Hands on Lab
Part 2 - MySQL Cluster

DBTech's Virtual Workshop on Distributed, Replicated, Embedded and Mobile Databases (Deliverable #28).

Antonio César Gómez Lora

Dpto. Lenguajes y Ciencias de la Computación

Universidad de Málaga

Introduction

We are going to create a MySQL Cluster using three machines. The first one (192.168.192.1) will contain the cluster manager. The second and third machines (192.168.192.2 and 3) will use their MySQL as the storage nodes.

We start now using the previously created and cloned virtual machines. Those machines have Debian (5.0.5/5.0.6), MySQL, PostgreSQL and PgPool-II installed.

Creating the cluster

So, our first step after starting the first machine and login as root is to edit the configuration file of the MySQL Cluster Manager. This file is located on /etc/mysql/ndb_mgmd.cnf (Fig. 1), so execute:

```
nano /etc/mysql/ndb_mgmd.cnf
```

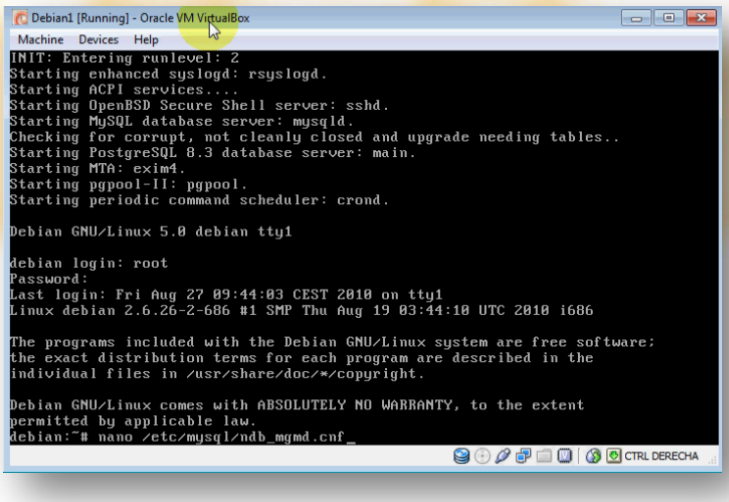

Fig. 1. ndb_mgmd.cnf

Include on this file the following sections:

```
[NDBD DEFAULT]
NoOfReplicas=2
DataMemory=32M
IndexMemory=16M

[MYSQLD DEFAULT]

[NDB_MGMD DAFULT]

[TCP DEFAULT]

[NDB_MGMD]
HostName=192.168.192.1

[NDBD]
HostName=192.168.192.2
DataDir=/var/lib/mysql-cluster
BackupDataDir=/var/lib/mysql-cluster/backup

[NDBD]
HostName=192.168.192.3
DataDir=/var/lib/mysql-cluster
BackupDataDir=/var/lib/mysql-cluster/backup

[MYSQLD]

[MYSQLD]
```

Fig. 2. ndb_mgmd.cnf content

These are:

- NDBD DEFAULT.- This section configures the global values of the cluster.
- MYSQLD DEFAULT, NDB_MGMD DEFAULT, TCP DEFAULT.- This sections loads the default values for the cluster manager and network protocol.
- NDB_MGMD.- Configuration of the cluster manager.
- NDBD.- One section for each replica; configuration of each replica.
- MYSQLD.- One section for each replica; configuration of MySQL on each replica.

Sections are mandatory, even those ones without any parameter. Each section can contain many different configuration parameters; please refer to MySQL Cluster Configuration section in MySQL manual.

On the example above some parameters can be omitted, because they have default values.

When you have edited the file, please exit nano by presing CTRL+X, then Y to confirm saving and ENTER to confirm name, Fig. 3.

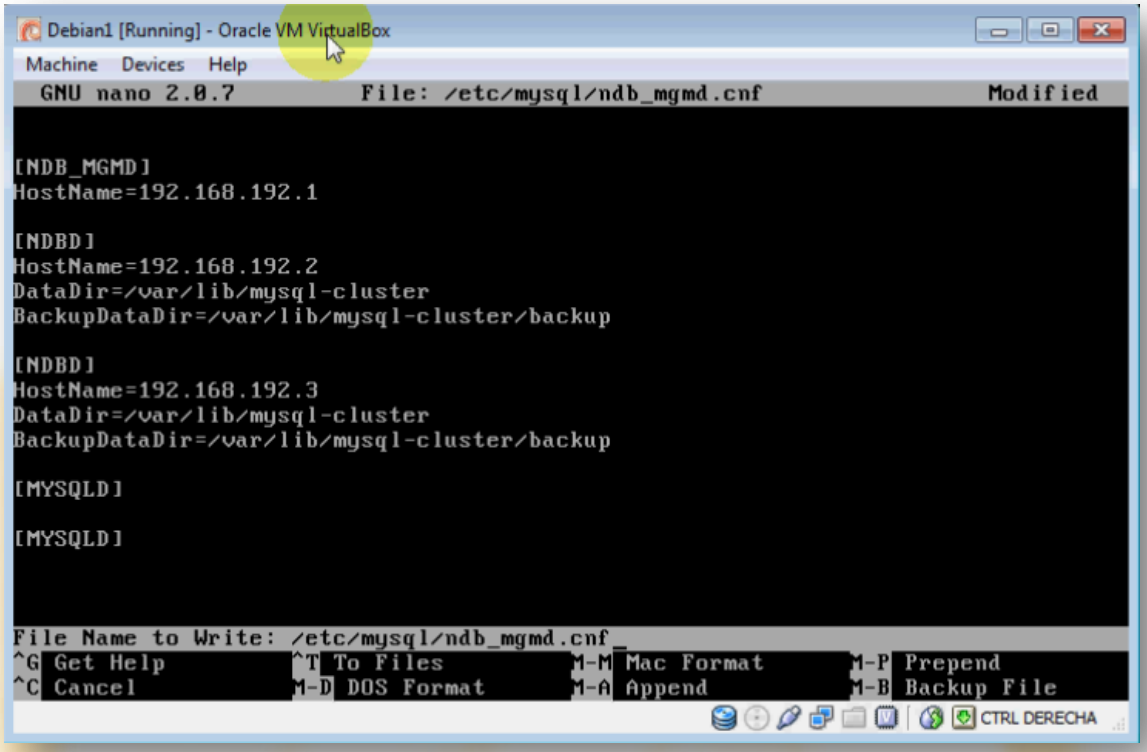

Fig. 3. ndb_mgmd.cnf edited.

