

Achieving Zero Downtime for Percona XtraDB Deployments

EXECUTIVE SUMMARY

MySQL deserves its top spot in database popularity contests. Intuitive and resilient; this workhorse is a long-time favorite of developers. It's tried and true for most companies, but when you're lucky enough to be part of a high-growth organization, suddenly the favorite is no longer a favorite. Scaling is much harder than anticipated. You need a database solution that helps, not hinders, that well-deserved, stellar growth. MySQL fork Percona sticks closely to standard SQL releases, and its storage solution XtraDB Cluster seems a perfect fit for this challenge. But they don't call them growing pains for nothing. Even with a new replication solution on your side, there are big challenges when implementing the Percona XtraDB Cluster.

Scaling

App developers clinch their teeth when conversations about scaling out arise. Extensive development time is needed to account for read/write splits. Assuring data consistency is difficult due to inevitable replication lag. And adding or removing servers requires careful coordination between app teams and DB teams to guarantee accurate changes to connection strings and app logic.

Failover

Replication and data traffic switch are pivotal concerns during a MySQL failover. Apps commonly send writes to more than one master in a multi-master set up running the risk of replication conflict. Making a new master can help, but intervening to make that happen requires downtime. TCP load balancers handle data traffic but not SQL connections. And because Percona XtraDB Cluster relies on a quorum, organizations are still expected to make config changes to prevent nodes from failing.

Visibility

To troubleshoot database issues, most organizations rely on slow logs and write logs. Both are valuable, but if an app is read-heavy, these logs only show write-related issues. Full db logging is needed to discern read performance, but that hurts the overall speed and performance of the database. So organizations find themselves caught between the desire to analyze performance and the need to maintain performance.

Performance

Query caching is obviously beneficial for apps with a large body of read-only data. But a workload that's being frequently written to has an additional challenge that the two most common approaches can't solve. MySQL's Query Cache isn't much help in that case because if anything is added midstream, cache invalidation occurs for the whole table. Memcached requires additional hours from teams to rewrite code and rework applications. And with no way to see cache usage stats built in, associates are still guessing at performance.

Connection Management

The free and open-source software LAMP makes it extremely popular as an application building block. But apps built on a LAMP stack present a new challenge in MySQL environments. Apps with lots of concurrent users produce churn and subsequent db effort to fight that churn. Connection surges can overload MySQL servers resulting in downtime. Application-level pooling isn't viable; app servers don't talk to each other or coordinate when each will connect to a server.

Scaling Writes

Partitioning a database, or sharding, spreads the load and reduces the amount of data in each table. This common design principal for scaling writes in MySQL environments carries its own challenges. Sharding must be added in the app server introducing new complexity, time and required maintenance intervals. If the app requires calls between two or more shards, the resulting bottleneck defeats the purpose of the shard.

Data Centers

Multi-data center deployments present another layer of complexity during a scaling operation. Percona XtraDB Cluster is based on synchronous replication, so scaling the nodes beyond one datacenter adversely affects app performance. Asynchronous replication using GTID (Global Transaction ID) is an option, but not one that offers clear, manageable tools for zero downtime across multiple environments.

Achieving Zero Downtime for Percona XtraDB Deployments

ScaleArc's database load balancing software meets and surpasses all these challenges. The additional, transparent layer living between app servers and database servers makes deployments seamless with no additional time, maintenance or manual intervention. The benefits overshadow anything else on the market today:

- Automatic read/write split
- No app, driver or database changes
- Built-in protection of load balancing logic for all servers
- Virtual IP for app servers assures high availability
- Built-in auto failover
- Traffic management proxies
- Real-time visibility of read and write traffic
- Proactive analytics pinpoint server-intensive queries and increase performance

To learn more about using ScaleArc for a seamless Percona XtraDB deployment, [click here to read the full white paper.](#) »