

HBase 的安装和基础编程

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。HBase 的目标是处理非常庞大的表,可以通过水平扩展的方式,利用廉价的计算机集群处理超过 10 亿行数据和数百万列元素组成的数据表。Hadoop 安装以后,不包含 HBase 组件,需要另外安装。

本章首先介绍 HBase 的安装方法,并介绍 HBase 的两种不同模式的配置方法,包括单机模式和伪分布式模式;其次介绍一些操作 HBase 常用的 Shell 命令;最后,介绍如何使用 Eclipse 开发可以操作 HBase 数据库的 Java 应用程序。

5.1 安装 HBase

本节介绍 HBase 的安装方法,包括下载安装文件、配置环境变量、添加用户权限等。

5.1.1 下载安装文件

HBase 是 Hadoop 生态系统中的一个组件,但是,Hadoop 安装以后,本身并不包含 HBase,因此,需要单独安装 HBase。登录 Linux 系统,在 Linux 系统(不是 Windows 系统)中打开火狐浏览器,访问本书官网的“下载专区”,在“软件”目录下下载 HBase 安装文件 hbase-2.2.2-bin.tar.gz。火狐浏览器会默认把下载文件都保存到当前用户的下载目录,由于本书全部采用 hadoop 用户登录 Linux 系统,所以,hbase-2.2.2-bin.tar.gz 文件会被保存到 /home/hadoop/下载/目录下。

需要注意的是,如果是在 Windows 系统下载安装文件 hbase-2.2.2-bin.tar.gz,则需要通过 FTP 软件上传到 Linux 系统的“/home/hadoop/下载/”目录下,这个目录是本书所有安装文件的中转站。

下载完安装文件后,需要对文件进行解压。按照 Linux 系统使用的默认规范,用户安装的软件一般都存放在 /usr/local/目录下。使用 hadoop 用户登录 Linux 系统,打开一个终端,执行如下命令:

```
$sudo tar -zxvf ~/下载/hbase-2.2.2-bin.tar.gz -C /usr/local
```

将解压的文件名 hbase-2.2.2 改为 hbase,以方便使用,命令如下:

```
$sudo mv /usr/local/hbase-2.2.2 /usr/local/hbase
```

5.1.2 配置环境变量

将 HBase 安装目录下的 bin 目录(即/usr/local/hbase/bin)添加到系统的 PATH 环境变量中,这样,每次启动 HBase 时就不需要到/usr/local/hbase 目录下执行启动命令,方便 HBase 的使用。使用 vim 编辑器打开 ~/.bashrc 文件,命令如下:

```
$vim ~/.bashrc
```

打开.bashrc 文件后,可以看到,已经存在如下 PATH 环境变量的配置信息,因为,之前在第 3 章安装 Hadoop 时,已经为 Hadoop 添加了 PATH 环境变量的配置信息:

```
$export PATH=$PATH:/usr/local/hadoop/sbin:/usr/local/hadoop/bin
```

这里需要把 HBase 的 bin 目录/usr/local/hbase/bin 追加到 PATH 中。当要在 PATH 中继续加入新的路径时,只要用英文冒号“:”隔开,把新的路径加到后面即可,追加后的结果如下:

```
$export PATH=$PATH:/usr/local/hadoop/sbin:/usr/local/hadoop/bin:/usr/
local/hbase/bin
```

添加后执行如下命令使设置生效:

```
$source ~/.bashrc
```

5.1.3 添加用户权限

需要为当前登录 Linux 系统的 hadoop 用户添加访问 HBase 目录的权限,将 HBase 安装目录下的所有文件的所有者改为 hadoop,命令如下:

```
$cd /usr/local
$sudo chown -R hadoop ./hbase
```

5.1.4 查看 HBase 版本信息

可以通过如下命令查看 HBase 版本信息,以确认 HBase 已经安装成功:

```
$/usr/local/hbase/bin/hbase version
```

执行上述命令以后,如果出现如图 5-1 所示的信息,则说明安装成功。

```
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
HBase 2.2.2
Source code repository git://6ad68c41b902/opt/hbase-rm/output/hbase revision=e6513a76c91cceda95dad7af246ac81d46fa2589
Compiled by hbase-rm on Sat Oct 19 10:10:12 UTC 2019
From source with checksum 4d23f97701e395c5d34db1882ac5021b
```

