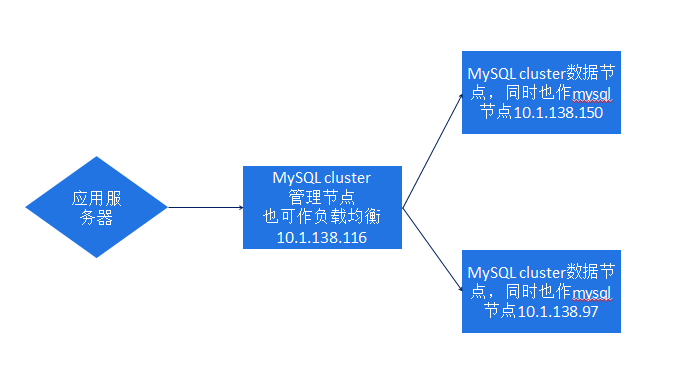
# 集群性能测试工具

1. **实验概述**

本次研讨的主题是集群性能测试工具。首先我们需要构建集群，这里我构建的是mysql集群。使用mysql官网提供的mysql-cluster实现集群。集群中应包括管理节点，数据节点和MySQL节点。管理节点负责管理、配置、监控整个集群。数据节点，使用内存存放数据，保存进数据节点的数据都会自动复制并存储到其他数据节点。mysql节点，也叫数据库节点，和我们平时使用的mysql相同，作为数据库使用。

由于条件限制我只使用了三台服务器，分别是10.1.138.116,10.1.138.97，10.1.138.150。其中116是管理节点，150和97都既是数据节点，又是mysql节点。

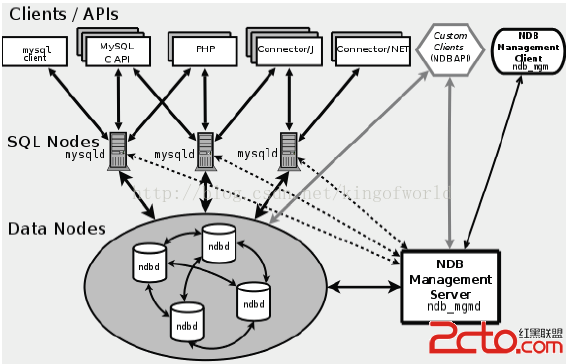
架构图及说明如图。



1. **Mysql集群概述：**

MySQL Cluster 是MySQL 官方集群部署方案，它的历史较久。支持通过自动分片支持读写扩展，通过实时备份冗余数据，是可用性最高的方案，声称可做到99.999%的可用性。

架构及实现原理：

****

缺点及限制：虽然最新的Cluster版本已经支持外键，但性能有问题（因为外键所关联的记录可能在别的分片节点中），所以建议去掉所有外键。Data Node节点数据会被尽量放在内存中，对内存要求大。

1. **Mysql集群安装实验步骤：**
2. 前期准备：保证服务器间可以互相ping通，有固定的ip地址。
3. 打开三台服务器的相关端口，MySQL默认的端口是3306， 群集管理节点的默认端口是1186， 数据节点的默认端口是2202。保证这三个端口没有其他程序占用。
4. 下载mysql cluster，由于官网比较慢，我使用的是镜像

|  |
| --- |
| Wget  <http://mirrors.sohu.com/mysql/MySQL-Cluster-7.5/> mysql-cluster-gpl-7.5.5-linux-glibc2.5-x86\_64.tar.gz |

1. 解压

# cd /usr/local

# tar xvf mysql-cluster-gpl-7.5.5-linux-glibc2.5-x86\_64.tar.gz

1. 将需要的文件取出

# cd mysql-cluster-gpl-7.5.5-linux-glibc2.5-x86\_64

# cp bin/ndb\_mgm\* /usr/local/bin

# cd /usr/local/bin

# chmod +x ndb\_mgm\*

1. 配置管理节点：新建配置文件并且初始化管理节点

# mkdir /var/lib/mysql-cluster

# mkdir /usr/local/mysql

# vi /var/lib/mysql-cluster/config.ini

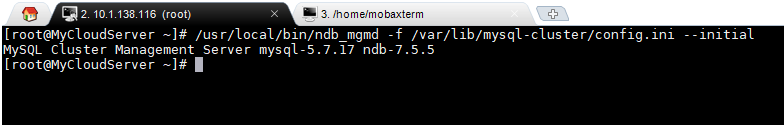
这里修改为：

|  |
| --- |
| [ndbd default]  NoOfReplicas=2  DataMemory=512M  IndexMemory=18M  [ndb\_mgmd]  HostName=10.1.138.116  DataDir=/var/lib/mysql-cluster  [ndbd]  HostName=10.1.138.97  DataDir=/var/lib/mysql-cluster  [ndbd]  HostName=10.1.138.150  DataDir=/var/lib/mysql-cluster  [mysqld]  HostName=10.1.138.97  [mysqld]  HostName=10.1.138.150 |

