



Tutorials

Tags

Forums

Linux Commands

Subscribe

ISPConfig

News

Q Tutorial search



Tutorials

How To Set Up A Load-Balanced MySQL Cluster

How To Set Up A Load-Balanced MySQL Cluster

Version 1.0

Author: Falko Timme

On this page

- [1 My Servers](#)
- [2 Set Up The MySQL Cluster Management Server](#)

This tutorial shows how to configure a MySQL 5 cluster with three nodes: two storage nodes and one management node. This cluster is load-balanced by a high-availability load balancer that in fact has two nodes that use the *Ultra Monkey* package which provides *heartbeat* (for checking if the other node is still alive) and *ldirectord* (to split up the requests to the nodes of the MySQL cluster).

In this document I use Debian Sarge for all nodes. Therefore the setup might differ a bit for other distributions. The MySQL version I use in this setup is 5.0.19. If you do not want to use MySQL 5, you can use MySQL 4.1 as well, although I haven't tested it.

This howto is meant as a practical guide; it does not cover the theoretical backgrounds. They are treated in a lot of other documents in the web.

This document comes without warranty of any kind! I want to say that this is not the only way of setting up such a system. There are many ways of achieving this goal but this is the way I take. I do not issue any guarantee that this will work for you!

1 My Servers

I use the following Debian servers that are all in the same network (192.168.0.x in this example):

- *sql1.example.com*: 192.168.0.101 MySQL cluster node 1
- *sql2.example.com*: 192.168.0.102 MySQL cluster node 2
- *loadb1.example.com*: 192.168.0.103 Load Balancer 1 / MySQL cluster management server
- *loadb2.example.com*: 192.168.0.104 Load Balancer 2

In addition to that we need a virtual IP address : 192.168.0.105. It will be assigned to the MySQL cluster by the load balancer so that applications have a single IP address to access the cluster.

Although we want to have two MySQL cluster nodes in our MySQL cluster, we still need a third node, the MySQL cluster management server, for mainly one reason: if one of the two MySQL cluster nodes fails, and the management server is not running, then the data on the two cluster nodes will become inconsistent ("*split brain*"). We also need it for configuring the MySQL cluster.

So normally we would need five machines for our setup:

2 MySQL cluster nodes + 1 cluster management server + 2 Load Balancers = 5

As the MySQL cluster management server does not use many resources, and the system would just sit there doing nothing, we can put our first load balancer on the same machine, which saves us one machine, so we end up with four machines.

2 Set Up The MySQL Cluster Management Server

First we have to download MySQL 5.0.19 (the **max** version!) and install the cluster management server (*ndb_mgmd*) and the cluster management client (*ndb_mgm* - it can be used to monitor what's going on in the cluster). The following steps are carried out on *loadb1.example.com* (192.168.0.103):

loadb1.example.com:

```
mkdir /usr/src/mysql-mgm
cd /usr/src/mysql-mgm
wget http://dev.mysql.com/get/Downloads/MySQL-5.0/mysql-max-5.0.19-linux-i686-  
glibc23.tar.gz/from/http://www.mirrorservice.org/sites/ftp.mysql.com/  
tar xvfz mysql-max-5.0.19-linux-i686-glibc23.tar.gz
cd mysql-max-5.0.19-linux-i686-glibc23
mv bin/ndb_mgm /usr/bin
mv bin/ndb_mgmd /usr/bin
chmod 755 /usr/bin/ndb_mg*
cd /usr/src
rm -rf /usr/src/mysql-mgm
```

Next, we must create the cluster configuration file, */var/lib/mysql-cluster/config.ini*:

loadb1.example.com:

```
mkdir /var/lib/mysql-cluster
cd /var/lib/mysql-cluster
vi config.ini
```

```
[NDBD DEFAULT]
NoOfReplicas=2

[MYSQLD DEFAULT]

[NDB_MGMD DEFAULT]

[TCP DEFAULT]

# Section for the cluster management node
[NDB_MGMD]
# IP address of the management node (this system)
HostName=192.168.0.103

# Section for the storage nodes
[NDBD]
# IP address of the first storage node
HostName=192.168.0.101
DataDir= /var/lib/mysql-cluster

[NDBD]
# IP address of the second storage node
HostName=192.168.0.102
DataDir=/var/lib/mysql-cluster

# one [MYSQLD] per storage node
[MYSQLD]
[MYSQLD]
```

Please replace the IP addresses in the file appropriately.