图 5-1 查看 HBase 版本信息

5.2 HBase 的配置

HBase 有 3 种运行模式,即单机模式、伪分布式模式和分布式模式。

(1) 单机模式:采用本地文件系统存储数据。

(2) 伪分布式模式:采用伪分布式模式的 HDFS 存储数据。

(3) 分布式模式:采用分布式模式的 HDFS 存储数据。

本书仅介绍单机模式和伪分布式模式。

在进行 HBase 配置之前,需要确认已经安装了 3 个组件:JDK、Hadoop、SSH。HBase 单机模式不需要安装 Hadoop,伪分布式模式和分布式模式需要安装 Hadoop。JDK、Hadoop 和 SSH 的安装方法,已经在第 3 章做了详细介绍,如果已经按照第 3 章的方法安装了 Hadoop,则这里不需要另外安装 JDK、Hadoop 和 SSH。

5.2.1 单机模式配置

1. 配置 hbase-env.sh 文件

使用 vim 编辑器打开/usr/local/hbase/conf/hbase-env.sh,命令如下:

```
$vim /usr/local/hbase/conf/hbase-env.sh
```

打开 hbase-env.sh 文件后,需要在 hbase-env.sh 文件中配置 JAVA 环境变量,在第 3 章已经配置了 JAVA_HOME=/usr/lib/jvm/jdk1.8.0_162,这里可以直接复制该配置信息到 hbase-env.sh 文件中。此外,还需要添加 Zookeeper 配置信息,配置 HBASE_MANAGES_ZK 为 true,表示由 HBase 自己管理 Zookeeper,不需要单独的 Zookeeper,由于 hbase-env.sh 文件中本来就存在这些变量的配置,因此,只需要删除前面的注释符号#并修改配置内容即可,修改后的 hbase-env.sh 文件应该包含如下两行信息:

```
export JAVA_HOME=/usr/lib/jvm/jdk1.8.0_162
export HBASE_MANAGES_ZK=true
```

修改完成以后,保存 hbase-env.sh 文件并退出 vim 编辑器。

2. 配置 hbase-site.xml 文件

使用 vim 编辑器打开并编辑/usr/local/hbase/conf/hbase-site.xml 文件,命令如下:

```
$vim /usr/local/hbase/conf/hbase-site.xml
```

在 hbase-site.xml 文件中,需要设置属性 hbase.rootdir,用于指定 HBase 数据的存储位置,如果没有设置,则 hbase.rootdir 默认为/tmp/hbase- \${user.name},这意味着每次重启系统都会丢失数据。这里把 hbase.rootdir 设置为 HBase 安装目录下的 hbase-tmp 文件夹,即/usr/local/hbase/hbase-tmp,修改后的 hbase-site.xml 文件中的配置信息如下:

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

保存 hbase-site.xml 文件,并退出 vim 编辑器。

3. 启动并运行 HBase

现在就可以测试运行 HBase,命令如下:

```
$cd /usr/local/hbase
$bin/start-hbase.sh          #启动 HBase
$bin/hbase shell             #进入 HBase Shell 命令行模式
```

进入 HBase Shell 命令行模式以后,用户可以通过输入 Shell 命令操作 HBase 数据库。成功启动 HBase 后会出现图 5-2 所示的界面。

```
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 2.2.2, re6513a76c91cceda95dad7af246ac81d46fa2589
, Sat Oct 19 10:10:12 UTC 2019
Took 0.0079 seconds
hbase(main):001:0>
```

图 5-2 进入 HBase Shell 模式

可以使用如下命令停止 HBase 运行:

```
$bin/stop-hbase.sh
```

需要说明的是,如果在操作 HBase 的过程中发生错误,可以查看{HBASE_HOME}目录(即/usr/local/hbase)下的 logs 子目录中的日志文件,寻找可能的错误原因,然后搜索网络资料寻找相关解决方案。

5.2.2 伪分布式模式配置

1. 配置 hbase-env.sh 文件

使用 vim 编辑器打开/usr/local/hbase/conf/hbase-env.sh,命令如下:

```
$vim /usr/local/hbase/conf/hbase-env.sh
```

