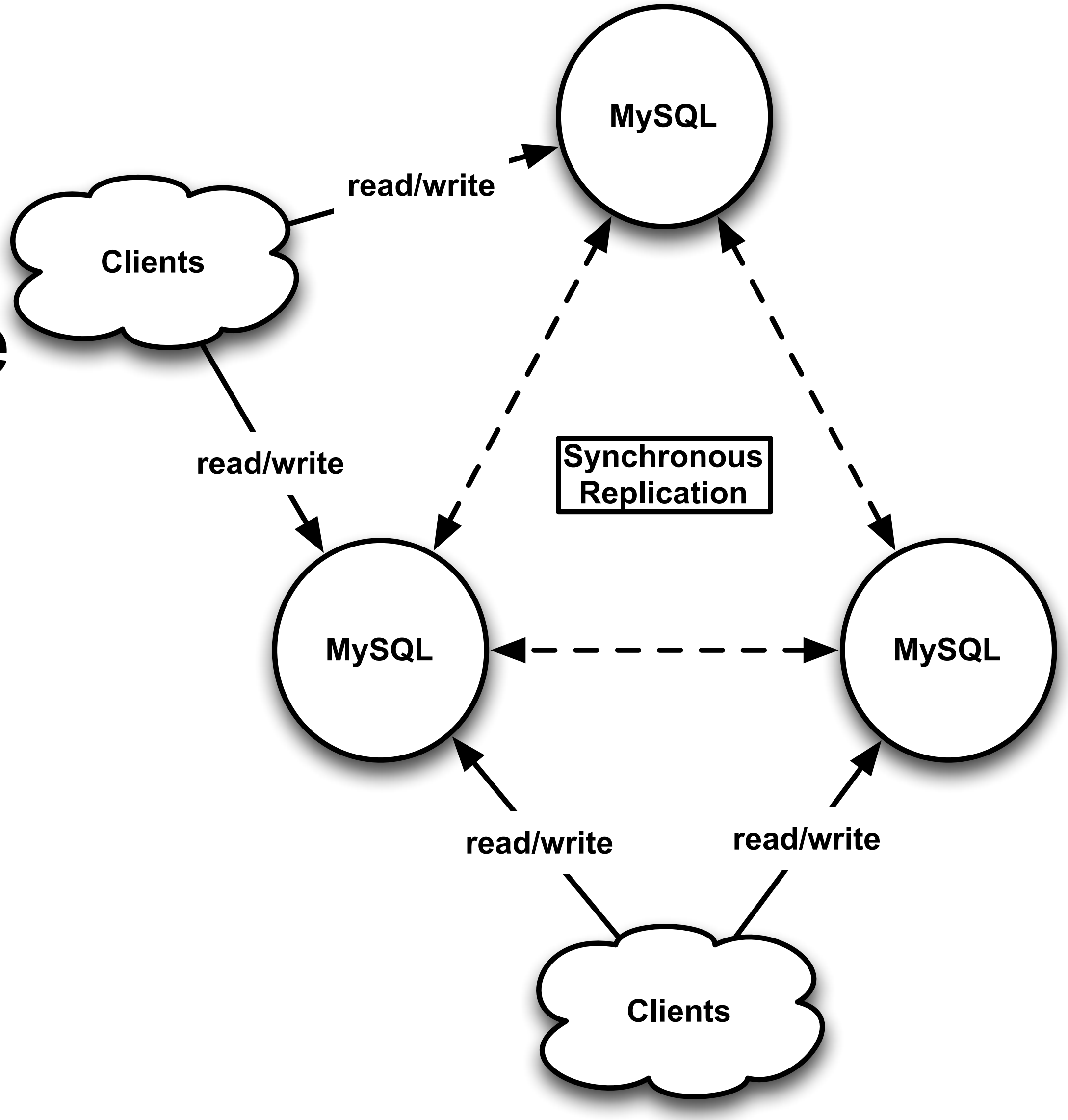
PERCONA

Migrating to XtraDB Cluster

Jay Janssen
Senior MySQL Consultant
June 6th, 2012

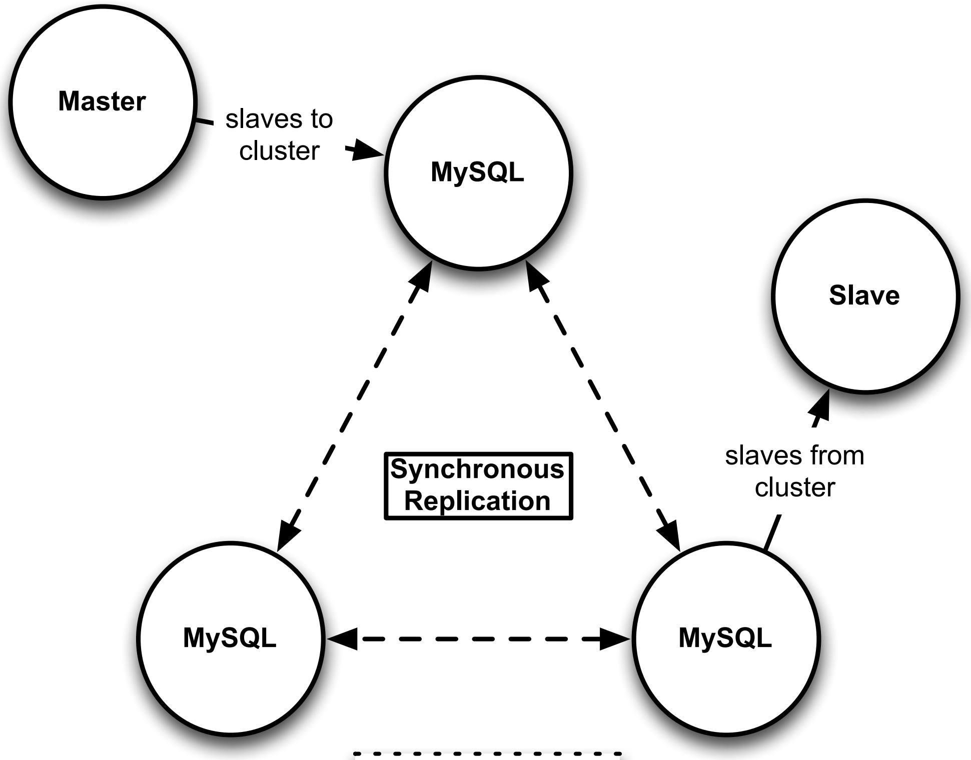
Overview of Xtradb Cluster

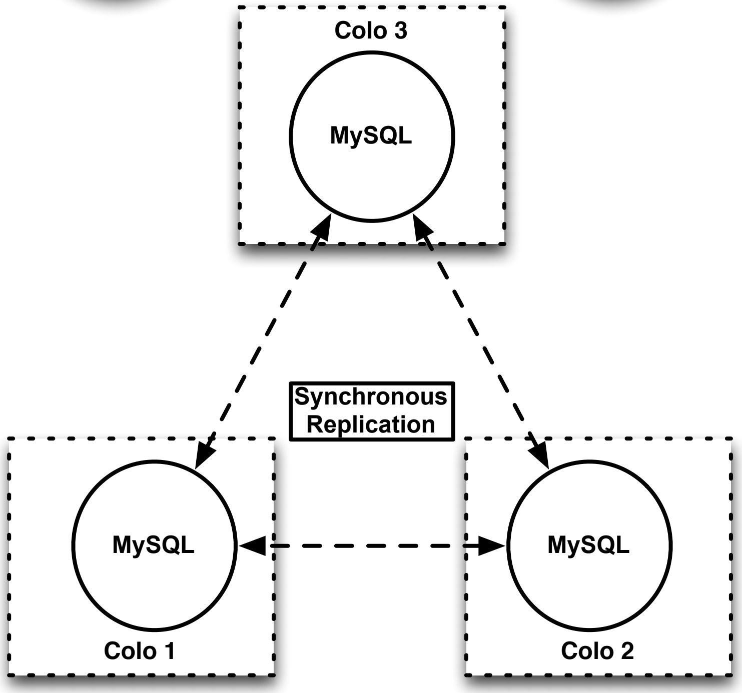
- Percona Server 5.5 + Galera Codership sync repl addon
- "Cluster of MySQL nodes"
  - Have all the data, all the time
  - Readable and writeable
- Established cluster:
  - Synchronizes new nodes
  - Handles node failures
  - Handles Node resync
  - Split brain protection (quorum)



XtraDB Cluster FAQ

- Standard MySQL replication
  - into or out of the cluster
- Write scalable to a point
  - all writes still hit all nodes
- LAN/WAN architectures
  - write latency ~1 RTT
- MyISAM experimental
  - big list of caveats
  - designed and built for Innodb





What you really want to know

- Is it production worthy?
  - Several production users of Galera
  - Looking for more early adopters to gain experience
  - The architecture is sound, code is good
  - Galera is several years old and at version 2.0
- What are the limitations of using Galera?
  - http://www.codership.com/wiki/doku.php?id=limitations

PERCONA

Configuring Xtradb Cluster

Cluster Replication Config

- Configured via wsrep_provider_options
- Can be a separate network from mysqld
- Default cluster replication port is 4567 (tcp)