Now start the second machine, Fig. 4.

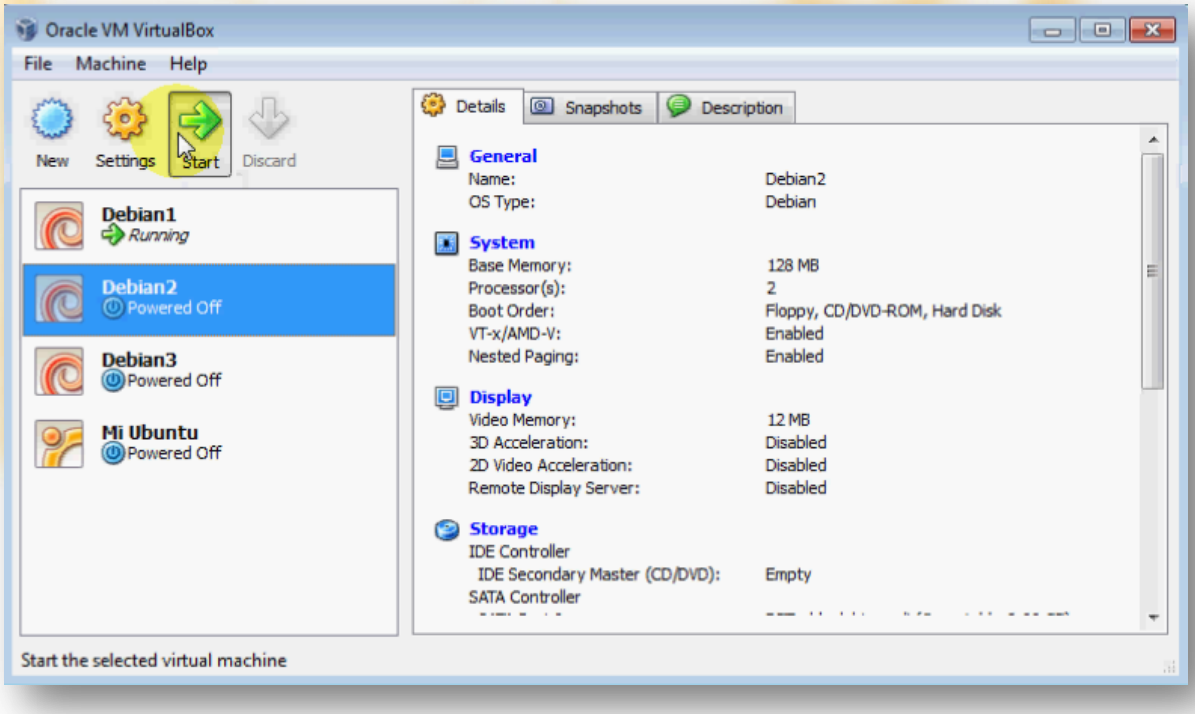

Fig. 4. Starting the second machine.

Version to be presented on
DBTech's Helsinki Meeting, Nov'10
Please feedback any comment to: cesar@lcc.uma.es

Login as root, Fig. 5.

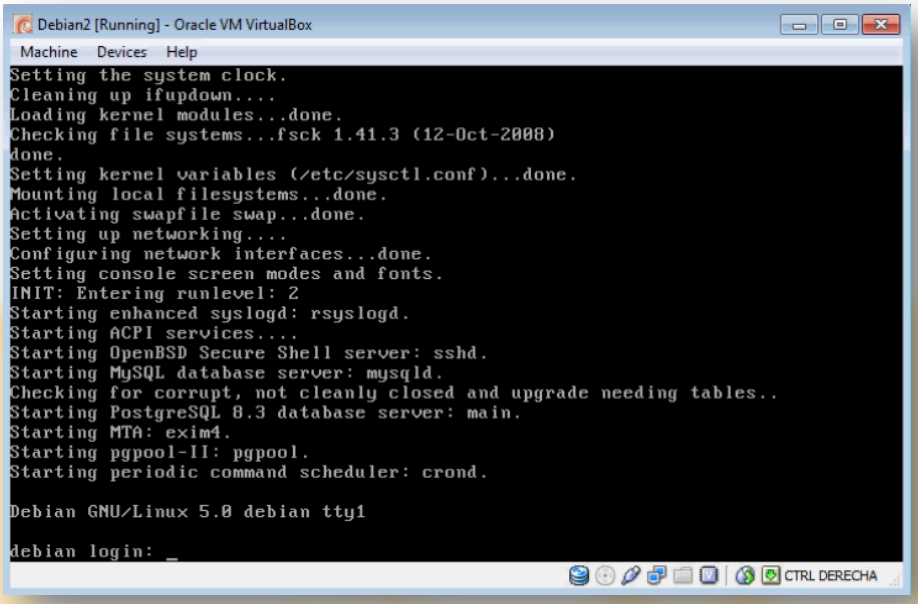

Fig. 5. Login.