打开 hbase-env.sh 文件后,需要在 hbase-env.sh 文件中配置 JAVA_HOME、HBASE_CLASSPATH 和 HBASE_MANAGES_ZK。其中,HBASE_CLASSPATH 设置为本机 hbase 安装目录下的 conf 目录(即/usr/local/hbase/conf)。JAVA_HOME 和 HBASE_MANAGES_ZK 的配置方法和上面单机模式配置方法相同。修改后的 hbase-env.sh 文件应该包含如下 3 行信息:

```
export JAVA_HOME=/usr/lib/jvm/jdk1.8.0_162
export HBASE_CLASSPATH=/usr/local/hbase/conf
export HBASE_MANAGES_ZK=true
```

修改完成以后,保存 hbase-env.sh 文件并退出 vim 编辑器。

2. 配置 hbase-site.xml 文件

使用 vim 编辑器打开并编辑/usr/local/hbase/conf/hbase-site.xml 文件,命令如下:

```
$vim /usr/local/hbase/conf/hbase-site.xml
```

在 hbase-site.xml 文件中,需要设置属性 hbase.rootdir,用于指定 HBase 数据的存储位置。在 HBase 伪分布式模式中,使用伪分布式模式的 HDFS 存储数据,因此,需要把 hbase.rootdir 设置为 HBase 在 HDFS 上的存储路径。根据第 3 章 Hadoop 伪分布式模式配置可以知道,HDFS 的访问路径为 hdfs://localhost:9000/,因为,这里设置 hbase.rootdir 为 hdfs://localhost:9000/hbase。此外,由于采用了伪分布式模式,因此,还需要将属性 hbase.cluster.distributed 设置为 true。修改后的 hbase-site.xml 文件中的配置信息如下:

```
<configuration>
  <property>
    <name>hbase.rootdir</name>
    <value>hdfs://localhost:9000/hbase</value>
  </property>
  <property>
    <name>hbase.cluster.distributed</name>
    <value>true</value>
  </property>
```

```
<property>
  <name>hbase.unsafe.stream.capability.enforce</name>
  <value>false</value>
</property>
</configuration>
```

保存 hbase-site.xml 文件,并退出 vim 编辑器。

3. 启动运行 HBase

首先登录 SSH,由于之前在第 3 章已经设置了无密码登录,因此这里不需要密码;然后切换至 /usr/local/hadoop,启动 Hadoop,让 HDFS 进入运行状态,从而可以为 HBase 存储数据,具体命令如下:

```
$ssh localhost
$cd /usr/local/hadoop
$./sbin/start-dfs.sh
```

输入命令 jps,如果能够看到 NameNode、DataNode 和 SecondaryNameNode 这 3 个进程,则表示已经成功启动 Hadoop。

然后启动 HBase,命令如下:

```
$cd /usr/local/hbase
$bin/start-hbase.sh
```

输入命令 jps,如果出现以下进程,则说明 HBase 启动成功:

```
Jps
HMaster
HQuorumPeer
NameNode
HRegionServer
SecondaryNameNode
DataNode
```

现在可以进入 HBase Shell 模式,命令如下:

```
$bin/hbase shell      #进入 HBase Shell 命令行模式
```

进入 HBase Shell 命令行模式以后,用户可以通过输入 Shell 命令操作 HBase 数据库。

4. 停止运行 HBase

最后可以使用如下命令停止运行 HBase:

```
$bin/stop-hbase.sh
```

如果在操作 HBase 的过程中发生错误,可以查看 {HBASE_HOME} 目录(即/usr/local/hbase)下的 logs 子目录中的日志文件,寻找可能的错误原因。

关闭 HBase 以后,如果不再使用 Hadoop,就可以运行如下命令关闭 Hadoop:

```
$cd /usr/local/hadoop  
$./sbin/stop-dfs.sh
```

需要注意的是,启动、关闭 Hadoop 和 HBase 时,一定要按照启动 Hadoop→启动 HBase→关闭 HBase→关闭 Hadoop 的顺序。

5.3 HBase 常用的 Shell 命令

在使用具体的 Shell 命令操作 HBase 数据之前,需要先启动 Hadoop,再启动 HBase 和 HBase Shell,进入 Shell 命令提示符状态,具体命令如下:

```
$cd /usr/local/hadoop  
$./sbin/start-dfs.sh  
$cd /usr/local/hbase  
$./bin/start-hbase.sh  
$./bin/hbase shell
```

