The Active MultiSync Controller of the Cubetree Storage Organization

Nick Roussopoulos, Yannis Kotidis, Yannis Sismanis
Advanced Communication Technology, Inc.*
{*nick,kotidis,isis*}*@act-us.com*

Abstract

The *Cubetree Storage Organization (CSO)*[1] logically and physically clusters materialized-views data, multi-dimensional indices on them, and computed aggregate values all in one compact and tight storage structure that uses a fraction of the conventional table-based space. This is a breakthrough technology for storing and accessing multi-dimensional data in terms of storage reduction, query performance and incremental bulk update speed. CSO has been extended with an Active MultiSync controller for synchronizing multiple concurrent access and continuous asynchronous online updates for a non-stop data warehouse.

1 The CSO Concept

The Relational On-Line Analytical Processing (ROLAP) is emerging as the dominant approach in data warehousing. In order to enhance query performance, the ROLAP approach relies on selecting and materializing in summary tables appropriate subsets of aggregate views which are then engaged in speeding up OLAP queries. However, a straight forward relational storage implementation of materialized ROLAP views is immensely wasteful on storage and incredibly inadequate on query performance and incremental update speed.

Cubetrees [RKR97] are collections of packed and compressed R-trees and shown to be the best alternative for storing and indexing at the same time collections of ROLAP aggregate views. The *Cubetree Storage Organization (CSO)* [ACT97] logically and physically clusters materialized-views data, e.g. summary tables, multi-dimensional indices on them, and computed aggregate values all in one compact and tight storage structure that uses a fraction of the conventional table-based space. The physical clustering and the ordering within the CSO permits sequential read and write operations during the creation and bulk incremental update in a *merge-pack* type of operation. It is this sequential I/O that utilizes the transfer rate of the disks and achieves maximum performance for both updates and queries.

CSO offers a breakthrough technology for storing and accessing derived materialized views. Each Cubetree structure is not attached to a specific materialized view or an index but acts like a *place holder* which stores multiple and possibly unrelated views and multi-dimensional indices without having to analyze and identify the dense and sparse area of the underlying multi-dimensional data. CSO clusters data on non-interleaved subspaces regardless of the skewness underlying data or the number of dimensions.

Given a set of OLAP aggregate views to be supported, the CSO optimizer selects the best placement on one or more Cubetrees for achieving maximum clustering. The underlying cost model is tailored to the CSO and uses a greedy algorithm first for selecting an optimal configuration and then an adaptive algorithm for refining the configuration. The warehouse administrator can tune the optimization to upper-bound either the duration of the incremental update window possible down-time for the warehouse or the maximum storage used.

2 Performance of the CSO

The CSO avoids the high cost of creation and maintenance by sorting and bulk incremental merge-packing of Cubetrees. The sorting portion of it is the dominant factor in this incremental bulk update mode of operation while the merge-packing portion achieves rates of 10-12 GB per hour on a single Seagate Cheetah disk. No other storage organization offers this scalable bulk merge-packing technology.

*1209 Goth Lane, Silver Spring, MD 20905, Tel: 301-384-3759, Fax: 301-384-3679, Url: *http://www.act-us.com*

[1]US Patent Pending

Experiments with various data sets showed that multi-dimensional range queries perform very fast. This performance is the result of the packing and compression of the shared common subspaces, and the data clustering which permits maximum non-interleaving serial I/O. In a recent test [KR98] using the TCP-D benchmark data the CSO implementation achieved at least a 2-1 storage reduction, a 10-1 better OLAP query performance, and a 100-1 faster updates over a conventional ROLAP storage organization of materialized views.[2]

3 The CSO MultiSync Controller and Online Maintenance

The CSO Active MultiSync Controller (AMC) further extends the performance of CSO by optimizing multiple concurrent queries and parallel disks. By pushing logic to the AMC, multiple I/O requests are scheduled to a single pass over serial portions of the disk. This decreases the number of asynchronous I/Os satisfying individual access requests and can increase the throughput performance by a large factor, especially for workloads that can be aligned for a small latency penalty.

Another feature of the CSO software is that the incremental maintenance can be done online while the data warehouse is operational using the previous version of the Cubetrees. Since the incremental update is a three-way merge of sequential I/O, namely the current Cubetrees, the delta Cubetrees carrying the updates, and the writing of the new Cubetrees during the merging, this three-way merge is best realized using three disks. From these two serial reads, only the one that reads the current Cubetree needs be synchronized with the query reads on the same disk because the other maintenance I/O, i.e. the reads on the delta Cubetrees and the writes of the new Cubetrees are doing serial I/O on their own disks. The maintenance read I/O can either be done in the background during low peaks of the system or Multisynched with the query I/O and thus, has negligible effect on query processing. This online maintenance of the CSO eliminates data warehouse down-time.

4 The CSO Software

The CSO software is offered on the Sun/Solaris and Intel/NT platforms. It comes in two configurations: a) as a Cubetree Datablade developed to run with the Informix Universal Server (IUS) and b) as a stand alone library. For the Datablade case, IUS and CSO harmoniously inter-operate in a tightly coupled configuration that has no overhead. The Datablade configuration takes advantage of the extensibility of Object-Relational database management system (O-RDBMS) and allows to include user-defined aggregate and statistical functions not normally offered by RDBMSs. The CSO Datablade offers the performance of the stand-alone CSO software, namely very fast OLAP aggregate queries and bulk-incremental update on them, but at the same time, transparent calls of the CSO software from within the SQL and the other APIs offered by IUS.

The stand alone configuration provides a browser-based GUI interface through which the user defines the set of views to be queried and optimized in the data warehouse. The interface also provides a query interface through which the user can specify range OLAP queries.

CSO has been developed with extensibility in mind. New data types for the dimension and measure attributes can be facilitated. Similarly, the aggregate functions on the measure attributes can be user-defined for specialized computation. Such functions can return either a scalar or a vector value. Other functions that extend the SQL aggregate functions have been developed and incorporated into the CSO libraries.

5 Sigmod 1999 Demo

ACT Inc. has prepared a demo and short presentations for the Demo Sessions of Sigmod in Philadelphia. The demos will run a benchmark data of roughly 50GB residing from the local disks of a Sun Ultra 60 server. It demonstrates CSOs real-time query performance for multiple concurrent queries. The audiences will also be able to write and run their own OLAP queries on the site. The demo also includes bulk incremental updates of roughly 100-150MB each. We will demonstrate the unaffected query execution offered by CSO while merge-packing each of these increments online.

References

[ACT97] ACT Inc. The Cubetree Datablade. http://www.act-us.com, August 1997.

[KR98] Y. Kotidis and N. Roussopoulos. An Alternative Storage Organization for ROLAP Aggregate Views Based on Cubetrees. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 249–258, Seattle, Washington, June 1998.

[RKR97] N. Roussopoulos, Y. Kotidis, and M. Roussopoulos. Cubetree: Organization of and Bulk Incremental Updates on the Data Cube. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 89–99, Tucson, Arizona, May 1997.

[2] Online demo available at *http://www.act-us.com*