Then we start the cluster management server:

loadb1.example.com:

```
ndb_mgmd -f /var/lib/mysql-cluster/config.ini
```

It makes sense to automatically start the management server at system boot time, so we create a very simple init script and the appropriate startup links:

loadb1.example.com:

```
echo 'ndb_mgmd -f /var/lib/mysql-cluster/config.ini' > /etc/init.d/ndb_mgmd
chmod 755 /etc/init.d/ndb_mgmd
update-rc.d ndb_mgmd defaults
```

Next >>

 [view as pdf](#) |  [print](#)

Share this page:

[Tweet](#)

[Follow @howtoforgecom](#)

[27.8K followers](#)

[Recommend 7](#)

[G+1](#) [5](#)

Sub pages

How To Set Up A Load-Balanced MySQL Cluster

[How To Set Up A Load-Balanced MySQL Cluster - Page 2](#)

[How To Set Up A Load-Balanced MySQL Cluster - Page 3](#)

[How To Set Up A Load-Balanced MySQL Cluster - Page 4](#)

[How To Set Up A Load-Balanced MySQL Cluster - Page 5](#)

[How To Set Up A Load-Balanced MySQL Cluster - Page 6](#)

[How To Set Up A Load-Balanced MySQL Cluster - Page 7](#)

[How To Set Up A Load-Balanced MySQL Cluster - Page 8](#)

Suggested articles

20 Comment(s)

Add comment

Name *

Email *

  **B** *I* 

p



I'm not a robot

reCAPTCHA
[Privacy](#) - [Terms](#)

Submit comment

Comments

From: Anonymous

Reply

This is rather unfortunate, but without foreign keys support and memory-only storage MySQL cluster is not a really viable solution for most RDBMS users.

From: Anonymous

Reply

The InnoDB engine supports foreign keys and works with the MySQL cluster so your comment is incorrect sir.

From: Anonymous

Reply

Actually you are wrong mate. While InnoDB in MySQL supports foreign keys, you can not use InnoDB while setting up the MySQL Cluster support described in this article. You can only use the NDB backend which is a simplified version of MyISAM. It also has the limitation of being completely memory resident.

From: Anonymous

Reply

NDB Cluster has *nothing* to do with MyISAM. NDB has a long history outside of MySQL, and it has no relationship whatsoever to MyISAM.

From:

Reply

We had MySQL Cluster running as the backend for a cluster of webapps and it had numerous problems. Many of them just from lack of needed features. I opened a lot of enhancement requests to MySQL about these. Some like all 'mysql' tables are not centralized but separate would nearly drive you insane. We had to try synchronizing the user tables between all the client nodes. Another issue is that when a user installs a webapp, if one of the client nodes happened to be down for maintenance at the time, that node would never learn of the new database the user just setup so if the load balancer directed him to that client node later on everything would error. There are just many architectural issues with MySQL Cluster that were never very well thought through. It has a long way to go to being enterprise ready. And performance was abysmal to boot.

From: Anonymous

Reply

Memory only storage is a significant limitation. I hope this is fixed in a future version.

From: Anonymous

Reply

Well, not anymore:

In MySQL 5.1, the memory-only requirement of MySQL Cluster is removed and operational data may now be accessed both on disk and memory. A DBA can specify that table data can reside on disk, in memory, or a combination of main memory and disk (although a single table can only be assigned to either disk or main memory). Disk-based support includes new storage structures - tablespaces – that are used to logically house table data on disk. In addition, new memory caches are in place to manage the transfer of data stored in tablespaces to memory for fast access to repeatedly referenced information.

From: Anonymous

Reply

"Memory only storage is a significant limitation. I hope this is fixed in a future version." This isn't a limitation to be fixed, but the fundamental tradeoff in MySQL Cluster Server's design: by accepting the limitation of being memory-based instead of disk based, it can be several orders of magnitude faster. If your data can't fit in RAM, and you don't need the performance, you should use one of the disk-based table types.