Although it was not a necessary parameter, as we have used it, we must create the BackupDataDir directory. Remember to declare mysql user as the owner, Fig. 6. Execute:

mkdir /var/lib/mysql-cluster/backup

chown -R mysql:mysql /var/lib/mysql-cluster

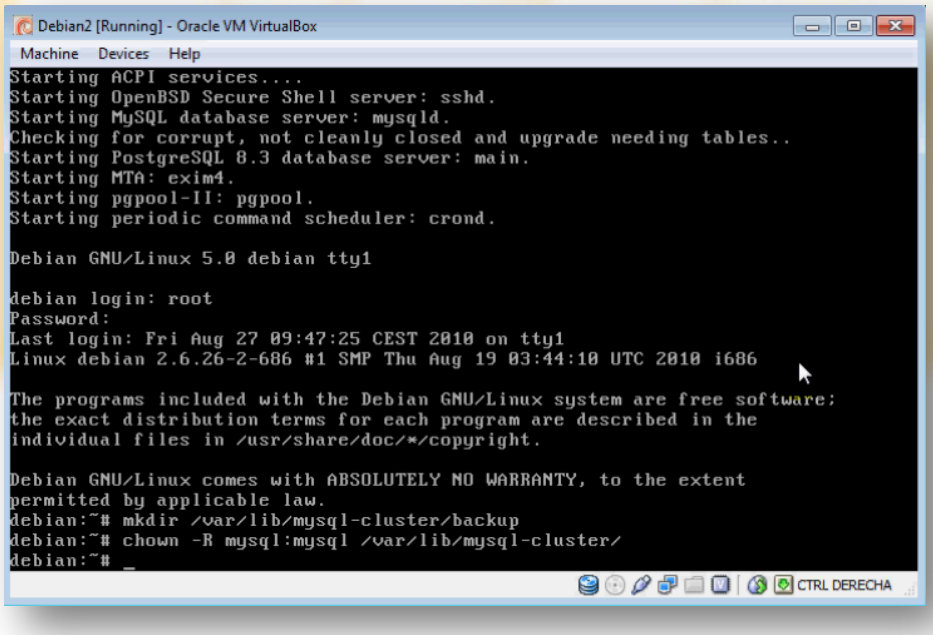

Fig. 6. Creating backup dir.

Version to be presented on
DBTech's Helsinki Meeting, Nov'10
Please feedback any comment to: cesar@lcc.uma.es

Now it is time to edit my.cnf (the mysql configuration file), Fig. 7.

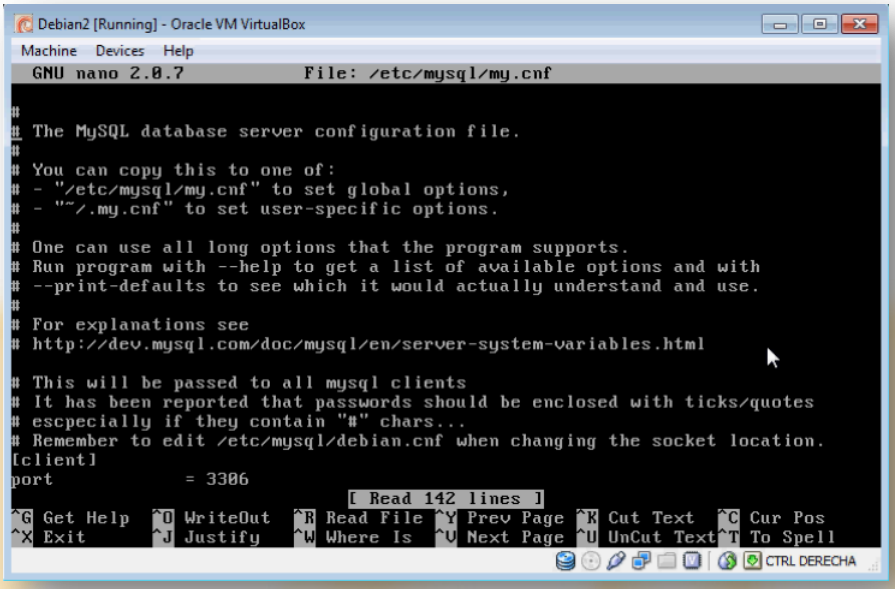

Fig. 7. my.cnf

At the end of [mysqld] section include:

ndbcluster

ndb-connectstring = 192.168.192.1

These lines tells MySQL to start ndb engine and declares where is the manager, Fig. 8.

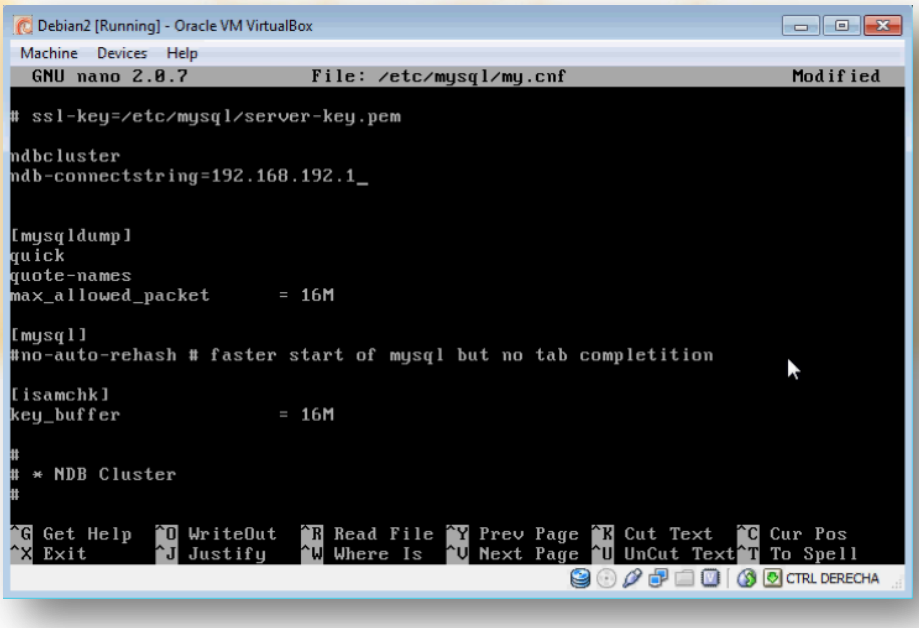

Fig. 8. ndbcluster on my.cnf

Version to be presented on
DBTech's Helsinki Meeting, Nov'10
Please feedback any comment to: cesar@lcc.uma.es

