南京邮电大学

云计算技术

题　　目：　HBase 安装和使用

学号　　B19040229

姓名　　[redacted]

日期　　2022 年 6 月 1 日

一、实验目的

（1） 官网下载与 Hadoop 版本匹配的 Hbase 版本；

（2） 了解学习 Hbase 的分布式存储原理；

（3） 熟悉 Hbase 的伪分布式的搭建；

（4） 学习掌握 HBase 的表设计；

二、 实验内容

（1） 下载与安装的 Hadoop 相匹配的 Hbase 安装包；

（2） 查看官方网文档进行 Hbase 的伪分布式的部署；

（3） 了解 Hbase 底层数据存储模型；

（4） 自定义一个表；

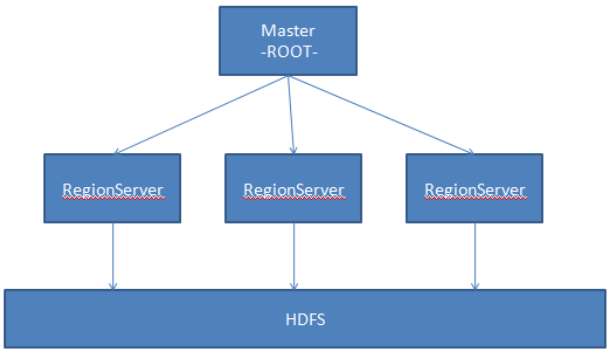
三、 实验步骤

I. 基础概念

Hbase 是一个分布式的数据库，同时它也是非结构化的；

- 使用 Zookeeper 来管理集群。
- 在架构层面上分为 Master（Zookeeper 中的 leader）和多个 RegionServer，

基本架构如图：



在 Hbase 的概念中，RegionServer 对应于集群中的一个节点，而一个 RegionServer 负责管理多个 Region。一个 Region 代 表一张表的一部分数据，所以在 Hbase 中的一张表可能会需要很多个 Region 来存储其数据， 但是每个 Region 中的数据并不是杂乱无章的，Hbase 在管理 Region 的时候会给每个 Region 定义一个 Rowkey 的范围，落在特定范围内的数据将交给特定的 Region，从而将负载分 摊到多个节点上，充分利用分布式的优点。另外，Hbase 会自动的调节 Region 处在的位置，如果一个 RegionServer 变得 Hot（大量的请求 落在这个 Server 管理的 Region 上），Hbase 就会把 Region 移动到相对空闲的节点，依次保证集群环境被充分利用。

LSM 树的简易模型是由两个树状结构组成，这两个树分别是 C0 和 C1。C0 比较小，并且全部存储于内存之中，而 C1 则存储于磁盘之上，一条新的记录先从 C0 中插入，插入时会判断 C0 中的内存阀值，如果超出了阀值，C0 中的某些数据片段会被迁移并合并到 C1 的树上。由于合并排序算法是批量的且是顺序存储，所以说速度是非常快的。当然这只是简易模

型，在实际应用时，是多余两层树结构的。

Hbase 采用的是 LSM 树的结构，这种结构的关键是：

- 每一次的插入操作都会先进入 MemStore（内存缓冲区）
- 当 MemStore 达到上限的时候，Hbase 会将内存中的数据输出为有序的 StoreFile 文件数据（根据 Rowkey、版本、列名排序，这里已经和列 簇无关了因为 Store 里都属于同一个列簇）。
- 这样会在 Store 中形成很多个小的 StoreFile，当这些小的 File 数量达到一个阀值的时 候，Hbase 会用一个线程来把这些小 File 合并成一个大的 File。

Row Key	column-family1		column-family2			column-family3
	column1	column2	column1	column2	column3	column1
key1						
key2						
key3						

