

实 验 报 告

学 号	1902001 1038	姓 名	岳宇轩	专业班级	2019 级计算机科学与技术专业 慧与卓越工程师班
课程名称	大数据导论			学期	2022 年秋季学期
任课教师	刘洁 刘艳艳	完成日期	2022.10.31	上机课时间	周一 56 节（双周）
实 验 名 称	实验 4 熟悉常用的 HBase 操作				

一、实验要求（10%）

1. 理解 HBase 在 Hadoop 体系结构中的角色；
2. 熟练使用 HBase 操作常用的 Shell 命令；
3. 熟悉 HBase 操作常用的 Java API。

二、实验内容及步骤（80%）

在实验开始之前，首先要进行 hbase 的安装和配置，这一步助教给的操作手册已经有很详细的说明了，按照说明一步一步来即可。

①在解压 hbase 时我没法在虚拟机里用 tar 命令直接解压，会报什么错误，我估计是助教在群里发的 7z 格式和 tar.gz 格式不一样的问题，所以我直接在 windows 上解压了之后用 xftp 传上去的

②修改 hbase-site.xml 时，我也是先在 windows 上改好了文件，调整好缩进之后再上传并覆盖 namenode 主机上的对应文件的

③在启动过程中，没有使用单点启动

使用 jps 指令看到以下结果：

```
hadoop@namenode hbase-2.2.21$ jps
1521 DataNode
2369 HMaster
1846 NodeManager
1341 NameNode
2541 HRegionServer
2669 Jps
2239 HQuorumPeer
hadoop@namenode hbase-2.2.21$
```

访问 <http://192.168.10.100:16010>，看到一下页面

Region Servers

ServerName	Start time	Last contact	Version	Requests Per Second	Num. Regions
datanode1,16020,1667193207048	Mon Oct 31 13:13:27 CST 2022	2 s	2.2.2	0	1
datanode2,16020,1667193206573	Mon Oct 31 13:13:26 CST 2022	0 s	2.2.2	0	1
namenode,16020,1667193208262	Mon Oct 31 13:13:28 CST 2022	2 s	2.2.2	0	0
Total 3				0	2

Backup Masters

ServerName	Port	Start Time
Total 0		

Tables

User Tables | System Tables | Snapshots

Peers

Peer Id	Cluster Key	Endpoint	State	IsSerial	Bandwidth	ReplicateAll	Namespaces	Exclude Namespaces	Table Cfs	Exclude Table Cfs
Total 0										

Tasks

Show All Monitored Tasks | **Show non-RPC Tasks** | Show All RPC Handler Tasks | Show Active RPC Calls | Show Client Operations

View as JSON

Start Time	Description	State	Status
Mon Oct 31 13:13:35 CST 2022	Initializing region hbase:meta,1,1588230740	COMPLETE (since 59sec ago)	Region opened successfully (since 59sec ago)
Mon Oct 31 13:13:35 CST 2022	Closing region 1588230740	COMPLETE (since 59sec ago)	Closed (since 59sec ago)
Mon Oct 31 13:13:47 CST 2022	Closing region 8e8e1c213b19a294d06b2c496kd7d	COMPLETE (since 47sec ago)	Closed (since 47sec ago)
Mon Oct 31 13:13:33 CST 2022	Master startup	COMPLETE (since 45sec ago)	Calling postStartMaster coprocessors (since 45sec ago)

Software Attributes

Attribute Name	Value	Description
HBase Version	2.2.2, revision=e6513a76c91cceda95dad7af246ac81d46fa2689	HBase version and revision
HBase Compiled	Sat Oct 19 10:10:12 UTC 2019, hbase-m	When HBase version was compiled and by whom
HBase Source Checksum	4d29f7701e395c5d34db1882ac5021b	HBase source MD5 checksum
Hadoop Version	2.8.5, revision=0b8464d75227fcae2c5e7f2410377b3d53d3d5f8	Hadoop version and revision
Hadoop Compiled	2018-09-10T11:00Z, jdu	When Hadoop version was compiled and by whom
Hadoop Source Checksum	9942ca5c745417c14e318835f420733	Hadoop source MD5 checksum
ZooKeeper Client Version	3.4.10, revision=-1	ZooKeeper client version and revision
ZooKeeper Client Compiled	03/23/2017 10:13 GMT	When ZooKeeper client version was compiled
ZooKeeper Quorum	namenode:2181 datanode1:2181	Addresses of all registered ZK servers. For more, see zk dump.