- Supports multicast
- Supports SSL
- Starting node needs to know a single node's ip that is up and running

Essential Galera settings

- [mysqld_safe]
  - wsrep_urls - possible urls to existing cluster nodes
- [mysqld]
  - wsrep_provider = /usr/lib64/libgalera_smm.so
  - wsrep_cluster_name - Identify the cluster
  - wsrep_node_name - Identify this node
  - wsrep_sst_method - How to synchronize nodes
  - binlog_format = ROW
  - innodb_autoinc_lock_mode=2
  - innodb_locks_unsafe_for_binlog=1 - performance

Other Galera Settings

- [mysqld]
  - **wsrep_provider_options** - cluster comm opts
    - wsrep_provider_options="gcache.size=<gcache size>"
    - http://www.codership.com/wiki/doku.php?id=galera_parameters
  - **wsrep_node_address**=<this node IP>
  - **wsrep_slave_threads** - apply writesets in parallel
  - wsrep_cluster_address - redundant with wsrep_urls
  - wsrep_notify_cmd - run on cluster state changes
  - wsrep_on - equivalent to SQL_LOG_BIN
- http://www.codership.com/wiki/doku.php?id=mysql_options_0.8

Possible Performance Tuning

- Single node durability can be disabled (?)
  - innodb_flush_log_at_trx_commit=2|0
  - safe as long as all cluster nodes don't go offline at once
- Other possibilities
  - log-bin, sync_binlog, innodb_support_xa = OFF
  - innodb_doublewrite = OFF?

