



浙江大學
ZHEJIANG UNIVERSITY

数据分析与算法设计课程大作业实验报告

实验名称 KV 查询

姓名

学号

专业

院所

2024 年 6 月 28 日

目录

1 实验介绍	3
2 实验准备：申请弹性公网 IP 和创建虚拟云	4
3 实验一：购买 MapReduce 服务	5
4 实验二：远程登录云服务器操作 HBase	6
5 实验三：创建预分 region 表	13
6 实验结束：云服务资源释放	15
7 实验总结与心得体会	15

1 实验介绍

实验简介

本课程为《大数据》实验课程，模拟高校学生系统，在华为 MRS 上进行 HBase 操作，通过创建表、对表进行 CURD 操作、创建预分 region 表及 Filter 过滤器的使用，让学员达到在 MRS 上熟练掌握 KV 查询。

实验流程



图 1: 实验流程图

实验环境

为了满足基于 HBase-MRS 实验需要，提前注册华为云账号并充值，确保通过 PUTTY 访问外网畅通。购买 MapReduce 1 个购买弹性 IP 1 个

2 实验准备：申请弹性公网 IP 和创建虚拟云

申请弹性公网 IP

1. 选择弹性公网 IP
2. 点击右上角的“购买弹性公网 IP”
3. 相关配置如图

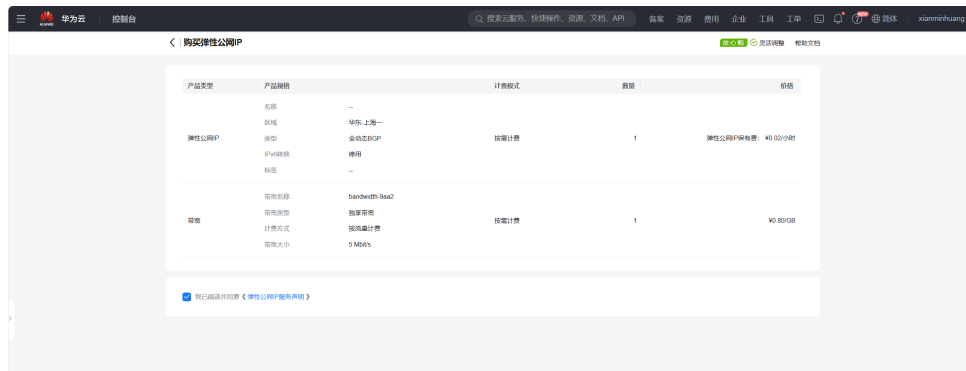


