HBase 实验报告

151180045 侯汶昕

1. 实验目的

熟悉基本的 HBase Shell 操作和 HBase 程序设计；编写 Java 程序建表。

2. 实验环境

实验中使用了两台虚拟机，虚拟机具有相同的环境和配置：

系统：ubuntu 16.04 LTS 64-bit

内存：1.9GB

软件版本为：Hadoop-2.9.1, HBase-2.1.1

3. 实验内容及结果

3.1 HBase 的安装及配置

3.1.1 从官网下载 HBase：

https://www.apache.org/dyn/closer.lua/hbase/2.1.1/hbase-2.1.1-bin.tar.gz

3.1.2 配置 hbase-site.xml

在文件中添加：

```
<property>
        <name>hbase.cluster.distributed</name>
        <value>true</value>
</property>
<property>
        <name>hbase.rootdir</name>
        <value>hdfs://host-0:9000/hbase</value>
</property>
```

*host-0 为 hosts 文件中设置的本地 IP 地址。

3.1.2 配置 hbase-env.sh

修改 JAVA_HOME 为本机的 JAVA 运行环境；将 export HBASE_MANAGES_ZK=true 的注释符号去掉，以便使用 HBase 自带的 ZooKeeper。

3.2 HBase Shell 的基本操作

3.2.1 打开 Shell 并查看所有的表

```
houwenxin@ubuntu:~/hbase_installs/hbase-2.1.1$ bin/hbase shell
2018-12-02 02:59:32,150 WARN  [main] util.NativeCodeLoader: Unable to load nativ
e-hadoop library for your platform... using builtin-java classes where applicabl
e
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
For Reference, please visit: http://hbase.apache.org/2.0/book.html#shell
Version 2.1.1, rb60a92d6864ef27295027f5961cb46f9162d7637, Fri Oct 26 19:27:03 PD
T 2018
Took 0.0127 seconds
hbase(main):001:0> list
TABLE
students
1 row(s)
Took 2.4055 seconds
=> ["students"]
hbase(main):002:0>
```

3.2.2 创建并描述表

```
hbase(main):008:0> create 'test', 'name', 'id', 'phone'
Created table test
Took 1.3688 seconds
=> Hbase::Table - test
hbase(main):009:0> describe 'test'
Table test is ENABLED
test
COLUMN FAMILIES DESCRIPTION
{NAME => 'id', VERSIONS => '1', EVICT_BLOCKS_ON_CLOSE => 'false', NEW_VERSION_BE
HAVIOR => 'false', KEEP_DELETED_CELLS => 'FALSE', CACHE_DATA_ON_WRITE => 'false'
, DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', MIN_VERSIONS => '0', REPLICAT
ION_SCOPE => '0', BLOOMFILTER => 'ROW', CACHE_INDEX_ON_WRITE => 'false', IN_MEMO
RY => 'false', CACHE_BLOOMS_ON_WRITE => 'false', PREFETCH_BLOCKS_ON_OPEN => 'fal
se', COMPRESSION => 'NONE', BLOCKCACHE => 'true', BLOCKSIZE => '65536'}
{NAME => 'name', VERSIONS => '1', EVICT_BLOCKS_ON_CLOSE => 'false', NEW_VERSION_
BEHAVIOR => 'false', KEEP_DELETED_CELLS => 'FALSE', CACHE_DATA_ON_WRITE => 'fals
e', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', MIN_VERSIONS => '0', REPLIC
ATION_SCOPE => '0', BLOOMFILTER => 'ROW', CACHE_INDEX_ON_WRITE => 'false', IN_ME
MORY => 'false', CACHE_BLOOMS_ON_WRITE => 'false', PREFETCH_BLOCKS_ON_OPEN => 'f
alse', COMPRESSION => 'NONE', BLOCKCACHE => 'true', BLOCKSIZE => '65536'}
{NAME => 'phone', VERSIONS => '1', EVICT_BLOCKS_ON_CLOSE => 'false', NEW_VERSION
_BEHAVIOR => 'false', KEEP_DELETED_CELLS => 'FALSE', CACHE_DATA_ON_WRITE => 'fal
se', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', MIN_VERSIONS => '0', REPLI
CATION_SCOPE => '0', BLOOMFILTER => 'ROW', CACHE_INDEX_ON_WRITE => 'false', IN_M
EMORY => 'false', CACHE_BLOOMS_ON_WRITE => 'false', PREFETCH_BLOCKS_ON_OPEN => '
false', COMPRESSION => 'NONE', BLOCKCACHE => 'true', BLOCKSIZE => '65536'}
3 row(s)
Took 0.2252 seconds
```

