**LeDrill测试环境部署文档**

**Version 1.0**

虚拟机配置信息

|  |  |  |
| --- | --- | --- |
|  | host1 | host2 |
| CPU | 2 cores | 2 cores |
| Memory | 8g | 8g |
| Disk | /dev/mapper/VolGroup-lv\_root 57g  mounted / | /dev/vda3 42g mounted /， /dev/mapper/data-data01 98g mounted /data |
| OS | CentOS release 6.2 (Final) | CentOS release 6.5 (Final) |
| IP | 192.168.11.123 | 192.168.11.121 |
| Hostname | cloudera01 | pmdrill |
| SSH Login | cloudera/cloudera | root/ YM!QAZ2wsx#EDC |
| Services Deployed  And Ports Listened | CDH Related:  Cloudera Manager: 7180  HDFS: 8020, 8022, 50010, 50020, 50070, 50075, 50090, 50475  MapReduce: 8021, 50030, 50060  ZooKeeper: 2181  Mdrill: 1107,6701,6801,6802,6803  Kafka: 9092 | Kafka Monitor: 8092  Tomcat: 8080  MySQL: 3306 |

20140717

Kafka, Zookeeper: 迁移到host2。

host1本身是一台包含CDH的虚拟机，当启动该虚拟机时，CDH的相关服务会自动启动，不需要再重新安装CDH。

host1的相关配置包括：系统配置，CDH配置，mdrill部署与配置，Kafka部署与配置

系统配置

1. 配置IP，子网掩码和网关（192.168.11.123/24, 192.168.11.1）

该主机上可用网卡设备名称为eth1，所以网络配置对象为eth1。

以root用户身份修改文件/etc/sysconfig/network-scripts/ifcfg-eth1，内容如下：

DEVICE="eth1"

TYPE="Ethernet"

ONBOOT="yes"

NM\_CONTROLLED="no"

BOOTPROTO=none

HWADDR="52:54:00:00:00:A9"

IPADDR=192.168.11.123

PREFIX=24

GATEWAY=192.168.11.1

DNS1=202.106.0.20

以root用户身份修改文件/etc/udev/rules.d/70-persistent-net.rules，内容如下：

SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?\*", ATTR{address}=="52:54:00:00:00:a9", ATTR{type}=="1", KERNEL=="eth\*", NAME="eth1"

以root身份使用如下命令重启网络服务

service network restart

注意：文件ifcfg-eth1中的DEVICE，HWADDR要分别与文件70-persistent-net.rules中的NAME，ATTR{address}保持一致，否则重启网络服务会出错。

1. 配置主机名cloudera01

以root用户身份修改文件/etc/sysconfig/network，将其中的HOSTNAME选项修改为

HOSTNAME=cloudera01

以root身份使用如下命令重启网络服务

service network restart

1. 修改两台主机的hosts文件，方便通过主机名相互访问

在两台主机的/etc/hosts文件末尾追加如下两行：

192.168.11.123 cloudera01

192.168.11.121 pmdrill

1. 设置时区，同步系统时间

cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime

/usr/sbin/ntpdate cn.pool.ntp.org

CDH配置

1. 在本地浏览器中输入<http://192.168.11.123:7180/>访问Cloudera Admin Console， 登录用户名和密码均为cloudera。
2. 为了节省host1的系统资源，删除测试环境中不需要的服务，只保留HDFS, MapReduce和ZooKeeper。
3. 配置各项服务数据存放路径

Admin Console主页面 -> Services -> *Service-Name* -> Configuration -> View and Edit -> 搜索框中搜索属性名称 –> 设置值 -> Save Changes -> Actions -> Restart。

其中*Service-Name*表示hdfs1, mapreduce1或者zookeeper1。

需要修改的具体配置如下：

HDFS各角色数据存放路径：

NameNode Data Directories: /data/hadoop/dfs/nn

HDFS Checkpoint Directory : /data/hadoop/dfs/snn

DataNode Data Directory: /data/hadoop/dfs/dn

MapReduce各角色数据存放路径：

JobTracker Local Data Directory: /data/hadoop/mapred/jt

TaskTracker Local Data Directory: /data/hadoop/mapred/local

ZooKeeper数据存放路径：

Data Directory: /data/zookeeper/data

Transaction Log Directory: /data/zookeeper/dataLog

1. 修改各项服务角色日志输出路径。

可以按如下流程进行操作，配置完成需要重启服务。

Admin Console主页面 -> Services -> *Service-Name* -> Configuration -> View and Edit -> 搜索框中搜索属性名称 –> 设置值 -> Save Changes -> Actions -> Restart。