图 2: 购买弹性公网 IP

4. 查看购买的弹性公网 IP

创建虚拟云

1. 进入虚拟私有云控制台后，点击右上角的“创建虚拟私有云”。
2. 配置基本信息

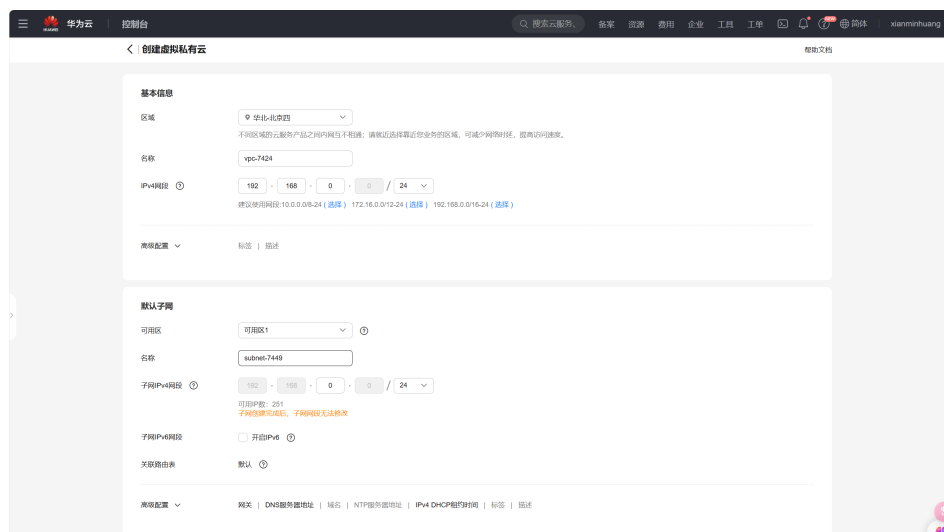


图 3: 配置虚拟云

3. 查看创建的 VPC



图 4: 查看创建的 VPC

3 实验一：购买 MapReduce 服务

1. 打开 MRS 服务申请集群
2. 集群配置：配置如下

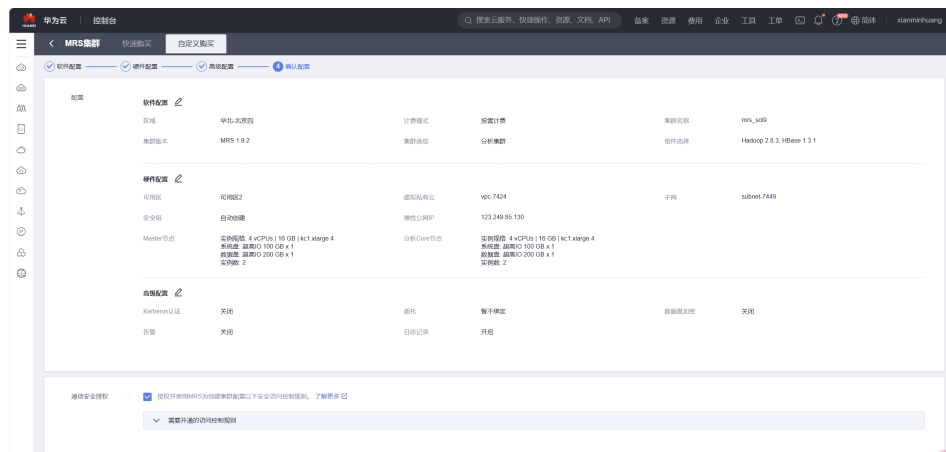


图 5: 集群配置

3. 点击集群名称，“前往 Manager”。输入用户名 admin 及密码, 点击“登录”，进入 MRS Manager

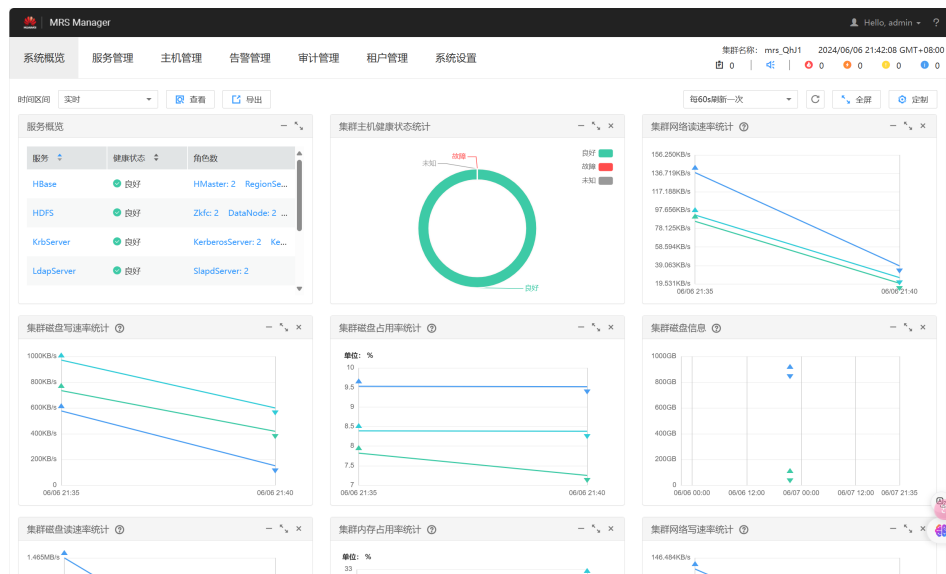


图 6: 进入 MRS Manager

4 实验二：远程登录云服务器操作 HBase

1. 配置安全组，选择“节点管理”，点击含有“master1”的节点，在弹出页面中选择“安全组”，点击“更改安全组规则”，选择“入方向规则”，点击“一键放通”，确认即可。
2. 远程登录 master 节点，在安全组配置项，点击右上方“远程登录”，选择 cloudshell 登录。输入密码，点击连接即可。

