

《大规模分布式系统》实验报告

——Hbase

姓名：刘佰川 专业：计算机科学与技术（数据科学方向）学号：16307130214

0. 实验环境

VMware+Ubuntu18.04+Hadoop2.9.1+Java1.8.0_201+Hbase2.0.5

1. 实验要求

- I. 在网上查找资料，简单介绍 Hbase 原理
- II. 安装 Hbase，叙述安装测试过程

2. 实验过程

I. Hbase 简介

回答助教提出的三个问题

A. Hbase 处理超过一台服务器 memory 的方法

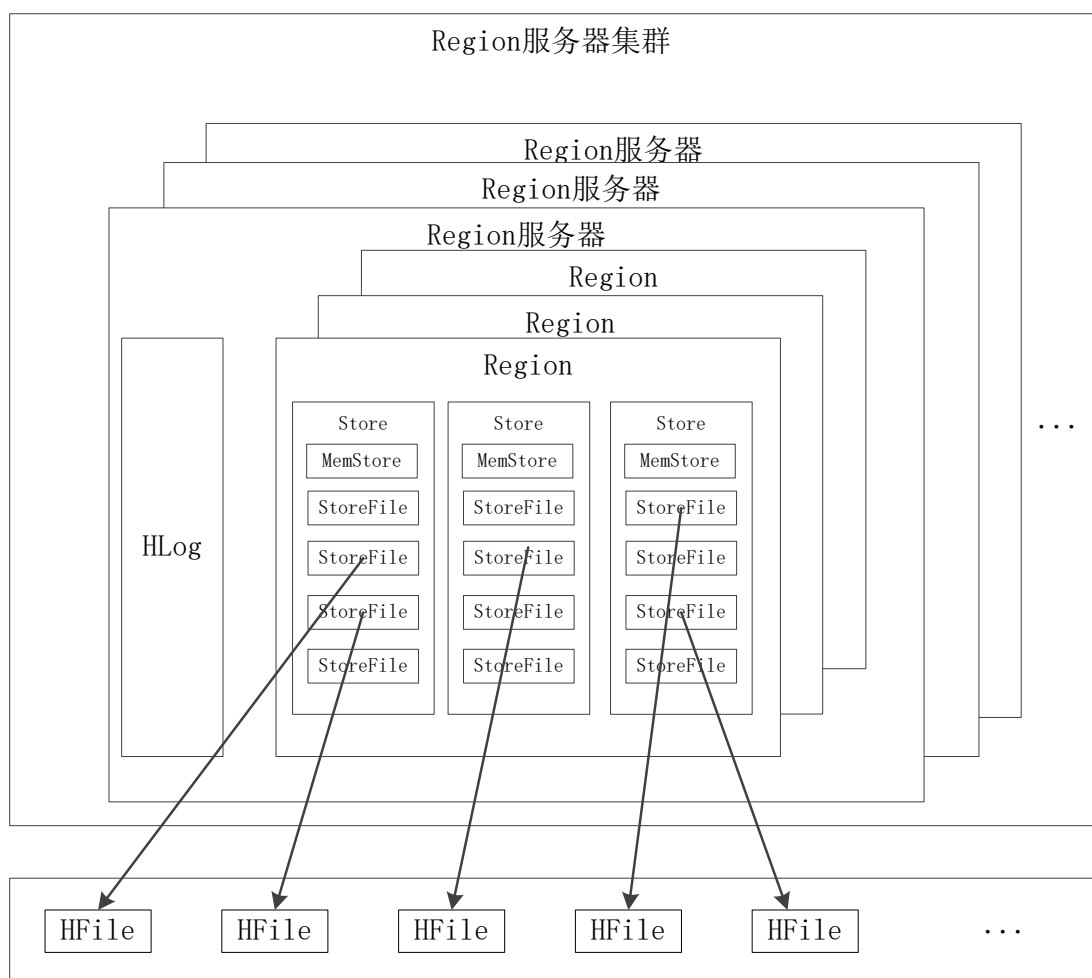
HBase 是一个高可靠、高性能、面向列、可伸缩的分布式数据库，是谷歌 BigTable 的开源实现，主要用来存储非结构化和半结构化的松散数据。通过阅读谷歌 BigTable 的介绍，对比 HBase 和 BigTable 的底层技术对应关系（即文件存储系统 HDFS vs GFS，海量数据处理 Hadoop MapReduce vs MapReduce，协同服务管理 Zookeeper vs Chubby）结合课程内容来解释 Hbase 的问题。

