

MySQL-Cluster

Version 0.1

22.06.2006

Korbinian Pauli

<http://www.korbinian-pauli.de>

Inhaltsverzeichnis

Hintergrundwissen.....	1
Problemstellung.....	1
Cluster einrichten.....	2
ndbd-Nodes.....	2
mysqld-Nodes.....	2
Managment-Node.....	2
Backup und Restore.....	4
Cluster erweitern.....	5

Es gibt bereits eine Vielzahl von Howtos, wie man schnell einen MySQL-Cluster installiert. Von daher soll das Hauptaugenmerk dieses Artikels darin liegen, wie man einen Cluster pflegt, erweitert, etc. Für die Tests kam `mysql-5.1.11-beta-linux-i686-glibc23.tar.gz` zum Einsatz auf einem Debian Sarge.

1. Hintergrundwissen

(Zeichnungen)

2. Problemstellung

Es soll erstmal ein ein kleiner MySQL-Cluster bestehend aus 5 Servern aufgebaut werden:

- 1 Managment-Server
- 2 mysqld-Server
- 2 Storage-Server (ndbd)

Hier zur Veranschaulichung nochmal die Übersicht mit IP-Adressen:

- 192.168.1.1: Managment-Server
- 192.168.1.11: mysqld-Node
- 192.168.1.12: mysqld-Node
- 192.168.1.21: ndbd-Node
- 192.168.1.22: ndbd-Node

