Choosing a MySQL High Availability Solution

Marcos Albe, Percona Inc.
Live Webinar
June 2017

Agenda

- **What is availability**
- Components to build an HA solution
- HA options in the MySQL ecosystem
- Failover/Routing tools
- Percona’s picks

What is availability

- Uninterrupted delivery of a service (a.k.a. Uptime)
- With reasonable response times (SLAs)
- Guaranteeing consistency (the C in ACID)

Uptime

Percentil	Max downtime
99%	3.65 days
99.5%	1.83 days
99.9%	8.76 hours
99.99%	52.56 minutes
99.999%	5.25 minutes
99.9999%	31.5 seconds

Estimated levels of availability

Method	Level of Availability
Simple replication	98-99.9%
Master-Master/MMM	99%
SAN	99.5-99.9%
DRBD, MHA, Tungsten Replicator	99.9%
NDBCluster, Galera Cluster	99.999%

Agenda

- What is availability
- **Components to build an HA solution**
- HA options in the MySQL ecosystem
- Failover/Routing tools
- Percona’s picks

Components for HA

- Redundancy (no SPoF)
- Durability (recovery/identity)
- Clustering (monitoring/failover)
- Performance (latency)

HA is Redundancy

- RAID: disk crashes? Another works
- Clustering: server crashes? Another works
- Power: fuse blows? Redundant power supplies
- Network: Switch/NIC crashes? 2nd network route
- Geographical: Datacenter offline/destroyed? Computation to another DC

Durability

- Data stored on disks
  - Is it really written to the disk?
  - being durable means calling fsync() on each commit
- Is it written in a transactional way to guarantee atomicity, crash safety, integrity?

Clustering

- Load balancers and Proxies
- Monitor health of replication components
- Direct traffic to the appropriate node based on status or other rules

Performance

- HA always implies some performance overhead
- To cope with overhead we need to have a resonable base performance
  - Good queries/schema design
  - Good configuration
  - Good hardware
  - Good connectivity

HA for databases

- HA is harder for databases
- Hardware resources and data need to be redundant
- Constantly changing data
- Operations can continue uninterrupted
  - Not by restoring a new/backup server
- Uninterrupted: measured in percentiles

Agenda

- What is availability
- Components to build an HA solution
- **HA options in the MySQL ecosystem**
- Failover/Routing tools
- Percona’s picks

Redundancy through XA

- Client writes to 2 independent but identical databases
  - HA-JDBC (http://ha-jdbc.github.io/)
  - Coordinated two-phase commit
- No replication anywhere
- Many pitfalls and known bugs

Redundancy through Shared Storage

- Requires specialized hardware (SAN)
- Complex to operate (specially for DBAs)
- One set of data is your single point of failure
- Cold standby
- Failover 1-30 minutes
- Not scale-out
- Active/Active solutions: Oracle RAC, ScaleDB

Redundancy through disk replication

- DRBD
  - Linux administration vs. DBA skills
- Synchronous
- Failover: 0.5 - 30 minutes
- Second set of data inaccessible for use
- Not scale-out
- Performance hit: worst case is ~60%

Redundancy through MySQL replication

- MySQL replication
- Galera Cluster / InnoDB Cluster
- MySQL Cluster (NDBCLUSTER)
- Tungsten Replicator
- Computing/storage requirements are multiplied
- Huge potential for scaling out

MySQL replication

- Statement based
- Row based became available in 5.1, and the default in 5.7
- Asynchronous
- GTID/UUID in 5.6
- MTS per schema in 5.6
- MTS intra schema in 5.7

Semi-sync replication

- Slave acknowledges transaction event only after written to relay log
- Timeout occurs? master reverts to async replication; resumes when slaves catch up
- It scales, Facebook runs semi-sync: http://yoshinorimatsunobu.blogspot.com/2014/04/semi-synchronous- replication-at-facebook.html
- Affected by latency

Galera Cluster

- Inside MySQL, a replication plugin (wsrep)
- Replaces MySQL replication (can work alongside it too)
- Virtually Synchronous
- True multi-master, active-active solution
- No slave lag or integrity issues
- Automatic node provisioning
- WAN performance: 100-300ms/commit, works in parallel

Galera Cluster (2)

- Minimum 3 nodes are recommended
- It’s elastic
  - Automatic node provisioning
  - Self healing / Quorum
- Slowest node drives performance
- Scales reads, NOT writes
- Has some limitations (InnoDB only, transaction size, transportable tablespaces)