In this file you also find a commented [MYSQL_CLUSTER], Fig. 9, include after the comment:

[MYSQL_CLUSTER]

ndb-connectstring = 192.168.192.1

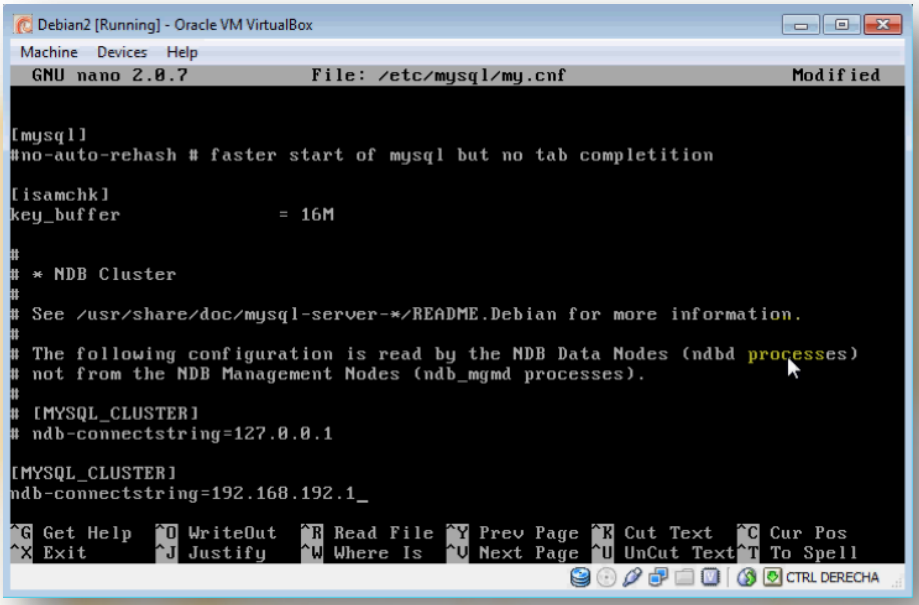

Fig. 9. ndb-connectstring on my.cnf

Now start the third virtual machine, Fig. 10.

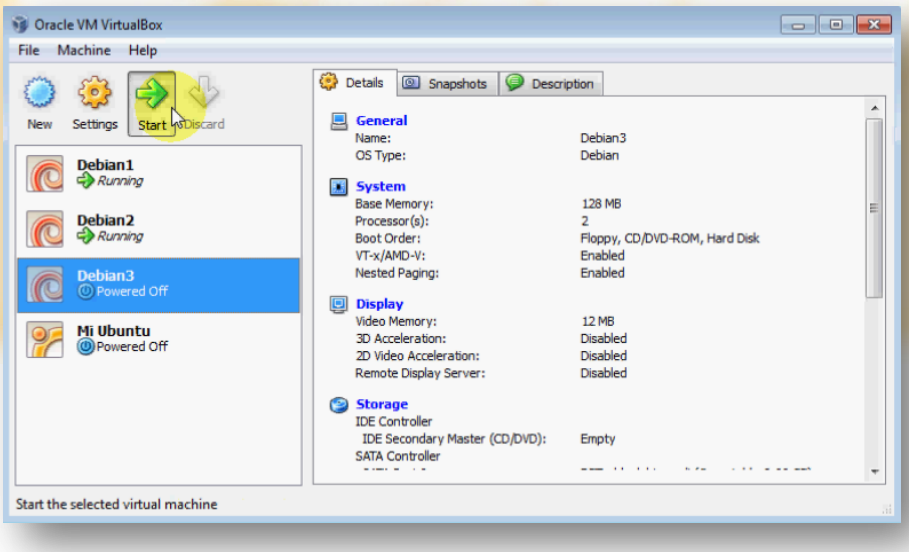

Fig. 10. Starting the third machine.

And now repeat the process, that is, create backup directory and edit my.cnf file.

Version to be presented on
DBTech's Helsinki Meeting, Nov'10
Please feedback any comment to: cesar@lcc.uma.es

When you have properly configured the third machine, go to the first machine and restart the cluster manager service, Fig. 11. Execute:

/etc/init.d/mysql-ndb-mgm restart

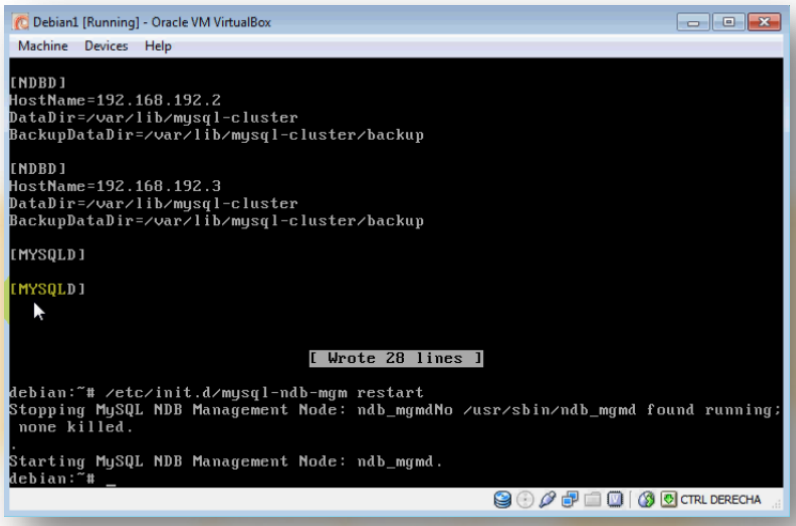

Fig. 11. Restarting service mysql-ndb-mgm

When service restarts, execute:

ndb_mgm

The cluster manager console, Fig. 12, and execute:

show;