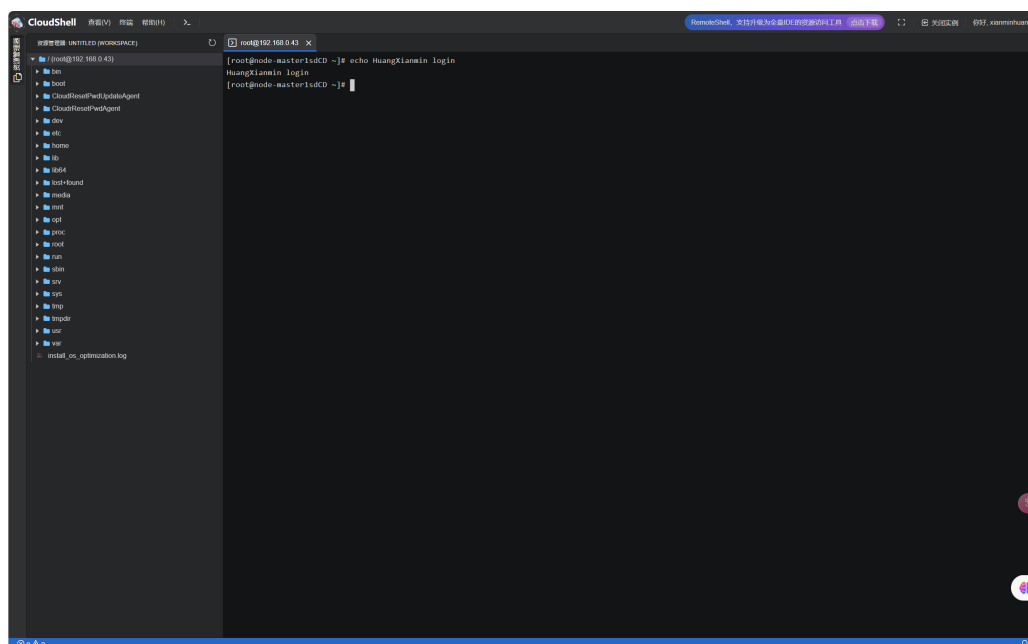


图 7: 远程连接登入

3. 设置环境变量

```
# source env_file
# hbase shell
```

4. 创建基表

```
# create 'cx_table_stu01', 'cf1'
```

5. list: 显示所有的表

```
# create 'cx_table_stu01', 'cf1'
```

如下图，创建了一个名为 cx_table_stu01 的空表，list 指令显示了当前有一个表 cx_table_stu01

```
hbase(main):001:0> create 'cx_table_stu01', 'cf1'
2024-06-06 21:59:18,814 INFO [main] client.HBaseAdmin: Created cx_table_stu01
0 row(s) in 1.8040 seconds

=> Hbase::Table - cx_table_stu01
hbase(main):002:0> list
TABLE
cx_table_stu01
1 row(s) in 0.0110 seconds
```

图 8: 创建和显示表

6. 增加数据

```
# put 'cx_table_stu01','20200001','cf1:name','tom'
# put 'cx_table_stu01','20200001','cf1:gender','male'
# put 'cx_table_stu01','20200001','cf1:age','20'
# put 'cx_table_stu01','20200002','cf1:name','hanmeimei'
# put 'cx_table_stu01','20200002','cf1:gender','female'
# put 'cx_table_stu01','20200002','cf1:age','19'
# scan 'cx_table_stu01'
```

如下图，向表中添加了两条数据。第一条数据的 rowkey 为 20200001，包含 name、gender、age 三个列族，第二条数据的 rowkey 为 20200002，包含 name、gender、age 三个列族。scan 指令显示了表中的所有数据

```
=> ["cx_table_stu01"]
hbase(main):003:0> put 'cx_table_stu01','20200001','cf1:name','tom'
0 row(s) in 0.1130 seconds

hbase(main):004:0> put 'cx_table_stu01','20200001','cf1:gender','male'
0 row(s) in 0.0180 seconds

hbase(main):005:0> put 'cx_table_stu01','20200001','cf1:age','20'
0 row(s) in 0.0180 seconds

hbase(main):006:0> put 'cx_table_stu01','20200002','cf1:name','hanmeimei'
0 row(s) in 0.0140 seconds

hbase(main):007:0> put 'cx_table_stu01','20200002','cf1:gender','female'
0 row(s) in 0.0110 seconds

hbase(main):008:0> put 'cx_table_stu01','20200002','cf1:age','19'
0 row(s) in 0.0160 seconds
```