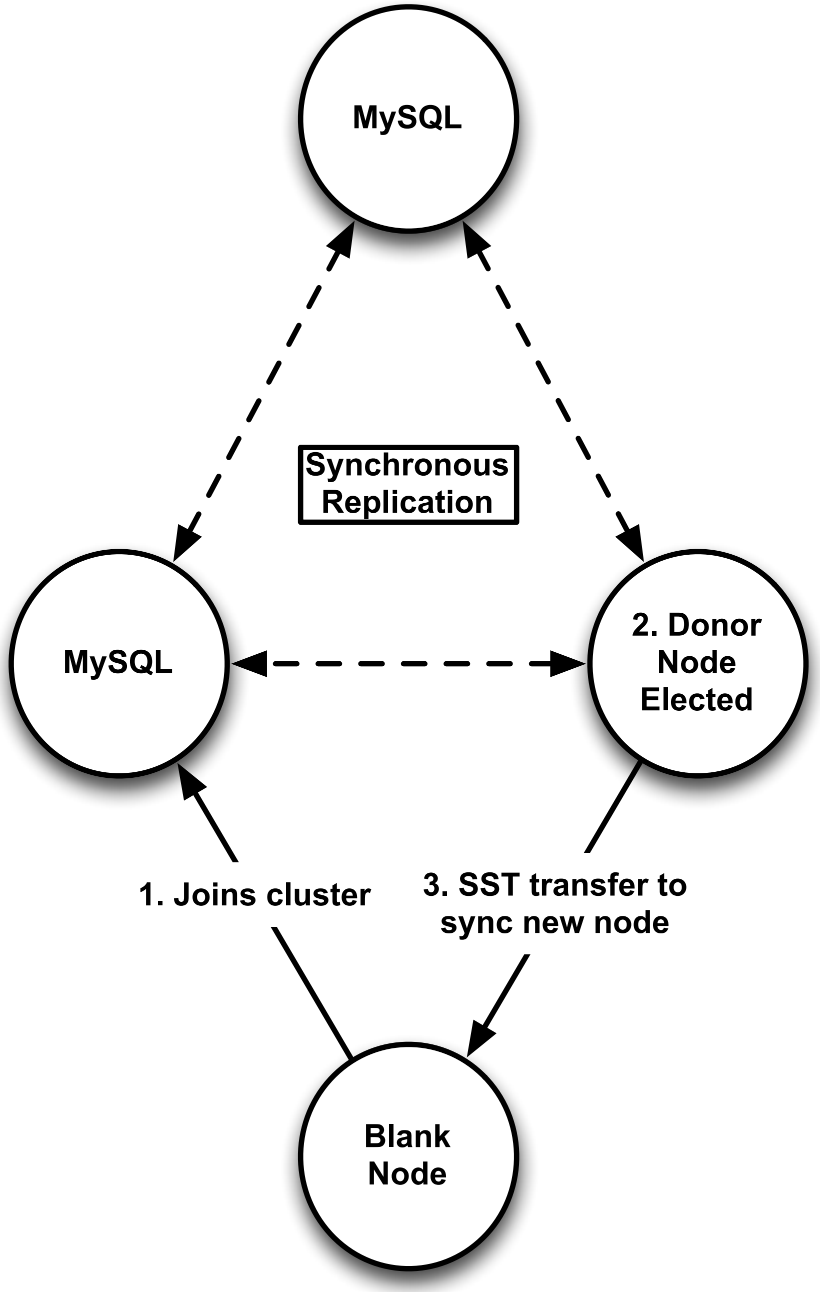
Example configuration

```
[mysqld_safe]
wsrep_urls=gcomm://192.168.70.2:4567, \
    gcomm://192.168.70.3:4567, \
    gcomm://192.168.70.4:4567, \
    gcomm://  # Only use this before the cluster is formed

[mysqld]
datadir=/var/lib/mysql
binlog_format=ROW

wsrep_cluster_name=trimethylxanthine
wsrep_node_name=percona1
wsrep_node_address=192.168.70.2
wsrep_provider=/usr/lib64/libgalera_smm.so

wsrep_sst_method=xtrabackup

wsrep_slave_threads=2

innodb_locks_unsafe_for_binlog=1
innodb_autoinc_lock_mode=2
innodb_buffer_pool_size=128M
innodb_log_file_size=64M
```

