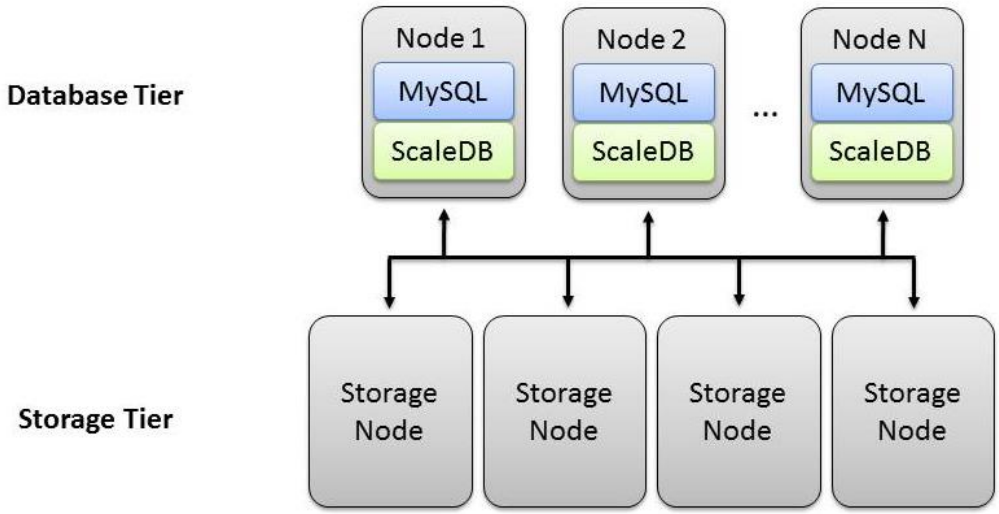
## ScaleDB: Business Intelligence with MySQL/MariaDB

As more and more companies recognize the benefits of running business intelligence (BI) against Big Data, they are now demanding next to real-time BI against the data as it is generated. BI applications require fast insertion rates to manage millions of inserts per second. These inserts generate petabytes of data that needs to be managed and queried.

The need for efficient BI processes is within every industry and the difference between an operational systems and BI systems shortens. A call center for a Telco company that records phone calls needs to satisfy queries by caller id, may, at the same time, need to efficiently query the same data to identify fraud.

ScaleDB boosts MySQL/MariaDB performance by running a single logical database across a cluster of servers. This enables inserts to be spread across multiple commodity servers or cloud machines while enabling any node to handle real-time queries against the entire data set. As demonstrated below, ScaleDB consists of 2 tiers – a database tier and a storage tier.




Each node in the database tier includes a MySQL/MariaDB server instance and a ScaleDB database engine. The applications interact with the MySQL/MariaDB instances but the data is managed by ScaleDB transparently. Rather than update or query the data using a single database instance, multiple database nodes are updating and querying the shared data concurrently. ScaleDB synchronizes all the updates and queries of the different database nodes and the users see a single consistent view of the data.

In the storage tier, the data is stripped across multiple nodes. Each node manages a portion of the data and the collection of storage nodes manages the entire collection of data. This configuration is similar to a RAID device that distributes I/Os across multiple disk drives. ScaleDB's storage tier distributes I/Os across multiple storage nodes. When a database node needs to update or query the data, it will interact with the specific storage node—or nodes—that store that data.

This approach manages large data sets efficiently. The cluster processes the data by a collection of database and storage nodes that offer a significant amount of resources (such as CPUs and memory) compared to a single database instance. The user applications continue to work against the

MySQL/MariaDB instance without the need to change the applications. Scaling the cluster is transparent to the applications – to scale, a user adds database or storage instances and there is no need to partition or shard the data.

For BI applications, ScaleDB provides fast data load and special purpose index mechanisms that delivers millions of inserts per second.

A major difference in managing BI queries when large data sets are considered is that SQL processing is offloaded from the database nodes to the storage nodes. This process pushes the SQL processing as close to the data as possible and allows for multiple storage nodes to operate in parallel. This reduces CPU consumption on the database server and reduces bandwidth as only rows that satisfy the queries are transferred to the database nodes.

**Conclusion**

ScaleDB transforms MySQL/MariaDB to a dynamic cluster of database and storage nodes that efficiently manage very large sets of data. For large BI applications, ScaleDB offers extremely fast data loads and efficient index mechanisms. Business Intelligence queries are pushed from the database tier to the storage tier to provide near real time performance when large amounts of data are evaluated. These processes deliver the efficiencies that the datacenter needs while leveraging the MySQL/MariaDB ecosystem and environment.