Adaptive Virtual Partitioning: Further Developments

Alexandre A. B. Lima[1], Marta Mattoso[1], Patrick Valduriez[2]

[1] Computer Science Department, COPPE, Federal University of Rio de Janeiro – Brazil
{assis, marta}@cos.ufrj.br
[2] INRIA and LIRMM, Montpellier – France
Patrick.Valduriez@inria.fr

Categories and Subject Descriptors: Information Systems [**Miscellaneous**]: Databases

Keywords: Parallel Databases, Database Cluster, Query Processing

1. INTRODUCTION

Adaptive Virtual Partitioning (AVP) is a parallel query processing approach which gracefully adapts to load unbalancing [Lima et al. 2004]. This paper describes the further developments on AVP after the initial proposal. It is organized as follows: Section 2 describes the first database cluster middlewares which we built based on AVP. Section 3 shows the results we obtained with real-world (non-synthetic) queries and datasets. Section 4 describes how the full database replication issue (present in our initial implementations of database cluster middlewares) was dealt with. In Section 5 we describe our initial efforts on using AVP on a data grid environment. Section 6 concludes and gives some future directions.

2. INITIAL IMPLEMENTATIONS OF AVP

AVP provides a way to break up a query into many sub-queries with no need to predetermine the best virtual partition size before query execution. The algorithm adapts the partition size according to each sub-query response time. However, this does not guarantee good performance during query processing, in particular, when the initial load (virtual partitions) assigned to participating nodes is unbalanced (with some nodes finishing their tasks before other ones). A task reassignment mechanism can correct an initially unbalanced workload distribution, each task being a sub-query. The "many queries per node" approach proposed by AVP makes it possible to implement dynamic load balancing during query execution using black-box DBMS, i.e., with non-intrusive techniques.

We used AVP to implement a database cluster query processor prototype. We implemented AVP and techniques for dynamic task reassignment and run experiments on a 64-node cluster. Very good results were obtained and super-linear speedup could be achieved even in scenarios with severe load skew. All techniques and results are described in [Lima 2004].

The dynamic load balancing mechanism implemented so far was based on *help offerings*: each time a node becomes idle, it sends help offering messages to all other nodes. A busy node that accepts a help offer sends half of its remaining workload to the offering node. In order to avoid data transfers between nodes during query processing, the prototype was implemented by using full database replication: each cluster node has to have a copy of the entire database. This way, the only information a node needs

in order to help another one is the limits of the new virtual partition it has to process.

The excellent results motivated us to develop ParGRES [Mattoso et al. 2006]: a fully functional database cluster middleware to process OLAP queries using intra-query parallelism. Besides providing intra-query parallelism with dynamic load balancing for database clusters, ParGRES has also an SQL3 query parser and can be used on any database cluster that employs DBMS with a JDBC driver. We also developed a JDBC driver for ParGRES, enabling it to be used by any Java application.

A major drawback found in our initial AVP-based query processors is the full database replication requirement, which imposes a severe limit to the database size: it cannot be bigger than the smallest disk capacity of the nodes involved in query processing. To ease references to these implementations, we call them AVP-FR (*AVP with Full Replication*) in the rest of this document. Developments were made in order to overcome AVP-FR limitations.

3. AVP-FR PERFORMANCE WITH REAL-WORLD DATABASES

So far, all experiments we did with AVP-FR were based on synthetic databases and queries produced according to the TPC-H Benchmark [TPC 2004]. In [Paes et al. 2008], we evaluate AVP-FR performance with a real-world OLAP application, from the Brazilian Institute of Geography and Statistics - IBGE. We focus on analyzing its non-intrusive database design approach. We performed several experiments and obtained linear and almost always super-linear speedup on queries frequently issued against IBGE databases. We noticed that, by using only a 4-node PC cluster, queries that take about 15 minutes drop to just half a minute. Since queries are *ad-hoc*, we did not perform any fine tuning nor any optimization on the DBMS. Application migration costs and investments on this solution are negligible, being restricted to acquiring an off-the-shelf PC cluster. These results can be further improved if caching and other optimizations are used. Extensive results show that AVP-FR has proved to be a very cost-effective alternative solution for OLAP query processing in real scenarios.

