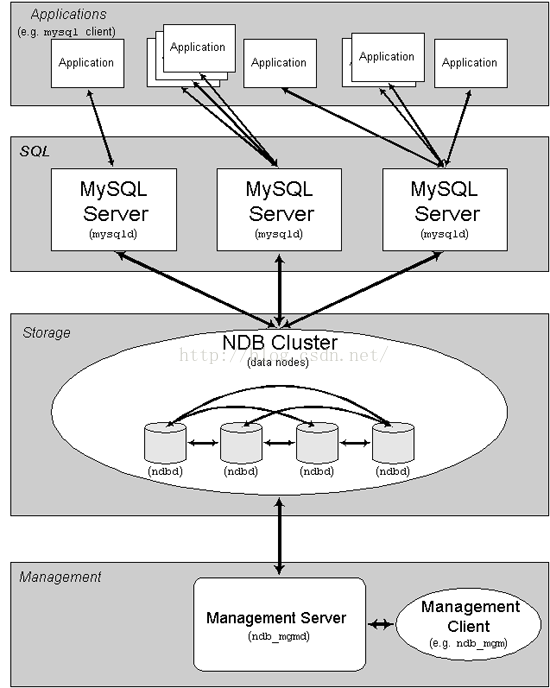
参考网站：<https://blog.csdn.net/c_fangfujiang/article/details/51332328>

MySQL簇是一种技术，该技术允许在无共享的系统中部署“内存中”数据库的簇。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。

MySQL簇将标准的MySQL服务器与名为NDB的“内存中”簇式存储引擎集成了起来。在我们的文档中，术语NDB指的是与存储引擎相关的设置部分，而术语“MySQL簇”指的是MySQL和NDB存储引擎的组合。

MySQL簇由一组计算机构成，每台计算机上均运行着多种进程，包括MySQL服务器，NDB簇的数据节点，管理服务器，以及（可能）专门的数据访问程序。关于簇中这些组件的关系，请参见下图：



所有这些程序一起构成了MySQL簇。将数据保存到NDB簇存储引擎中时，表将保存在数据节点内。能够从簇中所有其他MySQL服务器直接访问这些表。因此，在将数据保存在簇内的工资表应用程序中，如果某一应用程序更新了1位雇员的工资，所有查询该数据的其他MySQL服务器能立刻发现这种变化。

对于MySQL簇，保存在数据节点内的数据可被映射，簇能够处理单独数据节点的故障，除了少数事务将因事务状态丢失而被放弃外，不会产生其他影响。由于事务性应用程序能够处理事务失败事宜，因而它不是问题源。

通过将MySQL簇引入开放源码世界，MySQL为所有需要它的人员提供了具有高可用性、高性能和可缩放性的簇数据管理。

MySQL簇的基本概念

NDB是一种“内存中”存储引擎，它具有可用性高和数据一致性好的特点。

能够使用多种故障切换和负载平衡选项配置NDB存储引擎，但以簇层面上的存储引擎开始最简单。MySQL簇的NDB存储引擎包含完整的数据集，仅取决于簇本身内的其他数据。

下面，我们介绍了设置由NDB存储引擎和一些MySQL服务器构成的MySQL簇的设置方法。

目前，MySQL簇的簇部分可独立于MySQL服务器进行配置。在MySQL簇中，簇的每个部分被视为1个节点。

注释：在很多情况下，术语“节点”用于指计算机，但在讨论MySQL簇时，它表示的是进程。在单台计算机上可以有任意数目的节点，为此，我们采用术语簇主机。

有三类簇节点，在最低的MySQL簇配置中，至少有三个节点，这三类节点分别是：

管理(MGM)节点：这类节点的作用是管理MySQL簇内的其他节点，如提供配置数据、启动并停止节点、运行备份等。由于这类节点负责管理其他节点的配置，应在启动其他节点之前首先启动这类节点。MGM节点是用命令ndb\_mgmd启动的。

数据节点：这类节点用于保存簇的数据。数据节点的数目与副本的数目相关，是片段的倍数。例如，对于两个副本，每个副本有两个片段，那么就有4个数据节点。没有必要有一个以上的副本。数据节点是用命令ndbd启动的。

SQL节点：这是用来访问簇数据的节点。对于MySQL簇，客户端节点是使用NDB簇存储引擎的传统MySQL服务器。典型情况下，SQL节点是使用命令mysqld –ndbcluster启动的，或将ndbcluster添加到my.cnf后使用mysqld启动。

