Configuring a Percona MySQL Cluster

Percona XtraDB Cluster is a high performance active-active database cluster used in high availability close-to-MySQL scenarios

About this document

This document shows the installation procedure for Percona and connects a hybris platform.

Audience: hybris administrators, system administrators, consultants, developers

Validity: 5.2.0

Based on hybris version: 5.2.0

Rate the Document :

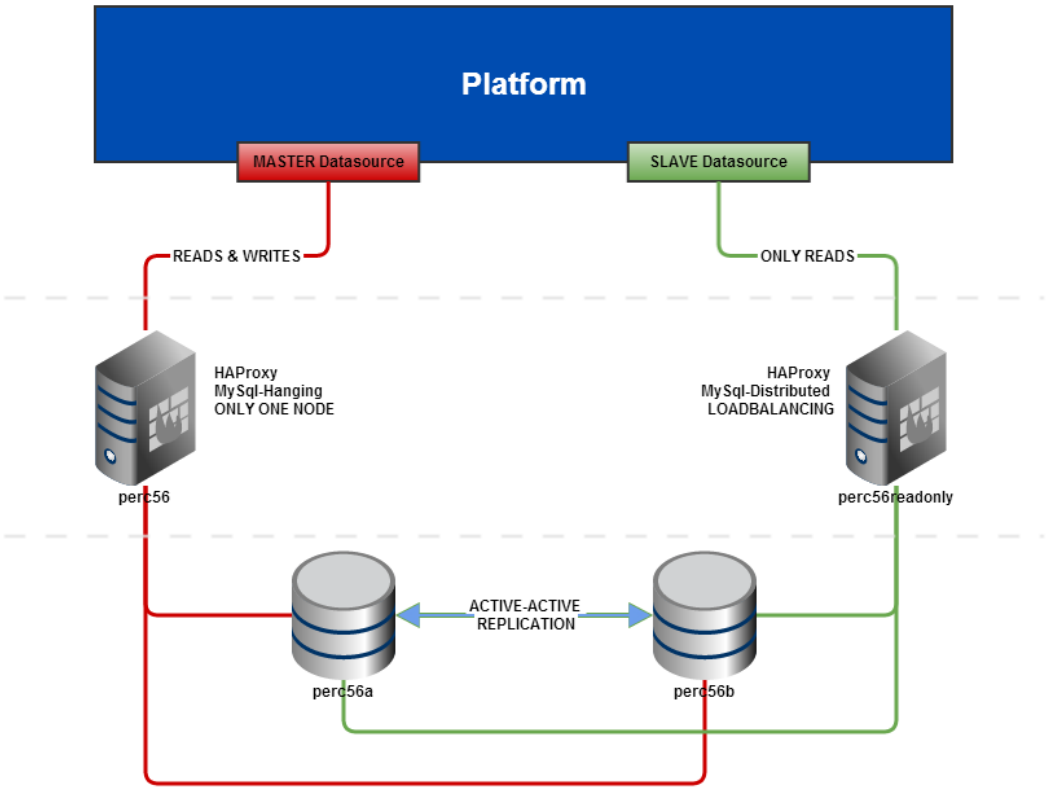
Your Rating: ☆☆☆☆☆ Results: ★★★★☆ 8 rates

See also

- http://www.percona.com/doc/percona-xtradb-cluster/5.6/howtos/virt_sandbox.html

Percona offers in comparison to MySQL :

- 100% binary and API compatibility with MySQL (version numbers are aligned)
- Enterprise support by an international company Percona
- active-active clustering
- a "real" backup tool (called xtrabackup)
- ~20% higher performance than MySQL



Naming Convention

The following naming convention for nodes and proxies is used throughout this installation guide:

Separate machines

- **perc56a** is Percona cluster node 1
- **perc56b** is Percona cluster node 2
- **percquorum** is Percona arbitrator node used to provide an even number of nodes, for details see Percona documentation

Separate IP addresses (virtual IP's)

- **perc56** is the Virtual IP ensuring "hanging" behavior (prefer one node to avoid race conditions)
- **perc56readonly** is the Virtual IP ensuring round robin access dedicated to read use.

Hardware Recommendations

hybris recommends a powerful set of machines for cluster node 1 and 2 with at least two network interface cards (cluster and data network) because these machines are the database servers. Hardware sizing should be done according to MySQL / Percona specs, depending on the expected workload. Both cluster nodes keep a copy of the full database and each needs to fulfill the performance requirements separately. Please plan accordingly: the cluster described here is twice as expensive as a single node (since two nodes with data load).

The arbitrator node (**percquorum**) does not get any load except network load. A minimum set of cores (2) and RAM (2GB) are recommended, it requires one network interface card to reach the cluster node and a second one just to monitor.