4. AVP-PR: DEALING WITH THE FULL REPLICATION

The main advantages of using AVP-FR are flexibility for node allocation during query processing tasks and high cluster availability as any node can process any query. Furthermore, it provides a good basis for dynamic load balancing through workload redistribution, as any node has local access to the entire database. On the other hand, the main disadvantage of AVP-FR is the poor disk utilization it provides. Calling sc_i the storage capacity of node i, the database size must be equal to $\min(sc_i)$, where i=1, ..., n, and n=number of cluster nodes. It makes AVP-FR suitable just for small and medium-sized databases.

A more realistic approach is the Hybrid Design (HD) proposed in [Röhm et al. 2000]. HD proposes a physical partition of the largest and most accessed tables and fully replicates small tables. Thus, intra-query parallelism can be achieved with lesser disk space requirements. We started combining AVP and HD. Our prototype was adapted and the results obtained are described in [Furtado et al. 2005] and [Furtado et al. 2008]. Disk space utilization significantly decreased with the new approach. Besides, the performance obtained was similar to the one achieved with AVP-FR in absence of load unbalancing during query processing. We call the new approach *Adaptive Hybrid Partitioning* (AHP).

The simple use of HD, however, has a major drawback: as each table partition is allocated to only one cluster node, dynamic load balancing is not possible without inter-node data transfers. Then, in [Lima et al. 2009] we propose a distributed database design strategy for database clusters based on physical and virtual data partitioning combined with replication techniques. To address the limitation of static physical partitioning we take advantage of replicas and propose a dynamic query load balancing strategy.

Basically, we use the *chained declustering* [Hsiao and DeWitt 1990] technique to replicate partitions

obtained through HD among cluster nodes. Thus, it became feasible for us to present a load balancing technique that, combined with AVP, makes it possible to dynamically redistribute tasks from busy cluster nodes to idle nodes that contain replicas of their data partitions. This way, our solution makes it possible to obtain intra-query parallelism during heavy-weight query execution with dynamic load balancing while avoiding the overhead of full replication. We call the new approach AVP-PR (*AVP with Partial Replication*). We run new experiments with AVP-PR, which achieved excellent speedup [Lima et al. 2009].

5. AVP FOR DATA GRIDS

Grid computing, and more recently cloud computing, have gained much interest in enterprise information systems, thus making data management in these contexts critical.

Ideally, a grid or cloud database solution must respect database autonomy (i.e. avoid database or application migration) while taking advantage of distributed and parallel computing. This can be achieved through the development of a middleware layer between the user applications and the databases. Such a middleware should provide for distributed and parallel query processing with non-intrusive techniques, considering DBMS as black-box components; hence, there is no need for database or application migration.

However, the migration from clusters to grids poses many challenges for OLAP parallel query processing as it must be dealt with in two levels: grid level, which requires the distribution of tasks between grid nodes; and cluster level, which requires the re-distribution of those tasks between cluster nodes (considering the typical scenario where each grid node is a PC cluster). The additional level demands special attention to load balancing and final result composition.

In [Kotowski et al. 2008], we propose a middleware solution to OLAP query processing in grids, called GParGRES, which employs AVP-FR to provide transparent inter and intra-query processing. We call this AVP implementation gAVP-FR (where "g" stands for "grid"). We consider a typical grid environment with multiple clusters at different sites. Thus, database replication and parallel query processing must be addressed at two levels: grid level and cluster level. Compared with the database cluster approach, where the database is replicated at a single site, gAVP-FR enables the database to be replicated at multiple sites of the grid, thus increasing data availability and quality of service. For instance, if one grid site is unavailable, it is still possible to run OLAP queries using other sites.

We partially implemented the middleware as grid services on Grid5000 [Bolze et al. 2006], a large and flexible configurable grid platform in France. In [Kotowski et al. 2008] we present preliminary experimental results obtained with two clusters of Grid5000 using queries of the TPC-H Benchmark. The results show linear or almost linear speedup in query execution, as more nodes are added in all tested configurations.

6. CONCLUSION AND FUTURE WORK

We further developed AVP as a basis for high-performance OLAP query processing in database clusters. AVP proved to be a very efficient non-intrusive technique. When combined with dynamic load balancing, excellent performance can be obtained even in the presence of severe load skew conditions. The initial full database replication requirement issue has been addressed and now AVP can be used with partially replicated databases, reducing disk consumption. Some steps were taken in order to implement our techniques in data grid environments.