簇配置包括对簇中单独节点的配置，以及设置节点之间的单独通信链路。对于目前设计的MySQL簇，其意图在于，从处理器的能力、内存空间和带宽来讲，存储节点是同质的，此外，为了提供单一的配置点，作为整体，簇的所有配置数据均位于1个配置文件中。

管理服务器(MGM节点)负责管理簇配置文件和簇日志。簇中的每个节点从管理服务器检索配置数据，并请求确定管理服务器所在位置的方式。当数据节点内出现有趣的事件时，节点将关于这类事件的信息传输到管理服务器，然后，将这类信息写入簇日志。

此外，可以有任意数目的簇客户端进程或应用程序。它们分为两种类型：

标准MySQL客户端：对于MySQL簇，它们与标准的（非簇类）MySQL没有区别。换句话讲，能够从用PHP、Perl、C、C++、Java、Python、Ruby等编写的现有MySQL应用程序访问MySQL簇。

管理客户端：这类客户端与管理服务器相连，并提供了优雅地启动和停止节点、启动和停止消息跟踪（仅对调试版本）、显示节点版本和状态、启动和停止备份等的命令。

**搭建sql节点(4个),data节点(4个)，manage节点(1个)步骤**

//环境介绍

1、manage节点：10.10.54.154

2、sql/data节点：10.10.54.154/155/156/157

**//编译安装cmake2.8**

[154/155/156/157]

1.下载mysql-cluster cmake

shell> wget http://dev.mysql.com/get/Downloads/MySQL-Cluster-7.2/mysql-cluster-gpl-7.2.15-linux2.6-x86\_64.tar.gz

shell> wget http://www.cmake.org/files/v2.8/cmake-2.8.12.2.tar.gz

2.编译安装cmake2.8---(154/155/156/157)

shell> tar xvf cmake-2.8.12.2.tar.gz

shell> cd cmake-2.8.12.2

shell> ./configure

shell> make

shell> make install

3.编译安装mysql-cluster---(154/155/156/157)

shell> tar xvf mysql-cluster-gpl-7.2.15.tar.gz

shell> cd mysql-cluster-gpl-7.2.15

shell> cmake -DCMAKE\_INSTALL\_PREFIX=/usr/local/mysql \

-DMYSQL\_UNIX\_ADDR=/usr/local/mysql/tmp/mysql-cluster.sock \

-DDEFAULT\_CHARSET=utf8 \

-DDEFAULT\_COLLATION=utf8\_general\_ci \

-DEXTRA\_CHARSETS=all \

-DWITH\_EMBEDDED\_SERVER=0 \

-DWITH\_NDB\_JAVA=OFF \

-DWITH\_MYISAM\_STORAGE\_ENGINE=1 \

-DWITH\_INNOBASE\_STORAGE\_ENGINE=1 \

-DWITH\_MEMORY\_STORAGE\_ENGINE=1 \

-DWITH\_BLACKHOLE\_STORAGE\_ENGINE=1 \

-DWITH\_FEDERATED\_STORAGE\_ENGINE=1 \

-DWITH\_PARTITION\_STORAGE\_ENGINE=1 \

-DWITH\_NDBCLUSTER\_STORAGE\_ENGINE=1 \

-DENABLED\_LOCAL\_INFILE=1 \

-DMYSQL\_USER=mysql \

-DWITH\_DEBUG=0 \

-DWITH\_SSL=yes

4.shell> make && make install

**//mysql-cluster配置---(154/155/156/157)**

shell> mkdir /data/ndb

shell> cd /home/mysql-cluster-gpl-7.2.15

shell> cp support-files/mysql.server /etc/init.d/mysqld

shell> cp support-files/my-large.cnf /etc/my.cnf

shell> chown mysql.mysql /usr/local/mysql -R

shell> chmod 755 /usr/local/mysql/scripts/\*

shell> chmod 755 /etc/init.d/mysqld

shell> sed -i "/

mysqld

/a datadir=/data/ndb" /etc/my.cnf

shell> /usr/local/mysql/scripts/mysql\_install\_db --user=root --basedir=/usr/local/mysql --datadir=/data/ndb

shell> chown mysql:mysql -R /data/ndb

//data/sql节点配置---(154/155/156/157)

shell> cd /usr/local/mysql/bin/

shell> cp ndb\_mgm\* /usr/local/bin/

shell> vim /etc/my.cnf

-------------------------

[mysql\_cluster]

ndb-connectstring=10.10.54.154

[mysqld]

datadir=/data/ndb

ndbcluster

ndb-connectstring=10.10.54.154

-------------------------

//管理节点154配置

shell> mkdir -p /etc/ndb/config.ini

shell> vim /etc/ndb/config.ini

--------------------------------------

[NDBD DEFAULT]

