

Grid Services for Efficient Decentralized Indexation and Query Execution on Distributed Data Warehouses*

Pascal Wehrle, Anne Tchounikine, Maryvonne Miquel

Lyon Research Center for Images and Intelligent Information Systems (LIRIS)

UMR CNRS 5205 – INSA Lyon,

20 avenue Albert Einstein,

69621 Villeurbanne Cedex,

France

{pascal.wehrle, anne.tchounikine, maryvonne.miquel}@insa-lyon.fr

Abstract. Online analysis of large volumes of data consists an important part of today's business and scientific applications. Storing structured data in data warehouses following a multidimensional storage model provides efficient and reliable access to it. Distributed systems have become the first choice to cope with the increasing volume and complexity of data warehouses. In order keep up with this trend, large quantities of interconnected computing and storage resources are now being integrated into computing grids. Adapting data warehouse concepts to grids however requires decentralized management of their traditionally centralized storage and query execution model. The grid services architecture we present in this paper therefore uses a unique data identification to implement grid-wide data localization based on local indexes in combination with a catalog service. These structures integrate information on both materialized and computable data for cost-optimized distributed query execution.

1 Introduction

Modern business intelligence and analysis applications heavily rely on data warehouses to efficiently store large volumes of structured data and provide a fast and reliable access to it. As new applications are developed in the domain of data warehouse-based analysis, data warehouses are often required to deliver data to a variety of different user groups, each requesting remote access to specific data sets at custom levels of detail.

The classical architectural solution to these requirements is a central data warehouse with both storage and computing of analysis tasks performed by one single machine with remote access for all users. A common solution to overcome storage and computing capacity shortages is delegation of both tasks to distributed systems offering parallelized computing on distributed data. Clusters with centralized control and access provide high performance and availability. However, given high costs and varying load, most companies and organizations prefer to dynamically integrate existing resources from several distant sites to build distributed data warehouses. This requires systems that efficiently organize these resources in a transparent way. Computing grids implementing this approach are based on the collaboration of

*This work is supported by the French Ministry for Research ACI "Masse de données" project (<http://acimd.labri.fr/>)

independent grid services, the existing mechanisms for organizing and operating distributed data warehouses cannot be straightly applied. The main challenges are the internal management of the data distribution among grid nodes, starting from an initial distribution, and the subsequent query execution requiring active placement of both data and computing jobs on the grid.

Bioinformatics research is one of the application domains where grids are being introduced to analyse huge amounts of data using complex algorithms. The grid services architecture presented in this paper has been developed in the context of the Grid for Geno-Medicine¹ (GGM) project [1], an effort to provide a software architecture capable of managing heterogeneous and dynamic bio-medical data stored on a grid infrastructure for intensive analysis and processing purposes.

The rest of this paper is organized as follows: In section 2 we give an overview of the grid service layers we developed for grid-wide querying based on local indexes of stored data. Their interaction during cost-optimized distributed query execution are presented in section 3. Section 4 concludes with an overview of our future work on the system.

2 Proposed Grid Service Layers

The proposed grid services of our architecture are implemented on the middleware infrastructure provided by the Globus Toolkit 4.0 [2]. As illustrated in figure 1 these services are grouped into three distinct layers above the standard Globus data access and job execution services.

The initial distribution of warehouse data is made using derived horizontal partitioning methods introduced in [3] and [4], including materialized aggregated data. For efficient data localization and exchange we use an identification method that creates one namespace for all warehouse data distributed on grid storage nodes. A detailed description of the approach is given in [5]. We introduced chunk identifiers for warehouse data that refer to all replicas of a tuple of warehouse data on the grid. Each contiguous partition of data is stored in a separate file and identified as a block of chunks between lower and an upper limit chunks.

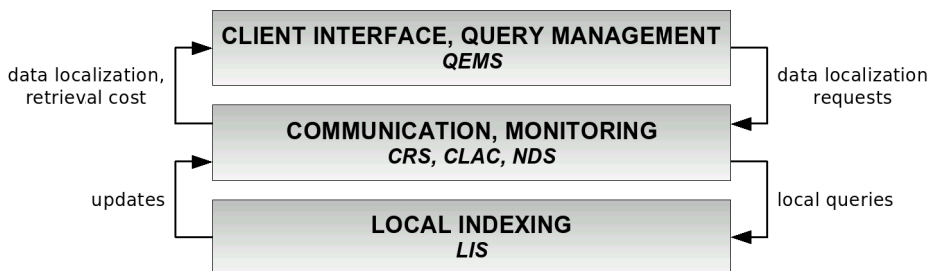


figure 1: the layered service architecture with main interactions

The blocks of chunks stored on a grid node are indexed by the “Local Indexing Service”(LIS), the base layer of our architecture. It offers a complete index of locally available data, both materialized and computable from materialized data. In order to achieve this the LIS builds a lattice structure that groups together data of the same

¹ http://liris.cnrs.fr/projets/ggm/GGM_English

aggregation level and uses a spatial X-tree index introduced in [6] to efficiently index blocks of data based on the multidimensional warehouse data model. Whenever files are materialized or deleted on the storage node, the index recursively updates all impacted computable data blocks on the index.