图 9: 向表中添加数据

7. Scan 方式查询数据

```
# scan 'cx_table_stu01',{COLUMNS=>'cf1'}
# scan 'cx_table_stu01',{COLUMNS=>'cf1:name'}
```

如下图，scan 指令可以查询表中的所有数据，也可以指定列族查询数据

```
=> ["cx_table_stu01"]
hbase(main):004:0> scan 'cx_table_stu01'
ROW                                COLUMN+CELL
20200001                          column=cf1:age, timestamp=1717682416029, value=20
20200001                          column=cf1:gender, timestamp=1717682385876, value=male
20200001                          column=cf1:name, timestamp=1717682377264, value=tom
20200002                          column=cf1:age, timestamp=1717682430541, value=19
20200002                          column=cf1:gender, timestamp=1717682426300, value=female
20200002                          column=cf1:name, timestamp=1717682420717, value=hanmeimei
2 row(s) in 0.0780 seconds

hbase(main):005:0> scan 'cx_table_stu01',{COLUMNS=>'cf1'}
ROW                                COLUMN+CELL
20200001                          column=cf1:age, timestamp=1717682416029, value=20
20200001                          column=cf1:gender, timestamp=1717682385876, value=male
20200001                          column=cf1:name, timestamp=1717682377264, value=tom
20200002                          column=cf1:age, timestamp=1717682430541, value=19
20200002                          column=cf1:gender, timestamp=1717682426300, value=female
20200002                          column=cf1:name, timestamp=1717682420717, value=hanmeimei
2 row(s) in 0.0230 seconds

hbase(main):006:0> scan 'cx_table_stu01',{COLUMNS=>'cf1:name'}
ROW                                COLUMN+CELL
20200001                          column=cf1:name, timestamp=1717682377264, value=tom
20200002                          column=cf1:name, timestamp=1717682420717, value=hanmeimei
2 row(s) in 0.0210 seconds
```

图 10: scan 查询数据结果

8. Get 方式查询数据

```
# get 'cx_table_stu01','20200001'
# get 'cx_table_stu01','20200001','cf1:name'
```

9. 指定条件查询数据

```
# scan 'cx_table_stu01',{STARTROW=>'20200001','LIMIT'=>2,STOPROW=>'20200002'}
```

如下图，get 指令可以查询指定 rowkey 的数据，也可以指定列族查询数据。scan 指令可以指定条件查询数据，可以指定起始行、终止行、查询条数等条件

```
hbase(main):007:0> get 'cx_table_stu01','20200001'
COLUMN                                CELL
cf1:age                               timestamp=1717682416029, value=20
cf1:gender                             timestamp=1717682385876, value=male
cf1:name                               timestamp=1717682377264, value=tom
1 row(s) in 0.0310 seconds

hbase(main):008:0> get 'cx_table_stu01','20200001','cf1:name'
COLUMN                                CELL
cf1:name                               timestamp=1717682377264, value=tom
1 row(s) in 0.0200 seconds

hbase(main):009:0> scan 'cx_table_stu01',{STARTROW=>'20200001','LIMIT'=>2,STOPROW=>'20200002'}
ROW                                   COLUMN+CELL
20200001                             column=cf1:age, timestamp=1717682416029, value=20
20200001                             column=cf1:gender, timestamp=1717682385876, value=male
20200001                             column=cf1:name, timestamp=1717682377264, value=tom
1 row(s) in 0.0140 seconds
```

图 11: get 与指定条件查询数据结果

10. 过滤器查询

```
# scan 'cx_table_stu01',{FILTER=>"ValueFilter(=,'binary:20')"}
# scan 'cx_table_stu01',{FILTER=>"ValueFilter(=,'binary:tom')"}
# scan 'cx_table_stu01',{FILTER => "ColumnPrefixFilter('gender')"}
# scan 'cx_table_stu01',{FILTER=>"ColumnPrefixFilter('name')
    AND ValueFilter(=,'binary:hanmeimei')"}}
```