Hbase 处理多台服务器采用一个 master 主服务器和许多 region 服务器（对应 bigtable 中的 tablet）。Hbase 一开始只有一个 Region，随着存储的内容增大，Region 不断分裂，Region 拆分操作非常快，接近瞬间，因为拆分之后的 Region 读取仍然是原存储文件。Region 服务器负责存储和维护分配给自己的 Region，处理来自客户端的读写请求。主服务器 Master 负责管理和维护 HBase 表的分区信息，维护 Region 服务器列表，所以客户端并不是直接从 Master 主服务器上读取数据，而是在获得 Region 的存储位置信息后，直接从 Region 服务器上读取文件。为了防止 Master 失效而导致系统崩溃，客户端通过 zookeeper

来获得 Region 位置信息，这种设计方式使得 Master 的负载很小。第二个问题是 Region 的定位，采用了三层结构定位，在 Zookeeper 文件中记录了 -ROOT- 表的位置信息，-ROOT 表记录了 .META. 表的 Region 位置信息（当 Hbase 很大时，.META. 表会被分裂成多个 Region），-ROOT- 表只能有一个 Region。 .META. 表保存了 HBase 中所有用户数据表的 Region 位置信息。

B. Hbase 如何读取数据

要知道 Hbase 如何读取文件，就应该知道 Region 服务器的结构。



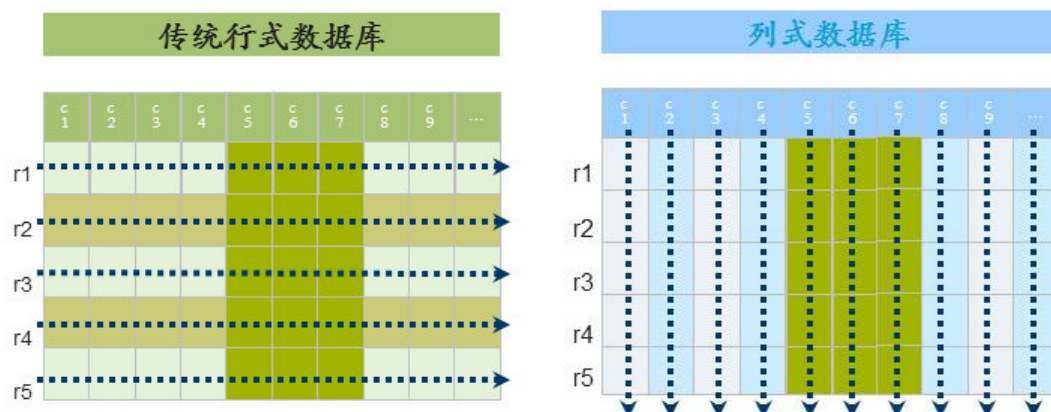
与 bigtable 的对应：Hlog->tablet log, MemStore->memtable, StoreFile->SSTable Files.

用户读写数据的过程，用户写入数据时，被分配到相应的 Region 服务器去执行，用户的数据首先被写入到 MemStore 和 Hlog 中，只有当操作写入 Hlog 之后，commit()调用才会将其返回给客户端，当用户读取数据时，Region 服务器会首先访问 MemStore 缓存，如果找不到，再去磁盘上面的

StoreFile 中寻找。

C. Hbase 是列数据库吗？行列数据库的区别

Hbase 是列数据库，采用面向列的存储，用一张图展示行列数据库的差别。



行式数据库存储格式按照‘行’的方式把一行各个字段的数据存储在一起，一行行连续存储。把一条数据的信息写到数据库中；或者对一条数据中的某些字段进行修改；或者删除整条数据一类的 OLTP 操作来说既直观也高效。若只对其中某些字段上的操作，行式数据库不分情况一律按照页面读取数据的方式，浪费了大量宝贵的 I/O。

