The Vertica Analytic Database – Introducing a New Era in DBMS Performance and Efficiency

- *Stored 20TB of data on $30,000 worth of hardware and queried it in under 4 minutes*
- *Queried 214x faster than a proprietary data warehouse hardware appliance*
- *Loaded data 3x faster than a major OLTP database*

Read about these benchmark results and the innovative Vertica Analytic Database architecture that produced them.

THE TIME FOR DATABASE INNOVATION IS NOW

As businesses become more analytical to gain competitive advantage and comply with new regulations, they are changing the demands placed on data warehouses and driving demand for new query-intensive database applications. The table below illustrates how data warehouse workloads have changed over recent years as they are pushed to answer more ad-hoc questions from more people analyzing vastly larger volumes of data in near real-time.

	Yesterday	Today
Data volume	100s GB	100s TB
Standard reports	100s	10,000s
Ad-hoc query users	10s	100s
Granularity	Aggregate	Detail records
Data loading	Nightly	Continuously
Access	Reports	Reports, plus...SOA apps, dashboards with alerts

Yet, during the last 30 years, there has been little database management system (DBMS) innovation to keep pace. Performing ad hoc queries on such large data volumes does not come naturally for existing DBMSs, which use a row-oriented design optimized for write-intensive transaction processing workloads rather than for read-intensive analytical workloads. Desperate for better performance, row-oriented DBMS customers spend millions of dollars annually on stop-gap measures such as adding DBA resources, creating and maintaining OLAP cubes or replacing their DBMS with expensive, proprietary data warehouse hardware.

Led by database technology pioneer Dr. Michael Stonebraker, Vertica has developed a break-through SQL database that offers all sizes of companies a competitively advantageous and cost-effective new way to analyze terabytes of information. Customer benchmarks show that the Vertica Analytic Database:

- Answers queries **50-200x faster** on average
- **Requires 10x-20x less storage** than other databases due to aggressive use of compression
- **Reduces DBMS hardware costs by up to 70%**

Vertica required just 5 servers, with 1TB of local disk each to store 21TB of CDR data and provide query results in less than 4 minutes on average.

This white paper describes the innovative architecture of the Vertica Analytic Database and customer benchmark results that show how it provides such remarkable performance compared to other data management solutions for read-intensive analytical database applications such as:

- Business Intelligence
- Data warehousing
- Call detail record analysis
- Network analysis
- Historical trade data analysis
- Web clickstream analysis
- RDF data analysis
- On-line customer self-service
- Fraud detection
- Basel II compliance
- Reg NMS compliance
- Customer and profitability analytics

THE VERTICA ANALYTIC DATABASE ADVANTAGE

Vertica believes it's time for significant innovation in the database industry and has developed a SQL database that provides blindingly fast query performance for databases scaling from hundreds of gigabytes to hundreds of terabytes. What makes the Vertica Analytic Database architecture so unique is that it combines the database industry's most significant recent innovations into a single database for the first time:

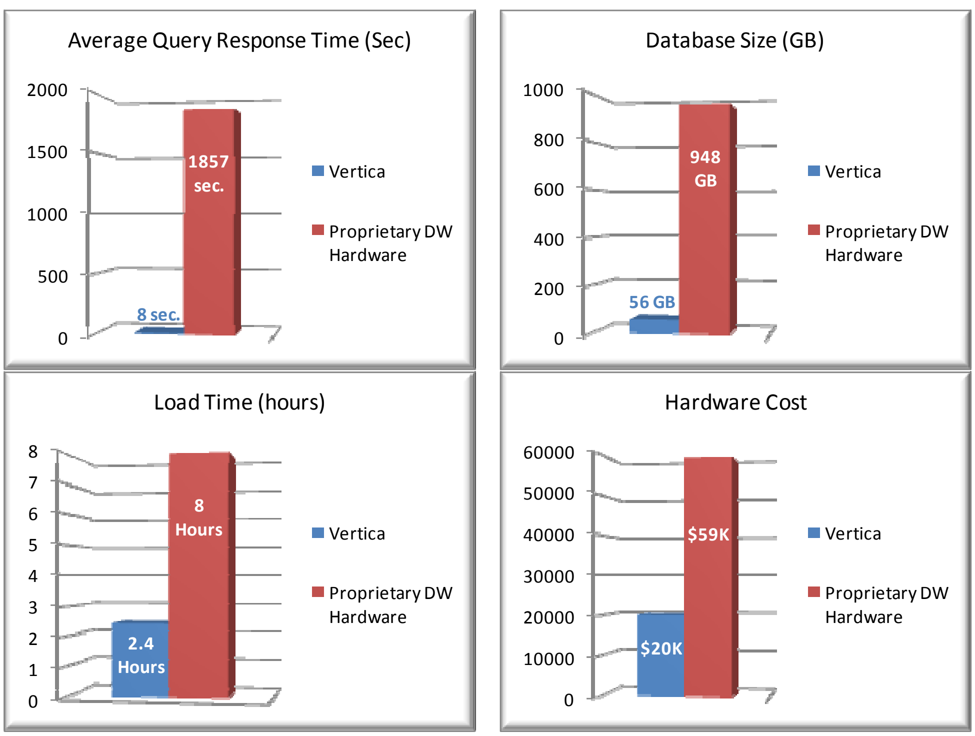
- Column architecture
- Aggressive data compression
- A hybrid transaction architecture that supports concurrent, parallelized data loading and querying
- Automatic physical database design
- Multiple physical sort orders ("projections")
- Automatic "log-less" recovery by query
- High availability without hardware redundancy
- Support for grid/shared-nothing hardware architecture to enable scaling "out" not "up"
- Connectivity to applications, ETL, and reporting via SQL, JDBC and ODBC

