# HBASE 实验报告

111172 徐鸿飞

## 1. 摘要

HBase 是一个开源的非关系型分布式数据库（NoSQL），它参考了谷歌的 BigTable 建模，实现的编程语言为 Java。它是 Apache 软件基金会的 Hadoop 项目的一部分，运行于 HDFS 文件系统之上，为 Hadoop 提供类似于 BigTable 规模的服务。

本次作业是 HBase 编程实践，主要涉及的是 HBase 的安装与配置，以及 HBase 的一些简单常用 Shell 命令的操作和使用 Java API 调用 HBase 进行操作。

## 2. 目的

本次作业的目的主要是学习 HBase 的安装与配置，以及一些 hbase shell 的常用命令和一些 Java api 的使用。

## 3. 软硬件环境

| | |
|---|---|
| 平台 | VMware 15 WorkStation |
| 硬件 | 2h4g centos 7 |
| 软件 | JAVA 8 |
| | Hadoop 2.10.0 |
| | HBase 1.6.0 |
| IDE | IDEA |

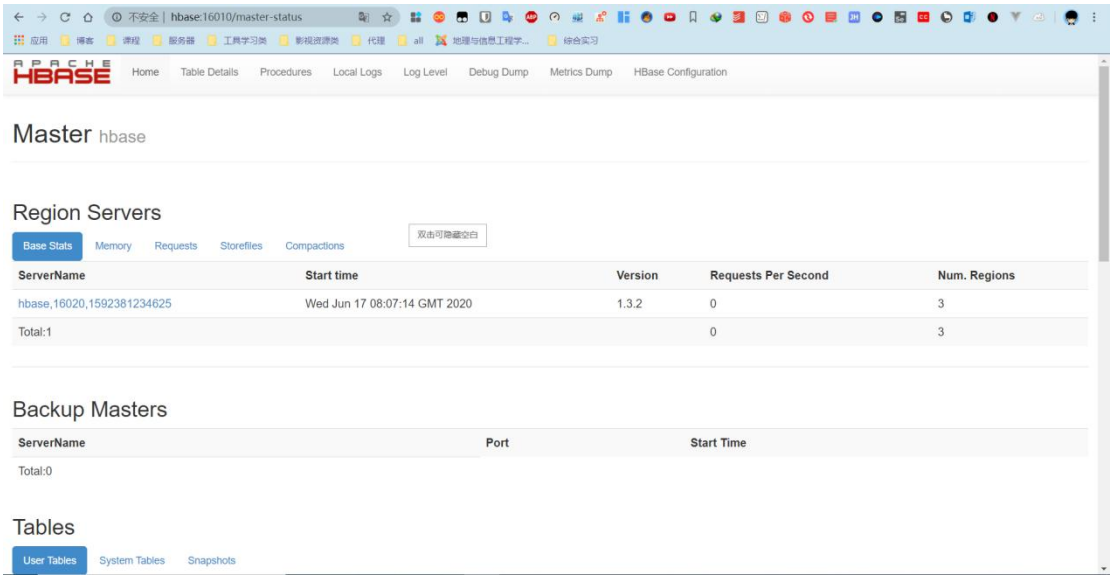
## 4. 数据量

都是一些简单数据，主要是尝试使用 hbase 存取数据。

# 5. 方法/算法

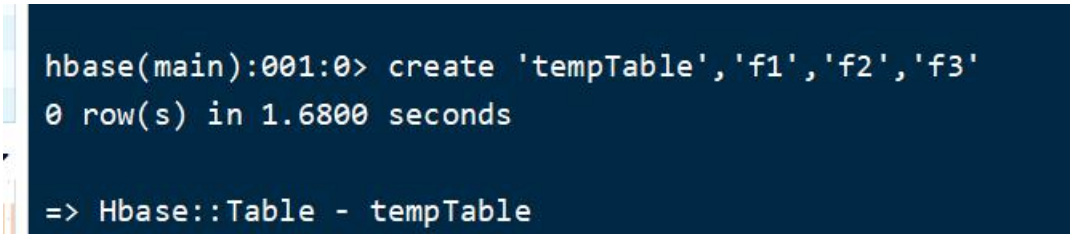
## 5.1. 首先是安装

安装参考了两篇博客：

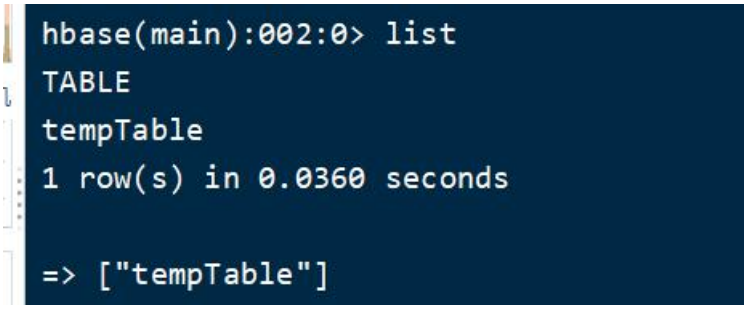



## 5.2. 常用 shell 命令

- create：创建表




- list：列出 HBase 中所有的表信息




- put：向表、行、列指定的单元格添加数据

一次只能为一个表的一行数据的一个列添加一个数据

```
hbase(main):003:0> put 'tempTable','r1','f1:c1','hello,hbase'
