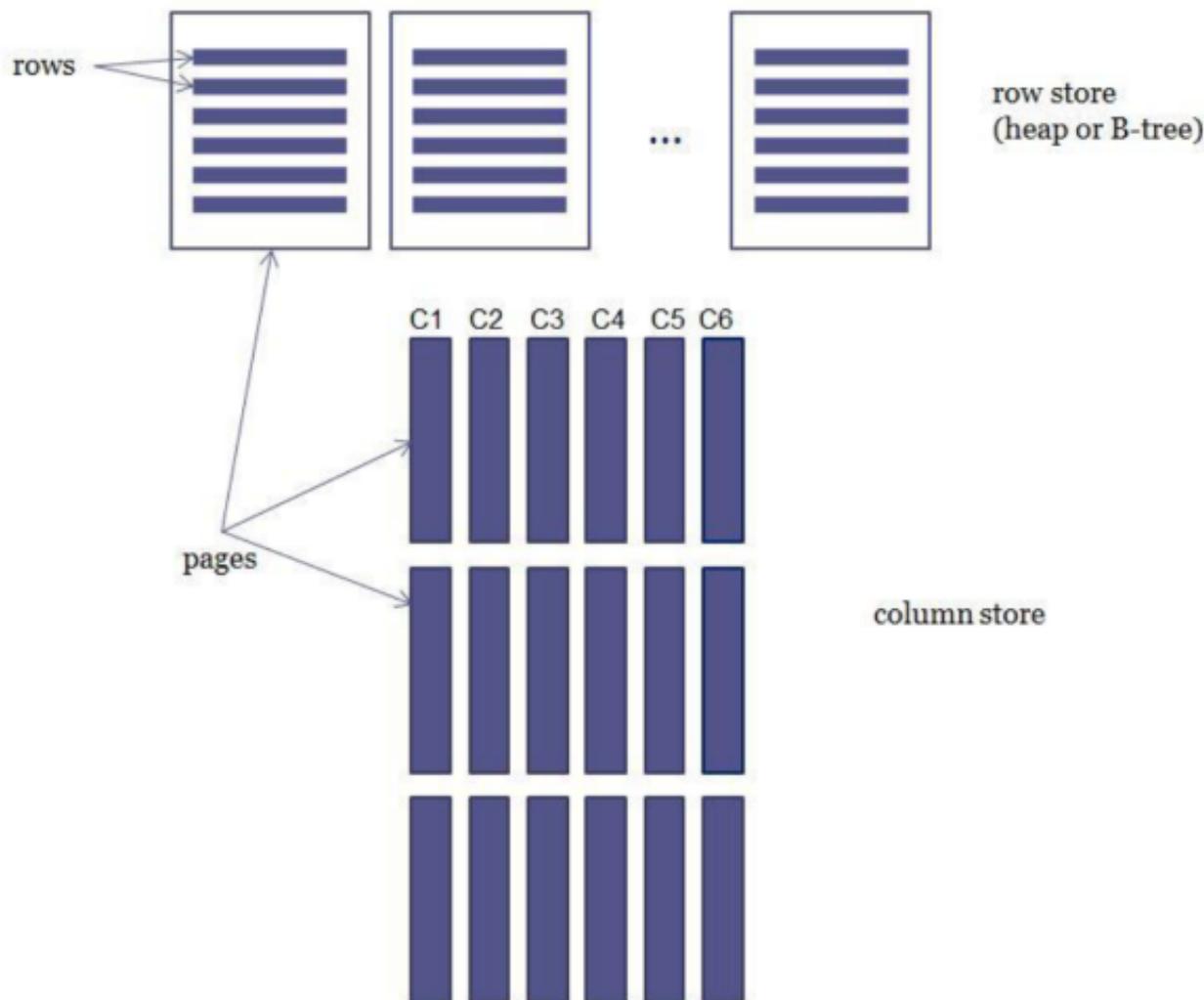


SQL Server 2012: Columnstore Index accelerates query 100X

by admin-Andrei - Saturday, December 11, 2010

<http://www.practicaldb.com/blog/sql-server-columnstore/>

Microsoft reused its patented **VertiPaq** column-oriented DB technology in upcoming SQL Server 2012 release by introducing columnstore indexes, where each columns stored in separate set of disk pages.