其中*Service-Name*表示hdfs1, mapreduce1或者zookeeper1。

下图是修改NameNode Log Directory和SecondaryNameNode Directory的日志输出路径的示例。

需要修改的所有配置如下：

HDFS各角色日志输出路径：

NameNode Log Directory: /data/hadoop/logs/hadoop-hdfs

SecondaryNameNode Log Directory: /data/hadoop/logs/hadoop-hdfs

DataNode Log Directory: /data/hadoop/logs/hadoop-hdfs

HttpFS Log Directory: /data/hadoop/logs/hadoop-httpfs

MapReduce各角色日志输出路径：

JobTracker Log Directory: /data/hadoop/logs/hadoop-0.20-mapreduce

TaskTracker Log Directory: /data/hadoop/logs/hadoop-0.20-mapreduce

ZooKeeper各角色日志输出路径：

ZooKeeper Log Directory: /data/zookeeper/logs

1. 配置mdrill所需的ZooKeeper各项参数。

可以按如下流程进行操作，配置完成需要重启ZooKeeper服务。

Admin Console主页面 -> Services ->zookeeper1 -> Configuration -> View and Edit -> 搜索框中搜索属性名称 –> 设置值 -> Save Changes -> Actions -> Restart。

需要修改的所有配置如下：

ZooKeeper配置项：黄色的是添加的配置；没有黄色的是默认值，检查是否与cdh中的配置相符。

# The number of milliseconds of each tick

tickTime=2000

# The number of ticks that the initial synchronization phase can take

initLimit=10

# The number of ticks that can pass between sending a request and getting an acknowledgement

syncLimit=5

# The directory where the snapshot is stored.

dataDir=/data/tmp/zookeeper/

# The port at which the clients will connect

clientPort=2181

# The number of snapshots to retain in dataDir

autopurge.snapRetainCount=100

# Purge task interval in hours. Set to "0" to disable auto purge feature

autopurge.purgeInterval=12

#for mdrill

maxClientCnxns=300

maxSessionTimeout=20000

mdrill部署与配置

1. 安装依赖库

将下面两个文件上传到host1（zeromq-tested.zip中包含zeromq-2.1.7.tar.gz和jzmq-master.zip）的Downloads目录下。

alimama-adhoc.tar.gz.0623

zeromq-tested.zip

制作安装脚本

# vi install-libs.sh

添加如下内容

#!/bin/bash

# yum required libs

yum -y install libtool

yum -y install gcc-c++

yum -y install uuid-devel

yum -y install libuuid-devel

cd ~/Downloads/

unzip zeromq-tested.zip

# install zeromq

cd ~/Downloads/

tar -xvf zeromq-2.1.7.tar.gz

cd zeromq-2.1.7

./autogen.sh

./configure

make

make install

# install jzmq

cd ~/Downloads/

unzip jzmq-master.zip

cd jzmq-master

./autogen.sh

./configure

make

make install

echo "export LD\_LIBRARY\_PATH=/usr/local/lib" > /etc/profile.d/mdrill.sh

source /etc/profile.d/mdrill.sh

# chmod +x install-libs.sh

在host1上执行install-libs.sh

./install-llibs.sh

测试jzmq是否装好。

cd ~/Downloads/jzmq-master/perf

sh local\_lat.sh tcp://127.0.0.1:5000 1 100

另外启动控制台：

cd ~/Downloads/jzmq-master/perf

sh remote\_lat.sh tcp://127.0.0.1:5000 1 100

message size: 1 [B]

roundtrip count: 100

mean latency: 275.0 [us]