1. 使用配置文件初始化管理节点

# /usr/local/bin/ndb\_mgmd -f /var/lib/mysql-cluster/config.ini --initial

出现MySQL Cluster Management Server mysql-5.7.16 ndb-7.5.5即为成功。

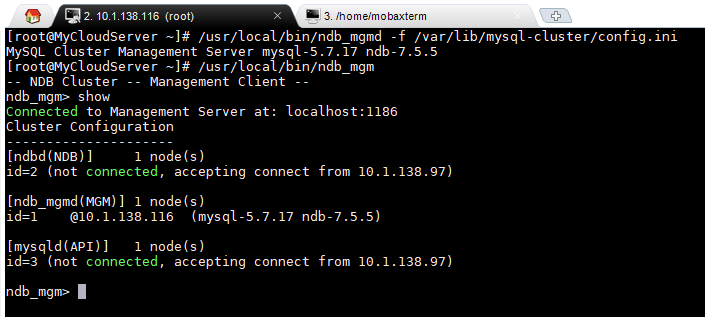


1. /usr/local/bin目录下使用ndb\_mgm命令

# /usr/local/bin/ndb\_mgm

ndb\_mgm>show

到此为止管理节点配置完毕，接下去配置数据和sql节点



1. 安装配置数据和mysql节点

新增用户组mysql和用户msyql

# groupadd mysql

useradd -g mysql -s /bin/false mysql

mkdir /var/lib/mysql-cluster

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

1. 解压

#cd /usr/local

# tar xvf mysql-cluster-gpl-7.5.5-linux-glibc2.5-x86\_64.tar.gz

1. 创建链接方便访问

# ln -s /usr/local/mysql-cluster-gpl-7.5.5-linux-glibc2.5-x86\_64 /usr/local/mysql

1. 初始化数据库

#./mysqld --user=mysql --basedir=/usr/local/mysql --datadir=/usr/local/mysql/data/ --initialize

如果初始化成功之后，系统会提示一个随机生成的数据库密码，此时需要记住这个密码.

1. 修改权限

chown -R root .

chown -R mysql data

chgrp -R mysql .

cp support-files/mysql.server /etc/rc.d/init.d/

chmod +x /etc/rc.d/init.d/mysql.server

chkconfig --add mysql.server

1. 配置数据节点

# vi /etc/my.cnf

|  |
| --- |
| [mysqld]  ndbcluster  ndb-connectstring=10.1.138.116  [mysql\_cluster]  ndb-connectstring=10.1.138.116 |