Hbase **表结构**

主要表结构元素：

1. **Row Key:**和其他 NoSQL 一样，是用来检索的记录的主键。存储时，数据按照 Key 的字典序进行排序。
2. **列族 Column-family:** Hbase 表中的每个列，都归属与某个列族。列族是表的 chema 的一部分(而列不是)，必须在使用表之前定义。列名都以列族作为前缀。例如 courses:history，courses:math 都属于 courses 这个列族。
3. **单元 Cell：** HBase 中通过 row 和 columns 确定的为一个存贮单元称为 cell。由{row key, column(=<family> + <label>), version} 唯一确定的单元。cell 中的数据是没有类型的，全部是字节码形式存贮。
4. 时间戳 timestamp：每个 cell 都保存着同一份数据的多个版本。版本通过时间戳来索引。时间戳的类型是 64 位整型。时间戳可以由 Hbase(在数据写入时自动)赋值，此时时间戳是精确到毫秒的当前系统时间。时间戳也可以由客户显式赋值。如果应用程序要避免数据版本冲突，就必须自己生成具有唯一性的时间戳。每个 cell 中，不同版本的数据按照时间倒序排序，即最新的数据排在最前面。

II. Hbase 部署安装（命令行安装、Docker 安装）

A. Hbase 命令行安装

1. 访问 Hbase 官网下载安装

在 Window 本地下载完成 Hbase 后使用 FTP 工具上传至云服务器或者使用 wget 工具下载，解压并且运行 `tar -zxvf hbase-2.4.1.tar.gz -C /home`

HBase 常用目录结构：

bin：存放 HBase 所有可执行命令与脚本。

conf：HBase 配置文件存放目录。

hbase-webapps：存放 Web 应用的目录，这些应用可以查看 HBase 的运行状态

lib：存放 HBase jar 包，包括第三方依赖包以及与 Hadoop 相关的 jar 文件

logs：日志存放的目录。

2. 配置系统环境变量
   - 运行 `Vim /etc/profile`
   - 将下面内容追加：

```
export HBASE_HOME=/home/hbase
#在 path 里增加路径
export PATH=$PATH:$HBASE_HOME/bin
```

   - 运行 `source /etc/profile` 使其生效
3. 配置 hbase-env.sh

   在 hbase 的 conf 目录下输入：`vi hbase-env.sh`

   追加下面内容到末尾:

```
# jdk 路径
export JAVA_HOME=/opt/jdk1.8.0_221
# 使用外部 Zookeeper
export HBASE_MANAGES_ZK=false
```

4. 配置 hbase-site.xml

   输入 `vi hbase-site.xml` 进入 hbase 的 conf 目录下

   在<configuration></configuration>中增加:

```
<property>
  <name>hbase.rootdir</name>
  <value>hdfs://hadoop101:9000/hbase</value>
</property>
<!-- 默认是 false，单机模式-->
<property>
  <name>hbase.cluster.distributed</name>
  <value>true</value>
</property>
<!-- 0.98 后的新变动，之前版本没有.port,默认端口为 60000 -->
<property>
 <name>hbase.master.port</name>
 <value>16000</value>
</property>
<property>
    <name>hbase.zookeeper.property.clientPort</name>
    <value>2181</value>
</property>
<property>
    <name>hbase.zookeeper.quorum</name>
    <value>hadoop101</value>
</property>
<property>
 <name>hbase.zookeeper.property.dataDir</name>
```

```
25.  <value>/opt/zkpr/zkData</value>
26.</property>
```