1. 编程实现以下指定功能，并用 Hadoop 提供的 HBase Shell 命令完成相同任务：这里助教已经提供了可用的代码了，只要徐泽需要执行的操作取消对应的注释即可

(1) 列出 HBase 所有的表的相关信息，例如表名；

```
//实验一（1） 列出HBase所有的表的相关信息，例如表名；
listTables();
```

实验结果如下：

```

连接成功
Table:s1
Table:s2
2022-10-31 22:01:35,977 INFO [main] client.ConnectionManager$HConnectionImplementation (ConnectionManager.java:closeMasterService(2139)) - Closing master protocol: MasterService
2022-10-31 22:01:35,977 INFO [main] client.ConnectionManager$HConnectionImplementation (ConnectionManager.java:closeZooKeeperWatcher(1710)) - Closing zookeeper sessionId=0x842e4f67f10007
2022-10-31 22:01:35,985 INFO [main] zookeeper.ZooKeeper (ZooKeeper.java:close(684)) - Session: 0x842e4f67f10007 closed
2022-10-31 22:01:35,987 INFO [main-EventThread] zookeeper.ClientCnxn (ClientCnxn.java:run(519)) - EventThread shut down for session: 0x842e4f67f10007
关闭成功
Process finished with exit code 0

```

(2) 在终端打印出指定的表的所有记录数据；

```

//实验一（2） 在终端打印出指定的表的所有记录数据；这里以s1表为例
getData( tableName: "s1");

```

实验结果如下：

```

连接成功
行键:LiSi 列族:score 列:Math 值:100 时间戳:1667224878919
2022-10-31 22:03:16,886 INFO [main] client.ConnectionManager$HConnectionImplementation (ConnectionManager.java:closeZooKeeperWatcher(1710)) - Closing zookeeper sessionId=0x842e4f67f10009
2022-10-31 22:03:16,902 INFO [main] zookeeper.ZooKeeper (ZooKeeper.java:close(684)) - Session: 0x842e4f67f10009 closed
2022-10-31 22:03:16,904 INFO [main-EventThread] zookeeper.ClientCnxn (ClientCnxn.java:run(519)) - EventThread shut down for session: 0x842e4f67f10009
关闭成功

```

(3) 向已经创建好的表添加和删除指定的列族或列；

①添加

```

//实验一（3）向已经创建好的表添加和删除指定的列族或列；以s1表为例，put 's1','zhangsan','score:Math','69'
//先向s1中添加一条数据
insertRow( tableName: "s1", rowKey: "LiSi", colFamily: "score", col: "Math", val: "100");
//然后使用上面的getData函数查看是否添加成功
getData( tableName: "s1");

```

实验结果如下，可以看到数据成功插入了：

```

连接成功
行键:LiSi 列族:score 列:Math 值:100 时间戳:1667224648185
2022-10-31 21:57:29,268 INFO [main] client.ConnectionManager$HConnectionImplementation (ConnectionManager.java:closeZooKeeperWatcher(1710)) - Closing zookeeper sessionId=0x842e4f67f10004
2022-10-31 21:57:29,272 INFO [main] zookeeper.ZooKeeper (ZooKeeper.java:close(684)) - Session: 0x842e4f67f10004 closed
2022-10-31 21:57:29,273 INFO [main-EventThread] zookeeper.ClientCnxn (ClientCnxn.java:run(519)) - EventThread shut down for session: 0x842e4f67f10004
关闭成功

```

②删除

```

//最后删除这条数据
deleteRow( tableName: "s1", rowKey: "LiSi", colFamily: "score", col: "Math");
//再次查看
getData( tableName: "s1");

```

实验结果如下，可以看到表中没有数据了：

```

连接成功
2022-10-31 22:00:07,742 INFO [main] client.ConnectionManager$HConnectionImplementation (ConnectionManager.java:closeZooKeeperWatcher(1710)) - Closing zookeeper sessionId=0x842e4f67f10005
2022-10-31 22:00:07,746 INFO [main] zookeeper.ZooKeeper (ZooKeeper.java:close(684)) - Session: 0x842e4f67f10005 closed
2022-10-31 22:00:07,747 INFO [main-EventThread] zookeeper.ClientCnxn (ClientCnxn.java:run(519)) - EventThread shut down for session: 0x842e4f67f10005
关闭成功

```

(4) 清空指定的表的所有记录数据；

```

//实验一（4）清空指定的表的所有记录数据；可以通过shell list命令查看表是否存在
clearRows( tableName: "s1");
listTables();