5.3.1 在 HBase 中创建表

假设这里要创建一个表 student,该表包含 Sname、Ssex、Sage、Sdept、course 5 个字段。需要注意的是,在关系数据库(如 MySQL)中,需要先创建数据库,再创建表,但是,在 HBase 数据库中,不需要创建数据库,只要直接创建表就可以。在 HBase 中创建 student 表的 Shell 命令如下:

```
hbase>create 'student','Sname','Ssex','Sage','Sdept','course'
```

对于 HBase 而言,在创建 HBase 表时,不需要自行创建行键,系统会默认一个属性作为行键,通常是把 put 命令操作中跟在表名后的第一个数据作为行键。

创建完 student 表后,可通过 describe 命令查看 student 表的基本信息。describe 命令及其执行结果如图 5-3 所示。

可以使用 list 命令查看当前 HBase 数据库中已经创建的表,命令如下:

```
hbase>list
```

5.3.2 添加数据

HBase 使用 put 命令添加数据,一次只能为一个表的一行数据的一个列(也就是一个单

```

hbase(main):009:0> describe 'student'
DESCRIPTION
'student', {NAME => 'Sage', DATA_BLOCK_ENCODING => 'NONE', BLOOMFILTER => 'NONE', REPLICATION_SCOPE => '0', VERSIONS => '3', COMPRESSION => 'NONE', MIN_VERSIONS => '0', TTL => '2147483647', KEEP_DELETED_CELLS => 'false', BLOCKSIZE => '65536', IN_MEMORY => 'false', ENCODE_ON_DISK => 'true', BLOCKCACHE => 'true'}, {NAME => 'Sdept', DATA_BLOCK_ENCODING => 'NONE', BLOOMFILTER => 'NONE', REPLICATION_SCOPE => '0', VERSIONS => '3', COMPRESSION => 'NONE', MIN_VERSIONS => '0', TTL => '2147483647', KEEP_DELETED_CELLS => 'false', BLOCKSIZE => '65536', IN_MEMORY => 'false', ENCODE_ON_DISK => 'true', BLOCKCACHE => 'true'}, {NAME => 'Ssex', DATA_BLOCK_ENCODING => 'NONE', BLOOMFILTER => 'NONE', REPLICATION_SCOPE => '0', VERSIONS => '3', COMPRESSION => 'NONE', MIN_VERSIONS => '0', TTL => '2147483647', KEEP_DELETED_CELLS => 'false', BLOCKSIZE => '65536', IN_MEMORY => 'false', ENCODE_ON_DISK => 'true', BLOCKCACHE => 'true'}, {NAME => 'course', DATA_BLOCK_ENCODING => 'NONE', BLOOMFILTER => 'NONE', REPLICATION_SCOPE => '0', VERSIONS => '3', COMPRESSION => 'NONE', MIN_VERSIONS => '0', TTL => '2147483647', KEEP_DELETED_CELLS => 'false', BLOCKSIZE => '65536', IN_MEMORY => 'false', ENCODE_ON_DISK => 'true', BLOCKCACHE => 'true'}
1 row(s) in 0.0380 seconds

```

图 5-3 describe 命令执行结果

元格,单元格是 HBase 中的概念)添加一个数据,所以,直接用 Shell 命令插入数据效率很低,在实际应用中,一般都是利用编程操作数据。因为这里只要插入一条学生记录,所以,可以用 Shell 命令手工插入数据,命令如下:

```
hbase>put 'student','95001','Sname','LiYing'
```

上面的 put 命令会为 student 表添加学号为'95001'、名字为'LiYing'的一个单元格数据,其行键为 95001,也就是说,系统默认把跟在表名 student 后面的第一个数据作为行键。

下面继续添加 4 个单元格的数据,用来记录 LiYing 同学的相关信息,命令如下:

```

hbase>put 'student','95001','Ssex','male'
hbase>put 'student','95001','Sage','22'
hbase>put 'student','95001','Sdept','CS'
hbase>put 'student','95001','course:math','80'

