



# Percona XtraDB Cluster Installation and setup (basics)

Peter Boros  
Consultant

# Agenda

---

- Installing the first node of the cluster
- Connecting subsequent nodes to the cluster
- Installing HaProxy on the application server
- Testing with a real-world application: sysbench
- Breaking and fixing the cluster

# Agenda

---

- Installing the first node of the cluster
- Connecting subsequent nodes to the cluster
- Installing HaProxy on the application server
- Testing with a real-world application: sysbench
- Breaking and fixing the cluster

# Overview

---

- The goal of this talk to show Percona XtraDB Cluster basics with a hands-on approach.
- We will use freshly installed CentOS 6 machines. Those are vanilla installations.
- We will cover load balancing using HaProxy.

# Percona XtraDB cluster at a glance

---

- All nodes are equal
- All nodes have all the data
- Replication is (virtually) synchronous
  - Completely different from asynchronous mysql replication
- At least 3 nodes (or 2 + arbitrator)

# Packages on all database nodes

- Add Percona's yum repository

# rpm -Uhv [http://www.percona.com/downloads/percona-release/percona-release-0.0-1.x86\\_64.rpm](http://www.percona.com/downloads/percona-release/percona-release-0.0-1.x86_64.rpm)

- Install PXC packages

```
# yum -y install Percona-XtraDB-Cluster-server \
Percona-XtraDB-Cluster-client \
Percona-Server-shared-compat \
percona-xtrabackup
```

# Packages installed

## Installing:

Percona-Server-shared-compat  
replacing mysql-libs.x86\_64 5.1.61-4.el6  
Percona-XtraDB-Cluster-client  
Percona-XtraDB-Cluster-server  
percona-xtrabackup

## Installing for dependencies:

Percona-XtraDB-Cluster-galera  
libaio  
nc  
perl  
perl-Module-Pluggable  
perl-Pod-Escapes  
perl-Pod-Simple  
perl-libs  
perl-version  
rsync

# Configuring the nodes

- `wsrep_cluster_address=gcomm://`
  - Initializes a new cluster, new nodes can connect to this one
- `wsrep_cluster_address=gcomm://<IP addr>:4567`
  - Starts a new node, which will try to connect to the node specified
- `wsrep_urls`
  - Option for `[mysqld_safe]` section, not for `[mysqld]`, will set `wsrep_cluster_address` to a usable item on this list.
  - Example:  
`wsrep_urls=gcomm://node1:4567,gcomm://node2:4567,  
gcomm://node3:4567`



# Configuring the first node

```
[mysqld]
server_id=1
binlog_format=ROW
log_bin=mysql-bin
wsrep_cluster_address=gcomm:// ←
wsrep_provider=/usr/lib64/libgalera_smm
.so
datadir=/var/lib/mysql
```

We are starting a new cluster with node 'pxc1' as primary

```
wsrep_slave_threads=4
wsrep_cluster_name=pxctest
wsrep_sst_method=xtrabackup
wsrep_node_name=pxc1
```

```
log_slave_updates
```

```
innodb_locks_unsafe_for_binlog=1
innodb_autoinc_lock_mode=2
```

# Configuring subsequent nodes

```
[mysqld]  
server_id=1  
binlog_format=ROW  
log_bin=mysql-bin  
wsrep_cluster_address=gcomm://192.168.56.41  
wsrep_provider=/usr/lib64/libgalera_smm.so  
datadir=/var/lib/mysql
```

```
wsrep_slave_threads=4  
wsrep_cluster_name=pxctest  
wsrep_sst_method=xtrabackup  
wsrep_node_name=pxc2
```

```
log_slave_updates
```

```
innodb_locks_unsafe_for_binlog=1  
innodb_autoinc_lock_mode=2
```

The other nodes are joining to 'pxc1'



# Additional configuration for demo

---

- iptables disabled
  - service iptables stop
  - chkconfig --del iptables
- SELinux disabled in /etc/selinux/config



**Demo: building the cluster  
and destroying it  
and building it again**



**What we saw...**

# State transfer

- SST (Snapshot State Transfer)
  - Copies the whole data set
  - Different methods: xtrabackup, rsync, etc...
- IST (Incremental State Transfer)
  - Transactions incrementally replayed from gcache
  - You can use a manual backup created with xtrabackup using --galera-info option of innobackupex

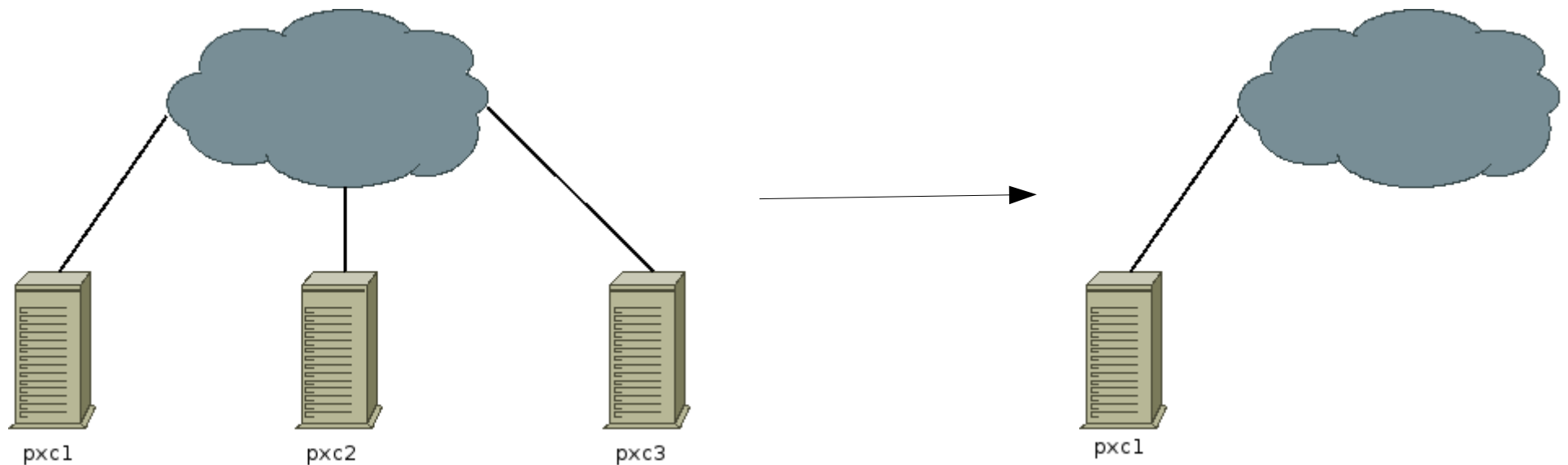
<http://www.mysqlperformanceblog.com/2012/08/02/avoiding-sst-when-adding-new-percona-xtradb-cluster-node/>

