大规模分布式系统第四次作业——HBase的安装与原理

16300200020 张言健

1. 安装过程

1.1 安装环境

服务器：ubuntu 16.04

软件：hbase 2.0.5

1.2 安装步骤

下载解压缩

```
wget https://mirrors.tuna.tsinghua.edu.cn/apache/hbase/2.0.5/hbase-2.0.5-bin.tar.gz
sudo cp hbase-2.0.5-bin.tar.gz /usr/local
sudo tar -zxf hbase-2.0.5-bin.tar.gz
```

配置环境，编辑**conf**目录下的**hbase-env.sh**，添加

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/
```

配置**conf**目录下的**hbase-site.xml**，添加

```
<configuration>
    <property>
        <name>hbase.rootdir</name>
        <value>/usr/local/tmp/hbase</value>
    </property>
</configuration>
```

在**~/.bashrc**文件中添加

```
export HBASE_HOME=/usr/local/hbase
export  HBASE_CONF_DIR=$HBASE_HOME/conf
export  HBASE_CLASS_PATH=$HBASE_CONF_DIR
export  PATH=$PATH:$HBASE_HOME/bin
```

使.bashrc文件立即生效

```
source ~/.bashrc
```

启动HBase

```
start-hbase.sh
```

结果如下图，配置成功

```
root@localhost:/usr/local/hbase/conf# start-hbase.sh
running master, logging to /usr/local/hbase/logs/hbase-root-master-localhost.out
root@localhost:/usr/local/hbase/conf# jps
1521 Jps
1222 HMaster
root@localhost:/usr/local/hbase/conf#
```

2. HBase原理

一、HBASE处理超过一台服务器的mem的方法

HBase使用Region的方法来对超过一台服务器的mem进行处理。以下是详细介绍：

HBase将表会切分成小的数据单位叫region，分配到多台服务器。托管region的服务器叫做RegionServer。一般情况下，RgionServer和HDFS DataNode并列配置在同一物理硬件上，RegionServer本质上是HDFS客户端，在上面存储访问数据，HMaster分配region给RegionServer,每个RegionServer托管多个region。

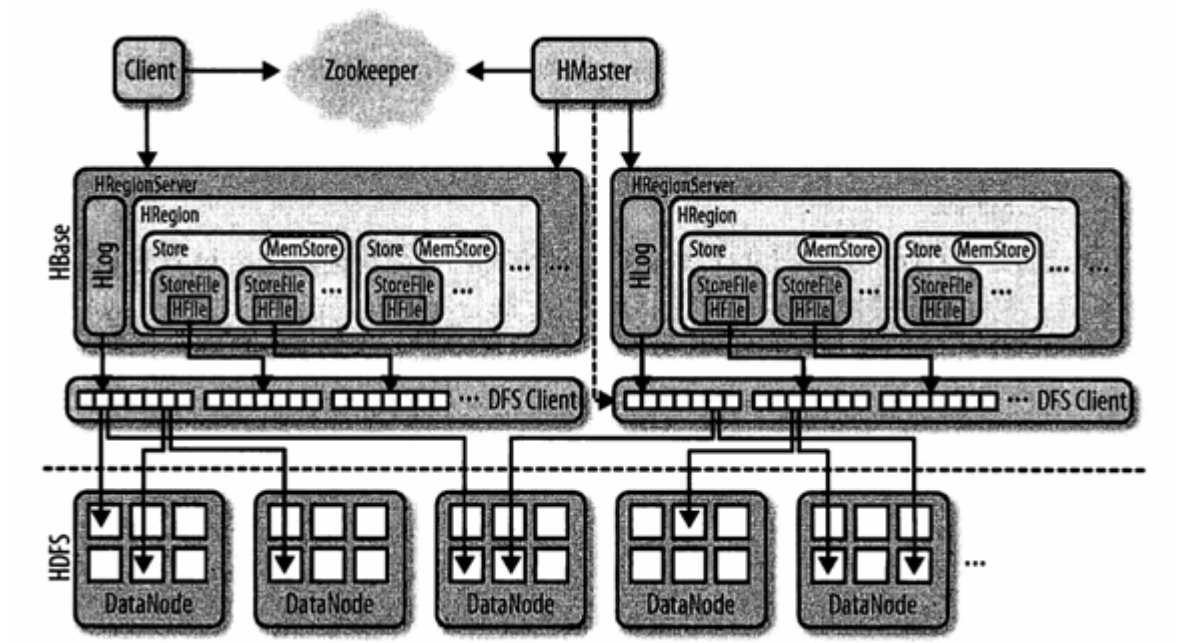
HBase中的两个特殊的表，-ROOT-和.META.，用来查找各种表的region位置在哪。-ROOT-指向.META.表的region，.META.表指向托管待查找的region的RegionServer。

zookeeper负责跟踪region服务器，保存root region的地址。

Client负责与zookeeper子集群以及HRegionServer联系。