```

5.3.3 查看数据

HBase 中有两个用于查看数据的命令。

(1) get 命令: 用于查看表的某个单元格数据。

(2) scan 命令: 用于查看某个表的全部数据。

例如,可以使用如下命令返回 student 表中 95001 行的数据:

```
hbase>get 'student','95001'
```

get 命令的执行结果如图 5-4 所示。

下面使用 scan 命令查询 student 表的全部数据:

```
hbase(main):024:0> get 'student','95001'
COLUMN                                CELL
Sage:                                timestamp=1442912525676, value=20
Sdept:                               timestamp=1442912586483, value=CS
Sname:                               timestamp=1442912495442, value=LiYing
Ssex:                                timestamp=1442912510852, value=male
course:math                          timestamp=1442912802499, value=80
5 row(s) in 0.0080 seconds
```

图 5-4 get 命令的执行结果

```
hbase>scan 'student'
```

scan 命令的执行结果如图 5-5 所示。

```
hbase(main):025:0> scan 'student'
ROW                                COLUMN+CELL
95001                             column=Sage:, timestamp=1442912525676, value=20
95001                             column=Sdept:, timestamp=1442912586483, value=CS
95001                             column=Sname:, timestamp=1442912495442, value=LiYing
95001                             column=Ssex:, timestamp=1442912510852, value=male
95001                             column=course:math, timestamp=1442912802499, value=80
1 row(s) in 0.0120 seconds
```

图 5-5 scan 命令的执行结果

5.3.4 删除数据

在 HBase 中用 delete 以及 deleteall 命令进行删除数据操作,两者的区别: delete 命令用于删除一个单元格数据,是 put 的反向操作;而 deleteall 命令用于删除一行数据。

首先使用 delete 命令删除 student 表中 95001 这行中的 Ssex 列的所有数据,命令如下:

```
hbase>delete 'student','95001','Ssex'
```

delete 命令的执行结果如图 5-6 所示。

```
hbase(main):026:0> delete 'student','95001','Ssex'
0 row(s) in 0.0020 seconds

hbase(main):027:0> get 'student','95001'
COLUMN                                CELL
Sage:                                timestamp=1442912525676, value=20
Sdept:                               timestamp=1442912586483, value=CS
Sname:                               timestamp=1442912495442, value=LiYing
course:math                          timestamp=1442912802499, value=80
4 row(s) in 0.0120 seconds
```

图 5-6 delete 命令的执行结果

从中可以看出,95001 这行中的 Ssex 列的所有数据已经被删除。

然后使用 deleteall 命令删除 student 表中的 95001 行的全部数据,命令如下:

```
hbase>deleteall 'student','95001'
```

5.3.5 删除表

删除表需要分两步操作：第一步让该表不可用；第二步删除表。例如，要删除 student 表，可以使用如下命令：

```
hbase>disable 'student'
hbase>drop 'student'
```

5.3.6 查询历史数据

在添加数据时，HBase 会自动为添加的数据添加一个时间戳。在修改数据时，HBase 会为修改后的数据生成一个新的版本（时间戳），从而完成“改”操作，旧的版本依旧保留，系统会定时回收垃圾数据，只留下最新的几个版本，保存的版本数可以在创建表时指定。

为了查询历史数据，这里创建一个 teacher 表，在创建表时，需要指定保存的版本数（假设指定为 5），命令如下：

```
hbase>create 'teacher',{NAME=>'username',VERSIONS=>5}
```

插入数据，并更新数据，使其产生历史版本数据，需要注意的是，这里插入数据和更新数据都是使用 put 命令，具体如下：

```
hbase>put 'teacher','91001','username','Mary'
hbase>put 'teacher','91001','username','Mary1'
hbase>put 'teacher','91001','username','Mary2'
hbase>put 'teacher','91001','username','Mary3'
hbase>put 'teacher','91001','username','Mary4'
hbase>put 'teacher','91001','username','Mary5'
```

查询时默认情况下会显示当前最新版本的数据，如果要查询历史数据，需要指定查询的历史版本数，由于上面设置了保存版本数为 5，所以，在查询时指定的历史版本数的有效取值为 1~5，具体命令如下：

```
hbase>get 'teacher','91001',{COLUMN=>'username',VERSIONS=>5}
hbase>get 'teacher','91001',{COLUMN=>'username',VERSIONS=>3}
```

上述 get 命令执行后的结果如图 5-7 所示。

5.3.7 退出 HBase 数据库

退出数据库操作，输入 exit 命令即可，命令如下：

```
hbase>exit
```