SCALEDB
MANAGING OPERATIONAL ANALYTICS AT A LOW TCO

Recent years have witnessed a dramatic increase collecting data from various sources such as: sensors, devices, the internet, and from independent or connected applications.

These enormous amounts of data are the basis for many current and emerging applications that try to provide useful operational insights from the data. These applications share a common requirement - they require support for online real-time analysis of rapidly growing data streams.

Examples of streaming (and thus operational analytic) applications are:

- IoT Applications – where different sensors provide data that needs to be analyzed.
- Internet security – where Internet packets are considered to detect critical conditions.
- Financial applications - analysis of stock prices to discover trends, correlations and opportunities.
- Gaming – to discover and track behaviors of users during online gaming.
- Telco applications – analyze and track information relating telephone call usage.

Operational Analytics process new data sets that are very different from traditional data or analytics. The new data sets, called streaming or time-series data, are characterized with huge volume, velocity and their inherent correlation with time. As data managed by traditional databases or analytics is relatively static with no pre-defined notion of time, traditional databases are not well equipped to efficiently manage streaming data.

In particular, operational analytics streaming applications require new challenges from data management systems:

Velocity – data is arriving continuously and fast - collecting streams of events such as monitoring power usage, click streams of websites, Internet packets or data generated from an online game are all examples where millions of events per second may need to be ingested to a database. Other examples include smaller velocities of multiple streams that aggregate to high velocity. As traditional databases do not perform well when data appear at high velocity, Stream Processors evolved. These products support limited functionality that only provides a summary of the data or only consider a “window” consisting of the last n elements of the stream. The drawback of this approach is the limited querying opportunities – only the most recent data is available, the source data is lost, advanced query technics (such as joins) are not supported and the richness and simplicity of SQL are not available. Some users leverage memory databases to ingest the data streams. However, managing large data sets in memory is very expensive - a stream of a million events per second requires 360 Giga of RAM every hour (assuming that every event is 100 bytes long). This is more than what the largest Amazon instance currently offers and to maintain the data over time, a new instance would be required every half an hour. Other users have moved to a NoSQL solution. NoSQL trades database properties for performance. But even with NoSQL, scaling is complex and entails considerable efforts. This Cassandra implementation achieved 1M inserts/second in EC2 by running a cluster of 96 nodes in each zone.

Volume - streaming data is one of the main sources of what is called Big Data. With streaming data, data volume grows continuously and queries needs to evaluate unbounded data sets. With continues streams, queries may very quickly require scans over hundreds of millions and billions of rows. These types of queries are not performed well by traditional databases that use conventional indexing and for some users justify the migration away from the database platform. The new migrated to environment is a complex, three tiered architecture (called Lambda Architecture) that includes a batch layer (Hadoop), a serving layer (that indexes the batch views so that they can be queried), and the speed layer (Storm). The cost and complexities of the technologies around Hadoop are the reason for Gartner’s findings in latest surveys that the number of CIOs that think that Hadoop will replace their existing analytics infrastructure has plummeted over the last few years, and is now down to just 3% (see also more info here).

ScaleDB Operational Analytics Approach:

ScaleDB offers a completely different approach by extending the MySQL database to manage Streaming and Real-Time Data. The database extensions use special stream engine technologies that integrate with the existing database and storage kernel. This approach adds specialized stream features (including management of Time Series Data) to the existing database functionality such that stream data can be managed by the database while maintaining complete and unified database functionality.

ScaleDB transforms a single database instance (such as MySQL or MariaDB) to a cluster of database and storage services that provides scalability and high availability features that exceed the capabilities of a single database instance. A ScaleDB cluster is made of two tiers – a storage tier and a database tier. The database tier includes multiple MySQL instances that process shared data that is managed in the storage tier. Both tiers use commodity or virtual machines. For Streaming and Time-Series Data ScaleDB provides the following:

Velocity – A ScaleDB cluster ingests millions of inserts per second.

Volume – ScaleDB is a disk based solution. The data persists on the disks of the machines in the storage tier. A ScaleDB cluster is tuned to process Streaming and Time-Series Data such that data volume does not impact performance.

Query functionality – ScaleDB executes Business Intelligence (BI) types of queries over Streaming and Time-Series Data with “pushdown” technology. With the pushdown technology, queries are “pushed” from the database tier to the storage tier and are executed next to the data (similar to MapReduce). This approach allows for distributed processing where billions of rows are evaluated within few seconds.

Query rewind – As Streaming and Time-Series Data is unbounded, ScaleDB offers a rewind mechanism where queries are executed in a sequence of cycles to provide continues view on incoming data. This approach allows applying a SQL query to unbounded data stream.

Simplicity – Streaming and Time-Series Data is processed with the same ease and together with other data. The schema is described using SQL through one of the database nodes in the cluster and becomes available to use to all the nodes in the cluster. Scaling is done by adding nodes to the cluster without the need to redesign the schema or partition/shard the data. High Availability (HA) is transparent.

TCO – ScaleDB offers the best TCO – it is simple to develop, deploy and manage. It provides the highest level of performance and scalability and uses commodity hardware. It harnesses the MySQL echo system to support the most complex Streaming and Time-Series Data applications.

The current approach of stream processing systems is built separately from the database. The separation cause significant overhead in data access and data movement, and is not able to take advantage of the functionalities already offered by the database. ScaleDB combines both stored and streaming/time-series data in an integrated MySQL environment offering large scale real-time streaming applications Velocity and Volume at a very low TCO, enabling a new host of critical business operation analytics implementations.