DATABASE TRENDS AND APPLICATIONS
Solutions for the Information Project Team • www.dbta.com July 2008

End of the One Size Fits All DBMS?

By Guy Harrison

In last month's article, we discussed the emergence of the non-relational cloud databases such as BigTable, SimpleDB and SQL Server Data Services (SSDS). While these new data stores may well fill a niche in cloud-based applications, they lack most of the features demanded by enterprise applications; in particular transactional support and business intelligence capabilities.

While the cloud database technologies such as BigTable were emerging, a parallel effort to determine the next stages of DBMS evolution was being sponsored by a team of database experts including Mike Stonebraker. Stonebraker - famous in database circles as the creator of Ingres and Postgres - and his colleagues had been suggesting since at least 2005 that for every significant application type, a customized database design could deliver at least a ten times improvement in performance compared to today's One Size Fits All (OSFA) relational database.

Stonebraker and his team followed up with concrete designs for database systems optimized for Data Warehousing and OLTP application processing (H-Store).

These proposals not only address issues raised by the needs of future enterprise applications - they also provide a parallel path to the requirements of next generation "cloud" databases.

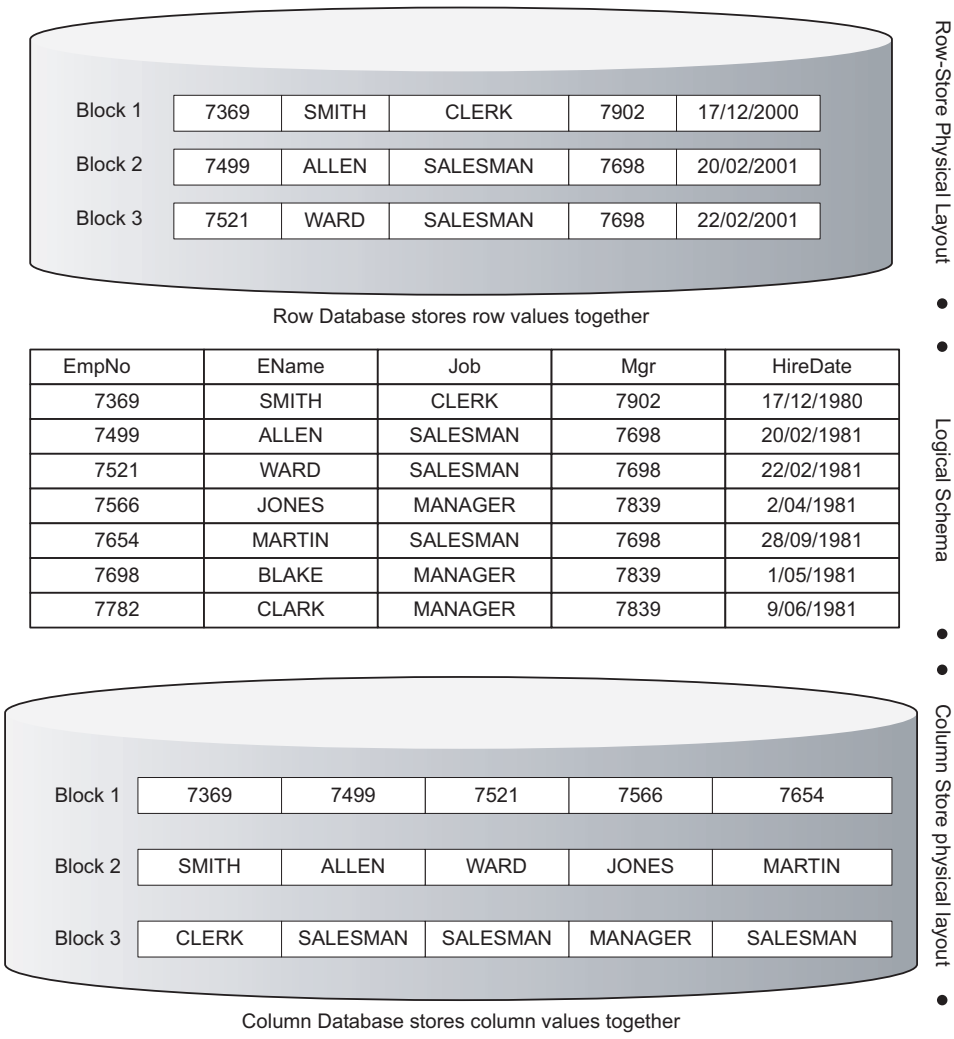

Figure 1: Column-oriented databases store values for a column together

C-Store: Next-Generation Data Warehouse

The C-Store model is based primarily on optimizing the DBMS for retrieving groups of columns rather than groups of rows. In today's relational database, it's inexpensive to retrieve all columns for a row, but very expensive to retrieve all the row values for a particular column. C-Store inverts this dynamic by optimizing for column retrievals.