1. 启动集群节点上面的服务启动mysql（成功会有success）

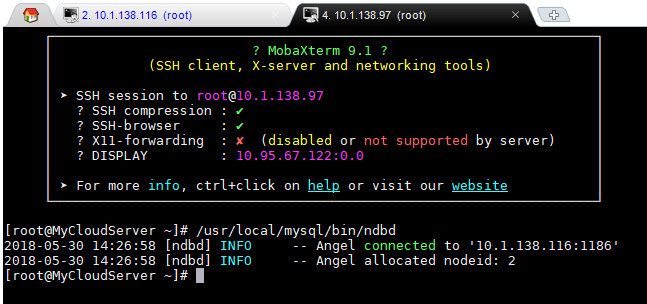
# /etc/init.d/mysql.server start

1. 启动mysql成功之后使用./mysql –uroot –p命令登录进mysql内然后进行密码修改等操作,集群数据库的密码需要相同，这里我修改为test。

# SET PASSWORD = PASSWORD('test');

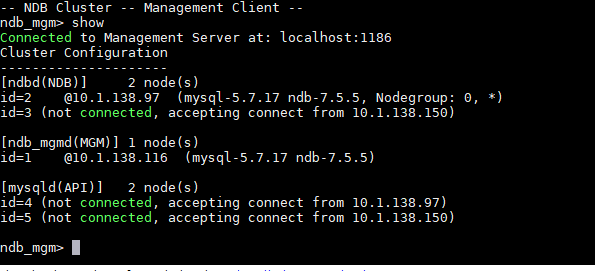
1. 启动/usr/local/mysql/bin/ndbd –initial

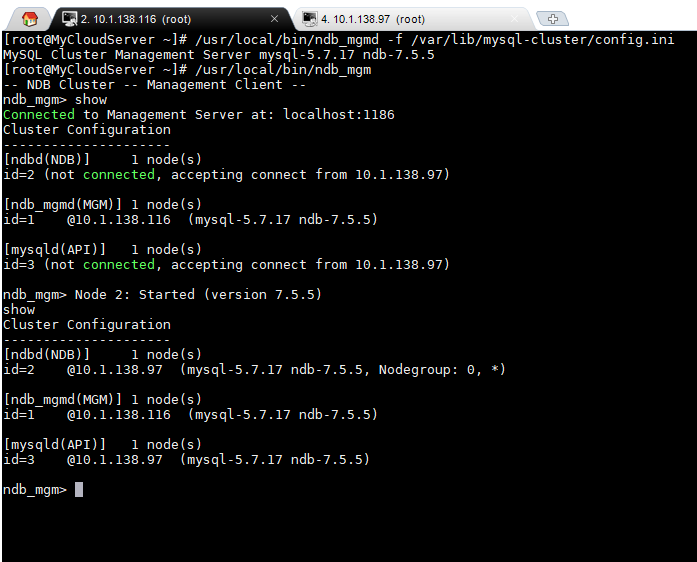
如果出现下述现象就成功了



1. 150如97配置步骤相同，当所有的节点配置完成，回到管理节点，使用上述说过的show查看，如下的类似显示，证明已经连接完成。

#此处截图对应的是只有一个97数据节点的情况。





1. **Sysbench性能测试工具概述**

集群完成后我们需要配置集群性能测试工具。我们使用sysbench。sysbench是一个开源的、模块化的、跨平台的多线程性能测试工具，可以用来进行CPU、内存、磁盘I/O、线程、数据库的性能测试。目前支持的数据库有MySQL、Oracle和PostgreSQL。当前功能允许测试的系统参数有：

|  |
| --- |
| file I/O performance （文件I / O性能）  scheduler performance （调度性能）  memory allocation and transfer speed （内存分配和传输速度）  POSIX threads implementation performance （POSIX线程执行绩效）  database server performance (OLTP benchmark) （数据库服务器性能） |

使用# sysbench –help 看到sysbench的详细使用方法。

sysbench [options]... [testname] [command]

（1）command

command是sysbench要执行的命令，包括prepare、run和cleanup，顾名思义，prepare是为测试提前准备数据，run是执行正式的测试，cleanup是在测试完成后对数据库进行清理。

（2）testname

testname指定了要进行的测试，在老版本（1.0以前）的sysbench中，可以通过--test参数指定测试的脚本；而在新版本中，--test参数已经声明为废弃，可以不使用--test，而是直接指定脚本。本实验使用的是0.4版本，也就是老版本，需要—test参数。