Converting Standalone MySQL to Xtradb Cluster

First a word about SST

- State Snapshot Transfer
  - full data copy to a needy node
  - methods supported:
    - rsync / rsync_wan, mysqldump, xtrabackup, skip. (pluggable)
- Donor is chosen as SST source
  - SST donation may block donor
  - Dedicated donor possible
- New cluster nodes get SST
- Node inconsistencies trigger SST
- Brief outages need not SST (IST)

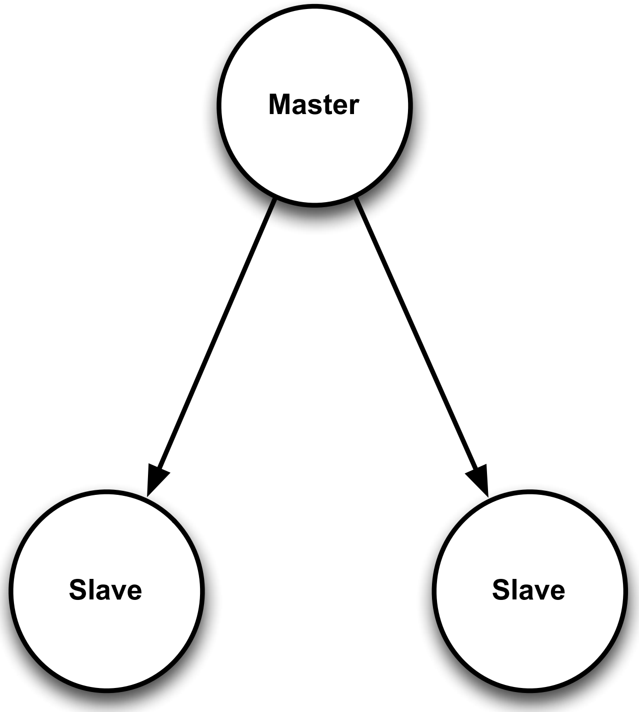


Method 1 - Single Node

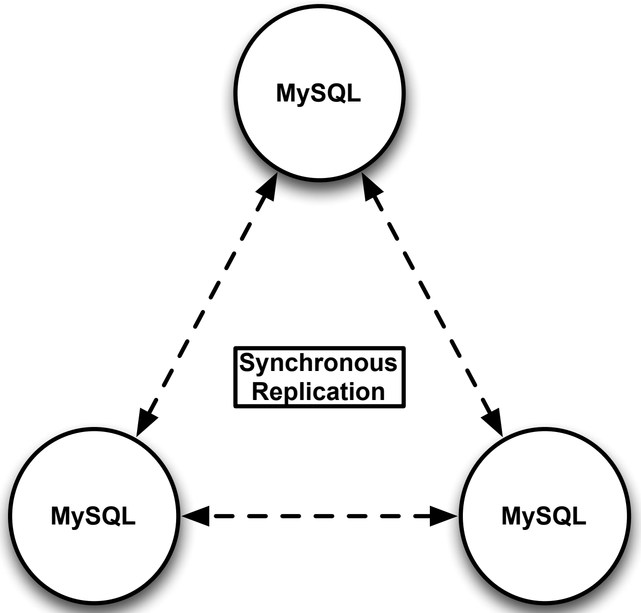
- Migrating a single server:
  - stop MySQL
  - replace the packages
  - add essential Galera settings
  - start MySQL
- A stateless, peerless node will form its own cluster
  - iff an empty cluster address is given (gcomm://)
- That node is the baseline data for the cluster
- Easiest from Percona Server 5.5

Method 2 - Blanket changeover

- All at once (with downtime):
  - Stop all writes, stop all nodes after replication is synchronized
  - skip-slave-start / RESET SLAVE
  - Start first node - initial cluster
  - Start the others with wsrep_sst_mode=skip
- The slaves will join the cluster, skipping SST
- Change wsrep_sst_mode != skip