Many of these innovations are already familiar to database architects, and Appendix I describes them in more detail. Through these innovations, Vertica has created a faster, simpler and more economical DBMS that is optimized for read-intensive applications such as data warehousing. The following benchmark studies show how the Vertica Analytic Database performs in practice.

To download Vertica and benchmark it yourself, please request a copy at www.vertica.com.

21 TB Call Detail Record Analysis Benchmark (20B calls)

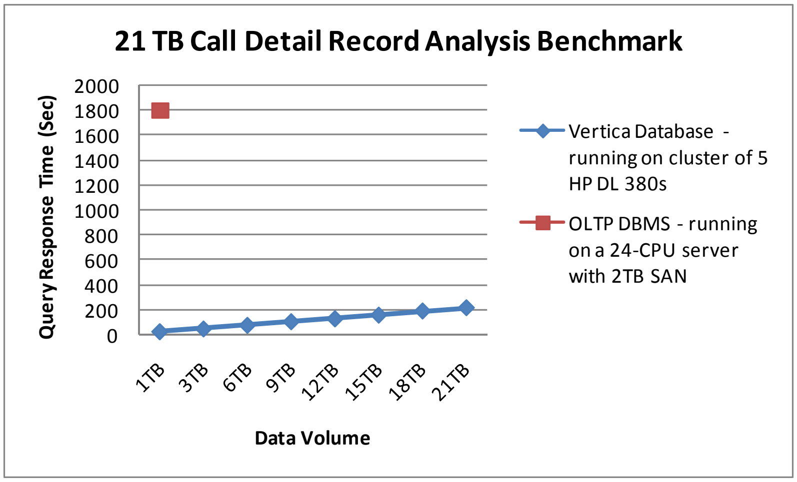
A wireless communications provider wanted to compare the performance of Vertica to a proprietary data warehouse hardware appliance. They used a 600GB data set and 14 billing-related queries for the comparison, then the test was scaled up to 21TB to see how Vertica handled the load.



600GB (600M Call) benchmark results

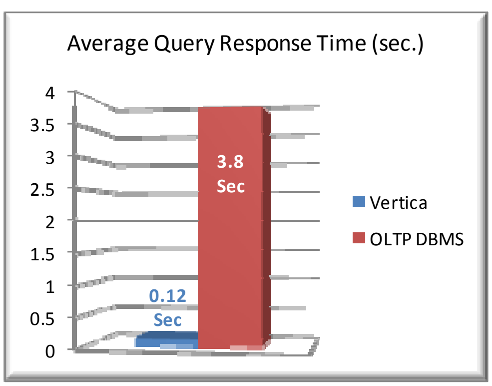
The data set featured one 200-column wide table (Vertica excels at searching wide tables as it only ever reads the columns in the query; row-stores such as those embedded within the appliance in this test, must scan the entire table regardless of which columns are in the query).

Vertica was run on a 4-node cluster of Linux servers (each with 2CPUs, 8GB RAM and 1TB of disk (4x2500GB SATA); the OLTP DBMS was run on a 4-CPU appliance with 24TB of disk (48 500GB drives) and 16GB of RAM. On hardware costing 50% less, Vertica answered queries 214x faster and loaded data nearly 3x faster. Compression enables Vertica to store almost 20x more data on 1/6th of the disk space. Since once could claim Vertica had a CPU count advantage in this test, Vertica was also run on a standalone server (with just 2 CPUs), and the query time increased from 8 seconds to 49 seconds (running 38x faster than the appliance).



