OKC MySQL Users Group

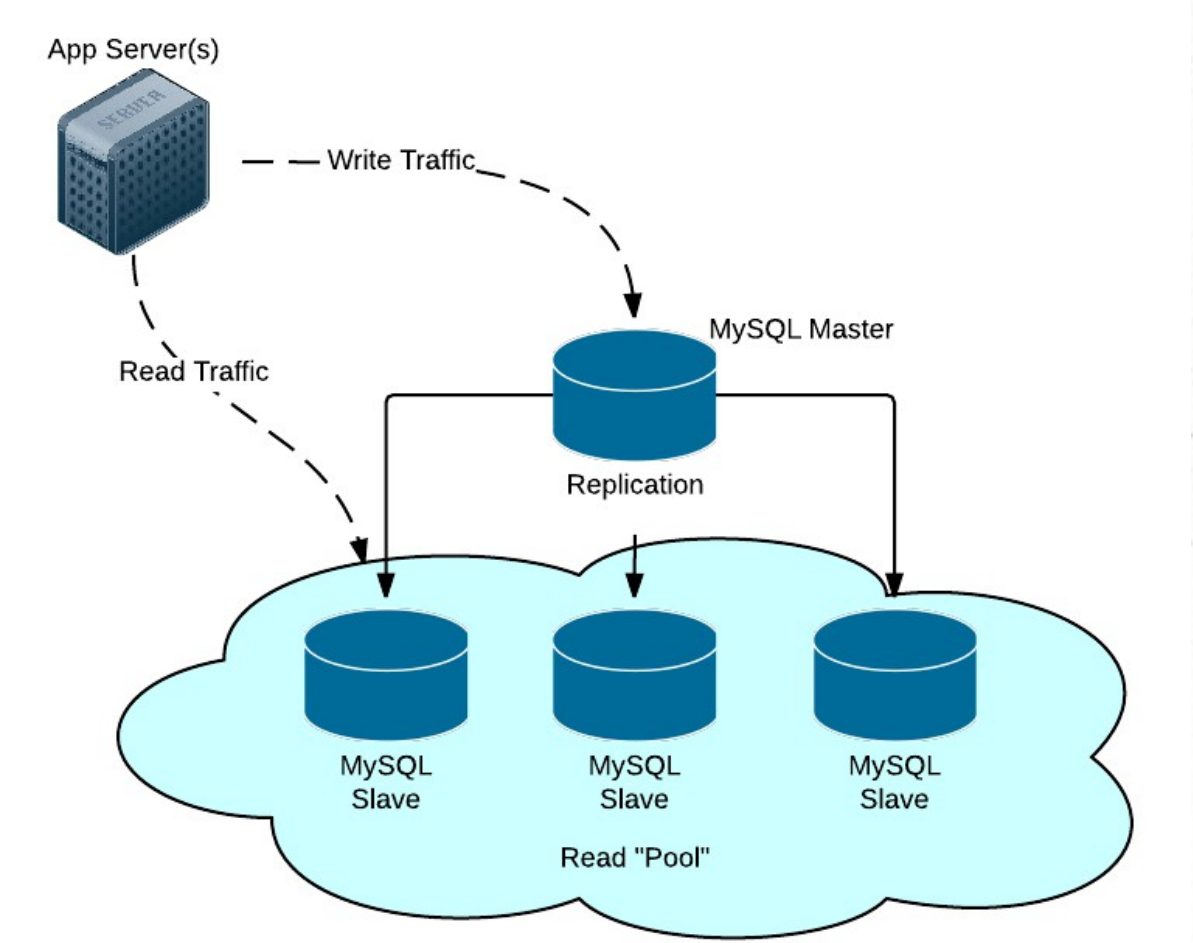
OKC MySQL

- Discuss topics about MySQL and related open source RDBMS
- Discuss complementary topics (big data, NoSQL, etc)
- Help to grow the local ecosystem through meetups and events

Sponsored By...

MySQL Group Replication

Standard Asynchronous Replication



How it works (high level)?

- Each server has a unique server-id
- There is a feature called **the binlog**
    - After each write, the statement is written to the binlog with the server-id
    - Note that what is written is configurable,but we'll touch on that later
- Slave servers connect to the master and it simply “dumps” the binlog contents to the slaves
- Slaves replay the events from servers with different server-ids to mirror the master
- NO KNOWLEDGE OF SLAVE STATUS ON THE MASTER

What is replication good for?

