Efficient OLAP Query Processing in Distributed Data Warehouses

Michael Akinde
Aalborg University
strategy@cs.auc.dk

Michael Böhlen
Aalborg University
boehlen@cs.auc.dk

Theodore Johnson
AT&T Labs–Research
johnsont@research.att.com

Laks V.S. Lakshmanan
University of British Columbia
laks@cs.ubc.ca

Divesh Srivastava
AT&T Labs–Research
divesh@research.att.com

The success of Internet applications has led to an explosive growth in the demand for bandwidth from ISPs. Managing an IP network includes complex data analysis (see, e.g., [1]) that can often be expressed as OLAP queries. For example, using flow-level traffic statistics data one can answer questions like: *On an hourly basis, what fraction of the total number of flows is due to Web traffic?* Current day OLAP tools assume the availability of the detailed data in a centralized warehouse. However, the inherently distributed nature of of the data collection (e.g., flow-level traffic statistics are gathered at network routers) and the huge amount of data extracted at each collection point (of the order of several gigabytes per day for large IP networks) makes such an approach highly impractical.

The natural solution to this problem is to maintain a distributed data warehouse, consisting of multiple local data warehouses (sites) adjacent to the collection points, together with a coordinator. In order for such a solution to make sense, we need a technology for *distributed processing of complex OLAP queries*. We have developed the Skalla system for this task. Skalla translates OLAP queries specified using relational algebra augmented with the GMDJ operator [2] into distributed query evaluation plans. Salient features of the Skalla approach are:

- the ability to handle complex OLAP queries involving correlated aggregates, pivots, etc.

- only partial results are ever shipped — never subsets of the detail data.

Skalla generates distributed query evaluation plans (for the coordinator architecture) as a sequence of rounds, where a round consists of: (i) each local site performing some computation and communicating the result to the coordinator, and (ii) the coordinator synchronizing the results and (possibly) communicating with the sites. The semantics of the subqueries generated by Skalla ensure that the amount of data that has to be shipped between sites is independent of the size of the underlying data at the sites. This is particularly important in our distributed OLAP setting. We note that such a bound does not exist for the distributed processing of traditional SQL join queries [3].

The Skalla evaluation scheme allows for a wide variety of optimizations that are easily expressed in the algebra and thus readily integrated into the query optimizer. The optimization schemes included in our prototype contribute both to the minimization of synchronization traffic and the optimization of the processing at the local sites. Significant features of the Skalla approach are the ability to perform both distribution-dependent and distribution-independent optimizations that reduce the data transferred and the number of evaluation rounds.

We conducted an experimental study of the Skalla evaluation scheme using TPC(R) data. We found the optimizations to be very effective, often reducing the processing time by an order of magnitude. The results demonstrate the scalability of the Skalla techniques and quantify the performance benefits of the optimization techniques that have gone into the Skalla system.

References

[1] R. Cáceres, N. Duffield, A. Feldmann, J. Friedmann, A. Greenberg, R. Greer, T. Johnson, C. Kalmanek, B. Krishnamurthy, D. Lavelle, P. Mishra, K. K. Ramakrishnan, J. Rexford, F. True, and J. van der Merwe. Measurement and analysis of IP network usage and behavior. *IEEE Communications*, May 2000.

[2] D. Chatziantoniou, M. Akinde, T. Johnson, and S. Kim. The MD-join: An operator for complex OLAP. In *Proc. ICDE*, 2001.

[3] D. Kossman The state of the art in distributed query processing. *ACM Computing Surveys*, 32(4): 2000.