# MySQL and Openstack


Peter Boros

2015-04-16

# Where is MySQL used



- Used on the backend and as a guest
- On backend
  - Typically galera, a typical deployment have 3 “head nodes”
  - Master-slave (PRM) is there as well with Suse’s distribution
- As a guest

# Guest



- Could be ran directly on VMs
  - Storage performance characteristics can be significantly different
  - “Bad neighbors” effect
- Could be ran as a service using trove

# Guest



- Could be ran directly on VMs
  - Storage performance characteristics can be significantly different
  - “Bad neighbors” effect
- Could be ran as a service using trove

# On the backend

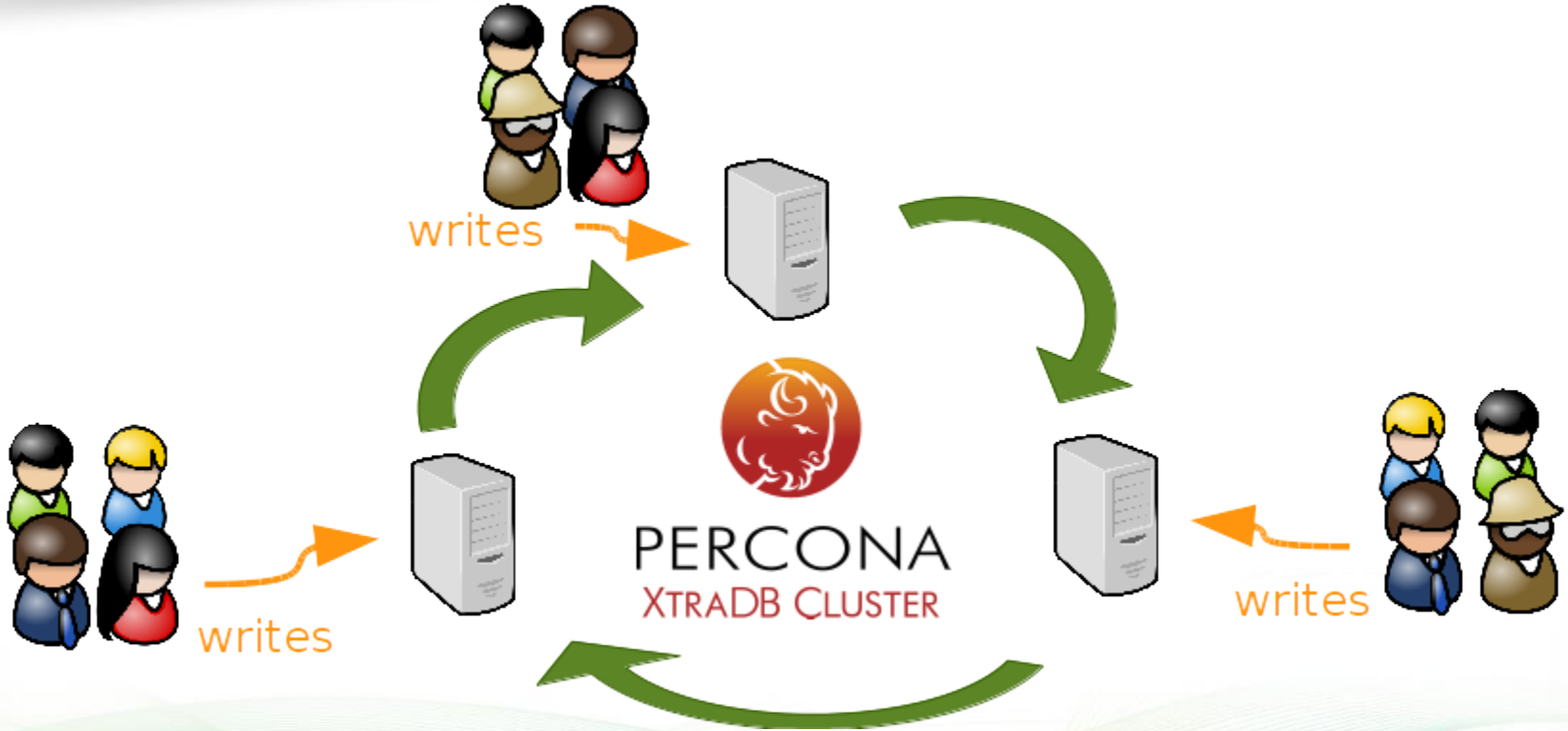
# Galera flavors



- MySQL-wsrep
  - released by codership
- MariaDB Galera Cluster
  - MariaDB + galera library
- Percona XtraDb Cluster
  - Percona Server + galera library