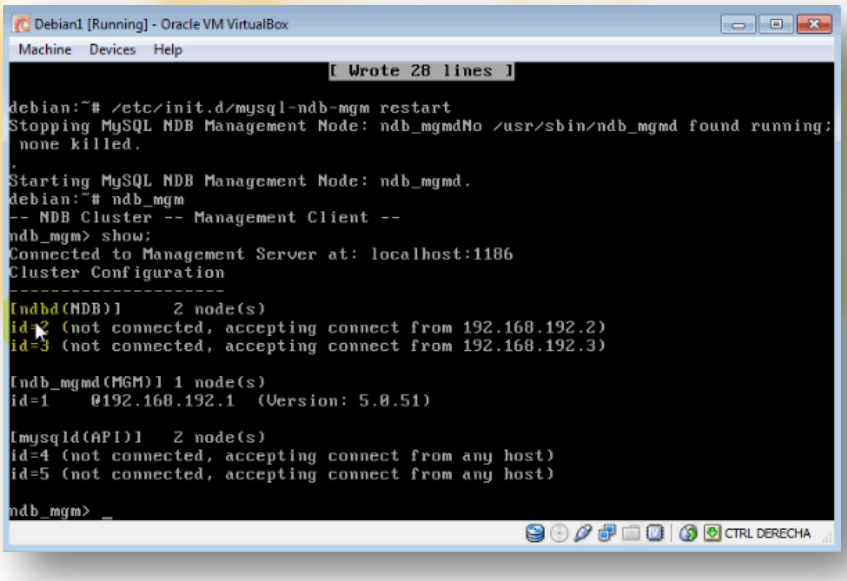

Fig. 12. ndb_mgm accepting connections

You can see that cluster manager is waiting for its storage nodes.

Restart services mysql and mysql-ndb on second and third machines, Fig. 13. On these machines execute:

/etc/init.d/mysql restart

/etc/init.d/mysql-ndb restart

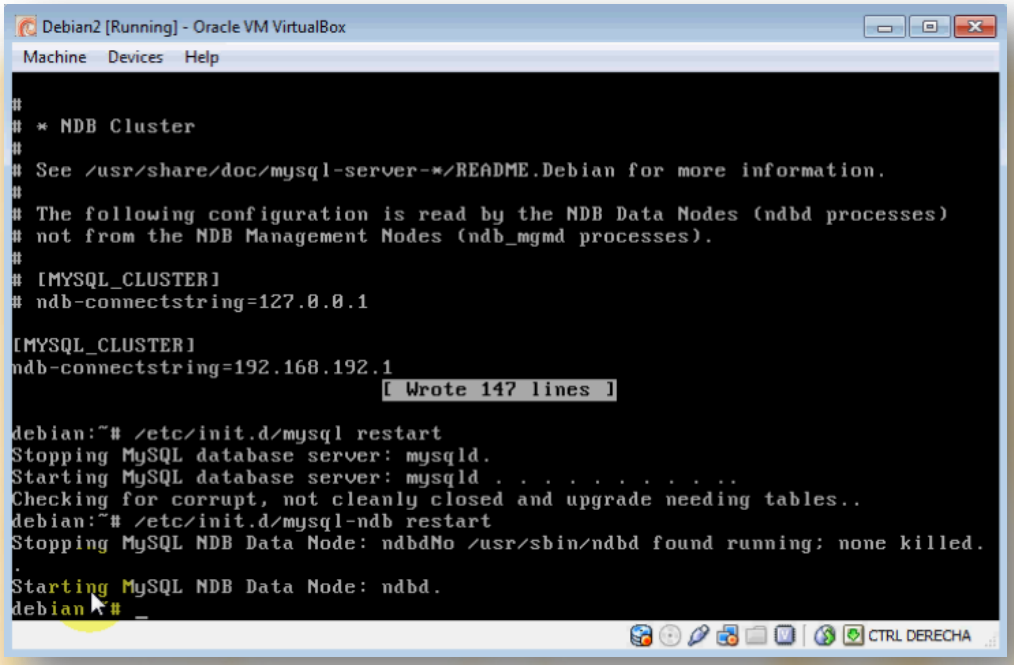

Fig. 13. Restarting mysql and mysql-ndb services

Executing show again on the cluster manager console, on the first machine, you will see that storage nodes are now connected, Fig. 14.

Fig. 14. Connected.

Version to be presented on
DBTech's Helsinki Meeting, Nov'10
Please feedback any comment to: cesar@lcc.uma.es

Testing the cluster

Now the cluster is working, let's test it. On the second machine start mysql interpreter (Fig. 15), execute:

mysql --user=root --password = dbtech

When connected execute:

create database clustertest;

use clustertest;

We are going to work on this database.