- Scaling reads (very common use case)
- Dedicated backup machines (take backups from slave server)
- Datacenter / Geographic redundancy
- Disaster recovery

Concerns with replication

- Slaves out of sync with masters
  - Non-deterministic statements
  - Users accidentally writing to slave (i.e. human error)
- Replication delay
  - There is inherent delay as the process is asynchronous
  - This can be compounded
- Underpowered slaves
  - Masters are multi-threaded, slaves are single threaded
  - This means slaves often need to be **more** powerful than masters

Replication Alternatives

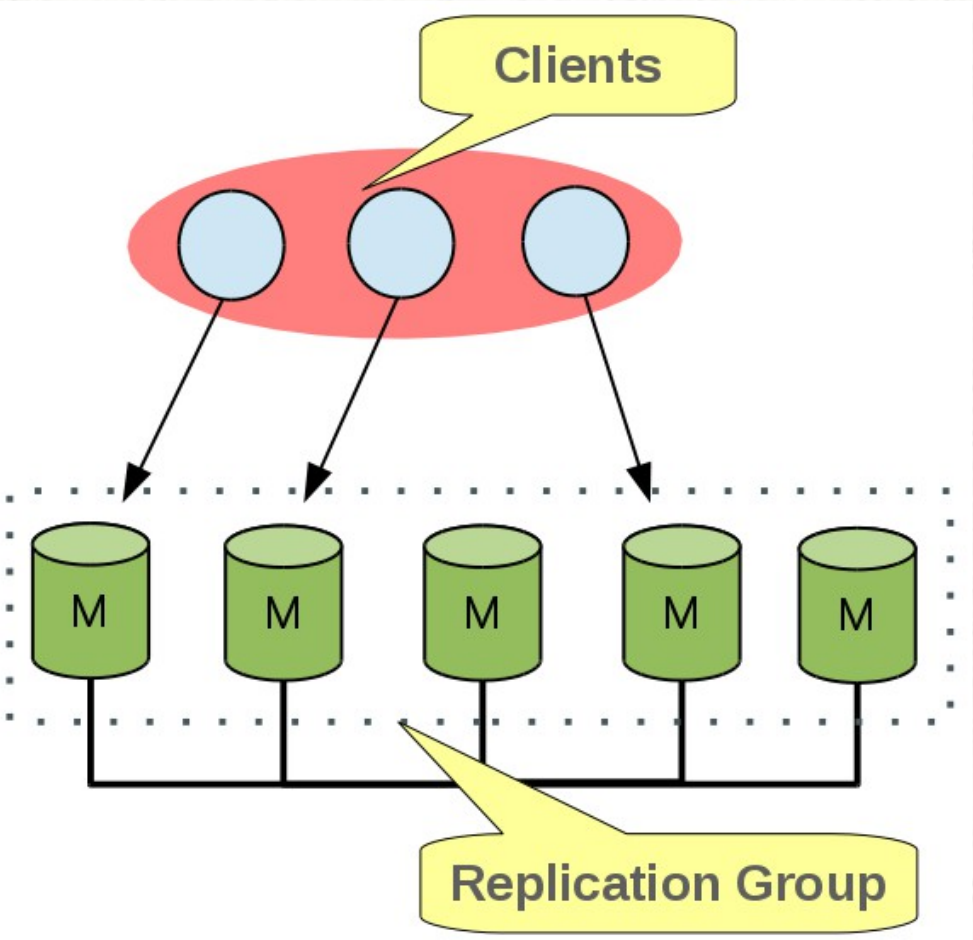
- Semi-sync replication
  - Master blocks until at least one slave acknowledges the transaction has been flushed to disk
- Disk based replication
  - DRDB – physical disk block replication
  - Slave is clone of master based on disk blocks
  - Replication is not handled via MySQL

Enter synchronous replication...

MySQL Group Replication

- GA in 5.7.17 (December, 2016 release)
- Plugin bundled with the release
- Native protocol written from the ground up
- Allows for synchronous writes and multi-writer

How it looks?



How it works?

- Write is issued to a single node
- InnoDB handles it like a normal local transaction first (looks for conflicts, etc)
- Assuming local InnoDB can commit, then it does certification
    - Ensures no other conflicts on majority of other nodes
    - Queues transaction and blocks on other nodes
- Once transaction is certified, local commit happens
- Other nodes process transactions in the queue

Some Concerns

- Increased write latency for certification
  - Slows down to lowest common denominator
  - Can work on WAN, but better for LAN
- Large transactions lead to more conflicts (especially in multi-writing)
  - Not great with synchronous replication
  - Better suited for small, OLTP workloads
- It isn't really synchronous?!
  - Other nodes have a “queue” of pending transactions
  - Can still serve stale reads

Percona XtraDB Cluster (PXC)

- Built using Galera plugin for clustering
- Very similar concept to Group Replication
- Auto-seeding of nodes (SST, IST)
- Battle tested in production for 5+ years (since 5.5 release)
- Includes Percona Server enhancements (configuration options, performance, etc)

Same Concerns

- Write latency penalty
- Large transactions are problematic
- ...

Demo Time

- Look at some MySQL Group / PXC setups

Questions?

(I know you have them, that's why you came today...
so don't be shy)

Thanks for coming!

Website: http://okcmysql.org

Twitter: @okcmysql

Email: mike@okcmysql.org