

# MySQL 集群搭建详解

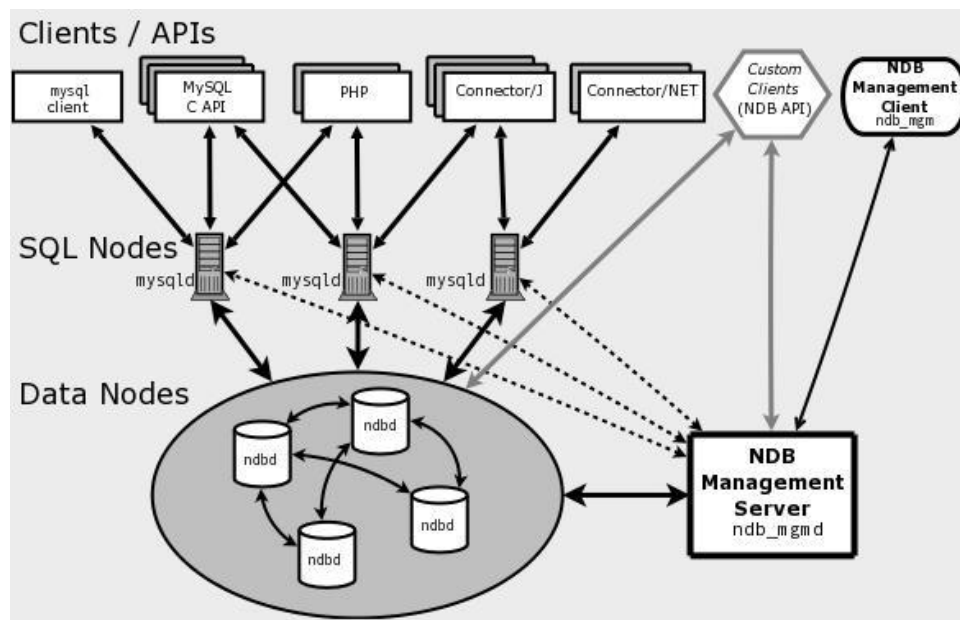
本文将搭建一个最简化的 MySQL Cluster 系统，配置方法中的所有命令都是以 root 账户运行。

## 概述

MySQL Cluster 是 MySQL 适合于分布式计算环境的高实用、可拓展、高性能、高冗余版本，其研发设计的初衷就是要满足许多行业里的最严酷应用要求，这些应用中经常要求数据库运行的可靠性要达到 99.999%。MySQL Cluster 允许在无共享的系统中部署“内存中”数据库集群，通过无共享体系结构，系统能够使用廉价的硬件，而且对软硬件无特殊要求。此外，由于每个组件有自己的内存和磁盘，不存在单点故障。

实际上，MySQL 集群是把一个叫做 NDB 的内存集群存储引擎集成与标准的 MySQL 服务器集成。它包含一组计算机，每个都跑一个或者多个进程，这可能包括一个 MySQL 服务器，一个数据节点，一个管理服务器和一个专有的一个数据访问程序。

MySQL Cluster 能够使用多种故障切换和负载平衡选项配置 NDB 存储引擎，但在 Cluster 级别上的存储引擎上做这个最简单。以下为 MySQL 集群结构关系图，



MySQL 从结构看，由 3 类节点 (计算机或进程) 组成，分别是：

## 管理节点:

用于给整个集群其他节点提供配置、管理、仲裁等功能。理论上通过一台服务器提供服务就可以了。

## 数据节点:

MySQL Cluster 的核心，存储数据、日志，提供数据的各种管理服务。2 个以上 时就能实现集群的高可用保证，DB 节点增加时，集群的处理速度会变慢。

## SQL 节点(API):

用于访问 MySQL Cluster 数据，提供对外应用服务。增加 API 节点会提高整个集群的并发访问速度和整体的吞吐量，该节点 可以部署在 Web 应用服务器上，也可以部署在专用的服务器上，也开以和 DB 部署在 同一台服务器上。

