大数据技术与应用 - HBase实践

1. 实验环境：
   1. Mac M1
2. 实验方法：
   1. 安装Hbase：官网下载Hbase 2.6.1安装包，解压缩至/usr/local
   2. 启动服务

qiuxiaotian@192 hbase-2.6.1 % bin/start-hbase.sh

running master, logging to

/usr/local/hbase-2.6.1/bin/../logs/hbase-qiuxiaotian-master-192.168.124.4.out

(base) qiuxiaotian@192 hbase-2.6.1 % bin/hbase-daemon.sh start thrift

running thrift, logging to

/usr/local/hbase-2.6.1/bin/../logs/hbase-qiuxiaotian-thrift-192.168.124.4.out

* 1. create table + put data：这个过程可以通过hbase shell，使用namespace\_create创建。插入数据可以step by step插入，但是为了结合作业2，直接利用python happyhbase + hbase thrift编写建表、插入、删除等方法
  2. 建表

def create\_table(self):

try:

self.connect.create\_table(

'Orders',

{

'Order Detail': dict(), # 定义 Order Detail 列族

'Transaction': dict() # 定义 Transaction 列族

}

)

except Exception as e:

print(f'{e}')

* 1. 插入数据

def put(self, table\_name, data:pd.DataFrame):

"""插入数据"""

table = self.connect.table(table\_name)

if not table:

raise ValueError("获取table失败")

col\_name = data.columns

for row in data.values.tolist():

row\_key, insert\_data = str(uuid.uuid4()), {}

for index, item in enumerate(row):

# 将非字节类型（如 int、float）转换为字节类型

insert\_data[col\_name[index]] = str(item).encode('utf-8')

table.put(row\_key, insert\_data)

* 1. 数据准备

def read\_from\_xlsx(file\_path):

"""从题目所给的xlsx里读去数据"""

df = pd.read\_excel(file\_path)

df\_aggr = [ item for item in df.head(0).columns]

col\_detail = df.iloc[0].tolist()

real\_data = df.iloc[2:] # 实际的数据

real\_col, aggr\_name\_tmp = [], ""

for index, aggr\_name in enumerate(df\_aggr):

if not aggr\_name.startswith('Unnamed'):

aggr\_name\_tmp = aggr\_name

real\_col.append(aggr\_name\_tmp + ":" + col\_detail[index])

real\_data.columns = real\_col

return real\_data

1. 实验结果

通过scan获取数据：

def scan\_all(self, table\_name):

"""扫描整个表并打印所有行"""

table = self.connect.table(table\_name)

if not table:

raise ValueError("获取table失败")

for key, data in table.scan():

print(f"Row Key: {key}, Data: {data}")