测试时使用的脚本为lua脚本，可以使用sysbench自带脚本，也可以自己开发。本实验使用自带脚本。

（3）options

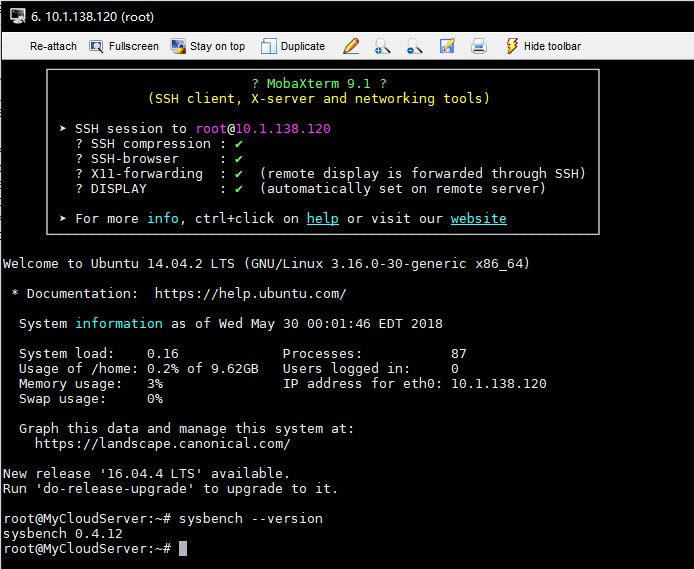
sysbench的参数有很多，MySQL连接信息参数有：

|  |
| --- |
| --mysql-host：MySQL服务器主机名，默认localhost；如果在本机上使用localhost报错，提示无法连接MySQL服务器，改成本机的IP地址应该就可以了。  --mysql-port：MySQL服务器端口，默认3306  --mysql-user：用户名  --mysql-password：密码  MySQL执行参数  --oltp-test-mode：执行模式，包括simple、nontrx和complex，默认是complex。simple模式下只测试简单的查询；nontrx不仅测试查询，还测试插入更新等，但是不使用事务；complex模式下测试最全面，会测试增删改查，而且会使用事务。可以根据自己的需要选择测试模式。  --oltp-tables-count：测试的表数量，根据实际情况选择  --oltp-table-size：测试的表的大小，根据实际情况选择  --threads：客户端的并发连接数  --time：测试执行的时间，单位是秒，该值不要太短，可以选择120  --report-interval：生成报告的时间间隔，单位是秒，如10  mysql options:  --mysql-host=[LIST,...] MySQL server host [localhost]  --mysql-port=N MySQL server port [3306]  --mysql-socket=STRING MySQL socket  --mysql-user=STRING MySQL user [sbtest]  --mysql-password=STRING MySQL password []  --mysql-db=STRING MySQL database name [sbtest]  --mysql-table-engine=STRING storage engine to use for the test table {myisam,innodb,bdb,heap,ndbcluster,federated} [innodb]  --mysql-engine-trx=STRING whether storage engine used is transactional or not {yes,no,auto} [auto]  --mysql-ssl=[on|off] use SSL connections, if available in the client library [off]  --myisam-max-rows=N max-rows parameter for MyISAM tables [1000000]  --mysql-create-options=STRING additional options passed to CREATE TABLE [] |

1. **Sysbench性能测试工具安装及测试步骤：**
2. 性能测试最好不要在安装mysql服务的服务器上运行，所以我使用另一个单独的服务器10.1.138.120.它是ubuntu系统。
3. 安装sysbench

#apt-get install sysbench

1. 运行sysbench –version测试是否安装成功。



1. 可以分别执行以下性能测试：

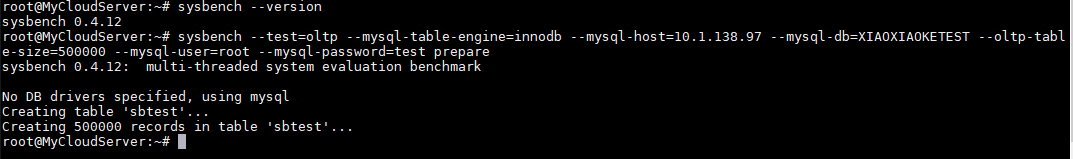
#sysbench --test=oltp --mysql-table-engine=innodb --mysql-host=10.1.138.97 --mysql-db=XIAOXIAOKETEST --oltp-table-size=500000 --mysql-user=root --mysql-password=test prepare

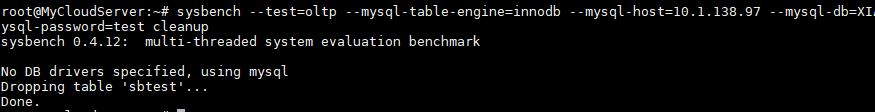
#sysbench --num-threads=16 --test=oltp --mysql-table-engine=innodb --mysql-host=10.1.138.97 --mysql-db=XIAOXIAOKETEST --oltp-table-size=500000 --mysql-user=root --mysql-password=test run

#sysbench --num-threads=16 --test=oltp --mysql-table-engine=innodb --mysql-host=10.1.138.97 --mysql-db=XIAOXIAOKETEST --oltp-table-size=500000 --mysql-user=root --mysql-password=test cleanup