如下图，过滤器查询可以通过指定条件查询数据，可以通过值、列族、列名等条件查询数据

```

hbase(main):002:0> scan 'cx_table_stu01', {FILTER => "ColumnPrefixFilter('gender')"}
ROW
20200001      column=cf1:gender, timestamp=1717682385876, value=male
20200002      column=cf1:gender, timestamp=1717682426300, value=female
2 row(s) in 0.0140 seconds

hbase(main):003:0> scan 'cx_table_stu01',{FILTER=>"ValueFilter(=,'binary:20')"}
ROW
20200001      column=cf1:age, timestamp=1717682416029, value=20
1 row(s) in 0.0390 seconds

hbase(main):004:0> scan 'cx_table_stu01',{FILTER=>"ValueFilter(=,'binary:tom')"}
ROW
20200001      column=cf1:name, timestamp=1717682377264, value=tom
1 row(s) in 0.0180 seconds

hbase(main):005:0> scan 'cx_table_stu01', {FILTER => "ColumnPrefixFilter('gender')"}
ROW
20200001      column=cf1:gender, timestamp=1717682385876, value=male
20200002      column=cf1:gender, timestamp=1717682426300, value=female
2 row(s) in 0.0180 seconds

hbase(main):006:0> scan 'cx_table_stu01',{FILTER=>"ColumnPrefixFilter('name') AND ValueFilter(=,'binary:hanmeimei')"}
ROW
20200002      column=cf1:name, timestamp=1717682420717, value=hanmeimei
1 row(s) in 0.0190 seconds

```

图 12: 过滤器查询结果

11. 查询多版本数据，Hbase 可以存储历史版本的数据，通过设定版本号 VERSIONS 值，版本号为几就是存储几个版本。

增加数据执行命令：

```

# put 'cx_table_stu01','20200001','cf1:name','ZhangSan'
# put 'cx_table_stu01','20200001','cf1:name','LiSi'
# put 'cx_table_stu01','20200001','cf1:name','WangWu'

```

然后get时指定多版本查询：

```

# get 'cx_table_stu01','20200001',{COLUMNS=>'cf1',VERSIONS=>5}

```

如下图，添加不同版本的数据，然后查询多版本数据，可以看到 get 指令只查询了 20200001 版本的数据

```

hbase(main):011:0> scan 'cx_table_stu01'
ROW
20200001      column=cf1:age, timestamp=1717682416029, value=20
20200001      column=cf1:gender, timestamp=1717682385876, value=male
20200001      column=cf1:name, timestamp=1717683734351, value=WangWu
20200002      column=cf1:age, timestamp=1717682430541, value=19
20200002      column=cf1:gender, timestamp=1717682426300, value=female
20200002      column=cf1:name, timestamp=1717682420717, value=hanmeimei
2 row(s) in 0.0280 seconds

hbase(main):012:0> get 'cx_table_stu01','20200001',{COLUMNS=>'cf1',VERSIONS=>5}
COLUMN      CELL
cf1:age      timestamp=1717682416029, value=20
cf1:gender   timestamp=1717682385876, value=male
cf1:name     timestamp=1717683734351, value=WangWu
1 row(s) in 0.0520 seconds

```

图 13: 多版本查询

12. 查看表的属性，执行语句：

```
#desc 'cx_table_stu01'
```

如下图，可以看到表的属性，包括表名、列族、版本数等信息

```
hbase(main):013:0> desc 'cx_table_stu01'
Table cx_table_stu01 is ENABLED
cx_table_stu01
COLUMN FAMILIES DESCRIPTION
(NAME => 'cf1', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'false', KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', COMPRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true', BLOCKSIZE => '65536', REPLICATION_SCOPE => '0')
1 row(s) in 0.3760 seconds
```

图 14: 表属性查看

- 查看多个版本的数据: 需要修改表的 VERSIONS 值（或者在建表的时候就指定 VERSIONS 值）

```
# alter 'cx_table_stu01',{NAME=>'cf1','VERSIONS'=>5}
# put 'cx_table_stu01','20200001','cf1:name','ZhangSan'
# put 'cx_table_stu01','20200001','cf1:name','LiSi'
# put 'cx_table_stu01','20200001','cf1:name','WangWu'
# get 'cx_table_stu01','20200001',{COLUMN=>'cf1',VERSIONS=>5}
```