# NDB 引擎

MySQL Cluster 使用了一个专用的基于内存的存储引擎——NDB 引擎，这样做的好处是速度快， 没有磁盘 I/O 的瓶颈，但是由于是基于内存的，所以数据库的规模受系统总内存的限制， 如果运行 NDB 的 MySQL 服务器一定要内存够大，比如 4G，8G，甚至 16G。NDB 引擎是分布式的，它可以配置在多台服务器上来实现数据的可靠性和扩展性，理论上 通过配置 2 台 NDB 的存储节点就能实现整个数据库集群的冗余性和解决单点故障问题。

### 缺陷

- 基于内存，数据库的规模受集群总内存的大小限制
- 基于内存，断电后数据可能会有数据丢失，这点还需要通过测试验证。
- 多个节点通过网络实现通讯和数据同步、查询等操作，因此整体性受网络速度影响，因此速度也比较慢

## 2.2 优点

- 多个节点之间可以分布在不同的地理位置，因此也是一个实现分布式数据库的方案。
- 扩展性很好，增加节点即可实现数据库集群的扩展。
- 冗余性很好，多个节点上都有完整的数据库数据，因此任何一个节点宕机都不会造成服务中断。

实现高可用性的成本比较低，不象传统的高可用方案一样需要共享的存储设备和专用的软件才能实现，NDB 只要有足够的内存就能实现。

本文将搭建一个最简化的 MySQL Cluster 系统，配置方法中的所有命令都是以 root 账户运行。这个 MySQL Cluster 包含一个管理结点、两个数据结点、两个 SQL 结点，这五个结点会分别安装在五个虚拟机上，虚拟机的名称和 IP 如下所示：

管理结点	mysql-mgm	192.168.124.141
数据结点 1	mysql-ndbd-1	192.168.124.142
数据结点 2	mysql-ndbd-2	192.168.124.143
SQL 结点1	mysql-sql-1	192.168.124.144
SQL 结点2	mysql-sql-2	192.168.124.145

### 一、公共配置

请在三个虚拟机上分别配置此处的配置项。

#### 1. 安装虚拟机

虚拟机操作系统安装 CentOS 6.4 的 x86\_64 版本，使用 NAT 网络，并且还要安装 vmware-tools，具体安装方法此处不详述。

#### 2. 拷贝 mysql cluster

下载以下版本的 MySQL-Cluster：

`http://cdn.mysql.com/Downloads/MySQL-Cluster-7.3/mysql-cluster-gpl-7.3.4-linux-glibc2.5-x86_64.tar.gz`

下载得到的压缩包拷贝至虚拟机的/root/Downloads 目录，然后在 shell 中运行以下命令：

1. `cd /root/Downloads`
- 2.
3. `tar -xvzf mysql-cluster-gpl-7.3.4-linux-glibc2.5-x86_64.tar.gz`
- 4.
5. `mv mysql-cluster-gpl-7.3.4-linux-glibc2.5-x86_64 /usr/local/mysql`

### 3. 关闭安全策略

关闭 iptables 防火墙(或者打开防火墙的 1186、3306 端口), 在 Shell 中运行以下命令:

1. `chkconfig --level 35 iptables off`

关闭 SELinux, 在 Shell 中运行以下命令:

1. `gedit /etc/selinux/config`

将 config 文件中的 SELINUX 项改为 disabled, 修改后的 config 文件的内容如下:

1. `# This file controls the state of SELinux on the system.`
- 2.
3. `# SELINUX= can take one of these three values:`
- 4.
5. `# enforcing - SELinux security policy is enforced.`
- 6.
7. `# permissive - SELinux prints warnings instead of enforcing.`
- 8.
9. `# disabled - No SELinux policy is loaded.`
- 10.
11. `SELINUX=disabled`
- 12.
13. `# SELINUXTYPE= can take one of these two values:`
- 14.
15. `# targeted - Targeted processes are protected,`
- 16.
17. `# mls - Multi Level Security protection.`
- 18.
19. `SELINUXTYPE=targeted`

最后重启系统

## 二、配置管理结点(192.168.124.141)

### 1. 配置 config.ini 配置文件

在 shell 中运行以下命令：

1. `mkdir /var/lib/mysql-cluster`
2.
3. `cd /var/lib/mysql-cluster`
4.
5. `gedit config.ini`

