ISIT312 Big Data Management

HBase Operations

Dr Janusz R. Getta

School of Computing and Information Technology -
University of Wollongong

HBase Operations

Outline

HBase shell

HBase provides extensible JRuby-based (Java Interactive Ruby - JIRB) shell as a feature to execute some commands

The shell is a typical Read–Eval–Print-Loop (REPL) shell also known as a language shell

It is a simple, interactive computer programming environment that takes single user inputs evaluates them, and returns the result to the user; a program written in a REPL environment is executed piecewise

It means that HBase shell allows for computation of Ruby scripts and brings all features enabled in JIRB shell

It allows to process a script of HBase commands saved in a file `file-name.hb` in the following way:

```
source 'file-name.hb'
```
HBase shell

HBase Operations

Outline

Data definition commands

In Hbase, a set of data definition commands includes: `create`, `list`, `describe`, `disable`, `disable_all`, `enable`, `enable_all`, `drop`, `drop_all`, `show_filters`, `alter`, `alter_status`

Create a table `'student'` with a column family `'personal'`

```
create 'student','personal'                                    HBase shell
```

Show a structure of a table `'student'`

```
describe 'student'                                             HBase shell
```

Implement a column family `'personal'` in transient memory

```
alter 'student',{NAME=>'personal',IN_MEMORY=>true}             HBase shell
```

Add a column family `'uni'` to a table `'student'`

```
alter 'student',{NAME=>'uni',VERSIONS=>'4'}                    HBase shell
```

Data definition commands

Delete a column family `'uni'` from a table `'student'`

```
alter 'student','delete'=>'uni'                                    HBase shell
```

Add a column family `'university'` to a table student and allow for 5 versions in each cell in the column family

```
alter 'student',{NAME=>'university',VERSIONS=>5}                   HBase shell
```

Increase a number of allowed versions in a column family `'personal'`to 3

```
alter 'student',{NAME=>'personal',VERSIONS=>3}                     HBase shell
```

HBase Operations

Outline

- HBase shell
- Data definition commands
- Data manipulation commands
- HBase Java AP

Data Manipulation commands

In Hbase, a set of data manipulation commands includes: `count`, `put`, `get`, `delete`, `delete_all`, `truncate`, `scan`

Put a value `'James'` into a cell in a column family `'personal'`, qualification `'first-name'`, row key `'007'`,

```
put 'student','007','personal:first-name','James'
```
HBase shell

Put a value `'Bond'` into a cell in a column family `'personal'`, qualification `'last-name'`, row key `'007'`

```
put 'student','007','personal:last-name','Bond'
```
HBase shell

Put a value `'01-OCT-1960'` into a cell in a column family `'personal'`, qualification `dob'`, row key `'007'`,

```
put 'student','007','personal:dob','01-OCT-1960'
```
HBase shell

Data Manipulation commands

List the contents of a table `'student'`

```
scan 'student'                                        HBase shell
```

Put a value `'02-OCT-1960'` as the second version into a cell in a column family `'personal'`, qualification `dob'`, row key `'007'`,

```
put 'student','007','personal:dob','02-OCT-1960'      HBase shell
```

Get no more than `5` versions of a cell `'dob'` in a column family `'personal'` from a row `'007'` in a table `'student'`

```
get 'student','007',{COLUMN=>'personal:dob',VERSIONS=>5}      HBase shell
```

Get no more than `5` versions of a cell `'dob'` in a column family `'personal'`, from a table `'student'`

```
scan 'student',{COLUMN=>'personal:dob',VERSIONS=>5}      HBase shell
```

Data Manipulation commands

Get all column families in a row `'666'` in a table `'student'`

```
get 'student','666'                                          HBase shell
```

Get no more than 5 versions of values from all cells in a column family `'grade'` in a row `'666'` in a table `'student'`

```
get 'student','666',{COLUMN=>'grade',VERSIONS=>5}            HBase shell
```

Get no more than 5 versions of values from a cell `'CSCI235'` in a column family `'grade'` in a row `'666'` in a table `'student'`

```
get 'student','666',{COLUMN=>'grade:CSCI235',VERSIONS=>5}    HBase shell
```

