DIC Assignment 5

Accumulo

-Utkarsh Srivastava

**Description :**

In this assignment, you will use Twitter API to collect data of 30 NBA teams, and store the data in CSV format (comma separated) on your local machine. Accumulo will be utilized to create table and compute the popularity of NBA teams based on the collected data.  
**Steps**   
Once the virtualbox is installed and the Image is imported, we provide the permissions to the update script. The script clears previous data values and starts Hadoop, zookeeper and accumulo. Copy the jar to the location accumulo/bin and change the permission of the jar to be executed

1. First start the shell from bin/accumulo with the argumnets username : root and password : acc

**bin/tool.sh WordCount1.jar com.WLose acc guestvb /input output –u root –p acc**

1. Go to shell :   
    **bin/accumulo shell –u root –p acc**
2. Two users are created using creatable command :

createuser <username>

**createuser east**  
Enter new password for east:

**createuser west**

Enter new password for west:

1. If required two tables with the name Wintable and Losetable can also be created. These two tables will be populated by the jar later on with the number of numbers of win and numbers of lose.

createtable <tablename>

**createtable WinTable**

**createtable LoseTable**

1. Set the authorization permissions for both the users so that they are able to acces **only**  their respective domains within the same table

**userpermissions**

**grant Table.READ –t WinTable –u east**

**grant Table.READ -t WinTable–u west  
grant Table.READ –t LoseTable –u east**

**grant Table.READ -t LoseTable –u west**

1. Associate each user with that label using authorization and Finally the relevant data is downloaded using the python script.

**setauths –s east-u east**

**setauths –s west-u ewest**

After logging in using

**User <username>**

**scan –t <tablename>**

The following school of logic is implemented in the jar :

* All teams in east and west are hardcoded into different arrays
* A mutation is associated with the filename
* Under the map section, Given that the filename is the name of a team, the value of column family , qualifier and visibility is set according to whether the team belongs to the east or west zone . The value is initialized to 0. Thus, each mapper will be allotted one/more of the csv files and will initialize that team’s win and lose count to 0
* Now, the content of twitter is obtained from the value argument of the mapper method. Whenever the word win or lose is encountered , the values are updated accordingly in the wintable and losetable. This ensures that an increase in record succeeds an increase in count for that file.
* The same procedure is repeated for west set of teams as well.
* When the reduce runs on these tables after they have been grouped according to names , the counts of each record of same filename sum up
* The output is written using context.write in each table.

**Output :**

|  |  |  |
| --- | --- | --- |
|  | username: east |  |
| **East Conference** | | |
| **Team Name** | **Hashtag** | **Win** |
| Atlanta Hawks | #hawks | 12 |
| Boston Celtics | #celtics | 5 |
| Charlotte Bobcats | #bobcats | 2 |
| Chicago Bulls | #bulls | 2 |
| Detroit Pistons | #piston | 0 |
| Indiana Pacers | #pacers | 0 |
| Miami Heat | #miamiheat | 0 |
| Milawukee Bucks | #bucks | 0 |
| New Jersey Nets | #nets | 0 |
| New York Knicks | #knicks | 0 |
| Orlando Magic | #orlandomagic | 0 |
| Philadelphia 76ers | #76ers | 0 |
| Toronto Raptors | #raptors | 0 |
| Washington Wizards | #wizards | 0 |
|  | username: east |  |
| **East Conference** | | |
| **Team Name** | **Hashtag** | **Lose** |
| Atlanta Hawks | #hawks | 4 |
| Boston Celtics | #celtics | 2 |
| Charlotte Bobcats | #bobcats | 1 |
| Chicago Bulls | #bulls | 0 |
| Detroit Pistons | #piston | 0 |
| Indiana Pacers | #pacers | 0 |
| Miami Heat | #miamiheat | 0 |
| Milawukee Bucks | #bucks | 0 |
| New Jersey Nets | #nets | 0 |
| New York Knicks | #knicks | 0 |
| Orlando Magic | #orlandomagic | 0 |
| Philadelphia 76ers | #76ers | 0 |
| Toronto Raptors | #raptors | 0 |
| Washington Wizards | #wizards | 0 |

|  |  |  |
| --- | --- | --- |
|  | username: west |  |
| **West Conference** | | |
| **Team Name** | **Hashtag** | **Win** |
| Dallas Mavericks | #mavs | 10 |
| Denver Nuggets | #nuggets | 5 |
| Golden State Warriors | #gswarriors | 4 |
| Houston Rockets | #rockets | 3 |
| LA Clippers | #clippers | 0 |
| LA Lakers | #lakers | 0 |
| Memphis Grizzlies | #grizzlies | 0 |
| Minnesota Timberwolves | #twolves | 0 |
| New Orleans Hornets | #hornets | 0 |
| Oklahoma City | #okcthunder | 0 |
| Phoenix Suns | #suns | 0 |
| Portland Trailblazers | #trailblazers | 0 |
| Sacramento Kings | #nbakings | 0 |
| San Antonio Spurs | #gospursgo | 0 |
| Utah Jazz | #utahjazz | 0 |

|  |  |  |
| --- | --- | --- |
|  | username: west |  |
| **West Conference** | | |
| **Team Name** | **Hashtag** | **Lose** |
| Dallas Mavericks | #mavs | 4 |
| Denver Nuggets | #nuggets | 3 |
| Golden State Warriors | #gswarriors | 1 |
| Houston Rockets | #rockets | 1 |
| LA Clippers | #clippers | 0 |
| LA Lakers | #lakers | 0 |
| Memphis Grizzlies | #grizzlies | 0 |
| Minnesota Timberwolves | #twolves | 0 |
| New Orleans Hornets | #hornets | 0 |
| Oklahoma City | #okcthunder | 0 |
| Phoenix Suns | #suns | 0 |
| Portland Trailblazers | #trailblazers | 0 |
| Sacramento Kings | #nbakings | 0 |
| San Antonio Spurs | #gospursgo | 0 |
| Utah Jazz | #utahjazz | 0 |

**FINAL REVIEW :**Using accumulo, we were not only able to use map reduce to analyse and sap information out of a huge amount of unformatted data, But more importantly, we were able to ensure that limited access of each team and its statistics are allocated/restricted to only certain users. Accumulo seems to be extremely powerful at scaling and organizing data. However, it supports only basic authentication, authorization and creation and deletion commands. Thus, we need to reply on something more powerful such as java and python script to blend the commands with logic.