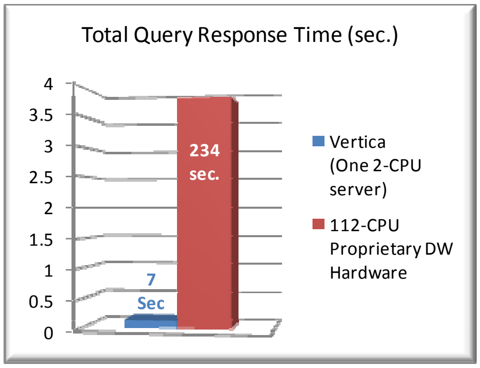
Then the test was scaled up to 21TB (on Vertica only) and run on a cluster of 5 servers (same spec as above). The graph to the left depicts the average query time for the Vertica Analytic Database as it grew from 3TB to 21TB (at each 3TB increment). The final database was 2.1TB in size—occupying 90% less disk space than the raw data, and the 21TB of data was loaded in 58 hours. As a reference point, the graph above also includes the query response time for a 1.2TB CDR database running on a 24-way hardware configuration costing $500,000—showing that Vertica is capable of querying 17.5x more data 8 times faster on 90% less expensive hardware.

Historical Trade Data Analysis ("Tickstore") Benchmark



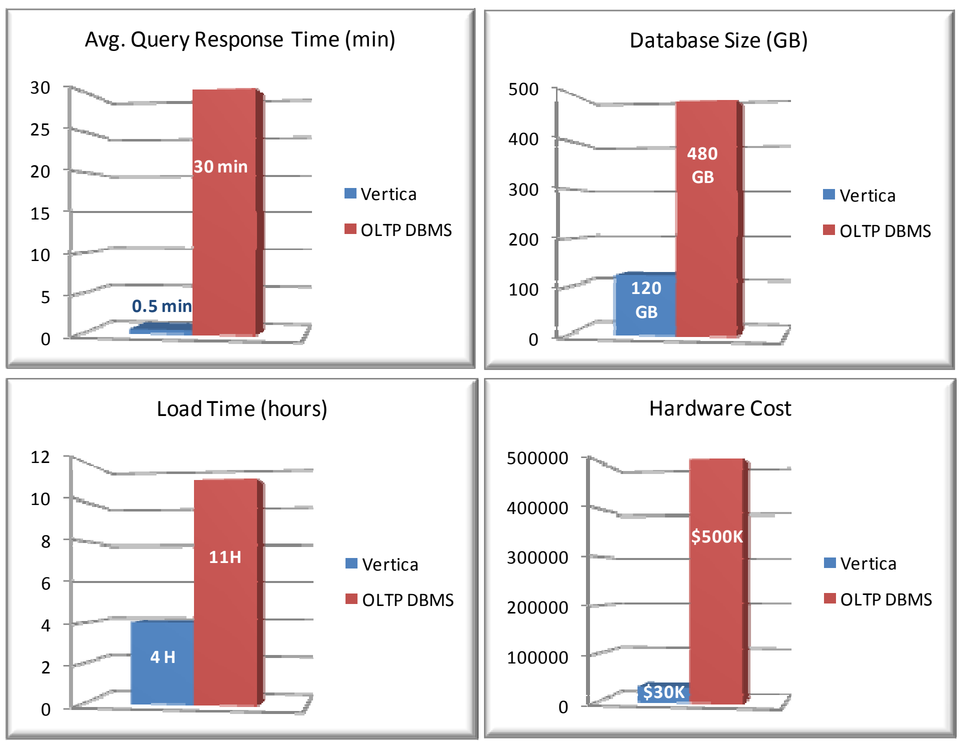
A global investment firm compared the speed with which Vertica can query historical trade and quote ("tick") data to the query response times of a financial services OLTP database. Performance is key in this domain as it allows investment firms to test and refine their predictive trading algorithms against larger historic data sets (years instead of weeks). The average query response time was 32x faster in Vertica, and the slowest OLTP DBMS query ran over 3,000x faster in Vertica. Both databases were run on identical hardware; a single 2-CPU server with 500GB of storage.