alter 指令可以修改表的属性，包括列族、版本数等信息。如下图，修改了表的版本数为 5，然后添加了不同版本的数据，查询多版本数据，可以看到 get 指令查询了 20200001 版本的数据

```
hbase(main):014:0> alter 'cx_table_stu01',{NAME=>'cf1','VERSIONS'=>5}
Updating all regions with the new schema...
0/1 regions updated.
1/1 regions updated.
Done.
0 row(s) in 2.9280 seconds

hbase(main):015:0> put 'cx_table_stu01','20200001','cf1:name','ZhangSan'
0 row(s) in 0.0170 seconds

hbase(main):016:0> put 'cx_table_stu01','20200001','cf1:name','LiSi'
0 row(s) in 0.0240 seconds

hbase(main):017:0> put 'cx_table_stu01','20200001','cf1:name','WangWu'
0 row(s) in 0.0100 seconds

hbase(main):018:0> get 'cx_table_stu01','20200001',{COLUMN=>'cf1',VERSIONS=>5}
COLUMN                                CELL
cf1:age                                timestamp=1717682416029, value=20
cf1:gender                             timestamp=1717682385876, value=male
cf1:name                               timestamp=1717684364142, value=WangWu
cf1:name                               timestamp=1717684358740, value=LiSi
cf1:name                               timestamp=1717684350661, value=ZhangSan
cf1:name                               timestamp=1717683734351, value=WangWu
1 row(s) in 0.0170 seconds
```

图 15: 多版本数据查询

- 删除数据

删除某列族下数据，执行命令：

```
# delete 'cx_table_stu01','20200002','cf1:age'
# get 'cx_table_stu01','20200002'
```

删除整行数据，执行命令：

```
# deleteall 'cx_table_stu01','20200002'
# get 'cx_table_stu01','20200002'
```

```
hbase(main):001:0> get 'cx_table_stu01','20200002'
COLUMN                                CELL
cf1:age                               timestamp=1717682430541, value=19
cf1:gender                             timestamp=1717682426300, value=female
cf1:name                               timestamp=1717682420717, value=hanmeimei
1 row(s) in 0.3560 seconds

hbase(main):002:0> delete 'cx_table_stu01','20200002','cf1:age'
0 row(s) in 0.0590 seconds

hbase(main):003:0> get 'cx_table_stu01','20200002'
COLUMN                                CELL
cf1:gender                             timestamp=1717682426300, value=female
cf1:name                               timestamp=1717682420717, value=hanmeimei
1 row(s) in 0.0100 seconds
```

图 16: 删除表中数据

15. 删除表

用drop命令可以删除表。但是在删除一个表之前必须先将其禁用。

第一步：disable “表名称”

```
# disable 'cx_table_stu01'
```

第二步：drop “表名称”

```
# drop 'cx_table_stu01'
```

```
hbase(main):002:0> disable 'cx_table_stu01'
2024-06-06 22:40:40,952 INFO [main] client.HBaseAdmin: Started disable of cx_table_stu01
2024-06-06 22:40:43,190 INFO [main] client.HBaseAdmin: Disabled cx_table_stu01
0 row(s) in 2.3500 seconds

hbase(main):003:0> drop 'cx_table_stu01'
2024-06-06 22:40:54,032 INFO [main] client.HBaseAdmin: Deleted cx_table_stu01
0 row(s) in 1.2400 seconds

hbase(main):004:0> list
TABLE
0 row(s) in 0.0160 seconds

=> []
```

图 17: 删除表

5 实验三：创建预分 region 表

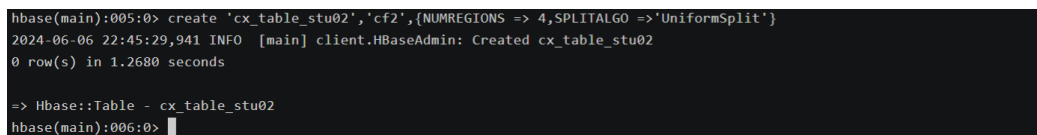
HBase 默认建表时只有一个 region，这个 region 的 rowkey 是没有边界的，即没有 startkey，也没有 endkey。在数据写入时，所有数据都会写入这个默认的 region，随着数据量的不断增加，此 region 已经不能承受不断增长的数据量，会进行 split，分成 2 个 region。在此过程中，会产生两个问题：

1. 数据往一个 region 上写，会有写热点问题。
2. region split 会消耗宝贵的集群 I/O 资源。

基于此我们可以在建表的时候，创建多个空 region，并确定每个 region 的起始和终止 rowkey，这样只要我们的 rowkey 设计能均匀的命中各个 region，就不会存在写热点问题，自然 split 的几率也会大大降低。hbase 提供了两种 pre-split 算法：HexStringSplit 和 UniformSplit，前者适用于十六进制字符的 rowkey，后者适用于随机字节数组的 rowkey。以 rowkey 切分，随机分为 4 个 region。以 rowkey 切分，随机分为 4 个 region。