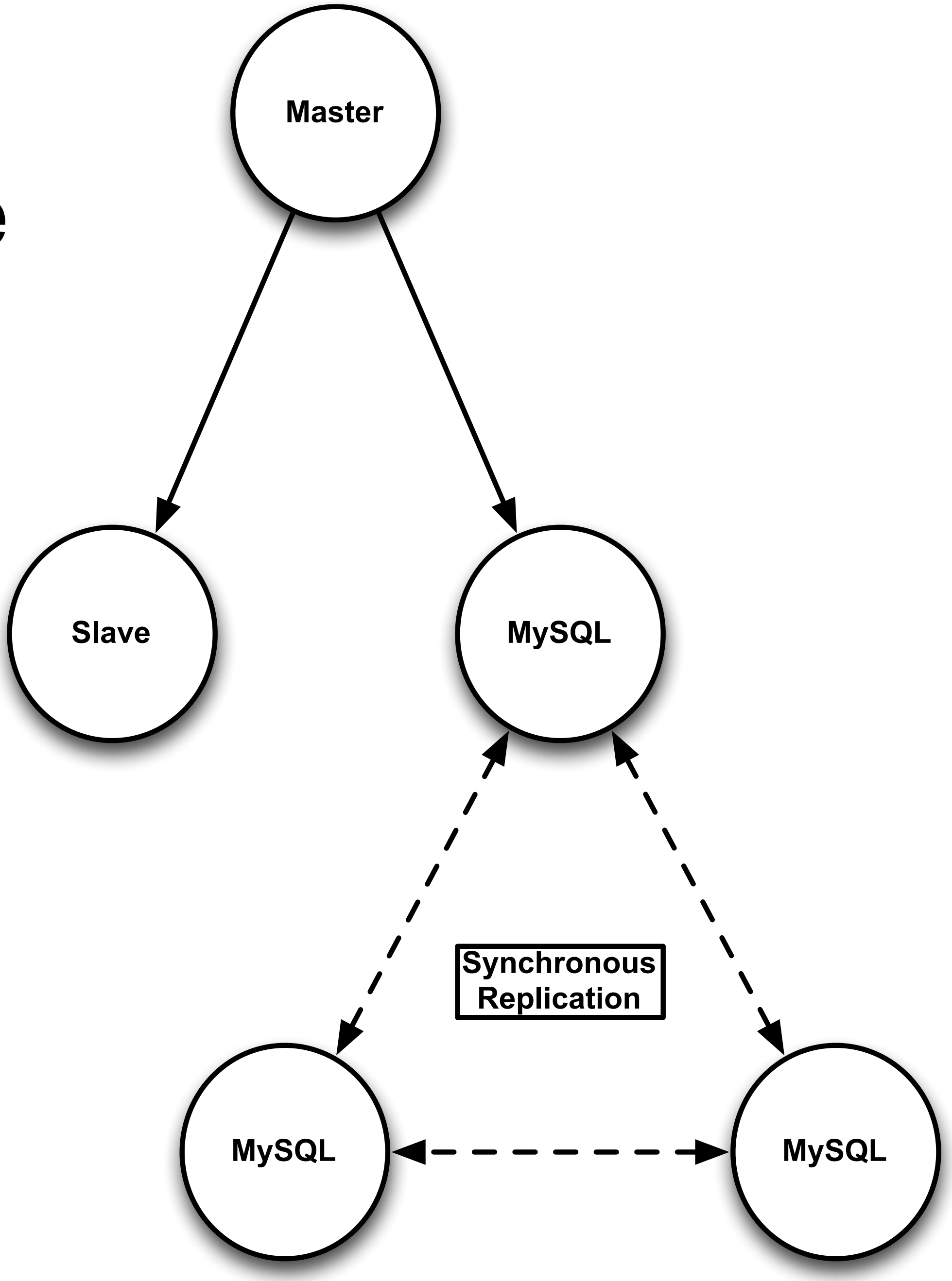


becomes



Method 3 - Slave cluster

- No downtime
  - Form new cluster from one slave
  - Node replicates from old master
    - log-slave-updates on this node
  - Test like any other slave
  - Move more slave nodes to cluster
  - Cut writes over to the cluster
  - Absorb master into cluster.
- Non-skip SST



PERCONA

Operational Considerations

Monitoring

- SHOW GLOBAL STATUS like ‘wsrep%’;
- Cluster integrity - same across all nodes
  - wsrep_cluster_conf_id - configuration version
  - wsrep_cluster_size - number of active nodes
  - wsrep_cluster_status - should be Primary
- Node Status
  - wsrep_ready - indicator that the node is healthy
  - wsrep_local_state_comment - status message
  - wsrep_flow_control_paused - replication lag
  - wsrep_local_send_q_avg - possible network bottleneck
- http://www.codership.com/wiki/doku.php?id=monitoring