Web Clickstream Analysis Benchmark



An on-line marketing analytics service provider compared the time required to execute a series of 217 cascading SQL statements that create temp tables and query them in order to analyze web site user behavior for a specific customer demographic segment. Running against 25M rows of clickstream and user data, Vertica executed the sequence in 7 seconds running on a single 2-CPU server—34 times faster than a proprietary data warehouse hardware appliance containing 112 CPU:Disk pairs.

1.2TB Call Detail Record Analysis Benchmark – (1.5B calls)



1.2 TB (1.5B Call) CDR benchmark results

A telecom provider was unable to manage more than 30 days of call detail records at a time due to database performance limitations. They decided to investigate whether Vertica could manage this data better and more cost effectively in order to have many more months' of CDRs on line for analysis.

They conducted a benchmark based on several days' worth of CDRs. Vertica was run on a 3-node cluster of Linux servers (each with 2CPUs, 16GB RAM and 1TB of disk (2x500GB SATA); the OLTP DBMS was run on a 24-CPU server (48GB RAM) connected to a 2TB SAN.

The Vertica query response time was 60x faster; load time was nearly 3x faster; the database used 75% less disk space and the hardware required to run Vertica cost 95% less.

IN SUMMARY: A DBMS ARCHITECTURE BUILT FOR ANALYSIS

Together, the key features of the Vertica Analytic Database create an elegant architecture for large-volume, high-speed analytics applications. Vertica's vertical partitioning by column, extensive use of compression, and hybrid storage model reduce the I/O required to execute queries. The types of analytic applications that use a small, arbitrary subset of columns per query are both common, and ideally suited for the vertical partitioning provided by the Vertica Analytic Database. At the same time, Vertica's data partitioning, which divides work across multiple nodes in a computer cluster, supports the very large data volumes to which these applications often grow. These analytic applications also typically support users from many disciplines, interrogating the database from multiple individual perspectives. Here, Vertica's ability to keep multiple physical projections of the data is a natural fit for such usage patterns. In summary, the architecture of the Vertica Analytic Database was designed specifically to handle the common characteristics of read-intensive analytics applications.

BENCHMARK THE VERTICA ANALYTIC DATABASE YOURSELF

Getting started with the Vertica Analytic Database is easy. It supports SQL, and integrates with ETL, analytical and reporting tools, and business intelligence applications via JDBC, ODBC and specific language bindings.

If you would like to learn more about the Vertica Analytic Database and how it can help your company analyze data more effectively, or if you would like to run your own benchmark test, please visit www.vertica.com to find out more.

Vertica Analytic Database System Requirements:

A cluster of 1 or more (at least 3 for production use) shared-nothing computers, each featuring:

- Dual-core 64-bit CPU (at least 1.6GHz)
- At least 2GB of RAM per CPU core
- 1TB of disk (e.g., 4x250GB drive, 10K RPM, SATA or SAS)
- Red Hat Enterprise Linux 5, SuSE 10 or Fedora Core 6

APPENDIX I: VERTICA ANALYTIC DATABASE – KEY INNOVATIONS

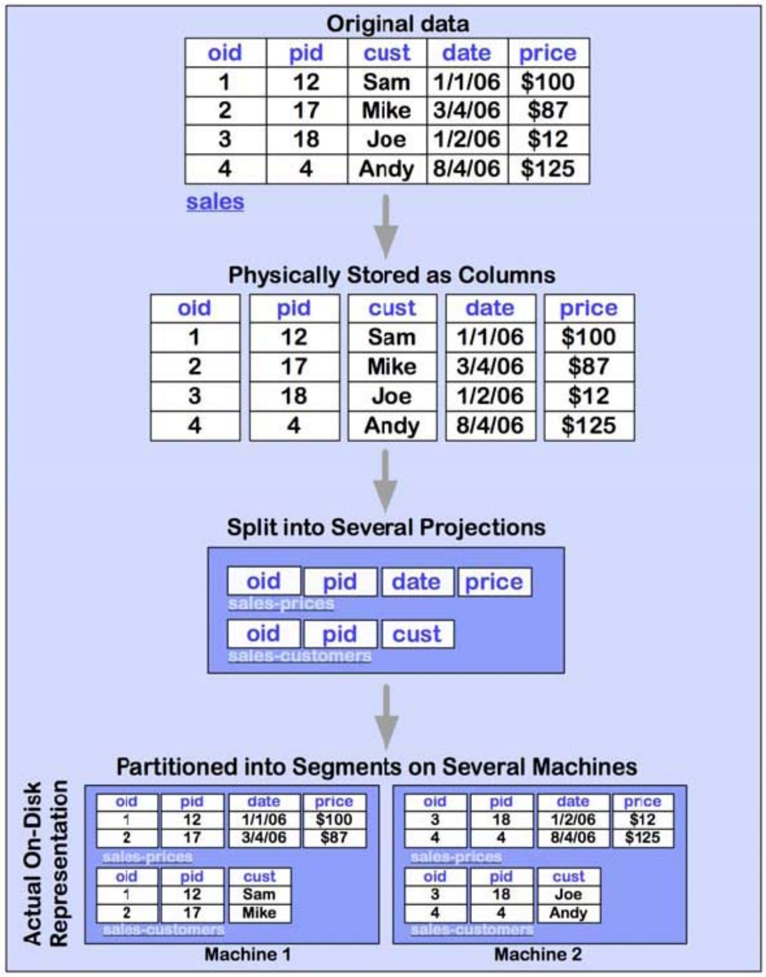
From a database developer perspective, the Vertica Analytic Database looks very standard; it supports SQL, ACID transactions, JDBC, ODBC and works with popular ETL and BI reporting products. Underneath the covers, it's a different story. Vertica is designed to aggressively economize disk I/O and is written natively to support grid computing. Vertica is a 21st-century solution for today's large-scale, read-intensive database applications, featuring ground-breaking architectural features such as:

Column Store Architecture. In the Vertica Analytic Database, data for each column is independently stored in contiguous blocks on disk. Column values are linked into rows implicitly, based on the relative position of values in each column. Unlike most databases where all columns for each row are stored together, Vertica only needs to retrieve those columns needed for a specific query, rather than all columns in the selected rows (see Figure 4 below). Vertica's vertical partitioning approach produces dramatic I/O savings for the large majority of queries that only retrieve a subset of columns. For example, consider a telecommunications company that has call detail records with 230 columns. Most queries touch fewer than 10 of these, reducing disk I/O time by over 50% compared to row-oriented databases and proprietary data warehouse hardware.

Aggressive Compression. Vertica employs multiple compression algorithms, depending on data type, cardinality, and sort order, to minimize the space occupied by a column. These include run length encoding, delta value encoding and integer packing for integer data, block-based dictionary encoding for character data, and Lempel-Ziv compression. Vertica automatically chooses a good algorithm for compressing data in each column based on a sample of the data. Compressing data by column often improves compression ratios because the data shares a common data type and value range. Run-length encoding (RLE), for example works best for columns of ordered data, or data with few distinct values compared to the number of rows. This ensures long runs of identical values, which RLE compresses quite well. Vertica demonstrates overall compression ratios ranging from 4x to 10x relative to the ASCII input data. The Vertica query engine processes data in compressed form.

Multiple Projections Stored. Instead of storing data in tables as defined in the logical schema, Vertica physically stores views of the table data, called projections. Each projection contains a subset of the columns of a table in a particular sort order. Rows in a projection consist of the value at the same position in each of the column stores comprising the projection (see Figure 4). Projections can also contain columns from multiple tables, thus materializing joins. To support ad hoc queries, every data element is guaranteed to appear in at least one projection. Vertica automatically selects appropriate projections to optimize query performance for the expected workload. Benefiting from large storage savings due to its extensive use of compression, the Vertica Analytic Database can maintain multiple projections with different and often overlapping sets of columns, in several different sort orders, to improve performance for a wide range of expected queries.