# Split Brain

---

- When only 1 node was up, the data was not usable
- Using 3 nodes guarantees that you can lose 1 node
- A node has to be able to access the majority of cluster node to serve the data

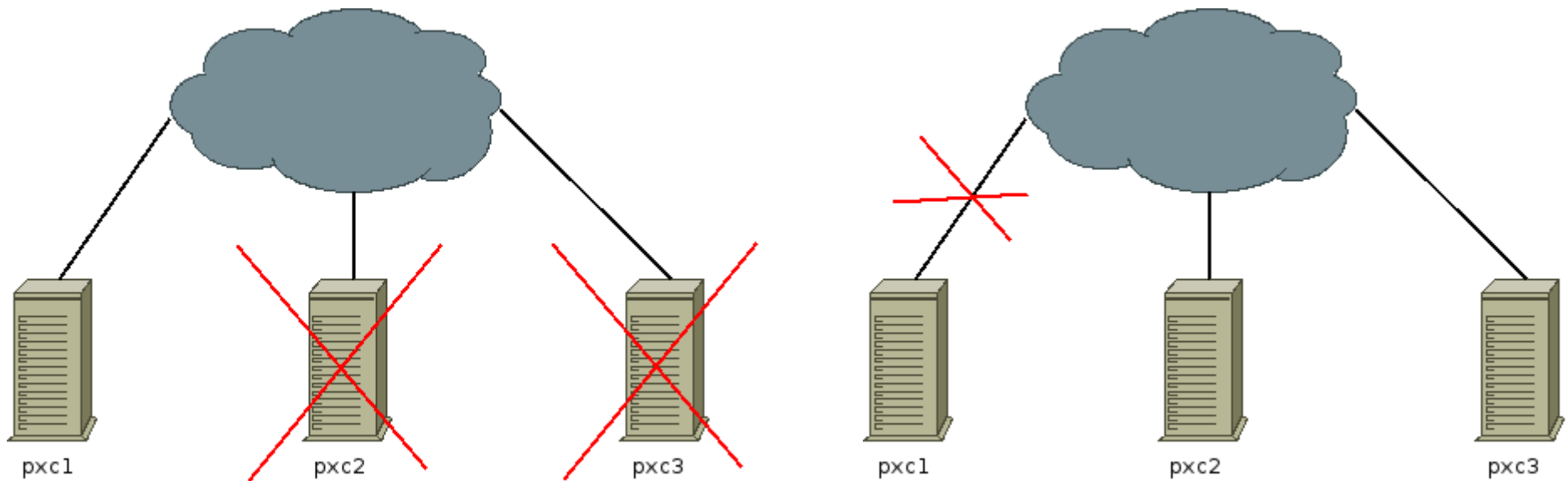
# What pxc1 saw



This does not necessarily mean pxc2 and pxc3 are dead...



# Possibilities



# Continuing operation in split brain mode

---

- The users (application servers) who can access pxc1 will write pxc1
- The users (application server) who can access pxc2 and pxc3 will write to that cluster
- This can be prevented by shutting down the node if it's not part of the group in majority
- garbd: galera arbitrator daemon, used in voting, but doesn't store data

# Load balancing

- Some application can use driver level load balancing by connecting to all nodes (JDBC)
- For the rest, external solution necessary
  - LVS
  - HaProxy
    - We will cover this
  - Any kind of load balancing software is usable, if it can balance TCP connections at least

# HaProxy configuration

---

```
backend pxc-back
mode tcp
balance leastconn
option httpchk
server pxc1 192.168.56.41:3306 check port 9200 inter 12000 rise 3 fall 3
server pxc2 192.168.56.42:3306 check port 9200 inter 12000 rise 3 fall 3
server pxc3 192.168.56.43:3306 check port 9200 inter 12000 rise 3 fall 3
```

```
backend pxc-onenode-back
mode tcp
balance leastconn
option httpchk
server pxc1 192.168.56.41:3306 check port 9200 inter 12000 rise 3 fall 3
server pxc2 192.168.56.42:3306 check port 9200 inter 12000 rise 3 fall 3 backup
server pxc3 192.168.56.43:3306 check port 9200 inter 12000 rise 3 fall 3 backup
```

# Application server node prepared for demo

---

- CentOS 6 base installation
- EPEL repo added
- HaProxy installed from EPEL repo
- Sysbench 0.5 package made by Frederic Descamps



# Load balancing demo



**Q&A**



PERCONA  
LIVE

[www.percona.com/live](http://www.percona.com/live)