1. 创建表执行命令：

```
# create 'cx_table_stu02','cf2',{NUMREGIONS => 4,SPLITALGO =>'UniformSplit'}
Region命名格式: [table],[region start key],[region id]
```



```
hbase(main):005:0> create 'cx_table_stu02','cf2',{NUMREGIONS => 4,SPLITALGO =>'UniformSplit'}
2024-06-06 22:45:29,941 INFO [main] client.HBaseAdmin: Created cx_table_stu02
0 row(s) in 1.2680 seconds

=> Hbase::Table - cx_table_stu02
hbase(main):006:0>
```

图 18: 创建表

2. 登录 HBase WebUI 查看表的分区情况

MRS Manager 界面，点击“HBase”服务，点击 HMaster(主) 进入 HBase UI, “User Tables” 下点击创建好的表名“cxtablestu02”

Table Attributes

Attribute Name	Value	Description
Enabled	true	Is the table enabled
Compaction	NONE	Is the table compacting

Table Regions

Name	Region Server	Start Key	End Key	Locality	Requests
cx_table_stu02,1717885128705.3869a257590b308b0598762d8ac776	node-ana-corewhF0001.mrs-v2gh.com,16020,1717880686153		@x00x00x00x00x00x00x00	0.0	0
cx_table_stu02,@x00x00x00x00x00x00x00,1717885128705.920ac25f760dc2eae6e543a852238	node-ana-corewhF0001.mrs-v2gh.com,16020,1717880686153	@x00x00x00x00x00x00x00	\x80x00x00x00x00x00x00	0.0	0
cx_table_stu02,\x80x00x00x00x00x00x00,1717885128705.f6977d463ab8ca97064037d3c0553d17	node-ana-corewhF0002.mrs-v2gh.com,16020,1717880686166	\x80x00x00x00x00x00x00	\xC0x00x00x00x00x00x00	0.0	0
cx_table_stu02,\xC0x00x00x00x00x00x00,1717885128705.f45ac30f10376178aa37ur9d89f4602	node-ana-corewhF0002.mrs-v2gh.com,16020,1717880686166	\xC0x00x00x00x00x00x00		0.0	0

Regions by Region Server

Region Server	Region Count
node-ana-corewhF0001.mrs-v2gh.com,16020,1717880686153	2
node-ana-corewhF0002.mrs-v2gh.com,16020,1717880686166	2

图 19: 查看该表分区情况

3. 指定 region 的 startKey 和 endKey

创建表执行命令：

```
# create 'cx_table_stu03', 'cf3', SPLITS => ['10000', '20000', '30000']
```

```
hbase(main):001:0> create 'cx_table_stu03', 'cf3', SPLITS => ['10000', '20000', '30000']
2024-06-06 22:54:23,820 INFO [main] client.HBaseAdmin: Created cx_table_stu03
0 row(s) in 1.5990 seconds

=> Hbase::Table - cx_table_stu03
hbase(main):002:0>
```

图 20: 创建表

同理，查看该表分区情况：

Table Attributes

Attribute Name	Value	Description
Enabled	true	Is the table enabled
Compaction	NONE	Is the table compacting

Table Regions

Name	Region Server	Start Key	End Key	Locality	Requests
cx_table_stu03,1717885962465.8062d803ff162b234cd97d4f17c0d41d5	node-ana-corewhF0002.mrs-v2gh.com,16020,1717880686166		10000	0.0	0
cx_table_stu03,10000,1717885962465.9f560890184bcd7f174625761028985d	node-ana-corewhF0001.mrs-v2gh.com,16020,1717880686153	10000	20000	0.0	0
cx_table_stu03,20000,1717885962465.9741cd9e9ebbbdd111cc0084c3daa	node-ana-corewhF0001.mrs-v2gh.com,16020,1717880686153	20000	30000	0.0	0
cx_table_stu03,30000,1717885962465.84c4784614b1bd92fc07be1ed9e26e7	node-ana-corewhF0002.mrs-v2gh.com,16020,1717880686166	30000		0.0	0

Regions by Region Server

Region Server	Region Count
node-ana-corewhF0001.mrs-v2gh.com,16020,1717880686153	2
node-ana-corewhF0002.mrs-v2gh.com,16020,1717880686166	2