Realtime Wsrep status

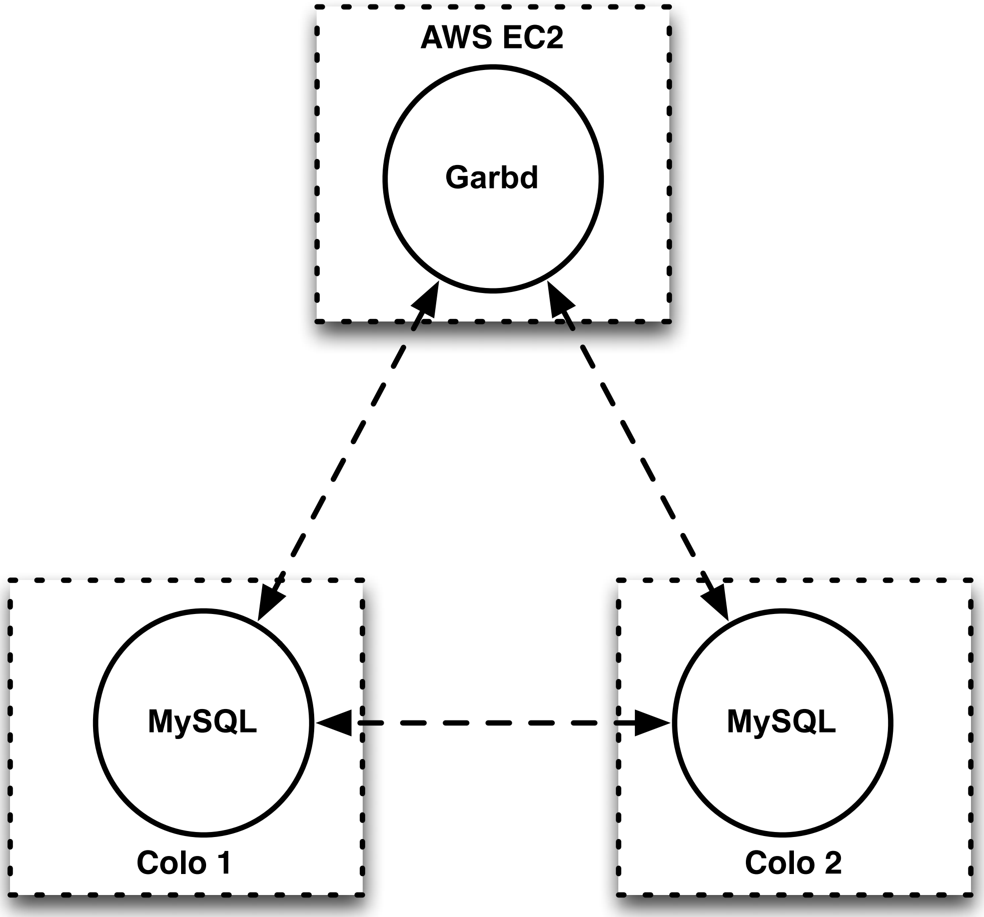
```
$ ./myq_status -t 1 -h 192.168.70.4 -u test2 -p test2 wsrep

Wsrep (Galera/Xtradb Cluster)                                   Replicated   Received
    time     state conf  rdy  ctd  cnt paus dist sent rcvq sndq wops wsize rops rsize
12:40:24     Donor   36   ON   ON    3    0  1.0    0    0    0    0     0  1.0 191.0
12:40:25     Donor   36   ON   ON    3    0  1.0    0    0    0    0     0  1.0 191.0
12:40:26     Donor   36   ON   ON    3    0  1.0    0    0    0    0     0  2.0 382.0
12:40:28     Donor   36   ON   ON    3    0  1.0    0    0    0    0     0  0.5 95.50
12:40:29     Donor   36   ON   ON    3    0  1.0    0    1    0    0     0    0     0
12:40:30     Donor   36   ON   ON    3    0  1.0    0    2    0    0     0    0     0
12:40:31     Donor   36   ON   ON    3    0  1.0    0    3    0    0     0    0     0
12:40:32     Donor   36   ON   ON    3    0  1.0    0    4    0    0     0    0     0
12:40:33     Donor   36   ON   ON    3    0  1.0    0    5    0    0     0    0     0
12:40:34     Donor   36   ON   ON    3    0  1.0    0    6    0    0     0    0     0
12:40:35     Donor   36   ON   ON    3    0  1.0    0    7    0    0     0    0     0
12:40:36     Donor   36   ON   ON    3    0  1.0    0    8    0    0     0    0     0
12:40:37     Donor   36   ON   ON    3    0  1.0    0    0    0    0     0  9.0 1.68K
12:40:38     Donor   36   ON   ON    3    0  1.0    0    0    0    0     0  1.0 191.0
12:40:39    Synced   36   ON   ON    3    0  1.0    0    0    0    0     0  3.0 207.0
12:40:40    Synced   36   ON   ON    3    0  1.0    0    0    0    0     0  1.0 191.0
12:40:41    Synced   36   ON   ON    3    0  1.0    0    0    0    0     0  1.0 191.0
12:40:42    Synced   36   ON   ON    3    0  1.0    0    0    0    0     0  1.0 191.0
12:40:43    Synced   36   ON   ON    3    0  1.0    0    0    0    0     0  1.0 191.0
12:40:44    Synced   36   ON   ON    3    0  1.0    0    0    0    0     0  1.0 191.0

https://github.com/jayjanssen/myq_gadgets
```

Maintenance

- Rolling package updates
- Schema changes
  - potential for blocking the whole cluster
  - Galera supports a rolling schema upgrade feature
    - http://www.codership.com/wiki/doku.php?id=rolling_schema_upgrade
    - Isolates DDL to individual cluster nodes
    - Won’t work if replication events become incompatible
  - pt-online-schema-change