Get no more than 5 versions of values from a cell `'dob'` in a column family `'grade'` in a row `'666'` in a table `'student'`

```
get 'student','007',{COLUMN=>'personal:dob',VERSIONS=>5}     HBase shell
```

Data Manipulation commands

Count total number of rows in a table `'student'`

```
count 'student'                                                    HBase shell
```

Get entire table `'student'`, one version per cell

```
scan 'student'                                                     HBase shell
```

Get entire table `'student'`, at most 5 versions per cell

```
scan 'student',{VERSIONS=>5}                                       HBase shell
```

Get all cells `'dob'` from in a column family `'personal'` from entire table `'student'`, at most `5` versions per cell

```
scan 'student',{COLUMNS=>'personal:dob',VERSIONS=>5}               HBase shell
```

Data Manipulation commands

Get all cells from the column families `'personal'` and `'university'` from entire table `'student'`

```
scan 'student',{COLUMNS=>['personal','university']}
```
HBase shell

Get at most 5 versions of cells 'dob' with timestamps in a range `[1,1502609828830]`, from a column family `'personal'` from entire table `'student'`

HBase shell
```
scan 'student',{COLUMNS=>'personal:dob',TIMERANGE=>[1,1502609828830],
                VERSIONS=>5}
```

Get at most 5 versions of cells 'dob' with timestamps in a range `[1,1502609828830]`, from a column family `'personal'` from entire table `'student'`

```
scan 'student',{COLUMNS=>'personal:dob',
                FILTER=>"TimestampsFilter(1,1502609828830)",VERSIONS=>5}
```
HBase shell

Data Manipulation commands

Get all cells whose name is `>=` than `'f'` in a table `'student'`

```
scan 'student',{FILTER=>"QualifierFilter(>=,'binary:f')"}
```

HBase shell

Get all rows from a table 'student' that have value of a cell `>=` than `'J'`

```
scan 'student',{FILTER=>"ValueFilter(>=,'binary:J')"}
```

HBase shell

Get all rows from a table `'student'` that have value of a cell in a range `['J','K']`

```
scan 'student',{FILTER=>"ValueFilter(>=,'binary:J') AND
               ValueFilter(<=,'binary:K')"}
```

HBase shell

Get all values of cells `'dob'` in a column family `'personal'` from rows in a table `'student'` where a cell `'dob'` has a value `'02-OCT-1960'`

```
scan 'student',{COLUMNS=>'personal:dob',FILTER=>
               "QualifierFilter(=,'binary:dob') AND
                                       ValueFilter(=,'binary:02-OCT-1960')"}
```

HBase shell

Data Manipulation commands

Delete a cell `'CSCI235'` from a column family `'student'` in a row `'666'` in a table `'student'`

```
delete 'student','666','grade:CSCI235'                                HBase shell
```

Delete entire row `'007'` from a table 'student'

```
deleteall 'student', '007                                             HBase shell
```

HBase Operations

Outline

HBase Java API

HBase Java Application Program Interface allows to access HBase tables from programs written in Java

The client APIs provide both data definition and data manipulation features

Creating a table `'my-table'` and column families `'Address'` and `'Name'`

```java
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.hbase.HBaseConfiguration;
import org.apache.hadoop.hbase.HColumnDescriptor;
import org.apache.hadoop.hbase.HTableDescriptor;
import org.apache.hadoop.hbase.TableName;
import org.apache.hadoop.hbase.client.HBaseAdmin;
```

```java
public class CreateTable {
  public static void main(String[] args) throws Exception {
    Configuration conf = HBaseConfiguration.create();
    HBaseAdmin admin = new HBaseAdmin(conf);
    HTableDescriptor tableDescriptor = new HTableDescriptor(TableName.valueOf("my-table"));
    tableDescriptor.addFamily(new HColumnDescriptor("Address"));
    tableDescriptor.addFamily(new HColumnDescriptor("Name"));
    admin.createTable(tableDescriptor);
    boolean tableAvailable = admin.isTableAvailable("my-table");
    System.out.println("tableAvailable = " + tableAvailable); } }
```