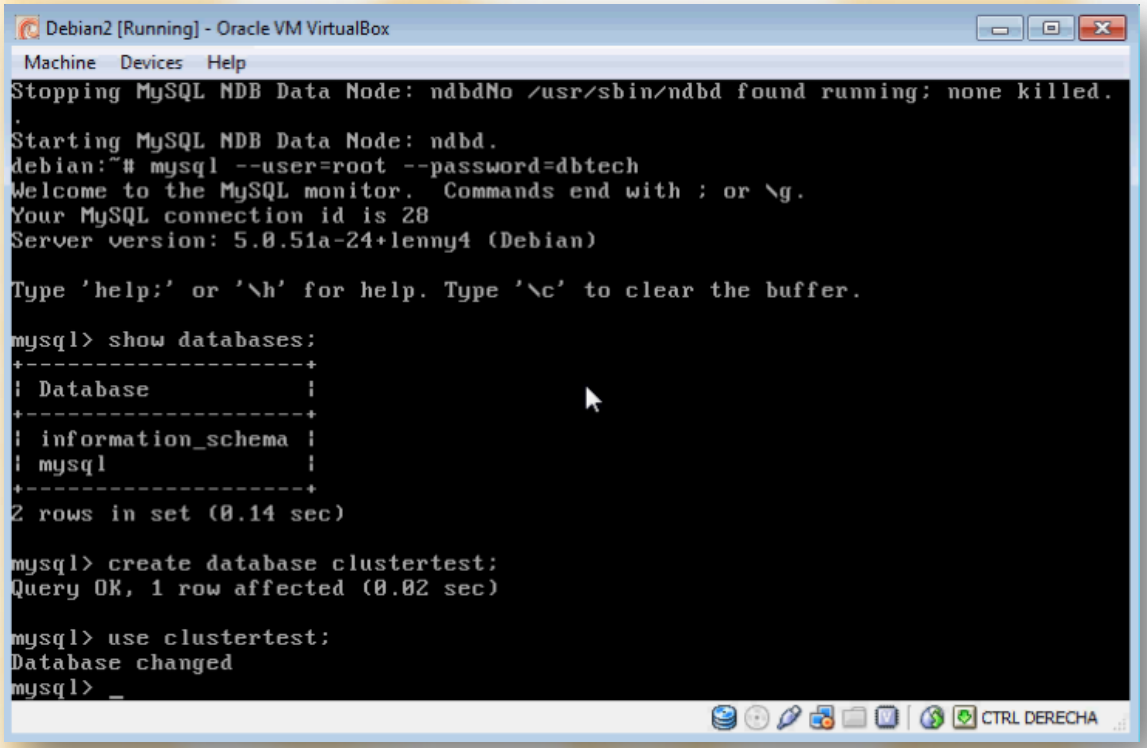

Fig. 15. Clustertest on second machine.

The easy way to test the cluster is to create a table and insert one tuple, and then verify it it has been replicated on the other storage node.

Execute:

create table test(i int) engine=ndbcluster;

insert into test values(1);

MySQL Cluster is implemented as a storage engine. You must specify the engine ndbcluster on table creation if you want to replicate it, Fig. 16.

Khaos
Version to be presented on
DBTech's Helsinki Meeting, Nov'10
Please feedback any comment to: cesar@lcc.uma.es

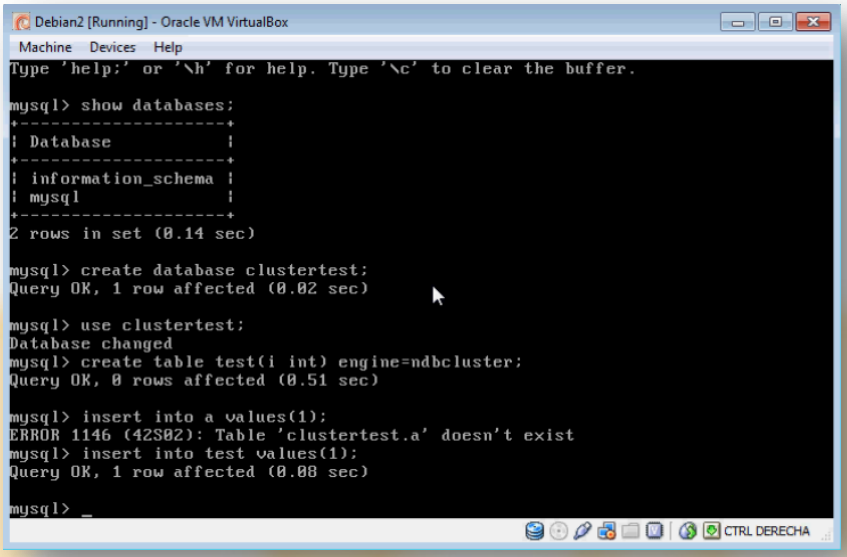

Fig. 16. Test table.

Now go to the third machine, start mysql interpreter, create clustertest database and use it as default, Fig. 17. Execute on third machine:

mysql --user=root --password=dbtech

create database clustertest;

use clustertest;

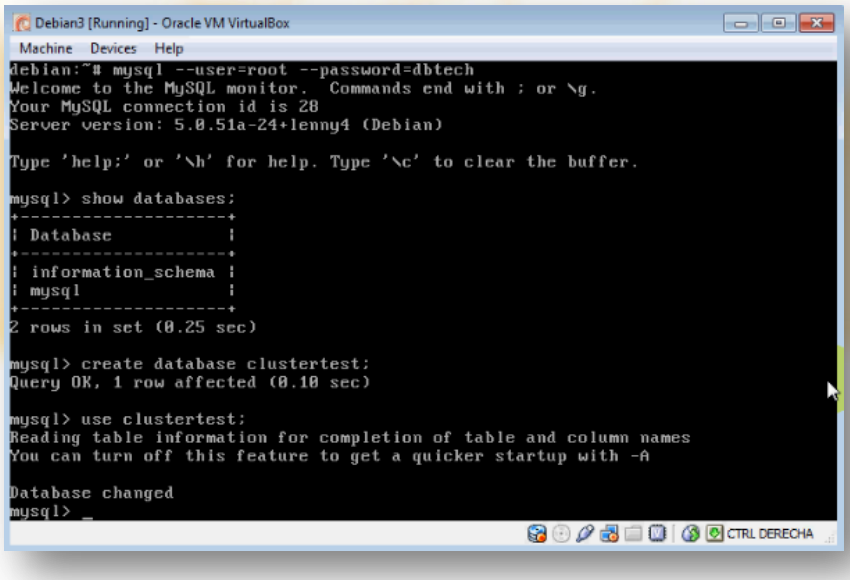

Fig. 17. Clustertest on third machine.