5. 修改 regionservers 文件

   在 hbase 的 conf 目录下输入：`vim regionservers` 追加 `hadoop101` 到文件末尾

6. 启动 Hbase

   ```
   [tanyongfeng@hadoop101 hbase]$ $HBASE_HOME/bin/start-hbase.sh
   ```

7. 验证执行 JPS 出现 HMaster。或者访问 http://hadoop101:60010 查看运行状态

B .Docker 安装

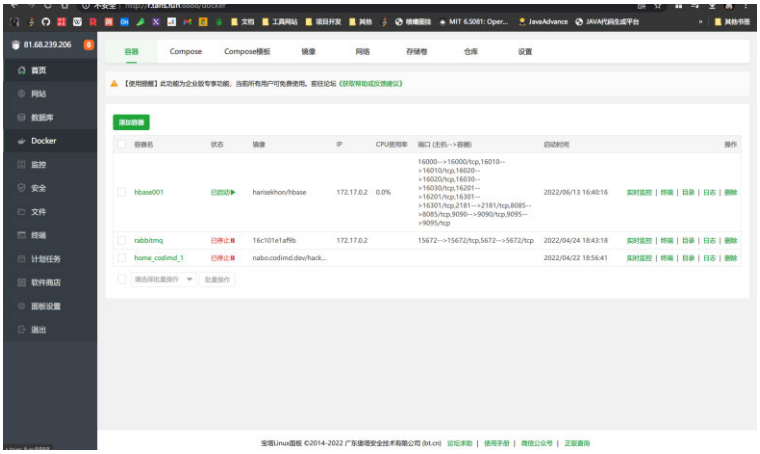
这里也使用 Docker 方法进行安装 HBase，也非常简便：

1. 拉取 HBase 镜像并启动：
   - `docker search hbase`
   - `docker pull harisekhon/hbase`
   - `docker run -d -p 2181:2181 -p 8085:8085 -p 9090:9090 -p 9095:9095 -p 16000:16000 -p 16010:16010 -p 16201:16201 -p 16301:16301 -p 16030:16030 -p 16020:16020 --name hbase001 harisekhon/hbase`

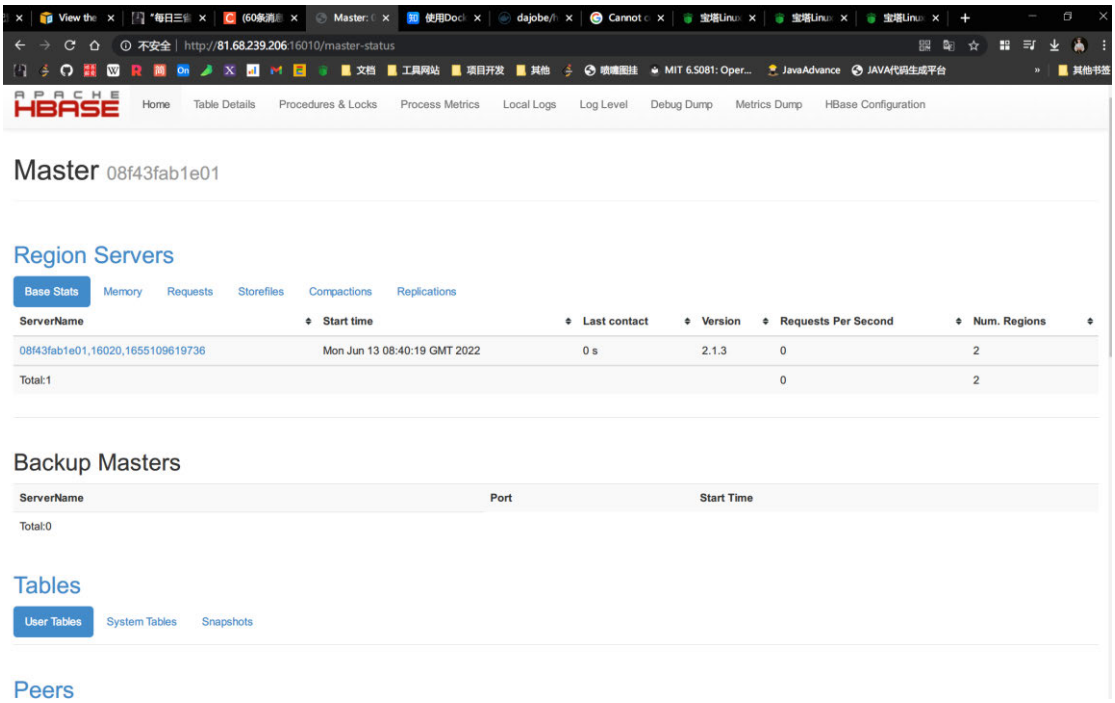
2. 服务器安全组放行服务器 16010 端口，或者在服务器直接使用 wget 命令访问。

```
[root@VM-4-6-centos ~]# wget localhost:16010
--2022-06-13 16:41:55--  http://localhost:16010/
Connecting to 127.0.0.1:7890... connected.
Proxy request sent, awaiting response... 200 OK
Length: 876 [text/html]
Saving to: 'index.html'

index.html                  100%[=======================================>]     876  --.-KB/s    in 0s

2022-06-13 16:41:55 (110 MB/s) - 'index.html' saved [876/876]
```

