development.

References