> **Read optimization via replica sets**
> Consider a separate read-only layer also knon as standard MySQL replication. The master (and therefore the write target) in this replication set can be set up as a Percona HA active-active cluster. This combines the scaling advantages of a MySQLreplication set

with the availability of the Percona active-active cluster. MySQL replicas are used to optimize read performance; write is mainly correlated to disk performance. To scale a write, the disk performance needs to be scaled up, scale out is not possible in MySQL. Because the hybris platform mostly executes read operations, the scaling with replicas is a viable option in this environment.

Higher availability via higher number of nodes
The described setup is the minimum in an active-active setup. If you want to achieve higher availability, for example, while servicing a node or while backing up, you need to increase the number of nodes (typical 3, so you are still high available while servicing one node). If you increase the number of cluster members, scale hardware, too. (A three node solution is *three* times as expensive as a single node solution and so on)

Services Used

Make sure that the cluster works seamlessly as you have to ensure not only the Percona product is needed (in order of installation and priority):

Service	Function
ntp / ntpdate	keeping the time in sync
Percona XtraDB Cluster 5.6	**providing the database**
haproxy / mysqlproxy	controlling the load balancing data flow
keepalived	maintaining the virtual ip fail over between the nodes
inetd / xinetd	making the cluster check available to the other node(s)
clustercheck.sh	monitoring the cluster status
transparent proxying (TPROXY)	make sure IP based acces restrictions are still working
firewall with e.g. ferm	ensure the security of the box

Installation Guide

Preparing the Machines

hybris uses default Debian stable as operating system.

1. Exchange ssh root keys (since the "valuable data" on the nodes is contained in the sql cluster and those credentials must be shared anyhow, this actually not lowerring the security). In this section we establish also a sync script for easy handling
   Click here to expand...

2. Add Percona repositories to the server repos on ALL nodes
   Click here to expand...
   a. add the repos:

/etc/apt/sources.list.d/percona.list

```
deb http://repo.percona.com/apt wheezy main
deb-src http://repo.percona.com/apt wheezy main
```

   b. add the valid Percona key to verify the signature of packages coming from Percona

```
> gpg --keyserver  keys.gnupg.net --recv-keys CD2EFD2A
> gpg -a --export CD2EFD2A | apt-key add -
```

c. change the priority to accept Percona packages in favor of maybe outdated debian version

/etc/apt/preferences.d/percona

```
Package: *
Pin: origin repo.percona.com
Pin-Priority: 990
```

d. make sure your repositories are uptodate

```
> apt-get update
```

3. Sync the time between the nodes and make them peer each other (already a drift of 0.5s is a major issue for the cluster)
   Click here to expand...
   a. install ntp

```
> apt-get install ntp lockfile-progs
```

   (lockfile-progs is needed in debian to ensure a proper restart)
   b. modify the ntp config file

/etc/ntp.conf

```
# select an adequate time server "near by"
server time1.fra.hybris.com iburst
server time2.fra.hybris.com iburst

# make sure to peer between servers
peer perc56a.database.fra.hybris.com iburst
peer perc56b.database.fra.hybris.com iburst
peer percquorum.database.fra.hybris.com iburst

# By default, exchange time with everybody, but don't allow configuration
and peering.
restrict -4 default kod notrap nomodify nopeer noquery
restrict -6 default kod notrap nomodify nopeer noquery

# allow peering
restrict perc56a.database.fra.hybris.com peer
restrict perc56b.database.fra.hybris.com peer
restrict percquorum.database.fra.hybris.com peer

# Local users may interrogate the ntp server more closely.
restrict 127.0.0.1
restrict ::1
```

   (this file is valid on all servers, the reference to the server itself is automatically ignored, so distribute on ALL servers without modification)

> the percquorum server could be used in different clusters, make sure it is peering to all nodes of all clusters it is servicing.

   c. Restart ntp

```
> service ntp restart
```

d. Verify correct operation

```
> ntpq -p
     remote           refid      st t when poll reach   delay   offset
jitter
==============================================================================
=====
*timec.fra.hybri 195.50.171.101   3 u    6   64  377    0.376    5.721
4.997
-timeb.fra.hybri 129.70.132.33    3 u   62   64  377    0.374   14.931
4.704
+perc56a.databas 10.8.1.148       4 u   26   64  377    0.789    2.112
4.491
+percquorum.data 10.8.2.149       4 u   37   64  377    0.829    1.962
3.758
```

4. make sure network confiuration is working
5. exchange ss

Installing Percona

Install Percona Cluster Node

1. Install Percona on the cluster nodes 1 and 2 :

```
apt-get install percona-xtradb-cluster-56
```

(it will ask you for the initial database password. The one you specified on node 1 will be important later in this document)