0 row(s) in 0.3650 seconds
```

- scan：浏览表的相关信息

```
hbase(main):004:0> scan 'tempTable'
ROW                                   COLUMN+CELL
 r1                                   column=f1:c1, timestamp=1592205301315, value=hello,hbase
1 row(s) in 0.0840 seconds
```

- get：通过表名、行、列、时间戳、时间范围和版本号来获得相应单元格的值

例子 3：

（1）从 tempTable 中，获取第 r1 行、第“f1:c1”列的值

```
hbase(main):005:0> get 'tempTable','r1',{COLUMN=>'f1:c1'}
COLUMN                                CELL
 f1:c1                                timestamp=1592205301315, value=hello,hbase
1 row(s) in 0.0730 seconds
```

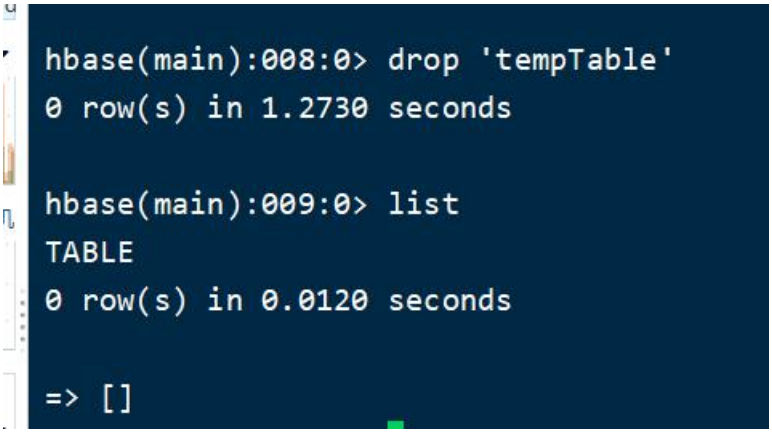
（2）从 tempTable 中，获取第 r1 行、第“f1:c3”列的值

```
hbase(main):006:0> get 'tempTable','r1',{COLUMN=>'f1:c3'}
COLUMN                                CELL
0 row(s) in 0.0060 seconds
```

- enable/disable：使表有效或无效

```
hbase(main):007:0> disable 'tempTable'
0 row(s) in 2.3110 seconds
```