HMaster负责在启动HBase时，把所有的region分配到每个HRegion Server上，也包括-ROOT-和.META.表。

HRegionServer负责打开region，并创建对应的HRegion实例。HRegion被打开后，它为每个表的HColumnFamily创建一个Store实例。每个Store实例包含一个或多个StoreFile实例，它们是实际数据存储文件HFile的轻量级封装。每个Store有其对应的一个MemStore，一个HRegionServer共享一个HLog实例。



一次服务基本的流程：

1. 客户端通过zookeeper获取含有-ROOT-的region服务器名。

2. 通过含有-ROOT-的region服务器查询含有.META.表中对应的region服务器名。
3. 查询.META.服务器获取客户端查询的行键数据所在的region服务器名。
4. 通过行键数据所在的region服务器获取数据。

二、HBASE如何读写数据

写操作流程

1. Client通过Zookeeper的调度，向RegionServer发出写数据请求，在Region中写数据。
2. 数据被写入Region的MemStore，直到MemStore达到预设阈值。
3. MemStore中的数据被Flush成一个StoreFile。
4. 随着StoreFile文件的不断增多，当其数量增长到一定阈值后，触发Compact合并操作，将多个StoreFile合并成一个StoreFile，同时进行版本合并和数据删除。
5. StoreFiles通过不断的Compact合并操作，逐步形成越来越大的StoreFile。
6. 单个StoreFile大小超过一定阈值后，触发Split操作，把当前Region Split成2个新的Region。父Region会下线，新Split出的2个子Region会被HMaster分配到相应的RegionServer上，使得原先1个Region的压力得以分流到2个Region上。

可以看出HBase只有增添数据，所有的更新和删除操作都是在后续的Compact历程中举行的，使得用户的写操作只要进入内存就可以立刻返回，实现了HBase I/O的高机能。

读操作流程

1. Client访问Zookeeper，查找-ROOT-表，获取.META.表信息。
2. 从.META.表查找，获取存放目标数据的Region信息，从而找到对应的RegionServer。
3. 通过RegionServer获取需要查找的数据。
4. Regionserver的内存分为MemStore和BlockCache两部分，MemStore主要用于写数据，BlockCache主要用于读数据。读请求先到MemStore中查数据，查不到就到BlockCache中查，再查不到就会到StoreFile上读，并把读的结果放入BlockCache。

寻址过程：client-->Zookeeper-->-ROOT-表-->.META.表-->RegionServer-->Region-->client

三、HBASE是列数据库吗？行、列存储数据库区别？

Hbase是一个面向列存储的分布式存储系统。以下为行、列数据库的区别：

行式数据库：

1. 数据是按行存储的
2. 没有建立索引的查询将消耗很大的io
3. 建立索引和视图需要花费一定的物理空间和时间资源
4. 面对大量的查询,复杂的查询,数据库必须被大量膨胀才能满足性能需求

列式数据库：

1. 数据是按列存储的,每一列单独存放
2. 数据既是索引
3. 只访问查询涉及的列,大量降低系统io
4. 每一列有一个线索来处理,支持查询的高并发
5. 数据类型一致,数据特征相似,高效的压缩