此时mysql-table-engine为innodb，所以进行的是单节点测试。

结果如





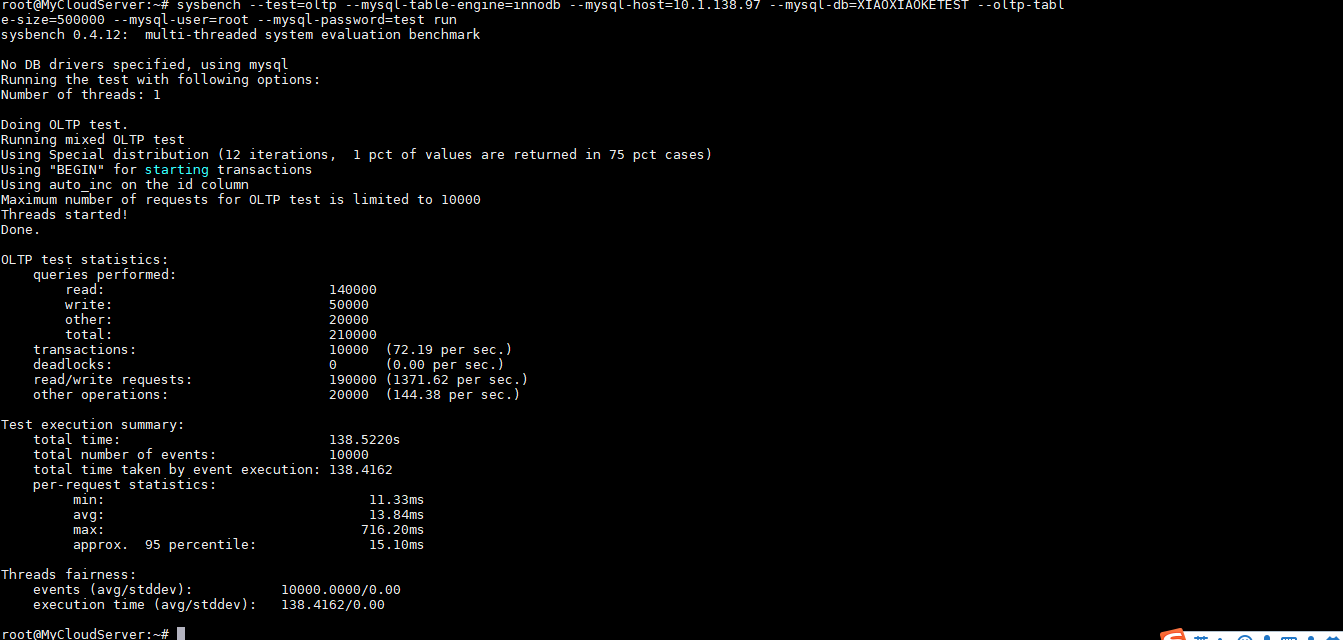
修改mysql-table-engine为ndbcluster，进行集群测试。

#sysbench --test=oltp --mysql-table-engine=ndbcluster --mysql-host=10.1.138.97 --mysql-db=XIAOXIAOKETEST --oltp-table-size=500000 --mysql-user=root --mysql-password=test prepare

#sysbench --test=oltp --mysql-table-engine=ndbcluster --mysql-host=10.1.138.97 --mysql-db=XIAOXIAOKETEST --oltp-table-size=500000 --mysql-user=root --mysql-password=test run

#sysbench --test=oltp --mysql-table-engine=ndbcluster --mysql-host=10.1.138.97 --mysql-db=XIAOXIAOKETEST --oltp-table-size=500000 --mysql-user=root --mysql-password=test cleanup

结果与上类似，run后的性能测试数据为



1. **总结**

1、在实际中需要分开数据节点和sql节点, 需要使用备份数据库，要分开管理节点和负载均衡器，负载均衡服务还需要做备份。

2、还未测试各种节点断开、服务器停止、数据库服务停止、是否会出现数据异常等等特殊情况。因为一旦数据不同步，想要恢复数据会比较麻烦。（之后经过测试，数据库服务停止之后，或者服务器停止之后，只要管理节点正常，数据会之后在开启之后同步过去，但是为了防止万一还是需要做好数据备份工作）

3、本来打算使用mysql cluster自带的FLEXASYNCH测试，但由于Oracle并不提供flexAsynch的二进制文件下载，仅仅在MySQL Cluster的源码包中提供这个测试工具的源码，所以必须手动编译MySQL Cluster才能获得flexAsynch的执行文件。相对比较麻烦，所以这里采用了sysbench。而此时需要注意，GitHub上最新版的sysbench有bug，代码与某些CPU不兼容，输入命令会出现illegal instruction的错误报告。这些不兼容的CPU如：3dnowprefetch，abm

Adx，avx2，bmi1，bmi2，cqm，cqm\_llc，cqm\_mbm\_local，cqm\_mbm\_total，cqm\_occup\_llc，fma，hle，intel\_pt，invpcid，movbe，rdseed，rtm，smap，tsc\_adjust。简单地说就是不适用大部分centos服务器。所以我才重新选用120这个ubuntu服务器进行sysbench测试。这是个坑，需要注意。这里的bug报告如：<https://bugzilla.redhat.com/show_bug.cgi?id=1459356>