NoOfReplicas=2

DataMemory=128M

IndexMemory=64M

[TCP DEFAULT]

portnumber=2202

[NDB\_MGMD]

NodeId=1

hostname=10.10.54.154

datadir=/var/lib/mysql-cluster

[NDBD]

NodeId=2

hostname=10.10.54.154

datadir=/data/ndb

BackupDataDir=/data/backup

[NDBD]

NodeId=3

hostname=10.10.54.155

datadir=/data/ndb

BackupDataDir=/data/backup

[NDBD]

NodeId=4

hostname=10.10.54.156

datadir=/data/ndb

BackupDataDir=/data/backup

[NDBD]

NodeId=5

hostname=10.10.54.157

datadir=/data/ndb

BackupDataDir=/data/backup

[MYSQLD]

NodeId=7

hostname=10.10.54.154

[MYSQLD]

NodeId=8

hostname=10.10.54.155

[MYSQLD]

NodeId=9

hostname=10.10.54.156

[MYSQLD]

NodeId=10

hostname=10.10.54.157

[MYSQLD]

NodeId=11

[MYSQLD]

NodeId=12

[MYSQLD]

NodeId=13

[MYSQLD]

NodeId=14

------------------------------------

//启动mysql-cluster

---启动顺序management node--data node---sql node

#启动管理节点

shell> ndb\_mgmd -f /var/lib/mysql-cluster/config.ini --initial

 ##--initial参数第一次启动时加上

#启动数据节点

shell> ndbd       #--initial ##初始化数据库，用在备份,恢复时

2014-02-27 18:14:23 [ndbd] INFO     -- Angel connected to '10.10.54.154:1186'

2014-02-27 18:14:23 [ndbd] INFO     -- Angel allocated nodeid: 2

#启动sql节点

shell> /etc/init.d/mysqld start

//查看mysql集群状态

shell> ndb\_mgm

ndb\_mgm> show

Cluster Configuration

---------------------

[ndbd(NDB)] 4 node(s)

id=2 @10.10.54.154  (mysql-5.5.35 ndb-7.2.15, Nodegroup: 0, \*)

id=3 @10.10.54.155  (mysql-5.5.35 ndb-7.2.15, Nodegroup: 0)

id=4 @10.10.54.156  (mysql-5.5.35 ndb-7.2.15, Nodegroup: 1)

id=5 @10.10.54.157  (mysql-5.5.35 ndb-7.2.15, Nodegroup: 1)

[ndb\_mgmd(MGM)] 1 node(s)

id=1 @10.10.54.154  (mysql-5.5.35 ndb-7.2.15)

[mysqld(API)] 8 node(s)

id=7 @10.10.54.154  (mysql-5.5.35 ndb-7.2.15)

id=9 @10.10.54.155  (mysql-5.5.35 ndb-7.2.15)

id=10 @10.10.54.156  (mysql-5.5.35 ndb-7.2.15)

id=11 @10.10.54.157  (mysql-5.5.35 ndb-7.2.15)

id=12 (not connected, accepting connect from any host)

id=13 (not connected, accepting connect from any host)

id=14 (not connected, accepting connect from any host)

id=15 (not connected, accepting connect from any host)

//测试mysql-cluster

1.管理机上创建表

mysql> create database d1;

mysql> use d1;

mysql> CREATE TABLE `t\_e` (

  `emp\_no` int(11) NOT NULL,

  `birth\_date` date NOT NULL,

  `first\_name` varchar(14) NOT NULL,

  `last\_name` varchar(16) NOT NULL,

  `gender` enum('M','F') NOT NULL,

  `hire\_date` date NOT NULL,

  PRIMARY KEY (`emp\_no`)

) ENGINE=ndbcluster DEFAULT CHARSET=utf8

2.mysql> insert into t\_e select \* from employees.employees;

##插入出错

ERROR 1297 (HY000): Got temporary error 233 'Out of operation records in transaction coordinator (increase MaxNoOfConcurrentOperations)' from NDBCLUSTER

#原因时插入数据量太大

mysql> insert into t\_e select \* from employees.employees limit 10000;

#插入成功

3.data节点上查看，数据已同步

//关闭mysql-cluster

1.先关闭sql节点

/etc/init.d/mysqld stop

2.在管理机上

shell> ndb\_mgm

ndb\_mgm> shutdown #关闭所有cluster进程