```

在这里我在助教的代码基础上根据注释的提示加了一行 listTables()

实验结果如下：

```

连接成功
2022-10-31 22:04:50,655 INFO [main] client.HBaseAdmin (HBaseAdmin.java:call(1378)) - Started disable of s1
2022-10-31 22:04:50,619 INFO [main] client.HBaseAdmin (HBaseAdmin.java:postOperationResult(1407)) - Disabled s1
2022-10-31 22:04:58,490 INFO [main] client.HBaseAdmin (HBaseAdmin.java:postOperationResult(963)) - Deleted s1
Table:s1数据已清空! Table:s2
2022-10-31 22:04:58,648 INFO [main] client.ConnectionManager$HConnectionImplementation (ConnectionManager.java:closeMasterService(2139)) - Closing master protocol: MasterService
2022-10-31 22:04:58,648 INFO [main] client.ConnectionManager$HConnectionImplementation (ConnectionManager.java:closeZooKeeperWatcher(1710)) - Closing zookeeper sessionId=0x842e4f67f1000a
2022-10-31 22:04:58,652 INFO [main] zookeeper.ZooKeeper (ZooKeeper.java:close(684)) - Session: 0x842e4f67f1000a closed
2022-10-31 22:04:58,654 INFO [main-EventThread] zookeeper.ClientCnxn (ClientCnxn.java:run(519)) - EventThread shut down for session: 0x842e4f67f1000a
关闭成功

```

(5) 统计表的行数。

```
//实验一（5）统计表的行数。  
countRows( tableName: "s2");
```

这里我之前把 s2 的数据全删了，所以结果是 0 行：

```
连接成功  
行数: 0  
2022-10-31 22:08:07,984 INFO [main] client.ConnectionManager$HConnectionImplementation (ConnectionManager.java:closeZooKeeperWatcher(1710)) - Closing zookeeper sessionId=0x842e4f67f1000f  
2022-10-31 22:08:07,989 INFO [main] zookeeper.ZooKeeper (ZooKeeper.java:close(684)) - Session: 0x842e4f67f1000f closed  
2022-10-31 22:08:07,991 INFO [main-EventThread] zookeeper.ClientCnxn (ClientCnxn.java:run(519)) - EventThread shut down for session: 0x842e4f67f1000f  
关闭成功
```

1. 编程实现以下指定功能，并用 Hadoop 提供的 HBase Shell 命令完成相同任务：

(1) 列出 HBase 所有的表的相关信息，例如表名；

```
hbase(main):003:0> list  
TABLE  
s1  
1 row(s)  
Took 0.0560 seconds  
=> ["s1"]  
hbase(main):004:0>
```

(2) 在终端打印出指定的表的所有记录数据；

```
hbase(main):004:0> scan 's1'  
ROW COLUMN+CELL  
zhangsan column=score:Math, timestamp=1667201793061, value=69  
1 row(s)  
Took 0.0464 seconds  
hbase(main):005:0> _
```

(3) 向已经创建好的表添加和删除指定的列族或列；

```
hbase(main):001:0> create 's1','score'  
Created table s1  
Took 2.4472 seconds  
=> Hbase::Table - s1  
hbase(main):002:0> put 's1','zhangsan','score:Math','69'  
Took 0.4103 seconds  
hbase(main):003:0> _
```

创建好的结果见第（2）问

```
hbase(main):005:0> delete 's1','zhangsan','score:Math'  
Took 0.0393 seconds  
hbase(main):006:0> scan 's1'  
ROW COLUMN+CELL  
0 row(s)  
Took 0.0307 seconds  
hbase(main):007:0>
```

删除操作之后，就看不到 zhangsan 的数据了

(4) 清空指定的表的所有记录数据；

(5) 统计表的行数。

```

hbase(main):007:0> create 's2','score'
Created table s2
Took 1.2523 seconds
=> Hbase::Table - s2
hbase(main):008:0> put 's2','wang1','score:English','79'
Took 0.0519 seconds
hbase(main):009:0> put 's2','wang2','score:Math','89'
Took 0.0175 seconds
hbase(main):010:0> _

```

先创建一个表 s2，在其中添加几行数据

```

hbase(main):010:0> count 's1'
0 row(s)
Took 0.1070 seconds
=> 0
hbase(main):011:0> count 's2'
2 row(s)
Took 0.0071 seconds
=> 2
hbase(main):012:0> _