3. 本地远程访问服务器查看:



Docker 镜像启动 Hbase



Hbase 管理界面

4. 进入 Hbase 执行相关命令

```
bash-4.4# ./shell.sh
bash: ./shell.sh: No such file or directory
bash-4.4# ./hbase shell
f2022-06-13 08:57:33,310 WARN  [main] util.NativeCodeLoader: Unable to load native-hadoop librar
y for your platform... using builtin-java classes where applicable
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 2.1.3, rda5ec9e4c06c537213883cca8f3cc9a7c19daf67, Mon Feb 11 15:45:33 CST 2019
Took 0.0047 seconds
hbase(main):001:0> status
1 active master, 0 backup masters, 1 servers, 0 dead, 2.0000 average load
Took 0.6443 seconds
hbase(main):002:0>
```

Demo1. Create 命令演示（建表语句）:

```
Took 0.0117 seconds
hbase(main):004:0> create 't1', {NAME => 'f1'}, {NAME => 'f2'}, {NAME => 'f3'}
Created table t1
Took 0.9091 seconds
=> Hbase::Table - t1
```

Demo2. Put 命令演示（添加记录）:

```
Took 0.1042 seconds
hbase(main):012:0> put 't1', 'tan', 'f1', '1'
Took 0.0588 seconds
hbase(main):013:0> put 't1', 'tan', 'f2', '1'
Took 0.0034 seconds
hbase(main):014:0> put 't1', 'tan', 'f3', '1'
Took 0.0056 seconds
```

Demo3. Get 命令演示（获取 row key 对应的键值对）

```
Took 0.0034 seconds
hbase(main):021:0> get 't1', 'tan'
COLUMN                          CELL
 f1:                            timestamp=1655114366110, value=1
 f2:                            timestamp=1655114370738, value=1
 f3:                            timestamp=1655114373520, value=1
1 row(s)
```

Demo4. Scan 命令演示（打印所有的键值对）:

```
Took 0.0373 seconds
hbase(main):022:0> scan 't1'
ROW                             COLUMN+CELL
 tan                            column=f1:, timestamp=1655114366110, value=1
 tan                            column=f2:, timestamp=1655114370738, value=1
 tan                            column=f3:, timestamp=1655114373520, value=1
1 row(s)
Took 0.0103 seconds
```

Demo5. List 命令演示（打印所有的数据库）

```
Took 0.0105 seconds
hbase(main):023:0> ls
NameError: undefined local variable or method `ls' for main:Object

hbase(main):024:0> list
TABLE
t1
1 row(s)
Took 0.0081 seconds
=> ["t1"]
hbase(main):025:0>
```

5. 参考资料

1. Docker 仓库
2. 使用 Docker 学习 Hbase
3. Hbase 入门
4. Hbase 命令大全

四、实验结果与分析

如上部分所示

五、实验总结

通过本次实验，对分布式存储设计原理有了一定的了解，同时通过使用普通命令安装以及 Docker 安装两种方法，熟悉 Hbase 的搭建。熟悉 Hbase shell 中常用的例如 list, get, put, scan 等基本命令的使用。同时不同于所学习的关系型数据库，Hbase 作为一种非结构化数据存储的数据库，在很多方面有着比较多的优点。