Architecture

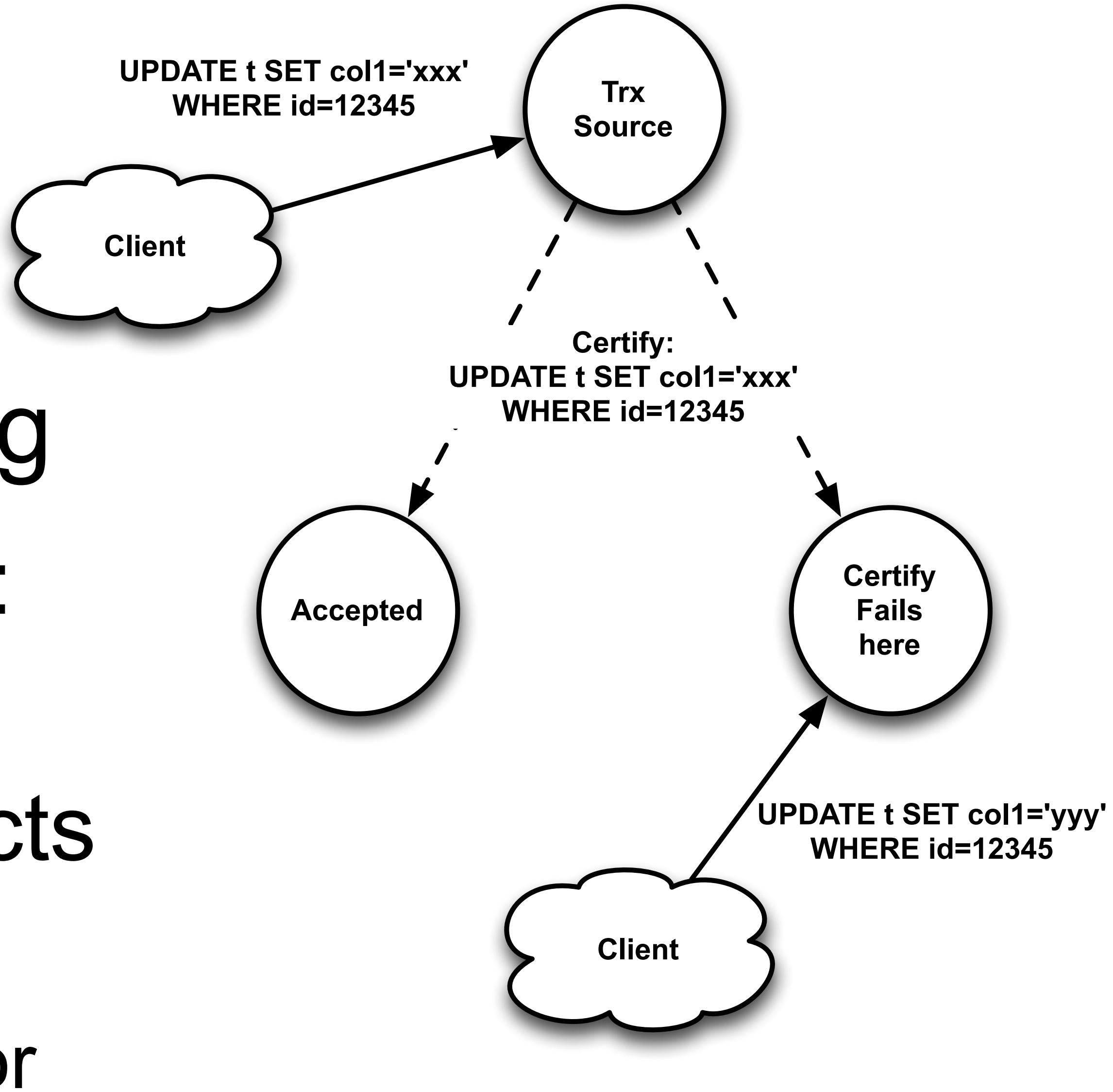
- How many nodes should I have?
  - >= 3 nodes for quorum purposes
    - 50% is not a quorum
  - garbd - Galera Arbitrator Daemon
    - Contributes as a voting node for quorum
    - Does not store data, but does replicate
- What gear should I get?
  - Writes as fast as your slowest node
  - Standard MySQL + Innodb choices
  - garbd could be on a cloud server



Application / Cluster Interactions

How Synchronous Writes Work

- Source node - pessimistic locking
  - Innodb transaction locking
- Cluster repl - optimistic locking
  - Before source returns commit:
    - certify trx on all other nodes
  - Nodes reject on locking conflicts
    - via locally running transactions
    - client gets rollback deadlock error
  - Commit succeeds if no conflicts on **any** node



Why does the Application care?