HBase Java API

Inserting data into HBase table 'my-table'

```java
public class PutRow {
  public static void main(String[] args) throws Exception {
    Configuration conf = HBaseConfiguration.create();
    HTable table = new HTable(conf, "my-table");
    Put put = new Put(Bytes.toBytes("007"));
    put.add(Bytes.toBytes("Address"),Bytes.toBytes("City"),Bytes.toBytes("Dapto"));
    put.add(Bytes.toBytes("Address"),Bytes.toBytes("Street"),Bytes.toBytes("Ellenborough'
    put.add(Bytes.toBytes("Name"),Bytes.toBytes("First"),Bytes.toBytes("James"));
    put.add(Bytes.toBytes("Name"),Bytes.toBytes("Last"),Bytes.toBytes("Bond"));
    table.put(put);
    table.flushCommits();
    table.close();
  }
}
```

HBase Java API

Getting data from HBase table 'my-table'

Java
```
import java.util.Map;
import java.util.NavigableMap
```

Java
```
public class GetRow {
    public static void main(String[] args) throws Exception {
        Configuration conf = HBaseConfiguration.create();
        HTable table = new HTable(conf, "my-table");
        Get get = new Get(Bytes.toBytes("007"));
        get.setMaxVersions(3);
        get.addFamily(Bytes.toBytes("Address"));
        get.addColumn(Bytes.toBytes("Name"), Bytes.toBytes("First"));
        get.addColumn(Bytes.toBytes("Name"), Bytes.toBytes("Last"));
```

Java
```
// Get a specific value
        Result result = table.get(get);
        String row = Bytes.toString(result.getRow());
```

HBase Java API

Getting data from HBase table 'my-table'

Java
```
String specificValue = Bytes.toString(result.getValue(Bytes.toBytes("Address"),
                                      Bytes.toBytes("City")));
System.out.println("Latest Address:City is: " + specificValue);
       specificValue = Bytes.toString(result.getValue(Bytes.toBytes("Address"),
                                      Bytes.toBytes("Street")));
System.out.println("Latest Address:Street is: " + specificValue);
       specificValue = Bytes.toString(result.getValue(Bytes.toBytes("Name"),
                                      Bytes.toBytes("First")));
System.out.println("Latest Name:First is: " + specificValue);
       specificValue = Bytes.toString(result.getValue(Bytes.toBytes("Name"),
                                      Bytes.toBytes("Last")));
System.out.println("Latest Name:Last is: " + specificValue)
```

HBase Java API

Java

```
// Traverse entire returned row
        System.out.println("Row key: " + row);
        NavigableMap>> map = result.getMap();
        for (Map.Entry>> navigableMapEntry : map.entrySet()) {
            String family = Bytes.toString(navigableMapEntry.getKey());
            System.out.println("\t" + family);
            NavigableMap> familyContents = navigableMapEntry.getValue();
            for (Map.Entry> mapEntry : familyContents.entrySet()) {
                String qualifier = Bytes.toString(mapEntry.getKey());
                System.out.println("\t\t" + qualifier);
                NavigableMap qualifierContents = mapEntry.getValue();
                for (Map.Entry entry : qualifierContents.entrySet()) {
                    Long timestamp = entry.getKey();
                    String value = Bytes.toString(entry.getValue());
                    System.out.printf("\t\t\t%s, %d\n", value, timestamp);
                }
            }
        }
        table.close();
    }
}
```

References

HBase shell commands, `https://learnhbase.wordpress.com/2013/03/02/hbase-shell-commands/`

HBase shell and General commands, `https://www.guru99.com/hbase-shell-general-commands.html#4`

HBase Java API, `https://dzone.com/articles/handling-big-data-hbase-part-4`

Kerzner M., Maniyam S., HBase Design Patterns, Packt Publishing 2014 (Available from UoW Library)

Jiang, Y. HBase Adminstration Cookbook, Pack Publishing, 2012 (Available from UoW Library)

Dimiduk N., Khurana A., HBase in Action, Mannig Publishers, 2013

Spaggiari J-M., O'Dell K., Architecting HBase Applications, O'Reilly, 2016