Physically, C-Store organizes data in projections (slices of a table that include specific rows) that are maintained in an optimal sort order. While this does optimize queries that can be satisfied by these projections, it definitely makes inserts updates and deletes much more expensive.

C-Store employs shared nothing clustering to provide scalability, parallelism and fault tolerance. Redundant copies of all data are kept, providing built-in high availability and reducing or eliminating the need for transaction log based recovery schemes. This redundancy is achieved through overlapping projections

- each of which might be optimized for unique queries - the combination of which allow complete recovery from the loss of any particular node.

Having a column orientation also comes in handy when compressing data. You can get much higher compression ratios when columns are stored together, since there will almost always be more repeating data within columns that across all columns. Furthermore, sorted projections make compression computationally less expensive since if a row is identical or almost identical to its predecessor, storing a compressed entry to that effect is simple. Indeed, some benchmarks have shown compression rates 200 times greater than what can be achieved with traditional row-based databases.

This architecture is very effective in optimizing read-only operations, but writes are now massively more expensive since creating a single new row now requires updates across all relevant projections. To overcome this, INSERTS, UPDATES and DELETES are buffered in a "writeable store" which is optimized for writes over reads and these changes are periodically applied in batch to the main "read-store."

C-Store supports standard SQL, support for read consistency and transactions. Furthermore, the logical representation of the data is largely unaffected; C-Store concentrates on optimizing the physical layout while still supporting fully normalized relational schemas.

Mike Stonebraker is also CTO of Vertica, which offers a commercial implementation of the C-Store concept. Vertica has recently announced a cloud-based offering in which Vertica databases are hosted in the Amazon EC2 cloud. The Amazon/Vertica column-oriented cloud database offers the usual cloud advantages: rapid deployment, scalability, redundancy and pay as you go. Vertica allows databases from 500GB to hundreds of terabytes can be provisioned almost immediately and scaled up as demand increases.

H-Store: the OLTP rewrite

H-Store is described by the Stonebraker group as a "complete re-write" of the OLTP DBMS.

Disk I/O remains the biggest bottleneck for DBMS systems; while Moore's Law is creating exponential growth in the memory and CPU capacity, I/O performance has improved only slightly. To avoid this, H-Store uses a memory-based model. Rather than guaranteeing data persistence by writing to a disk, persistence is guaranteed through replication across multiple machines. In-memory data can still be backed up to disk or tape, of course, but for normal operations no disk I/Os are required. If you need more memory than a single machine can support, you add more machines to the H-Store.

H-Store employs a hierarchical data model. While hierarchical organization is less flexible (and arguably less "correct") than the relational model, it allows for highly optimized partitioning and shared-nothing clustering, which in turn allows for scale out across large numbers of machines: a necessity as well as a virtue given the memory-based storage model.

H-Store radically simplifies the concurrency model employed by relational databases to avoid many of the overhead and contention issues that arise. Each H-Store instance is single-threaded which radically simplifies locking and latching, though multiple instances can be deployed on a single machine to take advantage of multiple CPUs.

H-Store transactions are made more atomic than in the relational database model by being encapsulated into a single stored procedure call, rather than being represented by a collection of separate SQL statements. This ensures that transaction durations are minimized (no think-time or network time within transactions) and further reduces locking issues.

The H-Store proposal abandons SQL in favour of a model that has more in common with pre-relational database languages or with ORM-based approaches. In fact, the Stonebraker team suggest the Ruby on Rails ActiveRecord Object Relational Modelling framework as a possible SQL replacement.

The Stonebraker/H-Store team declares that H-Store has delivered 80-times improvements in throughput on TPC-C benchmarks.

A Possible Future

The emergence of CloudDBs, together with the C-Store and H-Store proposals have allowed database experts - for the first time since the 1990s - to imagine a future in which the relational database of today has been depreciated in favor of new technologies.

The very simple CloudDB approach and the more sophisticated H-Store approach seem to represent the extreme ends of a architectural continuum that could satisfy most OLTP requirements. H-Store and CloudDB both offer scalability, economy and performance, with H-Store layering on transactional support and other features required by many enterprise applications.

The requirement that valuable business data does not become lost in the cloud can be provided by the establishment of C-Store-oriented data warehouses which are the ultimate recipients of data required for BI and OLAP purposes.

None of which is to suggest that today's One Size Fits All relational database is about to undergo rapid disruption. But for the first time in a long time, it's possible to imagine a world in which the relational database is not the database king.

Guy Harrison is chief architect for database solutions at Quest Software.