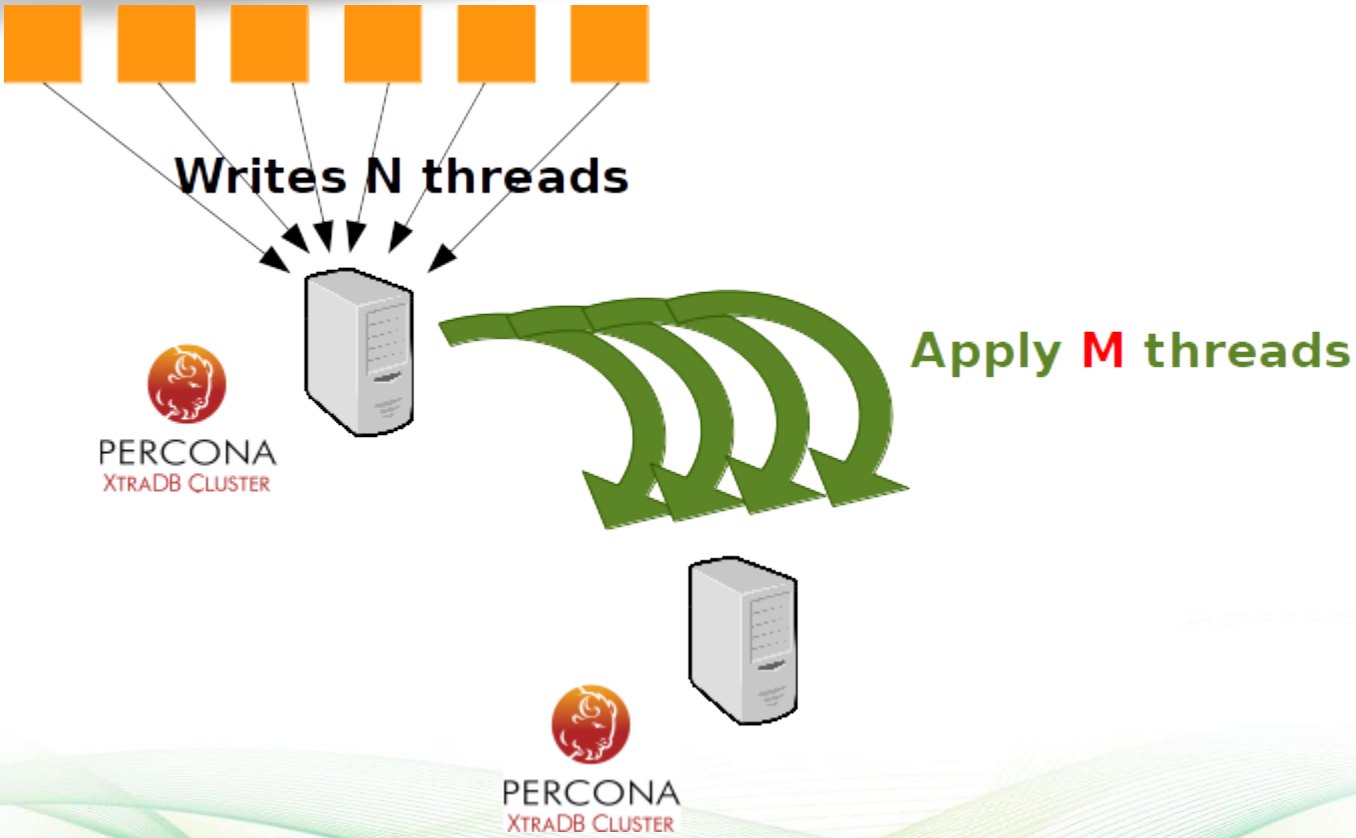
# PXC: write everywhere synchronously





# PXC: parallel replication





# Considerations



- Openstack backend components need to read their own writes
  - wsrep_sync_wait = 1
- All openstack components are fine with multi-writer setup
  - retry_on_deadlock decorator (will be moved to oslo.db)
  - Except neutron
- If you are seeing intermittent operational errors, configure connection pools

# As a guest

# Running in a VM



- IO characteristics can be different
  - Networked storage
- Solutions that are ensuring durability through network, not local durability through disk are typically better
  - Percona XtraDB Cluster
  - MySQL Cluster (NDB)
- “Bad neighbors” can result in inconsistent performance

# Trove



- DBaaS component of openstack
- Initial sponsors for rackspace and HP
- Incubated in Havana release, integrated in icehouse
- Supported databases as of Kilo (not complete)
  - MySQL, Percona Server, MariaDB
  - MongoDB
  - Cassandra
  - Couchbase
  - PostgreSQL
  - Redis