```

统计表的行数

```

hbase(main):012:0> truncate 's2'
Truncating 's2' table (it may take a while):
Disabling table...
Truncating table...
Took 2.1675 seconds
hbase(main):013:0> _

```

清空表中数据

2.实验要求如下

2. 现有以下关系型数据库中的表和数据，要求将其转换为适合于HBase存储的表并插入数据：

学生表 (Student)

学号 (S_No)	姓名 (S_Name)	性别 (S_Sex)	年龄 (S_Age)
2015001	Zhangsan	male	23
2015003	Mary	female	22
2015003	Lisi	male	24

课程表 (Course)

课程号 (C_No)	课程名 (C_Name)	学分 (C_Credit)
123001	Math	2.0
123002	Computer Science	5.0
123003	English	3.0

选课表 (SC)

学号 (SC_Sno)	课程号 (SC_Cno)	成绩 (SC_Score)
2015001	123001	86
2015001	123003	69
2015002	123002	77
2015002	123003	99
2015003	123001	98
2015003	123002	95

1) createTable(String tableName, String[] fields)

创建表，参数 tableName 为表的名称，字符串数组 fields 为存储记录各个域名称的数组。要求当 HBase 已经存在名为 tableName 的表的时候，先删除原有的表，然后再创建新的表。

主函数：

```
public static void main(String[] args) throws IOException {
    init();
    String[] fields = {"sex", "score"};

    createTable( myTableName: "s1", fields);

    close();
}
```

api 代码:

```
public static void createTable(String myTableName, String[] fields) throws IOException {
    //获取表名的唯一途径
    TableName tableName = TableName.valueOf(myTableName);

    if (admin.tableExists(tableName)) {
        //表存在, 删除
        System.out.println("table is exists!");
        //使表下线
        admin.disableTable(tableName);
        //删除
        admin.deleteTable(tableName);
    } else {
        //首先创建表描述器,
        HTableDescriptor descriptor = new HTableDescriptor(TableName.valueOf(myTableName));
        //通过表描述器添加列族信息
        for (String str : fields) {
            //因为列族需要管理列信息, 所以还需要创建列族描述器
            descriptor.addFamily(new HColumnDescriptor(str));
        }
        admin.createTable(descriptor);
        System.out.println("表" + myTableName + "创建成功!");
    }
}
```

实验结果如下:

```
连接成功
2022-10-31 22:11:20,475 INFO [main] client.HBaseAdmin (HBaseAdmin.java:postOperationResult(790)) - Created s1
表s1创建成功!
2022-10-31 22:11:20,475 INFO [main] client.ConnectionManager$HConnectionImplementation (ConnectionManager.java:closeMasterService(2139)) - Closing master protocol: MasterService
2022-10-31 22:11:20,475 INFO [main] client.ConnectionManager$HConnectionImplementation (ConnectionManager.java:closeZooKeeperWatcher(1710)) - Closing zookeeper sessionId=0x842e4f67f10010
2022-10-31 22:11:20,479 INFO [main] zookeeper.ZooKeeper (ZooKeeper.java:close(684)) - Session: 0x842e4f67f10010 closed
2022-10-31 22:11:20,481 INFO [main-EventThread] zookeeper.ClientCnxn (ClientCnxn.java:run(519)) - EventThread shut down for session: 0x842e4f67f10010
关闭成功
```

(2) addRecord(String tableName, String row, String[] fields, String[] values)

向表 tableName、行 row (用 S_Name 表示) 和字符串数组 files 指定的单元格中添加对应的数据 values。其中 fields 中每个元素如果对应的列族下还有相应的列限定符的话, 用 “columnFamily:column” 表示。例如, 同时向 “Math”、“Computer Science”、“English” 三列添加成绩时, 字符串数组 fields 为 { “Score:Math”, ” Score; Computer Science”, ” Score:English” }, 数组 values 存储这三门课的成绩。

主函数:

```

public static void main(String [] args) throws IOException{
    init();
    String row = "S_Name1";
    String fields[] = {"score:Math","score:Computer Science","score:English"};
    String values[] = {"10","90","80"};
    addRecord( tableName: "s1",row,fields,values);

    close();
}

```

api 代码:

```

public static void addRecord(String tableName,String row,String[] fields,String[] values) throws IOException {
    Table table = connection.getTable(TableName.valueOf(tableName));
    for(int i = 0;i != fields.length;i++){
        Put put = new Put(row.getBytes());
        String[] cols = fields[i].split( regex: ":" );
        put.addColumn(cols[0].getBytes(), cols[1].getBytes(), values[i].getBytes());
        table.put(put);
    }
    table.close();
}