看到 message size: 1 [B]….则说明配置成功。

1. 修改系统参数

在host1文件/etc/sysctl.conf末尾追加如下内容，如果sysctl.conf中已经有了其中的某些项配置，追加也没有关系，后面的配置起作用。

net.ipv4.ip\_local\_port\_range = 10000 65535

net.core.somaxconn = 4096

net.core.netdev\_max\_backlog=16384

net.ipv4.tcp\_max\_syn\_backlog=8192

net.ipv4.tcp\_syncookies=1

fs.file-max = 100000

// 使配置生效

# sysctl -p /etc/sysctl.conf

运行上面命令后，输出如下错误，不予理会。

error: "net.bridge.bridge-nf-call-ip6tables" is an unknown key

error: "net.bridge.bridge-nf-call-iptables" is an unknown key

error: "net.bridge.bridge-nf-call-arptables" is an unknown key

error: "net.ipv6.conf.default.disable\_ipv6" is an unknown key

error: "net.ipv6.conf.all.disable\_ipv6" is an unknown key

20140708 增加

vi /etc/security/limits.conf

#liuhk2

\* soft nofile 65535

\* hard nofile 65535

# pwd

/etc/security/limits.d

// 屏蔽下面这行

[root@psb-la-vm01 limits.d]# vi 90-nproc.conf

#\* soft nproc 1024

1. 安装mdrill

在host1主机上按顺序执行如下命令

cd ~/Downloads

mv alimama-adhoc.tar.gz.0623 alimama-adhoc.tar.gz

tar -xzf alimama-adhoc.tar.gz

mv alimama /data/

cd /data/alimama/adhoc-core/bin

chmod +x bluewhale

dos2unix bluewhale

# 创建mdrill使用的本地目录（与storm.yaml中的配置相符）

mkdir /data/storm

mkdir /data/mdrill

#HDFS中创建mdrill所需的目录结构，并修改所有者权限（与storm.yaml中的配置相符）

su hdfs

hadoop fs -mkdir –p /mdrill/tablelist

hadoop fs -chown -R root:root /mdrill

1. 启动/停止mdrill

mdrill要求的Python版本为2.6.6，使用如下命令检查系统中的Python版本是否符合要求。

python –V

进入mdrill的配置目录

cd /data/alimama/adhoc-core/conf

根据当前单机硬件配置，修改storm.yaml文件，具体内容详见附件。

进入mdrill的bin目录

cd /data/alimama/adhoc-core/binlsbluewhale create\_feedbackred.sql create\_feedback.sql create\_hitred.sql create\_hit.sql init-bluewhale.sh mdrill.jar start-bluewhale.sh stop-bluewhale.sh supervisor.log ui ui.log创建表（\*.sql的文件根据表的需求编写，详见《INSTALL.docx》）

./bluewhale mdrill create ./create\_feedback.sql

./bluewhale mdrill create ./create\_feedbackred.sql

./bluewhale mdrill create ./create\_hit.sql

./bluewhale mdrill create ./create\_hitred.sql

./bluewhale mdrill create ./create\_error.sql

启动Nimbus

nohup ./bluewhale nimbus >nimbus.log &

启动Supervisor

nohup ./bluewhale supervisor >supervisor.log &

启动UI

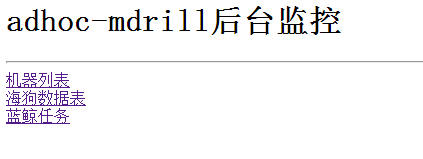
mkdir ./ui

nohup ./bluewhale mdrillui 1107 ../lib/adhoc-web-0.18-beta.jar ./ui >ui.log &

启动表

./bluewhale mdrill table feedback,feedbackred,hit,hitred,error

启动UI后在本地浏览器中输入<http://192.168.11.123:1107/mdrill.jsp> 登录mdrill后台监控web ui，可以看到如下页面。



点击海狗数据表链接可以看到如下页面，证明数据表已经启动了。



查看logs文件夹：

“如果有\*6901.log，一定是任务调度配置错了。”

--- 母延年（mdrill作者）

ll /data/alimama/adhoc-core/logs/

total 224652

-rw-r--r-- 1 root root 663 May 28 16:29 gc-6601.log

-rw-r--r-- 1 root root 377 May 28 15:38 gc-6701.log

-rw-r--r-- 1 root root 377 May 28 15:47 gc-6702.log

-rw-r--r-- 1 root root 377 May 28 15:28 gc-6703.log

-rw-r--r-- 1 root root 377 May 28 15:16 gc-6704.log

-rw-r--r-- 1 root root 377 May 28 15:29 gc-6705.log

-rw-r--r-- 1 root root 90612 May 28 13:58 supervisor.log

-rw-r--r-- 1 root root 21050272 May 28 17:22 worker-6601.log

-rw-r--r-- 1 root root 15480327 May 28 17:22 worker-6701.log

-rw-r--r-- 1 root root 15496944 May 28 17:22 worker-6702.log

-rw-r--r-- 1 root root 15539973 May 28 17:22 worker-6703.log

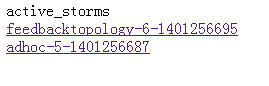
-rw-r--r-- 1 root root 15509500 May 28 17:22 worker-6704.log

-rw-r--r-- 1 root root 15568886 May 28 17:22 worker-6705.log

启动读任务

./bluewhale jar ./mdrill.jar com.alimama.mdrillImport.Topology feedbacktopology feedback,feedbackred,hit,hitred,error 2 256 1000 2014030520050