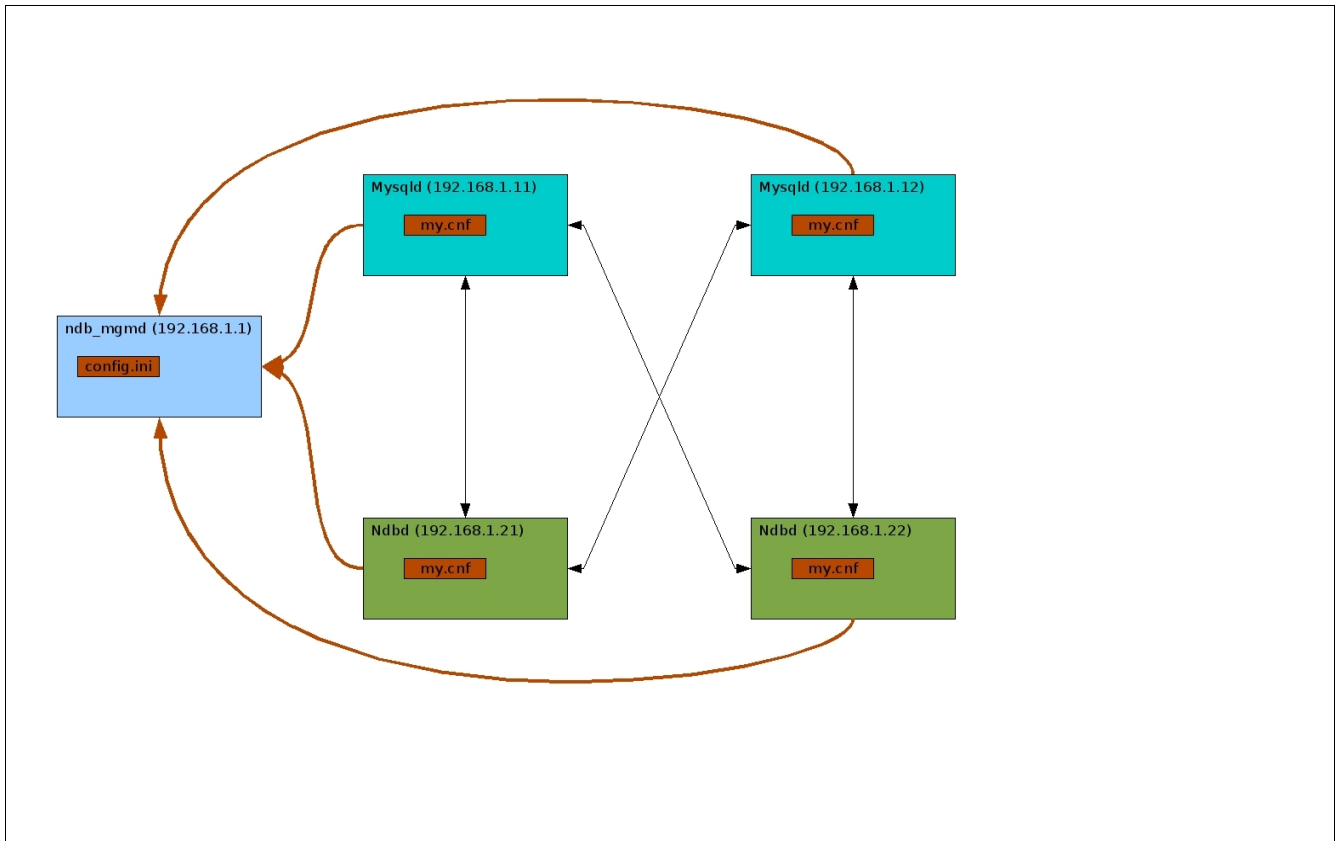


Abbildung 1: mysql-cluster einfach

Die Anzahl der Replicas, also die Anzahl der “gespiegelten” Daten ist in diesem Fall zwei. Später werde ich darauf eingehen, wie man weitere Replicas hinzufügt, um die Ausfallsicherheit noch weiter zu erhöhen. In wie weit das sinnvoll ist, soll hier nicht diskutiert werden. Der Administrator muss das für jeden Fall einzeln entscheiden.

3. Cluster einrichten

Hier also nochmal in Kürze, wie man die einzelnen ndbd-, mysqld- und ndb_mgmd-Nodes einrichtet.

3.1 ndbd-Nodes konfigurieren

Folgendes muss auf jedem ndbd-Node gemacht werden:

```
tar xvfz mysql-5.1.11-beta-linux-i686-glibc23.tar.gz
mv mysql-5.1.11-beta-linux-i686-glibc23 /usr/local/mysql
mkdir /var/lib/mysql-cluster
cd /usr/local/mysql
groupadd mysql
useradd -g mysql mysql
scripts/mysql_install_db --user=mysql
vi /etc/my.cnf
```

Die my.cnf sollte so aussehen:

```
[mysqld]
ndbcluster ndb-connectstring='host=192.168.1.1 # IP address of the management server
[mysql_cluster]
ndb-connectstring='host=192.168.1.1 # IP address of the management server
```

3.2 mysqld-Nodes konfigurieren

Folgendes muss auf jedem mysqld-Node gemacht werden:

```
tar xvfz mysql-5.1.11-beta-linux-i686-glibc23.tar.gz
mv mysql-5.1.11-beta-linux-i686-glibc23 /usr/local/mysql
mkdir /var/lib/mysql-cluster
cd /usr/local/mysql
groupadd mysql
useradd -g mysql mysql
scripts/mysql_install_db --user=mysql
cp support-files/mysql.server /etc/init.d/mysql.server
vi /etc/my.cnf
```

Die my.cnf sollte so aussehen:

```
[mysqld]
ndbcluster ndb-connectstring='host=192.168.1.1 # IP address of the management server
[mysql_cluster]
ndb-connectstring='host=192.168.1.1 # IP address of the management server
```

3.3 Management-Node konfigurieren

Zwar braucht man auf dem Management-Node nur die zwei Files `ndb_mgmd` und `ndb_mgm`, trotzdem ist es anzuraten, die ganze mysql-Distribution zu entpacken, da man später auch auf dem Management-Server noch eine mysqld-Instanz laufen lassen könnte, da die Hardware sonst sehr unterfordert ist.

Die config.ini im Verzeichnis `/var/lib/mysql-cluster` auf dem Management-Server:

```
[NDBD DEFAULT]
NoOfReplicas=2
DataDir= /var/lib/mysql-cluster
[MYSQLD DEFAULT]
[NDB_MGMD DEFAULT]
DataDir=/var/lib/mysql-cluster
[TCP DEFAULT]
[NDB_MGMD]
HostName=192.168.1.1
[NDBD]
HostName=192.168.1.21
[NDBD] HostName=192.168.1.22
[mysqld]
[mysqld]
```