配置文件 config.ini 内容如下：

1. `[ndbd default]`
2.
3. `NoOfReplicas=2`
4.
5. `DataMemory=80M`
6.
7. `IndexMemory=18M`
8.
9. `[ndb_mgmd]`
10.
11. `NodeId=1`
12.
13. `hostname=192.168.124.141`
14.
15. `datadir=/var/lib/mysql-cluster`
16.
17. `[ndbd]`
18.
19. `NodeId=2`
20.
21. `hostname=192.168.124.142`
22.
23. `datadir=/usr/local/mysql/data`
24.
25. `[ndbd]`
26.
27. `NodeId=3`

```
28.
29.     hostname=192.168.124.143
30.
31.     datadir=/usr/local/mysql/data
32.
33.     [mysqld]
34.
35.     NodeId=4
36.
37.     hostname=192.168.124.144
38.
39.     [mysqld]
40.
41.     NodeId=5
42.
43.     hostname=192.168.124.145
```

## 2. 安装管理结点

安装管理节点,不需要 mysqld 二进制文件,只需要 MySQL Cluster 服务端程序 (ndb\_mgmd) 和监听客户端程序 (ndb\_mgm)。在 shell 中运行以下命令:

```
1.     cp /usr/local/mysql/bin/ndb_mgm* /usr/local/bin
2.
3.     cd /usr/local/bin
4.
5.     chmod +x ndb_mgm*
```

## 三、配置数据结点(192. 168. 124. 142、192. 168. 124. 143)

### 1. 添加 mysql 组和用户

在 shell 中运行以下命令:

```
1.     groupadd mysql
2.
3.     useradd -g mysql mysql
```

### 2. 配置 my.cnf 配置文件

在 shell 中运行以下命令:

1. `gedit /etc/my.cnf`

配置文件 my.cnf 的内容如下:

```
1. [mysqld]
2.
3. basedir=/usr/local/mysql
4.
5. datadir=/usr/local/mysql/data
6.
7. socket=/usr/local/mysql/sock/mysql.sock
8.
9. user=mysql
10.
11. # Disabling symbolic-links is recommended to prevent assorted security
    risks
12.
13. symbolic-links=0
14.
15. [mysqld_safe]
16.
17. log-error=/var/log/mysql.log
18.
19. pid-file=/var/run/mysql/mysql.pid
20.
21. [mysql_cluster]
22.
23. ndb-connectstring=192.168.124.141
```

### 3. 创建系统数据库

在 shell 中运行以下命令:

```
1. cd /usr/local/mysql
2.
3. mkdir sock
4.
5. scripts/mysql_install_db --user=mysql --basedir=/usr/local/mysql --datadir=/usr
    /local/mysql/data
```

### 4. 设置数据目录

在 shell 中运行以下命令:

1. `chown -R root .`
- 2.
3. `chown -R mysql:mysql /usr/local/mysql/data`
- 4.
5. `chown -R mysql:mysql /usr/local/mysql/sock`
- 6.
7. `chgrp -R mysql .`

## 5. 配置 MySQL 服务

在 shell 中运行以下命令：

1. `cp support-files/mysql.server /etc/rc.d/init.d/`
- 2.
3. `chmod +x /etc/rc.d/init.d/mysql.server`
- 4.
5. `chkconfig --add mysql.server`

## 四、配置 SQL 结点(192.168.124.144、192.168.124.145)

### 1. 添加 mysql 组 and 用户

在 shell 中运行以下命令：

1. `groupadd mysql`
- 2.
3. `useradd -g mysql mysql`

### 2. 配置 my.cnf 配置文件

在 shell 中运行以下命令：

```
gedit /etc/my.cnf
```

配置文件 my.cnf 的内容如下：

1. `[client]`
- 2.
3. `socket=/usr/local/mysql/sock/mysql.sock`
- 4.
5. `[mysqld]`
- 6.



```
7.     ndbcluster
8.
9.     datadir=/usr/local/mysql/data
10.
11.    socket=/usr/local/mysql/sock/mysql.sock
12.
13.    ndb-connectstring=192.168.124.141
14.
15.    [mysql_cluster]
16.
17.    ndb-connectstring=192.168.124.141
```

### 3. 创建系统数据库

在 shell 中运行以下命令：

```
1.     cd /usr/local/mysql
2.
3.     mkdir sock
4.
5.     scripts/mysql_install_db --user=mysql --basedir=/usr/local/mysql --datadir=/usr/local/mysql/data
```