在UI中可以查看状态，点击“蓝鲸任务”。



adhoc-\*: 对应的是启动表的任务

feedbacktopology: 对应的是读实时表的任务

如果需要：

停止读任务

./bluewhale kill feedbacktopology

停止表

./bluewhale mdrill drop feedback,feedbackred,hit,hitred,error

Kafka部署配置启动与测试

1. 部署Kafka

将kafka\_2.8.0-0.8.1.tgz上传到host1的Downloads目录下。

cd ~/Downloads/

解压kafka\_2.8.0-0.8.1.tgz，得到文件夹kafka\_2.8.0-0.8.1。

tar zxf kafka\_2.8.0-0.8.1.tgz

创建kafka部署目录

mkdir /data/kafka

mv kafka\_2.8.0-0.8.1 /data/kafka

1. 配置Kafka

修改Kafka配置文件（已经有的项目修改，否则就增加）

cd /data/kafka/kafka\_2.8.0-0.8.1

vi config/server.properties 修改后的内容如下：

# The id of the broker. This must be set to a unique integer for each

broker.broker.id=4

#log.dirs=/tmp/kafka-logs

log.dirs=/data/tmp/kafkalog

auto.create.topics.enable=false

# The minimum age of a log file to be eligible for deletion

# log.retention.hours=168

log.retention.hours=72

zookeeper.connect=cloudera01:2181

#2014/7/14

zookeeper.connect=localhost:2181

kafka的broker在zk里面显示为空，即broker丢失

怀疑是网络不稳定，所以改成localhost

情况有好转，但是还是丢

Cloudera Manager中发现zk日志

Unexpected Exception:

java.nio.channels.CancelledKeyException

at sun.nio.ch.SelectionKeyImpl.ensureValid(SelectionKeyImpl.java:55)

at sun.nio.ch.SelectionKeyImpl.interestOps(SelectionKeyImpl.java:59)

at org.apache.zookeeper.server.NIOServerCnxn.sendBuffer(NIOServerCnxn.java:153)

at org.apache.zookeeper.server.NIOServerCnxn.sendResponse(NIOServerCnxn.java:1076)

at org.apache.zookeeper.server.FinalRequestProcessor.processRequest(FinalRequestProcessor.java:170)

at org.apache.zookeeper.server.SyncRequestProcessor.run(SyncRequestProcessor.java:138)

下一步：换台机器装zk试试

<http://zookeeper-user.578899.n2.nabble.com/HBase-dies-after-some-time-td7577294.html>

1. 启动Kafka

nohup bin/kafka-server-start.sh config/server.properties > sv.out &

如需要，停止Kafka

bin/kafka-server-stop.sh config/server.properties

1. 建立topic

bin/kafka-topics.sh --create --zookeeper cloudera01:2181 --replication-factor 1 --partitions 2 --topic feedback-topic

bin/kafka-topics.sh --create --zookeeper cloudera01:2181 --replication-factor 1 --partitions 2 --topic hit-topic

bin/kafka-topics.sh --create --zookeeper cloudera01:2181 --replication-factor 1 --partitions 2 --topic error-topic

1. 测试生产和消费

在一个终端中，输入如下命令，启动生产者。

bin/kafka-console-producer.sh --broker-list cloudera01:9092 --topic feedback-topic

之后在输入消息，回车。

在另一个终端中，输入如下命令，启动消费者。

bin/kafka-console-consumer.sh --zookeeper cloudera01:2181 --topic feedback-topic

可以看到生产者发送的消息。

1. 其他

列出所有topic

bin/kafka-topics.sh --list --zookeeper cloudera01:2181

查看topic的情况

bin/kafka-topics.sh --describe --zookeeper cloudera01:2181 --topic feedback-topic

Kafka Monitor部署与启动

1. 部署Kafka Monitor

上传文件夹kafka offset monitor到host2的/root/package目录下。

在host2上创建部署目录

mkdir /data/kmonitor/

将文件夹kafka offset monitor中的内容移动到/data/kmonitor/中

mv kafka\ offset\ monitor/\* /data/kmonitor/

ll /data/kmonitor/-rw-r--r-- 1 root root 57786330 Jun 18 17:59 KafkaOffsetMonitor-assembly-0.2.0.jar-rw-r--r-- 1 root root 1075 Jun 20 10:40 start-up.shcat /data/kmonitor/start-up.sh