1. To be able to use the slave data source in the platform, add the following configuration to your `local.properties` file:

```
db.url=jdbc:mysql://perc56.<youraddress>.com/<dbname>?useConfigs=maxPerformance&c
haracterEncoding=utf8
db.driver=com.mysql.jdbc.Driver
db.username=<username>
db.password=<password>
db.tableprefix=
mysql.optional.tabledefs=CHARSET=utf8 COLLATE=utf8_bin
mysql.tabletype=InnoDB
db.customsessionsql=SET SESSION TRANSACTION ISOLATION LEVEL READ COMMITTED;
mysql.allow.fractional.seconds=true
slave.datasource.1.db.url=jdbc:mysql://perc56read.<youraddress>.com/<dbname>?useC
onfigs=maxPerformance&characterEncoding=utf8
slave.datasource.1.db.driver=com.mysql.jdbc.Driver
slave.datasource.1.db.username=<username>
slave.datasource.1.db.password=<password>
slave.datasource.1.db.tableprefix=
slave.datasource.1.mysql.optional.tabledefs=CHARSET=utf8 COLLATE=utf8_bin
slave.datasource.1.mysql.tabletype=InnoDB
slave.datasource.1.db.customsessionsql=SET SESSION TRANSACTION ISOLATION LEVEL
READ COMMITTED;
slave.datasource.1.mysql.allow.fractional.seconds=true
```

2. Install an external disk as `/dev/vdb` for mysql data on nodes 1 and 2 in /etc/fstab:

```
LABEL=data      /media/data0    ext4    noatime,nobarrier,discard       0       1
```

```
/dev/vdb        197G 1015M  186G   1% /media/data0
```

3. Configure percona on nodes 1 and 2 in `/etc/mysql/my.cnf`:
   - set the log to `/var/syslog`

my.cnf

```
syslog
syslog-facility=local0
```

   - configure the cluster and nodes. Put this configuration on node 1

my.cnf

```
wsrep_cluster_address=gcomm://perc56b.<yourinternaladdress>.com,sqlquorum.<
yourinternaladdress>.com
wsrep_cluster_name=<yourclustername, for example perc_mygreatcluster>
wsrep_node_name=perc56a.<yourinternaladdress>.com
wsrep_data_home_dir=/var/lib/wsrep
wsrep_provider='/usr/lib/libgalera_smm.so'
wsrep_provider_options='gcache.size=1G'
wsrep_replicate_myisam=1
wsrep_slave_threads=8
wsrep_sst_method=xtrabackup
wsrep_sst_auth=root:{password}
```

and this configuration on node 2

my.cnf

```
wsrep_cluster_address=gcomm://perc56a.<yourinternaladdress>.com,sqlquorum.<
yourinternaladdress>.com
wsrep_cluster_name=<yourclustername, for example perc_mygreatcluster>
wsrep_node_name=perc56a.<yourinternaladdress>.com
wsrep_data_home_dir=/var/lib/wsrep
wsrep_provider='/usr/lib/libgalera_smm.so'
wsrep_provider_options='gcache.size=1G'
wsrep_replicate_myisam=1
wsrep_slave_threads=8
wsrep_sst_method=xtrabackup
wsrep_sst_auth=root:{password}
```

- binlog_format = ROW (this is the default setup)
- data dir on external disk

my.cnf

```
datadir=/srv/mysql
```

- remove the following parameters:

my.cnf

```
#innodb_locks_unsafe_for_binlog=1
#table_cache=1M
```

4. Copy `debian.cfg` to the root on node 2. Use the following command:

```
scp /etc/mysql/debian.cnf eit@perc56b
```

5. Create a **clustercheck** user in MySQL for monitoring purposes.

```
GRANT PROCESS ON *.* TO 'clustercheckuser'@'localhost' IDENTIFIED BY
'clustercheckpassword!';
```

6. Install the following necessary tools:

```
apt-get install haproxy xinetd httpcheck keepalived ntpdate
```

   - haproxy (load balancer) on nodes 1 and 2. To access haproxy statistics, use the following address pattern:

```
http://perc56.<yourinternaladdress>.com/haproxy/stats
```

   - xinetd on port 9200 (ENABLED = 1)
   - httpcheck, which checks if the server is running
   - keepalived (failover) on nodes 1 and 2
   - ntpdate to synchronize the time with an ntp server, set this parameter in /etc/default/ntpdate:

```
NTPSERVERS=<address of your ntp server>
```

7. Configure DHCP (DNS), to be able to use URLs to access nodes
8. To perform an operation specifically on the slave data source, use the following example code:

```
public static void doOnSlaveDataSource(final Runnable action)
        {
                final Tenant tenant = Registry.getCurrentTenantNoFallback();
                ((AbstractTenant) tenant).cancelForceMasterMode();
                tenant.activateSlaveDataSource();
                try
                {
                        requireNonNull(action).run();
                }
                finally
                {

Registry.getCurrentTenantNoFallback().deactivateAlternativeDataSource();
                }
        }
```