```

实验结果如下:

```

连接成功
2022-10-31 22:14:49,052 INFO [main] client.ConnectionManager$HConnectionImplementation (ConnectionManager.java:closeZooKeeperWatcher(1710)) - Closing zookeeper sessionId=0x842e4f67f10011
2022-10-31 22:14:49,056 INFO [main] zookeeper.ZooKeeper (ZooKeeper.java:close(684)) - Session: 0x842e4f67f10011 closed
2022-10-31 22:14:49,057 INFO [main-EventThread] zookeeper.ClientCnxn (ClientCnxn.java:run(519)) - EventThread shut down for session: 0x842e4f67f10011
关闭成功

```

(3) scanColumn(String tableName, String column)

浏览表 tableName 某一列的数据，如果某一行记录中该列数据不存在，则返回 null。要求当参数 column 为某一列族名称时，如果底下有若干个列限定符，则要列出每个列限定符代表的列的数据；当参数 column 为某一列具体名称（例如“Score:Math”）时，只需要列出该列的数据。

主函数:

```

public static void main(String [] args) throws IOException{
    init();
    scanColumn( tableName: "s1", column: "score");

    close();
}

```

api 代码:


```

public static void scanColumn(String tableName,String column)throws IOException{

    Table table = connection.getTable(TableName.valueOf(tableName));
    Scan scan = new Scan();
    scan.addFamily(Bytes.toBytes(column));
    ResultScanner scanner = table.getScanner(scan);
    for (Result result = scanner.next(); result != null; result = scanner.next()){
        showCell(result);
    }
    table.close();
}
}

```

实验结果如下：

```

连接成功
RowName:S_Name1
Timestamp:1667225688052
column Family:score
row Name:Computer Science
value:90
RowName:S_Name1
Timestamp:1667225688058
column Family:score
row Name:English
value:80
RowName:S_Name1
Timestamp:1667225688034
column Family:score
row Name:Math
value:10
2022-10-31 22:16:21,822 INFO [main] client.ConnectionManager$HConnectionImplementation (ConnectionManager.java:closeZooKeeperWatcher(1710)) - Closing zookeeper sessionId=0x842e4f67f10012
2022-10-31 22:16:21,829 INFO [main] zookeeper.ZooKeeper (ZooKeeper.java:close(684)) - Session: 0x842e4f67f10012 closed
2022-10-31 22:16:21,833 INFO [main-EventThread] zookeeper.ClientCnxn (ClientCnxn.java:run(519)) - EventThread shut down for session: 0x842e4f67f10012
关闭成功

```

(4) modifyData(String tableName, String row, String column)

修改表 tableName，行 row（可以用学生姓名 S_Name 表示），列 column 指定的单元格的数据。

目前 s1 表状态如下：

```

连接成功
行键:S_Name1 列族: scoreCo 列: Computer Science 值: 90 时间戳: 1667225688052
行键:S_Name1 列族: scoreEn 列: English 值: 80 时间戳: 1667225688058
行键:S_Name1 列族: scoreMa 列: Math 值: 50 时间戳: 1667228650677
2022-10-31 23:04:29,408 INFO [main] client.ConnectionManager$HConnectionImplementation (ConnectionManager.java:closeZooKeeperWatcher(1710)) - Closing zookeeper sessionId=0x842e4f67f1001d
2022-10-31 23:04:29,414 INFO [main] zookeeper.ZooKeeper (ZooKeeper.java:close(684)) - Session: 0x842e4f67f1001d closed
2022-10-31 23:04:29,416 INFO [main-EventThread] zookeeper.ClientCnxn (ClientCnxn.java:run(519)) - EventThread shut down for session: 0x842e4f67f1001d
关闭成功

```

修改代码如下：（将数学成绩改为 100）

```

public static void modifyData(String tableName,String row,String column,String val)throws IOException{

    Table table = connection.getTable(TableName.valueOf(tableName));
    Put put = new Put(row.getBytes());
    String[] cols = column.split( regex: " ");
    ///put.addColumn(column.getBytes(),null,val.getBytes());
    put.addColumn(cols[0].getBytes(), cols[1].getBytes(), val.getBytes());

    table.put(put);
    table.close();
}
}