#!/bin/bashnohup java -Xms512M -Xmx512M -Xss1024K -XX:PermSize=256m -XX:MaxPermSize=512m -cp KafkaOffsetMonitor-assembly-0.2.0.jar com.quantifind.kafka.offsetapp.OffsetGetterWeb --zk cloudera01:2181 --port 8092 --refresh 300.seconds --retain 3.days > km.out &

1. 启动Kafka Monitor

cd /data/kmonitor/

chmod +x start-up.sh

./start-up.sh

通过jps查看Kafka Monitor是否启动。

jps23772 OffsetGetterWeb在本地浏览器中输入<http://192.168.11.121:8092>可以访问Kafka Monitor的Web页面。

Stream部署

1. 安装mysql

yum install mysql mysql-server mysql-devel mysql-connector-odbc libdbi-dbd-mysql

1. 配置mysql

vi /etc/my.cnf

[mysqld]

datadir=/var/lib/mysql

socket=/var/lib/mysql/mysql.sock

user=mysql

# Disabling symbolic-links is recommended to prevent assorted security risks

symbolic-links=0

max\_allowed\_packet=128M

default-character-set=utf8

[client]

default-character-set=utf8

[mysqld\_safe]

log-error=/var/log/mysqld.log

pid-file=/var/run/mysqld/mysqld.pidi

1. 启动/停止mysql

# service mysqld status

# service mysqld start

# service mysqld stop

# chkconfig mysqld on

1. 创建数据库，导入数据

上传db\_avatarui.sql到host2的/root/package目录下。

登录mysql

mysql –uroot –p

创建db\_stream数据库

mysql> CREATE DATABASE IF NOT EXISTS db\_stream default charset utf8 COLLATE utf8\_general\_ci;

导入数据

mysql> use db\_stream;

source /root/package /db\_avatarui.sql;

创建用户骚扰度数据表

CREATE TABLE `db\_stream`.`disturbance` (

`id` INT NOT NULL AUTO\_INCREMENT,

`thedate` VARCHAR(45) NOT NULL,

`hit` INT NOT NULL,

`users` INT NOT NULL,

`percent` DOUBLE NOT NULL,

`inserttime` TIMESTAMP NOT NULL,

PRIMARY KEY (`id`));

1. 安装Tomcat

yum install tomcat6 tomcat6-webapps tomcat6-admin-webapps tomcat6-docs-webapp tomcat6-javadoc

1. 配置Tomcat

修改Tomcat日志输出位置

mkdir /data/tomcatlogs/chown –R tomcat:tomcat /data/tomcatlogs/

cd /usr/share/tomcat6/

rm logs

ln –s /data/tomcatlogs/ logs

ll logs lrwxrwxrwx 1 root root 17 May 23 09:41 logs -> /data/tomcatlogs/

1. 部署Stream项目到Tomcat

将Stream项目的WAR包Stream.war-0619上传到host2的/root/package目录下。

将WAR包移动到Tomcat的webapps目录下

mv /root/package/Stream.war-0619 /usr/share/tomcat6/webapps/Stream.war

启动停止Tomcat

service tomcat6 start

service tomcat6 stop

chkconfig tomcat6 on

启动Tomcat后，Stream项目的WAR包会自动解压到webapps目录下。

修改Stream项目的配置文件prod-config.properties，修改后的内容如下。

driverClassName=com.mysql.jdbc.Driver

#url=jdbc:mysql://127.0.0.1:3306/db\_avatarui?useUnicode=true&characterEncoding=utf-8

#url=jdbc:mysql://172.17.61.114:3306/db\_stream?useUnicode=true&characterEncoding=utf-8

url=jdbc:mysql://127.0.0.1:3306/db\_stream?useUnicode=true&characterEncoding=utf-8

username=root

#password=123456

password=

cpool.checkoutTimeout=5000

cpool.minPoolSize=5

cpool.maxPoolSize=25

cpool.maxIdleTime=120

cpool.maxIdleTimeExcessConnections=1800

cpool.acquireIncrement=10

mdrill.driverClassName=com.alimama.mdrill.jdbc.MdrillDriver

#note that no / at the end of url

#mdrill.url=jdbc:mdrill://10.100.149.232:1107

#mdrill.url=jdbc:mdrill://172.17.61.112:1107

mdrill.url=jdbc:mdrill://192.168.11.123:1107

# zookeeper configuration

zkAddress=192.168.11.123:2181

zkPath=/config/pm/hitconfig

重启Tomcat

service tomcat6 restart

在本地浏览器输入<http://192.168.11.121:8080/Stream/>即可访问Stream项目。

2014年7月2日星期三