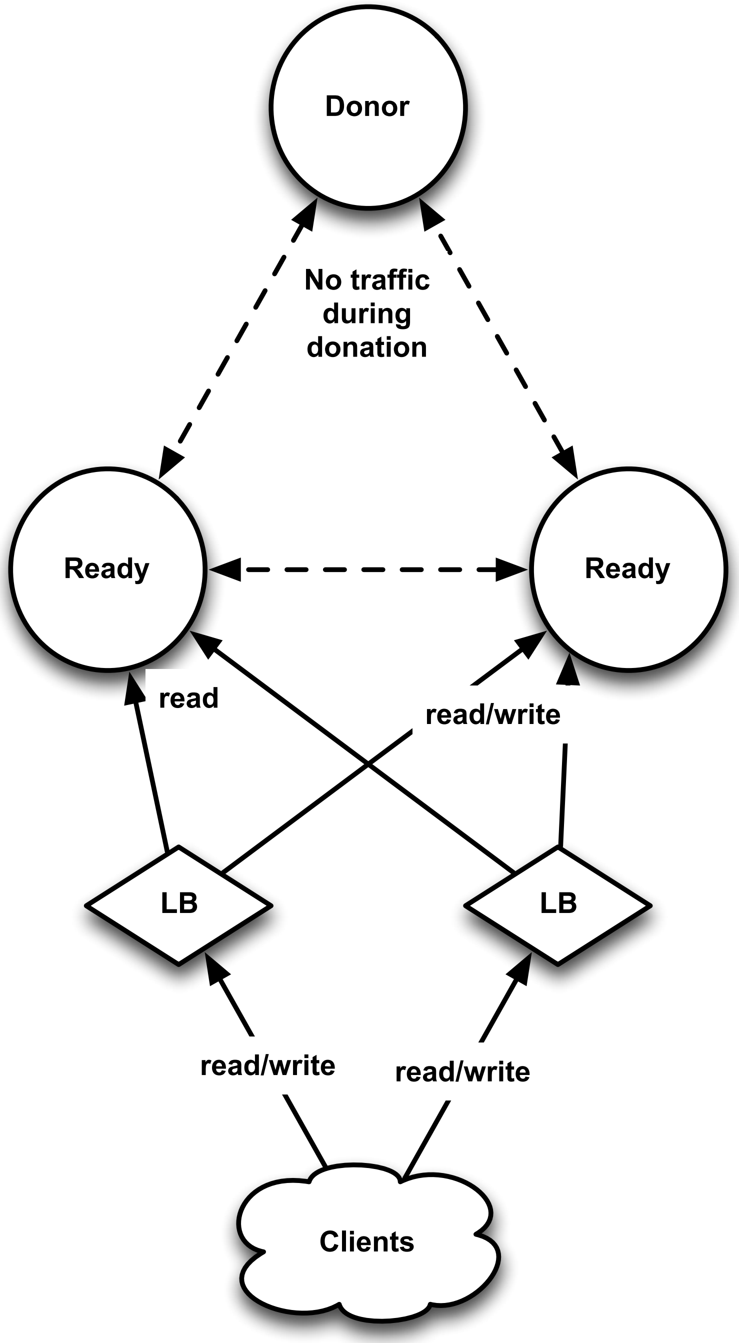
- Workload dependent!
- Write to all nodes simultaneously and evenly:
  - Increase of deadlock errors on data hot spots
- Can be avoided by
  - Writing to only one node at a time
    - all pessimistic locking happens on one node
  - Data subsets written only on a single node
    - e.g., different databases, tables, rows, etc.
    - different nodes can handle writes for different datasets
    - pessimistic locking for that subset only on one node

Application to Cluster Connects

- For writes:
  - Best practice: (any) single node
- For Reads:
  - All nodes load-balanced
    - Can be hashed to hit hot caches
    - Geo-affinity for WAN setups
  - Never worry about replication delay again!
- Be sure to monitor that nodes are functioning members of the cluster!

Load balancing and Node status

- Health check:
  - TCP 3306
  - SHOW GLOBAL STATUS
    - wsrep_ready = ON
    - wsrep_local_state_comment !~ m/ Donor/?
- Maintain a separate rotations:
  - Reads
    - RR or Least Connected all available
  - Writes
    - Single node with backups on failure



Load Balancing Technologies

- glbd - Galera Load Balancer
  - similar to Pen, can utilize multiple cores
  - No advanced health checking (tcp-only)
  - http://www.codership.com/products/galera-load-balancer
- HAProxy
  - httpchk to monitor node status
  - http://www.percona.com/doc/percona-xtradb-cluster/haproxy.html

HAProxy Sample config

```
listen cluster-writes 0.0.0.0:4306
  mode tcp
  balance leastconn
  option  httpchk

  server percona1 192.168.70.2:3306 check port 9200
  server percona2 192.168.70.3:3306 check port 9200 backup
  server percona3 192.168.70.4:3306 check port 9200 backup

listen cluster-reads 0.0.0.0:5306
  mode tcp
  balance leastconn
  option  httpchk

  server percona1 192.168.70.2:3306 check port 9200
  server percona2 192.168.70.3:3306 check port 9200
  server percona3 192.168.70.4:3306 check port 9200
```

Resources

- XtraDB Cluster homepage and documentation:
  - http://www.percona.com/software/percona-xtradb-cluster/
- Galera Documentation:
  - http://www.codership.com/wiki/doku.php
- Virtualbox 3 node test cluster:
  - https://github.com/jayjanssen/percona-cluster
  - http://www.mysqlperformanceblog.com/2012/04/12/testing-percona-xtradb-cluster-with-vagrant/
- http://www.mysqlperformanceblog.com/2012/01/12/create-3-nodes-xtradb-cluster-in-3-minutes/

PERCONA

Jay Janssen
@jayjanssen

Join us at Percona Live NYC - Oct 1-2 2012
http://www.percona.com/live/nyc-2012/