Figure 4: Vertica Analytic Database Architecture



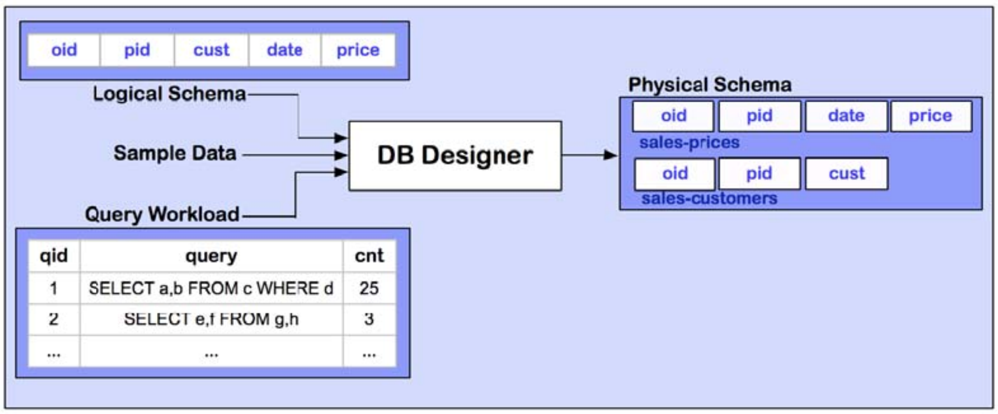
Shared-nothing Parallelism. The Vertica Analytic Database is a shared nothing system, designed to run on a collection of homogeneous nodes of a Linux cluster or grid connected by a TCP/IP network. Nodes contain commodity, multi-core processors with 2 to 4 GB of RAM per core. Storage can be directly attached to each node, or can be SAN-based. In the Vertica Analytic Database, parallelism is optimized for star or snowflake data models. Fact tables are range partitioned across the nodes of the cluster (see Figure 4). Dimension tables are typically replicated on each site of the cluster. Very large dimensions are partitioned on the same key as the fact table. This limits the need to share data among nodes during query execution. Queries can be initiated on any site. The query planner determines what work needs to be done to answer the query, distributes it to participating nodes, collects each node's partial result and prepares the final answer to be sent to the requestor.

K-Safe based Availability. The Vertica Analytic Database maintains multiple stored projections, which can also serve as redundant copies of the data for purposes of high availability. By imposing an additional constraint, namely, that the system guarantees that projections are partitioned such that each data element exists on multiple nodes, Vertica implements intelligent data mirroring as an integral part of the database. Vertica calls this K-Safety, where *k* is the number of node failures that a given set of Vertica projections will tolerate. Vertica guarantees K-Safety by building *k*+1 replicas of all segmented projections - where each replica has the same columns and partitioning key, though the sort order may differ – and offsetting the distribution of partitions across nodes for each replica. K-Safety allows requests for data owned by failed nodes to be satisfied by existing projections on surviving nodes, even though the optimal projection may no longer be available. Once the failed node is restored, Vertica uses the projections on the other nodes to automatically re-populate data on the previously failed node.

Automatic Physical Database Design. With the Vertica Analytic Database, users need only specify a logical database schema. Today, Vertica targets star or snowflake schemas, so the logical model must satisfy this constraint. Given a logical schema, the Vertica DB Designer automatically generates an appropriate physical database design based on that schema, a sample of representative data and queries, and a space budget for the database (see figure 5).

The DB Designer guarantees that any valid query on the schema can be answered by insuring that all data appears in at least one projection. The DB Designer chooses compression technique to use for each column. It determines which projections to build in order to optimize performance of the sample workload, which columns and joins they will contain, and how each projection should be sorted. It selects the appropriate partitioning key for each projection and guarantees that the projections satisfy the specified K-Safety level. By taking over the details of the physical design, the Vertica Analytic Database simplifies database implementation and allows database designers and administrators to focus on the best logical data model to meet their business needs.

Figure 5: Vertica DB Designer Process



<u>**Hybrid Storage Model.**</u> The Vertica Analytic Database caches all database updates to a queryable main memory cache called the Write-optimized Store (WOS) (see figure 6). The WOS organizes data into projections that are stored as collections of uncompressed, unsorted column arrays, maintained in update order. Periodically, an asynchronous background process on each node, called the Tuple Mover, migrates recent updates to permanent disk storage in the Read-optimized Store (ROS). Data in the ROS is sorted, compressed, and densely packed into variable length disk blocks, optimized for query performance. Appended data is simply added to the end of the appropriate column stores in the ROS. Data inserted into the middle of sorted projection causes the Tuple Mover to rebuild and rewrite disk blocks to maintain the sort order and dense data packing in the ROS. SQL queries can be issued to execute against data in the ROS only, or the ROS+WOS if real-time results are required.

The Vertica Analytic Database supports snapshot isolation for query processing, which means that queries and updates do not interfere with one another, and that read-only queries do not require locking. Updates are collected in time-based buckets of fixed duration called epochs. At fixed intervals, Vertica closes the current epoch and begins a new one. New updates are grouped in the new current epoch. Data in older epochs is available for query processing and eventual migration by the Tuple Mover to the ROS (see figure 4).

Vertica's hybrid storage model supports both bulk loads and trickle-feed updates. The Vertica Tuple Mover is regularly working in the background to drain the WOS and merge updates into the ROS to keep it current. While clearly designed for read-mostly application, this approach also works for near-real time data warehouses with high append data volumes, as long as the data latency requirement exceeds the epoch period.

Figure 6: Vertica's Hybrid Storage Model