Above is a "compressed" extraction from Microsoft publication and I think it is very relevant to the future of Data Visualization (DV) technologies. Traditionally RDBMS uses "row store" where heap or a B-tree contains multiple rows per page. The columns are stored in different groups of pages in the columnstore index. Benefits of this are:

- only the columns needed to solve a query are fetched from disk (this is often fewer than 15% of the columns in a typical fact table),
- it's easier to compress the data due to the redundancy of data within a column, and
- buffer hit rates are improved because data is highly compressed, and frequently accessed parts of commonly used columns remain in memory, while infrequently used parts are paged out.

"The columnstore index in SQL Server employs Microsoft's patented Vertipaq™ technology, which it shares with SQL Server Analysis Services and PowerPivot. SQL Server columnstore indexes don't have to fit in main memory, but they can effectively use as much memory as is available on the server. Portions of columns are moved in and out of memory on demand." SQL Server is the first major database product to support a pure Columnstore index. Columnstore recommended for fact tables in DW in datawarehouse, for large dimensions (say with more than 10 millions of records) and any large tables designated to be used as read-only.

"In memory-constrained environments when the columnstore working set fits in RAM but the row store working set doesn't fit, it is easy to demonstrate thousand-fold speedups. When both the column store and the row store fit in RAM, the differences are smaller but are usually in the 6X to **100X range** for star join queries with grouping and aggregation."

Your results will of course depend on your data, workload, and hardware. Columnstore index query processing is most heavily optimized for star join queries. OLTP-style queries, including point lookups, and fetches of every column of a wide row, will usually not perform as well with a columnstore index as with a B-tree index.

Columnstore compressed data with a factor of 4 to a factor of 15 compression with different fact tables. The columnstore index is a secondary index; the row store is still present, though during query processing it is often not needed, and ends up being paged out. A clustered columnstore index, which will be the master copy of the data, is planned for the future. This will give significant space savings.

Tables with columnstore indexes can't be updated directly using INSERT, UPDATE, DELETE, and MERGE statements, or bulk load operations. To move data into a columnstore table you can switch in a partition, or disable the columnstore index, update the table, and rebuild the index.

Columnstore indexes on partitioned tables must be partition-aligned. Most data warehouse customers have a daily, weekly or monthly load cycle, and treat the data warehouse as read-only during the day, so they'll almost certainly be able to use columnstore indexes.

You can also create a view that uses UNION ALL to combine a table with a column store index and an updatable table without a columnstore index into one logical table. This view can then be referenced by queries. This allows dynamic insertion of new data into a single logical fact table while still retaining much of the performance benefit of columnstore capability.

Most important for Data Visualization is this:

"Users who were using OLAP systems only to get fast query performance, but who prefer to use the T-SQL language to write queries, may find they can have one less moving part in their environment, reducing cost and complexity. Users who like the sophisticated reporting tools, dimensional modeling capability, forecasting facilities, and decision-support specific query languages that OLAP tools offer can continue to benefit from them. Moreover, they may now be able to use ROLAP against a columnstore-indexed SQL Server data warehouse, and meet or exceed the performance they were used to in the past with OLAP, **but save time by eliminating the cube building process**".

This sounds like Microsoft finally figured out of how to compete with Qlikview (technology-wise only, because Microsoft still does not have - may be intentionally(?) - DV product).

Update: In May 2011 Erik Hanson, Principal Program Manager Lead (for Query Processing and Storage at Microsoft/SQL) said that CTP3 of SQL Server 2012 will be released in "Summer of 2011" and it Columnstore Index will be finally available (typical Microsoft misbehavior: basically 8 months after it was pre-announced).

Andrei Pandre, Ph.D., VP of Data Visualization, Practical Computer Applications, Inc.