As ongoing work, we continue to investigate the use of AVP in data grids. More specifically, we are studying the use of AVP in Bioinformatics applications that take a long time to process and deal with huge amounts of distributed data. In this scenario, we intend to implement gAVP-PR: a grid version of AVP-PR, thus eliminating the full replication issue in data grids.

Besides, we are investigating the combination of AVP and cloud computing for OLAP query processing. The remote location of data and processing nodes, the virtually infinite pool of resources for query processing and the adaptive characteristics of AVP sounds like a very promising and interesting research field.

REFERENCES

Bolze, R., Cappello, F., Caron, E., Daydé, M., Desprez, F., Jeannot, E., Jégou, Y., Lanteri, S., Leduc, J., Melab, N., Mornet, G., Namyst, R., Primet, P., Quetier, B., Richard, O., Talbi, E.-G., and Touche, I. Grid'5000: A Large Scale And Highly Reconfigurable Experimental Grid Testbed. *International Journal of High Performance Computing and Networking* 20 (4): 481–494, 2006.

Furtado, C., Lima, A. A. B., Pacitti, E., Valduriez, P., and Mattoso, M. Physical and Virtual Partitioning in OLAP Database Clusters. In *Proceedings of the 17th Symposium on Computer Architecture and High Performance Computing*. Rio de Janeiro, Brazil, pp. 143–150, 2005.

Furtado, C., Lima, A. A. B., Pacitti, E., Valduriez, P., and Mattoso, M. Adaptive hybrid partitioning for OLAP query processing in a database cluster. *International Journal of High Performance Computing and Networking* 5 (4): 251–262, 2008.

Hsiao, H.-I. and DeWitt, D. J. Chained Declustering: A New Availability Strategy for Multiprocessor Database Machines. In *Proceedings of the Sixth International Conference on Data Engineering*. Los Angeles, USA, pp. 456–465, 1990.

Kotowski, N., Lima, A. A. B., Pacitti, E., Valduriez, P., and Mattoso, M. Parallel query processing for OLAP in grids. *Concurrency and Computation: Practice and Experience* 20 (17): 2039–2048, 2008.

Lima, A. A. B. *Intra-Query Parallelism in Database Clusters*. Ph.D. thesis, Federal University of Rio de Janeiro, 2004. (In Portuguese).

Lima, A. A. B., Furtado, C., Valduriez, P., and Mattoso, M. Parallel OLAP query processing in database clusters with data replication. *Distributed and Parallel Databases* 25 (1-2): 97–123, 2009.

Lima, A. A. B., Mattoso, M., and Valduriez, P. Adaptive Virtual Partitioning for OLAP Query Processing in a Database Cluster. In *Proceedings of the 19th Brazilian Symposium on Databases*. Brasília, Brazil, pp. 92–105, 2004.

Mattoso, M., Silva, G. Z., Lima, A. A. B., Baião, F. A., Braganholo, V. P., Aveleda, A., Miranda, B., Almentero, B. K., and Costa, M. N. ParGRES: Middleware para Processamento Paralelo de Consultas OLAP em Clusters de Banco de Dados. In *Proceedings of the 21st Brazilian Symposium on Databases – 2nd Demo Session*. Florianópolis, Brazil, pp. 19–24, 2006.

Paes, M., Lima, A. A. B., Valduriez, P., and Mattoso, M. High-Performance Query Processing of a Real-World OLAP Database with ParGRES. In *Proceedings of the International Conference on High Performance Computing for Computational Science*, J. M. L. M. Palma, P. Amestoy, M. J. Daydé, M. Mattoso, and J. C. Lopes (Eds.). Lecture Notes in Computer Science, vol. 5336. Springer, pp. 188–200, 2008.

Röhm, U., Böhm, K., and Schek, H.-J. OLAP Query Routing and Physical Design in a Database Cluster. In *Proceedings of the 7th International Conference on Extending Database Technology*. Konstanz, Germany, pp. 254–268, 2000.

TPC. TPC Benchmark H (Decision Support). `http://www.tpc.org/tpch`, 2004.