Version to be presented on
DBTech's Helsinki Meeting, Nov'10
Please feedback any comment to: cesar@lcc.uma.es

And now see if table test exists, execute (Fig. 18) on third machine:

select * from test;

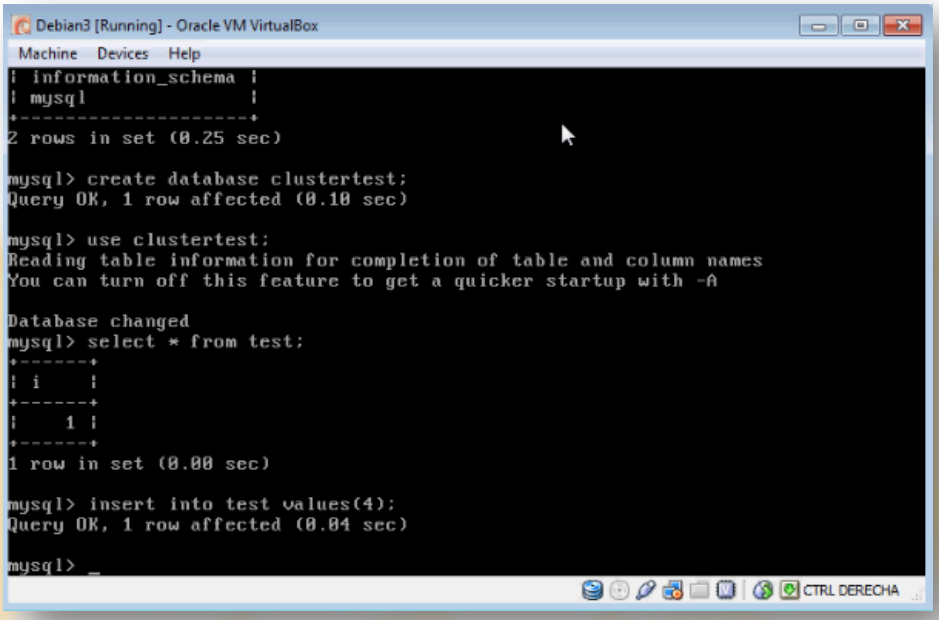

Fig. 18. Test table replicated.

You can now insert a tuple from the third machine, and its must be viewed on the second machine, Fig. 19.

Try to delete this new value from the second machine.

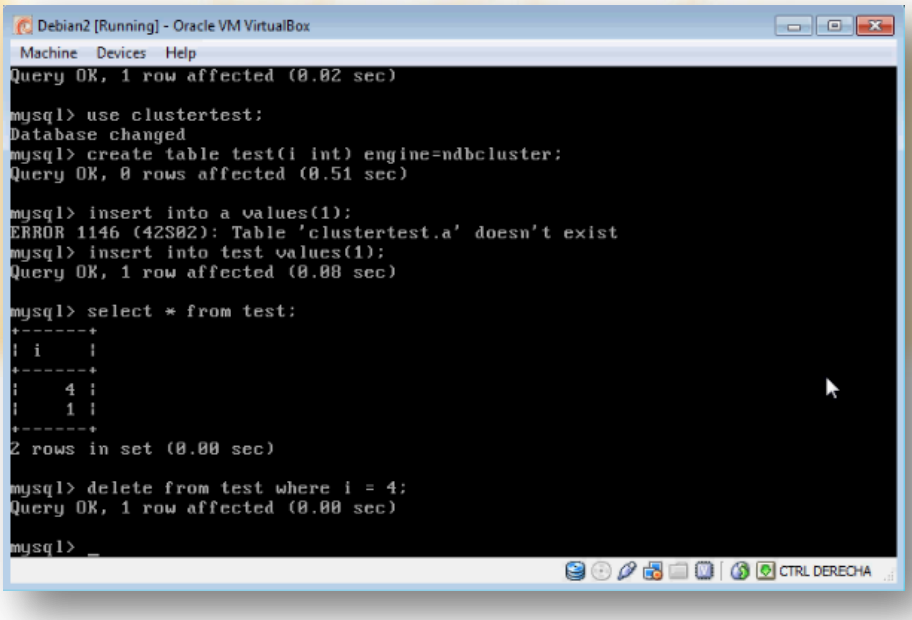

Fig. 19. Inserting and deleting on third machine.

And observe how it disappears also on the other machine, Fig. 20.

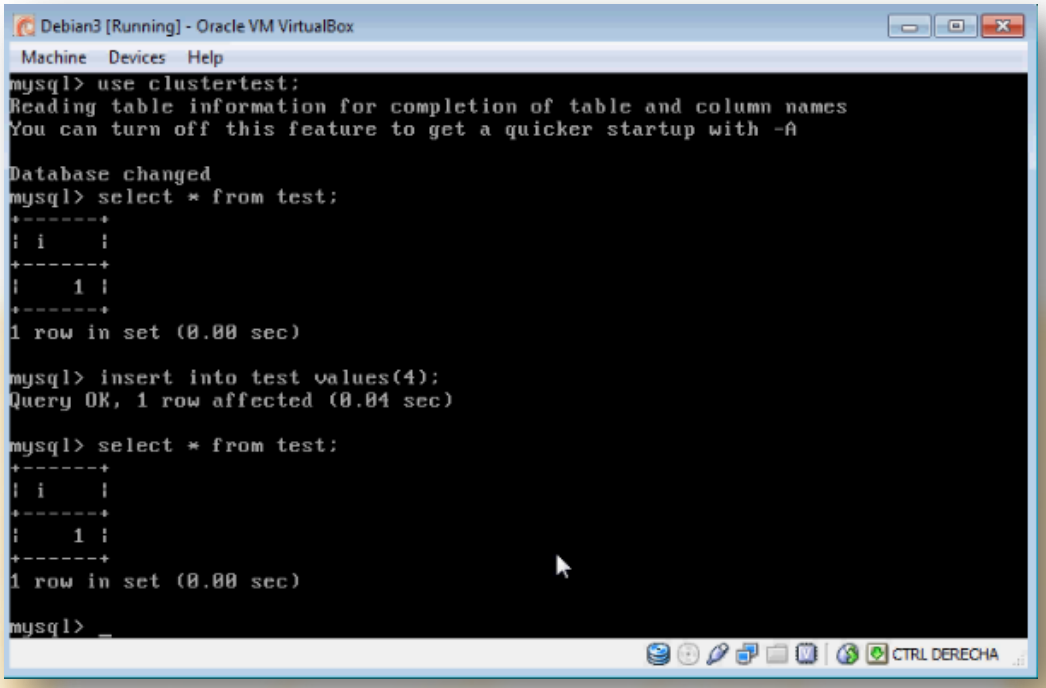

Fig. 20. Replicated again.