```

这里我直接使用助教提供的源代码跑不通会报错，观察之后发现主函数中需要修改如下两点：

①row 由 S_Name 改为 S_Name1，因为问题 2_(2)的 addRecord 添加的数据学生姓名是 S_Name1：

```
public static void main(String [] args) throws IOException{
    init();
    String row = "S_Name1";
    String fields[] = {"score:Math","score:Computer Science","score:English"};
    String values[] = {"10","90","80"};
    addRecord( tableName: "s1",row,fields,values);

    close();
}
```

②其次，在指定修改的列时不能只输入列名，还要指定列簇的名字，这也是 hbase 和关系数据库表的一大区别：

```
public static void main(String [] args) throws IOException{
    init();
    String row = "S_Name";
    // String fields[] = {"score:Math","score:Computer Science","score:English"};
    // String values[] = {"100","90","80"};
    modifyData( tableName: "s1", row: "S_Name1", column "score:Math", val: "100");

    close();
}
```

实验结果如下：

```
连接成功
行键:S_Name1 列族: scoreCo 列: Computer Science 值: 90 时间戳: 1667225688052
行键:S_Name1 列族: scoreEn 列: English 值: 80 时间戳: 1667225688058
行键:S_Name1 列族: scoreMa 列: Math 值: 100 时间戳: 1667228736200
2022-10-31 23:09:09,246 INFO [main] client.ConnectionManager$HConnectionImplementation (ConnectionManager.java:closeZooKeeperWatcher(1710)) - Closing zookeeper sessionId=0x842e4f67f1001f
2022-10-31 23:09:09,250 INFO [main] zookeeper.ZooKeeper (ZooKeeper.java:close(684)) - Session: 0x842e4f67f1001f closed
2022-10-31 23:09:09,251 INFO [main-EventThread] zookeeper.ClientCnxn (ClientCnxn.java:run(519)) - EventThread shut down for session: 0x842e4f67f1001f
关闭成功
Process finished with exit code 0
```

数学成绩已经成功修改为了 100 分

(5) deleteRow(String tableName, String row)

删除表 tableName 中 row 指定的行的记录。

主函数：

```

public static void main(String [] args) throws IOException{
    init();

    deleteRow( tableName: "s1", row: "S_Name1");

    close();

}

```

api 代码:

```

public static void deleteRow(String tableName,String row)throws IOException{
    Table table = connection.getTable(TableNames.valueOf(tableName));
    Delete delete = new Delete(row.getBytes());
    table.delete(delete);
    table.close();
}

```

实验结果如下:

```

连接成功
行数: 0
2022-10-31 23:13:32,144 INFO [main] client.ConnectionManager$HConnectionImplementation (ConnectionManager.java:closeZooKeeperWatcher(1710)) - Closing zookeeper sessionId=0x842e4f67f10022
2022-10-31 23:13:32,148 INFO [main] zookeeper.ZooKeeper (ZooKeeper.java:close(o84)) - Session: 0x842e4f67f10022 closed
2022-10-31 23:13:32,151 INFO [main-EventThread] zookeeper.ClientCnxn (ClientCnxn.java:run(519)) - EventThread shut down for session: 0x842e4f67f10022
关闭成功
Process finished with exit code 0

```

使用 `countRows("s1")` 可以看到, 数据已删除, 剩余 0 行。

三、心得总结（写出自己在完成实验过程中遇到的问题、解决方法，以及体会、收获等）（10%）

通过本次实验, 我学会了如何正确安装 hbase。 有了之前安装 hadoop 的经验, 此次安装并配置 hbase 明显比之前熟练了一些。值得注意的是助教提供的 hadoop 是 3.1.3 版本的, 因此需要使用对应版本的 hbase (即 2.2.2)。

在配置 hbase 的过程中, 需要灵活变通, 比如当我没有办法直接解压虚拟机上的文件时, 我会先在本地解压后直接用 xftp 将其上传到虚拟机上; 当我需要在虚拟机上用 vim 大规模修改 xml 文件而无法复制粘贴使, 我会在本地先修改好然后上传并覆盖原先的文件。

通过本次实验我也初步学习了 hbase shell 指令, 感觉和 sql 语句有相似的地方, 都需要制定表

名，但区别在于 hbase shell 是有列簇和时间戳的，并且时间戳会在插入时自动生成，并不需要手动输入。类似 sql，hbase shell 也有一系列对行列的操作和表级别的操作。

除了使用 hbase shell 之外，还可以使用 java 编程 api 实现对 hbase 的操作。在理解代码，进行 api 应用在主函数上的调整过程中，我也更加清楚了 hbase 的特点和机制。