- drop：删除表



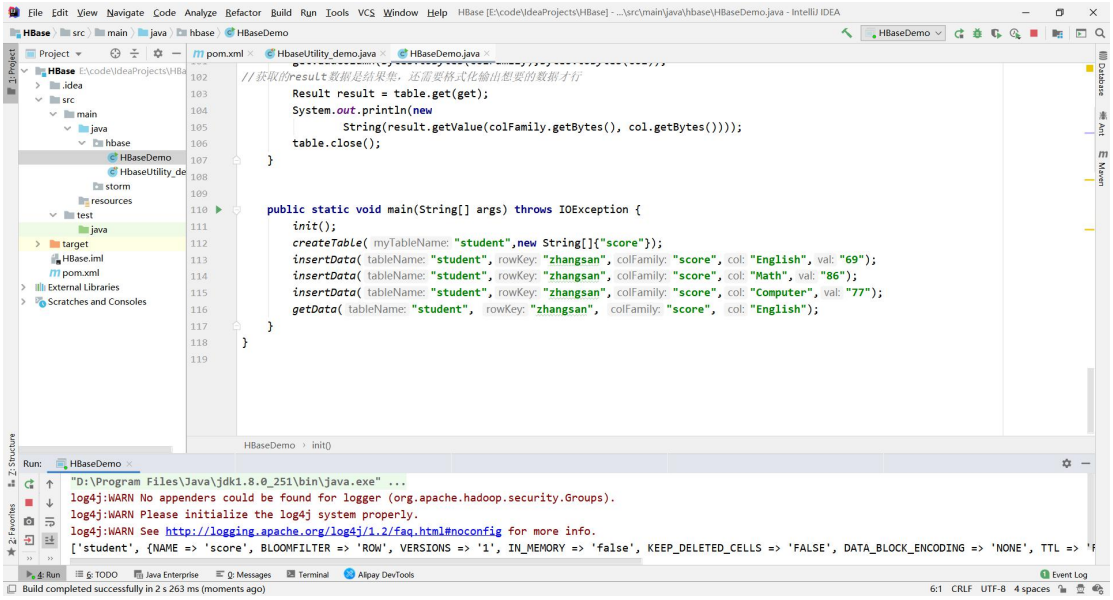

## 5.3. Java API 及应用实例

步骤：

先获取配置对象配置连接属性；

在根据属性获取 hbase 的连接；

最后根据链接获取 admin 来操作 hbase。




一个 HBase 集群主要包含一个 Master 主服务器和多个 Region 服务器；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，分配 Region，负载均衡；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。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 服务器上读取数据客户端并不依赖 Master，而是通过 Zookeeper 来获得 Region 位置信息，大多数客户端甚至从来不和 Master 通信，这种设计方式使得 Master 负载很小。

### 5.4. 碰到的问题

在安装与使用时，碰到了很多问题：

首先就是版本的问题，Hadoop 与 HBase 的版本不对应可能会导致 HBase 无法启动，最后在 http://hbase.apache.org/book.html#basic.prerequisites 中找到对应版本关系，使用了 Hadoop2.10.0+HBase1.6 的组合；

然后是配置的问题，网页无法访问，通过 hostnamectl set-hostname 修改了 hostname 之后可以了；

最后就是在调用 api 时会报错：Exception in thread "main" org.apache.hadoop.hbase.client.RetriesExhaustedException: Failed after attempts=36, exceptions:；经过查找是要改 hosts 文件。

## 6. 结果和分析

虽然 HBase 是一个分布式的 BigTable 数据库，但由于硬件限制，这次只使用了单机的模式来运行。虽然 HBase 可以存储很大的数据，但是他占的空间也很大，有时候很小的文件导入到 HBase 中也会占用很大的空间。

HBase 主要是用在分布式的存储上，单节点无法体现出他的优越性。

## 7. 结论

HBase 在对大数据进行随机读写和实时的访问时速度很快。在数据的拓展上很方便。

HBase 可以支持 Native Java API、HBase Shell、Thrift Gateway、REST Gateway、Pig、Hive 等多种访问接口，可以根据具体应用场合选择相应访问方式； HBase 实际上就是一个稀疏、多维、持久化存储的映射表，它采用行键、列键和时间戳进行索引，每个值都是未经解释的字符串。

HBase 采用分区存储，一个大的表会被分拆许多个 Region，这些 Region 会被分发到不同的服务器上实现分布式存储

## 8. 参考

Hadoop 安装：https://blog.csdn.net/twypx/article/details/81236403

HBase 安装：https://blog.csdn.net/twypx/article/details/104201410