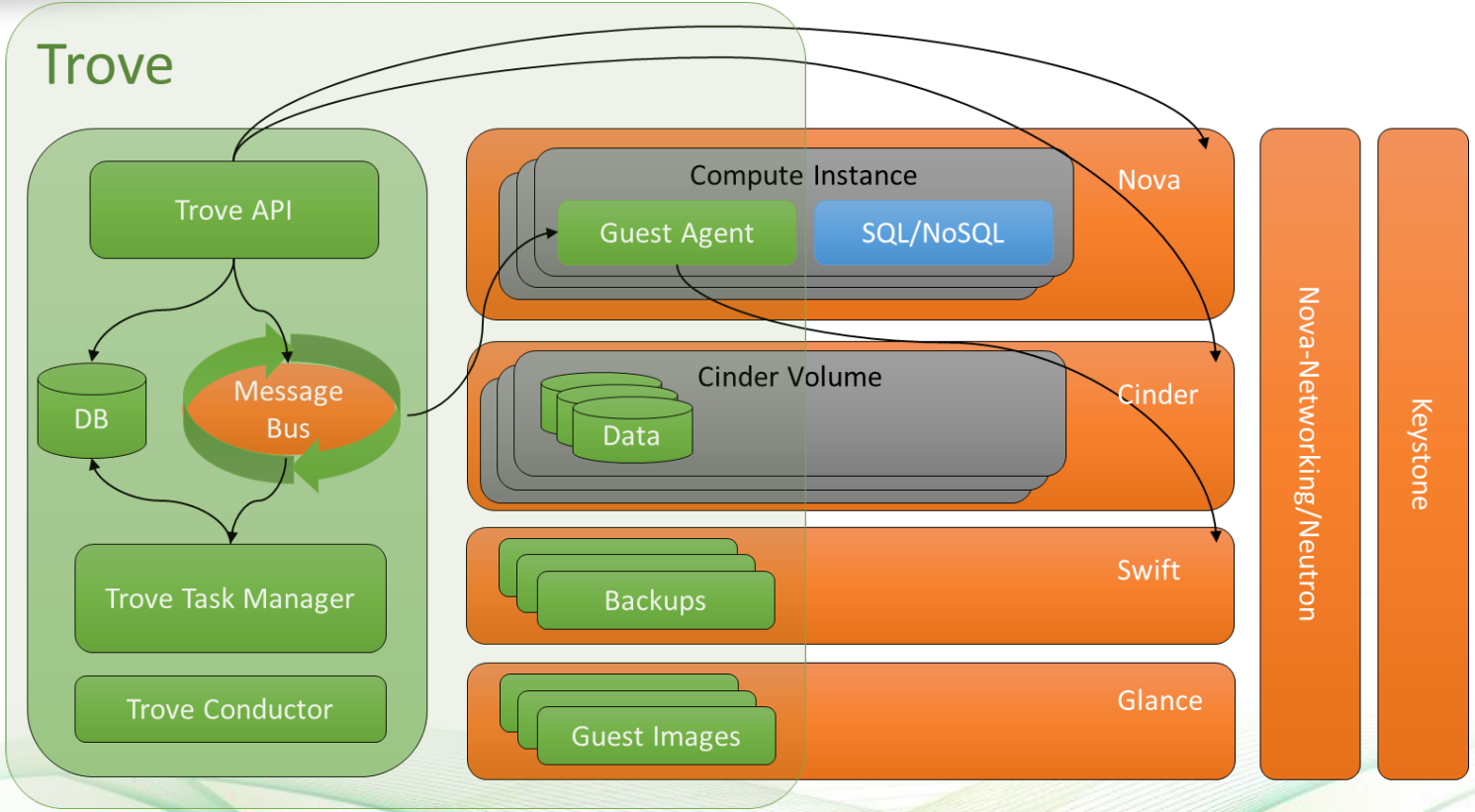
# Trove



- DBaaS component of openstack
- Initial sponsors for rackspace and HP
- Incubated in Havana release, integrated in icehouse
- Supported databases as of Kilo (not complete)
  - MySQL, Percona Server, MariaDB
  - MongoDB
  - Cassandra
  - Couchbase
  - PostgreSQL
  - Redis

# Trove architecture





# Using trove

# Creating an instance with trove



```
# trove create mysql1 101 --size 2 --
datastore percona --datastore_version 5.6
--databases database1 --users user1:
password
```

# Create a backup



```
# trove backup-create mysql1 backup1
```

# Create incremental backup



```
# trove backup-create mysql1 backup1.1 --
parent 5c1470cb-5c2c-4eb5-a47c-
839960676cdf
```

# Create an instance from backup



```
# trove create guest2 101 --size 2 --
backup 5c1470cb-5c2c-4eb5-a47c-
839960676cdf --
datastore percona
```

# Create a new slave



```
# trove create slave1 101 --size=2 --
datastore_version 5.6 --datastore percona
--replica_of mysql1
```

# Thanks