From: Anonymous

Reply

The cluster management software seems to be a single point of failure; that is, if the load balancer running this software goes down, doesn't the cluster either go down or end up with inconsistent data ("*split brain*", as referenced in the article)? I'm very new to clustering, so I'd be happy to learn why I'm wrong!

From: Anonymous

Reply

The storage and MySQL Server nodes are not dependent on the management server for their execution. Its purpose is only to manage the cluster. It may fail and be restarted any number of times without affecting the running MySQL Cluster.

From: Anonymous

Reply

You use Debian and install package from source. Don't store files in /usr/bin. Use /usr/local or /opt. Custom packages installed in /usr can be broken by Debian packaging system.

From: Anonymous

Reply

Actually, if you use chinstall instead of 'make instal', it'll add the package to your apt setup so the files won't be overwritten

From:

Reply

You can make each balancer a management server and eliminate a single point of failure. Install manager on both lb and add both to config.ini

Managment Server 1

[NDB_MGMD]

HostName=192.168.0.8 # the IP of the First Management Server

ID=1

Datadir=/var/lib/mysql-cluster

Managment Server 2

[NDB_MGMD]

HostName=192.168.0.9 # the IP of the Second Management Server

ID=2

Datadir=/var/lib/mysql-cluster

then on each data node modify my.cnf

[mysqld]

ndbcluster

ndb-connectstring = "host=192.168.0.8,host=192.168.0.9"

[ndb_mgm]

connect-string = "host=192.168.0.8,host=192.168.0.9"

[ndbd]

connect-string = "host=192.168.0.8,host=192.168.0.9"

make sure to run ndbd --initial

From:

Reply

I wanted to know whether or not there will be significant changes in performance if we run apache with load balancing enabled with mysql cluster.

Has anyone tried it before ?

Anjin

From: Elumalai Ranganathan

Reply

Thanks a lot! This document helped a lot in configuring MYSQL cluster. I have a query. I am going to configure web server on the nodes using Tomcat, Is it possible to use the mysql virtual ip for tomcat configuration...

From: Anonymous

Reply

<http://www.dancryer.com/2010/01/mysql-circular-replication>

This is part 1 of a three posts series:

- MySQL Load-Balanced Cluster Guide – Part 1 - setting up the servers themselves and configuring MySQL replication.

- MySQL Load-Balanced Cluster Guide – Part 2 - set up a script to monitor the status of your MySQL cluster nodes, which we'll use in the next guide to set up our proxy.

- MySQL Load-Balanced Cluster Guide – Part 3 - setting up the load balancer with HAProxy, using the monitoring scripts

From: Altaf Hussain

Reply

Very elaborative tutorial I must say. A noob coming here can do a lot after reading the tutorial !!

From: Driggs

Reply

How to add additional data node?

From: Majed

Reply

This is a very old tutorial, and needs to be updated

From: Muhammad Karam Shehzad

Reply

Question Regarding MySQL Cluster Ports Hello, I am on Linux platform with MySQL NDB 5.7. I am trying to monitor all traffic related to MySQL clustering - between data nodes, management node and sql nodes. To that end, I used netstat to list all open ports listening on my machine before starting MySQL cluster. Then, I started MySQL cluster and ran netstat again. I assumed that the ports that were listening the second time around, but not the first time, were related to MySQL clustering. But there are two problems with this. First, there could be ports opened by other processes between the two netstat runs. Second, MySQL might open other ports after I ran the netstat command the second time.

What is the best way to go about finding all ports being used by MySQL for clustering purposes?

I believe ephemeral ports are picked dynamically, so perhaps if I knew all the MySQL clustering related processes that would be running, I can figure out every port that they are using. Pointers will be very welcome.

Tutorials

How To Set Up A Load-Balanced MySQL Cluster

Sign up now!



Tutorial Info

Author: falko
Tags: mysql, high-availability

Share This Page

Tweet

Recommend 7

G+1 5