列式数据库是将同一个数据列的各个值存放在一起。插入某个数据行时，该行的各个数据列的值也会存放不同的地方。列式数据库大大地提高了只对某些字段的大数据量查询的效率。当然，列式数据库不是万能的，每次读取某个数据行时，需要分别从不同的地方读取各个数据列的值，然后合并在一起形成数据行。因此，如果每次查询涉及的数据量较小或者大部分查询都需要整行的数据，列式数据库并不适用。很多列式数据库还支持列族（column family，Bigtable 系统中称为 locality group），即将多个经常一起访问的数据列的各个值存放在一起。如果读取的数据列属于相同的列组，列式数据库可以从相同的地方一次性读取多个数据列的值，避免了多个数据列的合并。

II. Hbase 安装

从镜像网站下载安装包 hbase-2.0.5-bin.tar.gz

解压 tar xzf hbase-1.2.4-bin.tar.gz

现在进行 Hbase 环境配置，进入 hbase 下的 conf 目录

```
junjin@ubuntu:~$ cd hbase-2.0.5/conf
junjin@ubuntu:~/hbase-2.0.5/conf$
```

编辑目录下的 hbase-env.sh，修改 JAVA 路径

```
# The java implementation to use. Java 1.8+ required.
export JAVA_HOME=/usr/local/java/jdk1.8.0_201
```

修改 HBase 的主配置文件 hbase-site.xml

```
<configuration>
  <property>
    <name>hbase.rootdir</name>
    <value>/home/junjin/hbase_tmp</value>
  </property>
  <property>
    <name>hbase.zookeeper.property.dataDir</name>
    <value>/home/junjin/zookeeper_tmp</value>
  </property>
</configuration>
```

在 bin 目录下启动 Hbase

```
junjin@ubuntu:~$ cd hbase-2.0.5/bin
junjin@ubuntu:~/hbase-2.0.5/bin$ ./start-hbase.sh
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/junjin/hbase-2.0.5/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
```

出现报错，两个 jar 文件出现重复，根据报错提示路径进入文件夹删除任意一个 jar 文件即可。

```
junjin@ubuntu:~/hbase-2.0.5/bin$ ./start-hbase.sh
running master, logging to /home/junjin/hbase-2.0.5/logs/hbase-junjin-master-ubuntu.out
junjin@ubuntu:~/hbase-2.0.5/bin$ jps
16162 Jps
16087 HMaster
junjin@ubuntu:~/hbase-2.0.5/bin$ ./stop-hbase.sh
stopping hbase.....
.....
.....
```

登录之后，利用 jps 验证是否安装成功，出现 HMaster 则是安装成功。

现在安装 Hbase 的伪分布式

添加环境变量，设置 hbase_classpath(指向 hadoop 的配置文件目录)环境变量

```
# The java implementation to use. Java 1.8+ required.
export JAVA_HOME=/usr/local/java/jdk1.8.0_201
export HBASE_CLASSPATH=/usr/local/hadoop/etc/hadoop
```

hbase.rootdir 要配置为 hdfs 上的路径

```
<property>
  <name>hbase.cluster.distributed</name>
  <value>true</value>
</property>
```

启动 hbase，通过 jps 验证，发现多了 HQuorumPeer、HRegionServer

```
junjin@ubuntu:~/hbase-2.0.5/bin$ jps
4755 HMaster
4692 HQuorumPeer
2247 HRegionServer
4879 Jps
```

3. 参考资料

<https://www.jianshu.com/p/510e1d599123>

<https://www.jianshu.com/p/77d8ddc4e7b3>

<https://www.cnblogs.com/rrttp/p/7426771.html>

<https://blog.csdn.net/linlinv3/article/details/49465615>

《Bigtable: A Distributed Storage System for Structured Data》