3.4 MySQL-Cluster starten

Nun sind wir soweit den Management-Daemon zu starten. Dies wird mit dem Befehl `/usr/local/mysql/bin/ndb_mgmd -f /var/lib/mysql-cluster/config.ini` gemacht. Ob der Management-Daemon tatsächlich läuft überprüfen wir mit dem Befehl `ps ax|grep ndb_mgmd`. Läuft der Daemon können wir uns nun mit `/usr/local/mysql/bin/ndb_mgm` die Management-Console starten

```

- NDB Cluster - Management Client -
ndb_mgm> show
Connected to Management Server at: 192.168.1.1:1186
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=2 (not connected, accepting connect from 192.168.1.21)
id=3 (not connected, accepting connect from 192.168.1.22)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @192.168.1.1 (Version: 5.1.11)

[mysqld(API)] 2 node(s)
id=4 (not connected, accepting connect from any host)
id=5 (not connected, accepting connect from any host)

```

Mit dem Kommando `show` wird der Status des Clusters angezeigt. Wir sehen, dass nur der Management-Daemon aktiv ist und dieser auf Verbindungen von den zwei `ndb`-Daemons sowie auf zwei `mysqld`-Daemons wartet.

Da wir in der `config.ini` unter den Einträgen für die `mysqld` keine IP-Adressen angegeben haben, kann sich jeder x-beliebige `mysqld` Rechner im Netzwerk verbinden.

Jetzt können wir auf dem ersten Storage-Node (192.168.1.21) mit dem Befehl `/usr/local/mysql/bin/ndbd` einen `ndb`-Daemon starten. Auch hier sollte wieder kontrolliert werden, ob der Daemon wirklich läuft (`ps axfgrep ndbd`).

In der Management-Konsole schauen wir nun nach, ob sich der `ndbd` richtig beim `ndb_mgmd` angemeldet hat. Ich habe die Erfahrung gemacht, dass es immer ein wenig dauern kann, bis die Anzeige aktualisiert wird.

```

ndb_mgm> show
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=2 @192.168.1.21 (Version: 5.1.11, starting, Nodegroup: 0)
id=3 (not connected, accepting connect from 192.168.1.22)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @192.168.1.1 (Version: 5.1.11)

[mysqld(API)] 2 node(s)
id=4 (not connected, accepting connect from any host)
id=5 (not connected, accepting connect from any host)

```

Ist alles OK, starten wir den `ndbd` auch auf dem zweiten `ndb`-Node (192.168.1.22), wieder mit `/usr/local/mysql/bin/ndbd`

```

ndb_mgm> show
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=2 @192.168.1.21 (Version: 5.1.11, Nodegroup: 0, Master)
id=3 @192.168.1.22 (Version: 5.1.11, Nodegroup: 0)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @192.168.1.1 (Version: 5.1.11)

[mysqld(API)] 2 node(s)
id=4 (not connected, accepting connect from any host)
id=5 (not connected, accepting connect from any host)

```

Was sagt uns nun die Ausgabe?

Unser Storage-Cluster besteht aus 2 Nodes die zur Nodegroup 0 gehören, also die selben Daten enthalten.

Später werde ich noch darauf eingehen, wie man weitere Nodegroups hinzufügt, um die Speicherkapazität des Clusters zu erhöhen. Aber alles der Reihe nach.

Noch können wir keine SQL-Befehle absetzen, da noch kein mysqld läuft. Diesen starten wir auf dem ersten mysql-Node (192.168.1.11) mit dem Befehl `/etc/init.d/mysql.server start` und kontrollieren mit `ps axfgrep mysqld`, ob der Daemon richtig gestartet wurde. In der Management-Konsole muss die Ausgabe nun wie etwa folgt aussehen:

```
ndb_mgm> show
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=2 @192.168.1.21 (Version: 5.1.11, Nodegroup: 0, Master)
id=3 @192.168.1.22 (Version: 5.1.11, Nodegroup: 0)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @192.168.1.1 (Version: 5.1.11)

[mysqld(API)] 2 node(s)
id=4 @192.168.1.11 (Version: 5.1.11)
id=5 (not connected, accepting connect from any host)
```

Starten wir nun auch noch den zweiten mysqld (192.168.1.12). Damit steht nun der komplette Cluster und wir können mit ein paar kleinen Tests beginnen.

4. Backup und Restore

x

5. Cluster erweitern

Nun wird es interessant. Das bisherige Vorgehen findet man in beliebig vielen Howtos in ähnlicher Weise zu hauf.

5.1 Weitere Replicas hinzufügen