### 4. 设置数据目录

在 shell 中运行以下命令：

```
1.     chown -R root .
2.
3.     chown -R mysql:mysql /usr/local/mysql/data
4.
5.     chown -R mysql:mysql /usr/local/mysql/sock
6.
7.     chgrp -R mysql .
```

### 5. 配置 MySQL 服务

在 shell 中运行以下命令：

```
1.     cp support-files/mysql.server /etc/rc.d/init.d/
2.
3.     chmod +x /etc/rc.d/init.d/mysql.server
```

- 4.
5. `chkconfig --add mysql.server`

## 五、Cluster 环境启动

注意启动顺序：首先是管理节点，然后是数据节点，最后是 SQL 节点。

### 1. 启动管理结点

在 shell 中运行以下命令：

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

还可以使用 `ndb_mgm` 来监听客户端，如下：

```
ndb_mgm
```

### 2. 启动数据结点

首次启动，则需要添加 `--initial` 参数，以便进行 NDB 节点的初始化工作。在以后的启动过程中，则是不能添加该参数的，否则 `ndbd` 程序会清除在之前建立的所有用于恢复的数据文件和日志文件。

1. `/usr/local/mysql/bin/ndbd --initial`

如果不是首次启动，则执行下面的命令。

1. `/usr/local/mysql/bin/ndbd`

### 3. 启动 SQL 结点

若 MySQL 服务没有运行，则在 shell 中运行以下命令：

1. `/usr/local/mysql/bin/mysqld_safe --user=mysql &`

### 4. 启动测试

查看管理节点，启动成功

## 六、集群测试

## 1. 测试一

现在我们在其中一个 SQL 结点上进行相关数据库的创建, 然后到另外一个 SQL 结点上看看数据是否同步。

在 SQL 结点 1(192.168.124.144)上执行:

```
1. shell> /usr/local/mysql/bin/mysql -u root -p
2.
3. mysql>show databases;
4.
5. mysql>create database aa;
6.
7. mysql>use aa;
8.
9. mysql>CREATE TABLE ctest2 (i INT) ENGINE=NDB; //这里必须指定数据库表的引擎
   为 NDB, 否则同步失败
10.
11. mysql> INSERT INTO ctest2 () VALUES (1);
12.
13. mysql> SELECT * FROM ctest2;
```

然后在 SQL 结点 2 上看数据是否同步过来了

经过测试, 在非 master 上创建数据, 可以同步到 master 上

查看表的引擎是不是 NDB, >show create table 表名;

## 2. 测试二

关闭一个数据节点, 在另外一个节点写输入, 开启关闭的节点, 看数据是否同步过来。

首先把数据结点 1 重启, 然后在结点 2 上添加数据

在 SQL 结点 2(192.168.124.145)上操作如下:

```
1. mysql> create database bb;
2.
3. mysql> use bb;
4.
5. mysql> CREATE TABLE ctest3 (i INT) ENGINE=NDB;
```

```

6.
7.      mysql> use aa;
8.
9.      mysql> INSERT INTO ctest2 () VALUES (3333);
10.
11.     mysql> SELECT * FROM ctest2;

```

等数据结点 1 启动完毕，启动数据结点 1 的服务

```

1.      #/usr/local/mysql/bin/ndbd --initial#service mysqld start

```

然后登录进去查看数据

```

1.      # /usr/local/mysql/bin/mysql -u root -p

```

可以看到数据已经同步过来了，说明数据可以双向同步了。

## 七、关闭集群

1. 关闭管理节点和数据节点，只需要在管理节点 (ClusterMgm--134) 里执行：

```

1.      shell> /usr/local/mysql/bin/ndb_mgm -e shutdown

```

显示

```

1.      Connected to Management Server at: localhost:1186
2.
3.      2 NDB Cluster node(s) have shutdown.
4.
5.      Disconnecting to allow management server to shutdown.

```

2. 然后关闭 Sql 节点 (135, 136)，分别在 2 个节点里运行：

```

shell> /etc/init.d/mysql.server stop

```

```

Shutting down MySQL... SUCCESS!

```

注意：要再次启动集群，就按照第五部分的启动步骤即可，不过这次启动数据节点的时候就不要加“-initial”参数了。