The information provided by local indexes is assembled, amended and exchanged by the *communication and monitoring services* layer that provide efficient localization of requested warehouse data and up-to-date monitoring information. External interactions with this layer are executed through the “*Chunk Resolution Service*”(CRS). It is deployed on every storage node and executes localization queries for warehouse data blocks. In order to make every block of indexed data on the grid available through the network of CRS instances, each CRS publishes the identifiers of locally available data blocks to a catalog service, the “*Chunk Localization Array Catalog*”(CLAC). The CLAC stores a mapping between data block identifiers and a list of grid nodes from which the data block can be obtained. During query execution, the list of potential sources for a requested data block has to be amended with estimates of transfer and computing costs to make optimization choices possible. The “*Network Distance Service*”(NDS) [7] provides these current cost estimates based on network bandwidth, latency and CPU load of source and destination nodes. Optimization itself is controlled by the “*Query Execution Management Service*”(QEMS). Deployed on nodes designated as access points for users, it handles user query transformation, optimization and execution as detailed in the next section.

3 Query Execution Procedure

The interaction between the different grid services is best described following the processing of a user query. We distinguish three main phases in this process: The transformation and localization phase consists in transforming the user query and searching the requested data using the communication and monitoring layer services. The following planning phase utilizes the results to optimize the transfer and computation of results. At the end stands the execution phase with concurrent execution of the established distributed query plan.

First, the QEMS transforms the initial SQL user query into data block identifiers using the dimensions of the multidimensional warehouse model. A localization query is then sent to the nearest storage node's CRS which controls the execution of the localization process. The query is forwarded to the LIS on the same node which returns the list of locally materialized or computable blocks that satisfy at least part of the query. Missing parts of the requested data are then searched for by transmitting requests to the CLAC. Upon receiving the result list of source nodes for the requested data, the CRS tries to obtain direct references to the source data files containing the data blocks by polling the designated source nodes. At the same time it requests monitoring data from the NDS for each of them.

The final results of the localization phase are returned to the QEMS. During the subsequent optimization phase, the QEMS uses a cost metric inspired by the ones presented in [8]. When selecting a replica for a requested block r , it uses the weighted cost $c(f,r)$ for any source file f , which is proportional to the total cost of retrieval $tc(f)$ and the set of useful chunks $uc(f,r)$ contained in f that are also part of r :

$$c(f, r) = \frac{tc(f)}{|uc(f, r)|}, uc(f, r) = f \cap r \quad (1)$$

All source files are next inserted into a list sorted by their weighted cost. A greedy algorithm then computes a local minimum solution by iteratively selecting the file with the least cost and updating the cost values for the remainder of the list to avoid redundant transfers of the same data and to favor retrieval from multiple sources. The resulting distributed query plan contains independent jobs for each source file.

During the final execution phase, the QEMS launches file transfers and submits computing jobs according to the distributed execution plan. Jobs involving additional computing of aggregates are given a higher priority over simple data transfers. The query result is progressively assembled through a union operation on the partial results provided by each job and transferred to the client application.

4 Conclusion

In this paper we have given an overview of the layered grid services architecture designed to allow efficient decentralized management of distributed data warehouses on grid infrastructures. Based on local indexes of both materialized and computable data on storage nodes our identification method provides grid-wide localization of requested data through a set of communication and monitoring services. The gathered information is used for cost-optimized distributed query execution. The architecture was implemented and tested on the GGM project's grid testbed. Our current efforts are focused on the integration of a collaborative cache system interacting with local indexes to better manage data replication and temporary storage. We are also working on a suitable decentralized maintenance mechanism for the distributed data warehouse on the grid.

References

- [1] Brunie, L., Miquel, M., Pierson, J., Tchounikine, A., Dhaenens, C., Melab, N., Talbi, E., Hameurlain, A., Morvan, F.: Information grids: managing and mining semantic data in a grid infrastructure; open issues and application to geno-medical data. In 14th International Workshop on Database and Expert Systems Applications (DEXA'03), September 1-5 2003, Prague, Czech Republic. IEEE CS, pp. 509-516
- [2] Foster, I.T.: Globus Toolkit Version 4: Software for Service-Oriented Systems. In IFIP International Conference on Network and Parallel Computing, NPC 2005, November 30 - December 3 2005, Beijing, China. Springer, pp. 2-13
- [3] Özsu, M.T., Valduriez, P.: Principles of distributed database systems. Prentice Hall, 1991.
- [4] Bellatreche, L., Karlapalem, K., Mohania, M.K.: OLAP Query Processing for Partitioned Data Warehouses. In 1999 International Symposium on Database Applications in Non-Traditional Environments (DANTE '99), 28-30 November 1999, Kyoto, Japan. IEEE CS, pp. 35-42
- [5] Wehrle, P., Miquel, M., Tchounikine, A.: A Model for Distributing and Querying a Data Warehouse on a Computing Grid. In 11th International Conference on Parallel and Distributed Systems (ICPADS 2005), 20-22 July 2005, Fukuoka, Japan. IEEE CS, pp. 203-209
- [6] Berchtold, S., Keim, D.A., Kriegel, H.: The X-tree : An Index Structure for High-Dimensional Data. In Proceedings of the 22nd International Conference on Very Large Data Bases (VLDB'96), September 3-6 1996, Mumbai (Bombay), India. Morgan Kaufmann Publishers/Elsevier Science, pp. 28-39
- [7] Gossa, J., Pierson, J., Brunie, L.: Evaluation of network distances properties by NDS, the Network Distance Service. In Proceedings of Third International Workshop on Networks for Grid Applications (GridNets 2006), October 1-2, 2006, San Jose, California, U.S.A. IEEE CS.
- [8] Weng, L., Çatalyürek, Ü.V., Kurç, T.M., Agrawal, G., Saltz, J.H.: Servicing range queries on multidimensional datasets with partial replicas. In 5th International Symposium on Cluster Computing and the Grid (CCGrid 2005), 9-12 May, 2005, Cardiff, UK. IEEE CS, pp. 726-733