Group Replication

- Very much the same than Galera
- Built-in to MySQL; All Platforms
- A bit too-early for production (https://goo.gl/oKHm27)
- https://goo.gl/AbTRco for Vadim's (Percona's CTO) comparison of Galera and Group Replication
- https://goo.gl/emL9zX for Frederic Descamps (Oracle) comparing them

Tungsten

- MySQL writes binlog,Tungsten reads it and uses its own replication protocol
- Replaces MySQL Replication layer
- Per-schema multi-threaded slave
- Heterogeneous replication: MySQL <-> MongoDB <-> Postgres <-> Oracle
- Multi-master replication
- Multiple masters to single slave (multi-source replication)
- Other complex topologies

Agenda

- What is availability
- Components to build an HA solution
- HA options in the MySQL ecosystem
- **Failover/Routing tools**
- Percona’s picks

All in... sometimes it can get out of sync

- Changed information on slave directly
- Statement based replication
- Master in MyISAM, slave in InnoDB (deadlocks)
- --replication-ignore-db with fully qualified queries
- Binlog corruption on master
- Lack of primary keys
- read_buffer_size larger than max_allowed_packet
- PURGE BINARY LOGS issued and not enough files to update slave
- Bugs

Handling failure

- How to detect failure? Polling, monitoring, alerts, error returned to client side
- What to do? Direct requests to the spare nodes (or DCs)
- How to preserve integrity?
  - Async: Must ensure there is only one master at all times.
  - DRBD/SAN cold-standby: Must unmount disks and stop mysqld; then the opposite on promoted node.
- In all cases must ensure that 2 disconnected replicas or clusters cannot both commit independently. (split brain)

Tooling to handle failure

- **Orchestrator**
- **MySQL MHA**
- Tungsten Replicator
- 5.6: mysqlfailover, mysqlrpladmin
- Percona Replication Manager
- SeveralNines ClusterControl
- MariaDB Replication Manager
- MySQL MMM

Orchestrator

- Topology introspection, keeps state, continuous polling
- Smart picking of node to be promoted
- No manual promotions
- Flapping protection
- No checking for transactions on master
- Modify your topology — move slaves around
- Nice GUI, JSON API, CLI
- https://goo.gl/ELWM7S and https://goo.gl/Uy9I3c for more in-depth reviews

MHA

- Similar to Orchestrator
- Automated and manual failover options
- Choose new master by comparing slave binlog positions.
  - Fetch missing from master if possible.
- Can be used in conjunction with other solutions (example: https://goo.gl/Wds1es)
- No longer developed; Still maintained

Tooling for multi-master clusters

- Synchronous multi-master clusters like Galera require load balancers
- HAProxy
- MySQL Router
- MaxScale
- **ProxySQL**

ProxySQL

- Layer 7 router; Knows MySQL protocol
- Connection multiplexing
- Query caching, routing, rewriting and mirroring
- Zero downtime configuration changes
- SQL Firewall (preventing injections)
- Stats about workload

Agenda

- What is availability
- Components to build an HA solution
- HA options in the MySQL ecosystem
- Failover/Routing tools
- **Percona's picks**

Conclusions (1)

- Simpler is better
- MySQL replication > DRBD > SAN
- Async replication = no latency; good for WAN
- Loss-less Semi-sync replication = very little risk of data loss; latency bound
- Sync multi-master = no failover required; latency bound
- Multi-threaded slaves help in disk/latency bound workloads
- Galera provides these two with good performance & stability

Conclusions (2)

- MySQL replication is amazing if you know it (and monitor it) well enough
- Large sites run just fine with semi-sync + tooling for automated failover (either MHA or Orchestrator)
- Galera Cluster is well tested and great choice for (virtually) synchronous replication
- Don’t forget the need for a load balancer: ProxySQL is nift

Percona Live Europe Call for Papers & Registration are Open!

Championing Open Source Databases

- MySQL, MongoDB, Open Source Databases
- Time Series Databases, PostgreSQL, RocksDB
- Developers, Business/Case Studies, Operations
- September 25-27th, 2017
- Radisson Blu Royal Hotel, Dublin, Ireland

Submit Your Proposal by July 17th!

www.percona.com/live/e17

marcos.albe@percona.com

We're Hiring! www.percona.com/about-us/careers/