Proposed exercises

1. If you click on VirtualBox network icon (on the bottom of each virtual machine status bar) you can view network interfaces and disconnect and reconnect whenever you want. Disconnect a virtual machine and try to determine what happens with replicated data if both replicas continue modifying the replicated data. First disconnect cluster manager, and after testing it, disconnect the first storage node and finally the second storage node. If cluster lost its integrity regenerate the replicated table. If a data node cannot reconnect then restart the services properly (as it has been done in this document).
2. Start transactions on storage nodes and try to analyze what kind of transactional model it follows.

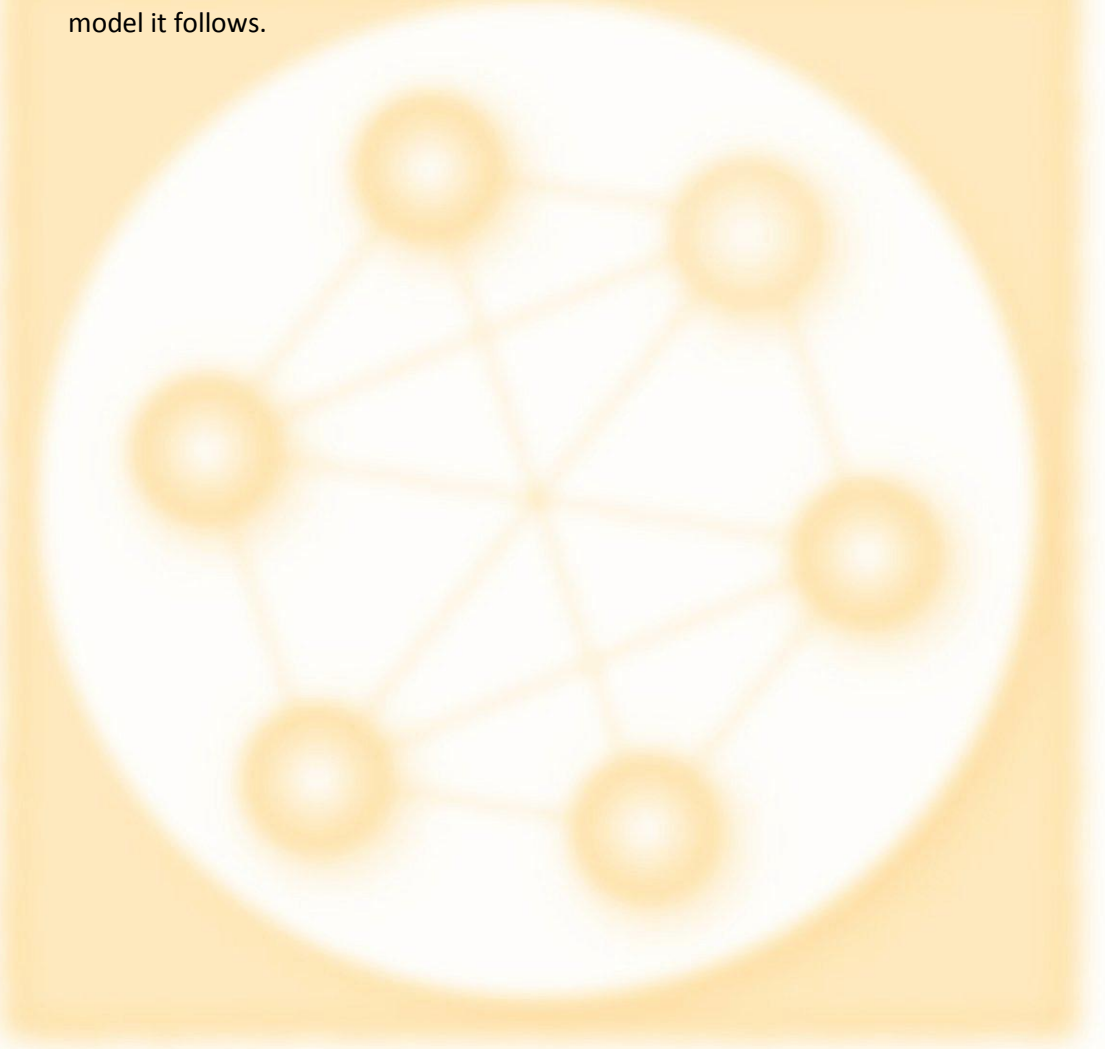

Figure Index

Fig. 1. ndb_mgmd.cnf 1
Fig. 2. ndb_mgmd.cnf content 2
Fig. 3. ndb_mgmd.cnf edited. 3
Fig. 4. Starting the second machine. 3
Fig. 5. Login 4
Fig. 6. Creating backup dir. 4
Fig. 7. my.cnf 5
Fig. 8. ndbcluster on my.cnf 5
Fig. 9. ndb-connectstring on my.cnf 6
Fig. 10. Starting the third machine. 6
Fig. 11. Restarting service mysql-ndb-mgm 7
Fig. 12. ndb_mgm accepting connections 7
Fig. 13. Restarting mysql and mysql-ndb services 8
Fig. 14. Connected. 8
Fig. 15. Clustertest on second machine. 9
Fig. 16. Test table. 10
Fig. 17. Clustertest on third machine. 10
Fig. 18. Test table replicated. 11
Fig. 19. Inserting and deleting on third machine. 11
Fig. 20. Replicated again 12