图 21: 查看该表分区情况

6 实验结束：云服务资源释放

释放 MapReduce 服务

1. 进入 MRS 控制台，点击集群后的删除按钮

释放网络资源 VPC

1. 进入 VPC 控制台界面
2. 删除 IP: 点击“弹性公网 IP 和带宽”，选择弹性公网 IP，选中所有 IP，点击释放按钮。
3. 删除安全组规则: 点击“访问控制”，选择安全组，点击安全组名称，选择入方向规则，点击列表后的删除按钮。

7 实验总结与心得体会

在这次 KV 查询实验中，我通过模拟高校学生系统，在华为 MRS 平台上进行了 HBase 操作，主要包括创建表、CURD 操作、创建预分 region 表及使用 Filter 过滤器等内容。整个实验过程让我深刻理解了 HBase 的基础操作以及在实际应用中的重要性。

首先，实验准备阶段，我体验了从申请弹性公网 IP 到创建虚拟私有云的完整过程。在此过程中，我通过华为云控制台完成了相关配置，成功购买了弹性公网 IP 和虚拟私有云。这不仅让我熟悉了华为云平台的操作，也让我认识到在大数据处理环境中，网络资源的配置和管理是基础且关键的一步。通过这一过程，我深刻体会到了云计算资源的灵活性和便捷性，同时也学会了如何高效地管理和使用这些资源。

在购买 MapReduce 服务并配置集群时，我进一步加深了对大数据处理框架的理解。通过配置集群并进入 MRS Manager，我了解了集群管理的基本操作。这一步骤让我认识到，合理配置和管理集群对于大数据处理的效率和性能至关重要。在实际操作中，我学会了如何根据实验要求配置集群参数，以及如何通过 MRS Manager 进行集群管理和监控。

实验二的重点是远程登录云服务器并进行 HBase 操作。这部分内容让我对 HBase 的基本操作有了全面的了解。从配置安全组、远程登录 master 节点到设置环境变量，再到创建表、增加数据、查询数据、删除数据等一系列操作，每一步都让我对 HBase 的工作原理有了更深入的理解。例如，通过创建一个简单的表并向其中添加数据，我体会到了 HBase 在处理大规模数据时的高效性和灵活性。此外，通过使用不同的查询命令（如 scan 和 get），我学会了如何根据具体需求灵活查询数据。而在使用过滤器进行数据查询时，我认识到了 HBase 在数据过滤和查询方面的强大功能，这对于实际应用中快速定位和提取数据非常有帮助。

实验三中创建预分 region 表的操作让我认识到在大数据处理过程中如何避免写热点问题。通过预先定义多个 region 并设置 rowkey 的起始和终止值，我们可以有效分散数据写入的压力，避免单一 region 过载问题。这不仅提高了数据写入的效率，也减少了 region split 对集群资源的消耗。通过这一操作，我深刻体会到在实际应用中，合理设计和管理数据存储结构对于提高系统性能至关重要。

在实验过程中遇到的一些问题和挑战也让我受益匪浅。例如，在配置虚拟私有云时，由于对华为云平台的操作不够熟悉，我曾一度在网络配置上遇到困难。但通过查阅相关资料并不断尝试，我最终解决了这些问题。这不仅增强了我的问题解决能力，也让我对华为云平台有了更深入的了解。

此外，在操作 HBase 时，由于对 HBase 的命令不够熟悉，我曾在数据查询和过滤器使用上遇到了一些问题。但通过反复练习和实验，我逐渐掌握了这些命令的使用方法。这让我认识到，在学习和掌握新技术时，动手实践和反复操作是非常重要的。

通过这次实验，我不仅掌握了 HBase 的基本操作和使用技巧，也对大数据处理的整个流程有了更全面的理解。从申请和配置云资源，到管理和操作 HBase，再到解决实际操作中遇到的问题，每一步都让我受益匪浅。这些知识和技能不仅对我今后的学习和工作非常有帮助，也让我对大数据处理技术充满了兴趣和信心。

总的来说，这次 KV 查询实验不仅让我掌握了许多实用的技能，也让我对大数据处理技术有了更深刻的理解和认识。在未来的学习和工作中，我将继续深入学习和探索大数据处理技术，不断提升自己的专业能力和水平。通过这次实验，我深刻体会到，只有不断实践和总结，才能真正掌握和运用所学的知识，实现自己的职业目标和价值。