ScaleDB
Scaling MySQL to New Heights

ScaleDB for MySQL – Clustering Technology

ScaleDB for MySQL is a database server engine designed specifically for clustering. It leverages a shared-disk architecture that overcomes the limitations of the traditional shared-nothing approach. When coupled with MySQL, it provides a highly scalable and available database solution for all your business applications. ScaleDB's shared-disk architecture enables multiple MySQL servers, or nodes in the cluster, to simultaneously process data from a single database that is stored on a network drive.

Cluster-Level Load Balancing

Because each node has full access to the entire collection of data, it can satisfy any database request. This compares to shared-nothing, where each server specializes on only a section of the data. The benefit of the shared-disk approach is that the application can request data from any available node, instead of waiting for the node that owns a specific piece of data, as you would with shared-nothing.

This ability to work with any ScaleDB node means that you have cluster-level load balancing. In other words whatever happens to be the busy workload at the time can be spread across the entire cluster. Shared-nothing database clusters are much more rigid, being tuned for a specific type of workload, so the application ends up waiting for various servers depending upon usage.

Eliminates Data Shipping

Another benefit of enabling any node to satisfy any database operation is that it minimizes the shipping of data between nodes that you get with shared-nothing. In a shared-nothing system, a query that involves two or more nodes, requires that one node processes the query, then batches the data and ships it to the second node to process. This is repeated for each additional node involved. This same process occurs with distributed 2-phase commit, except that the nodes lock the data and wait for the entire transaction to commit. However, ScaleDB's shared-disk architecture enables any single node to handle even the most complex query or 2-phase commit by itself. This eliminates serialization, synchronized locking and data shipping, all of which can have a very negative impact on performance.

Delivers High-Availability

Because any ScaleDB node can handle any database request, the loss of one or more nodes doesn't cause the application to fail. On the contrary, the load is seamlessly distributed across the remaining servers, and depending upon the load, the database may slow down, but it won't fail. Contrast this to a shared-nothing database, where the failure of a single server causes the application to fail until you can bring a back-up server online which generally takes about half an hour. Then even if the back-up is 100% up to date, which it generally isn't, that server's performance may remain painfully slow for hours while the server warms-up its cache by processing real transactions.

ScaleDB
Scaling MySQL to New Heights

Solving the Data Consistency Challenge

For a database to be consistent, it must deliver the same results to all requestors at any time. Why is this important? If you have $20M in your account and you wire $15M to another account and then simultaneously go to a teller and withdraw $15M, the bank should know that you only have $5M left. A shared-nothing database makes copies of the data on slaves for fail-over. If you start using these slaves for read access, you run the risk of inconsistent data. ScaleDB uses a single trough of data that all nodes in the cluster manipulate. Because there is only one set of data, and since ScaleDB uses transactions, data consistency problems are a thing of the past.

Dynamic Scalability

ScaleDB's shared-disk architecture enables dynamic scalability without interrupting the database. Since you have one trough of data and multiple nodes manipulating that data, it doesn't matter how many nodes you are using. You can dynamically add a node and your processing throughput increases. Remove a node and your throughput drops. But in both cases, the database keeps running, assuming you have at least one node active. You don't have to shut down and repartition the database like you do with shared-nothing databases. This capability is very helpful for mission-critical applications that must keep planned downtime to a minimum. It is also extremely valuable to cloud computing, which promises scalability on demand, something ScaleDB delivers.

Lower Total Cost of Ownership (TCO)

ScaleDB's shared-disk clustering eliminated the need to partition and re-partition your data. The process of partitioning data is very time consuming, since you need to split the data such that your database achieves optimal performance under a specific usage pattern. As that usage pattern changes, or data grows, you must repartition the data. ScaleDB eliminates this expensive process. ScaleDB also enables you to extract every bit of performance from your servers. It doesn't rely on slaves that simply wait for the master node to fail, like shared-nothing. As a result, you need fewer servers. These advantages combine to result in a dramatic reduction in the total cost of ownership as a result of using ScaleDB.

Summary

ScaleDB utilizes its modern shared-disk architecture to deliver the following advantages:

- High-availability that seamlessly handles node failure without interruption
- The database adapts to shifting workloads via cluster-level load balancing
- Eliminates data shipping, resulting in increased performance
- Ensures data consistency, delivering correct information every time
- Dynamic scalability that grows with your needs
- Lower Total Cost of Ownership (TCO)

ScaleDB
3723 Haven Avenue
Menlo Park, CA 94025
☎ (650) 587-8787
(650) 587-1571
info@scaledb.com
www.scaledb.com