3.2.3 添加记录并扫描表

```
hbase(main):011:0> put 'test', 'row1', 'name', 'Wenxin'
Took 0.0778 seconds
hbase(main):012:0> put 'test', 'row2', 'name', 'Shuai'
Took 0.0291 seconds
hbase(main):013:0> put 'test', 'row1', 'id', '151180045'
Took 0.0065 seconds
hbase(main):014:0> put 'test', 'row2', 'id', '151180042'
Took 0.0112 seconds
hbase(main):015:0> put 'test', 'row1', 'phone', '123456'
Took 0.0172 seconds
hbase(main):016:0> put 'test', 'row2', 'phone', '654321'
Took 0.0347 seconds
hbase(main):017:0> scan 'test'
ROW                                             COLUMN+CELL
 row1                                           column=id:, timestamp=1543749284864, value=151180045
 row1                                           column=name:, timestamp=1543749260569, value=Wenxin
 row1                                           column=phone:, timestamp=1543749305456, value=123456
 row2                                           column=id:, timestamp=1543749294852, value=151180042
 row2                                           column=name:, timestamp=1543749269120, value=Shuai
 row2                                           column=phone:, timestamp=1543749313279, value=654321
2 row(s)
Took 0.1751 seconds
```

```
hbase(main):018:0> scan 'test', {COLUMNs=>'name'}
NameError: uninitialized constant COLUMNs

hbase(main):019:0> scan 'test', {COLUMNS=>'name'}
ROW                                             COLUMN+CELL
 row1                                           column=name:, timestamp=1543749260569, value=Wenxin
 row2                                           column=name:, timestamp=1543749269120, value=Shuai
2 row(s)
Took 0.0789 seconds
hbase(main):020:0> get 'test', 'row1'
COLUMN                                          CELL
 id:                                            timestamp=1543749284864, value=151180045
 name:                                          timestamp=1543749260569, value=Wenxin
 phone:                                         timestamp=1543749305456, value=123456
1 row(s)
Took 0.2079 seconds
```

3.2.4 删除表

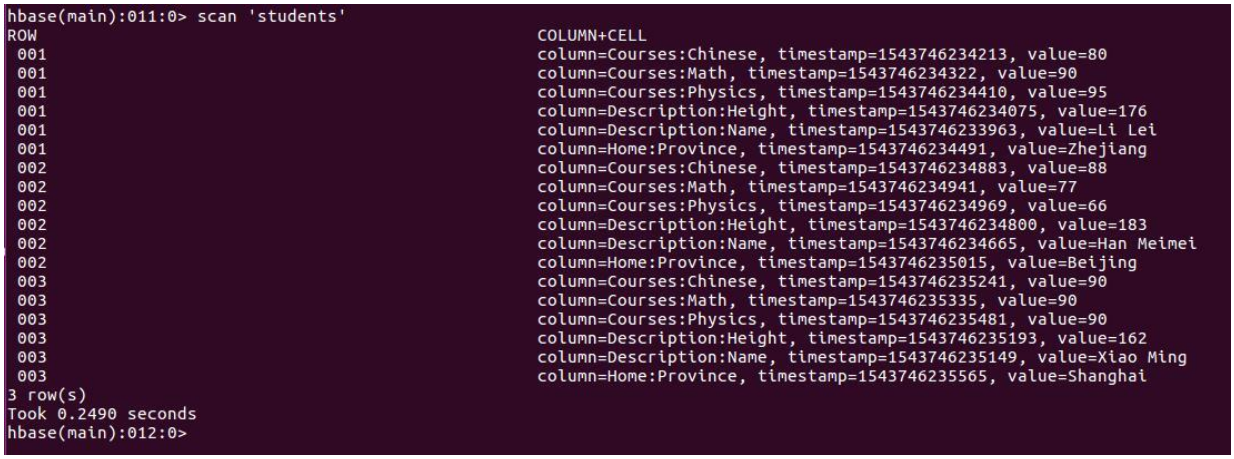
```
hbase(main):023:0> disable 'test'
Took 30.6214 seconds
hbase(main):024:0> drop 'test'
Took 1.5400 seconds
```

3.3 编写 Java 程序，创建如下的 students 表

ID	Description		Courses			Home
	Name	Height	Chinese	Math	Physics	Province
001	Li Lei	176	80	90	95	Zhejiang
002	Han Meimei	183	88	77	66	Beijing
003	Xiao Ming	162	90	90	90	Shanghai

运行结果：

右下图可见，该程序成功的创建了 students 表。



4. 实验总结

本次实验主要是为了熟悉 HBase 的 Shell 操作和编程接口，基本实现上没有什么难度，不过细节方面还是有很多需要注意的地方。比如 hbase-env.sh 和 hbase-site.xml 文件的配置，运行 HBase 前要先启动 HDFS，Java 程序需要添加 Java 库的依赖，将 hbase-site.xml 文件放入 class path 中等。还有一点就是很多书上和网上提供的接口都已经不适用新版本的 HBase 了，因此需要习惯自己去官网查看文档查看用法。