# Hands On Exercise for Cloudera Hadoop-Developer Track

# Table of Contents

# Exercise 1: Setting up the Environment

**Step1:** Installing VMware Player from Training Bundle.

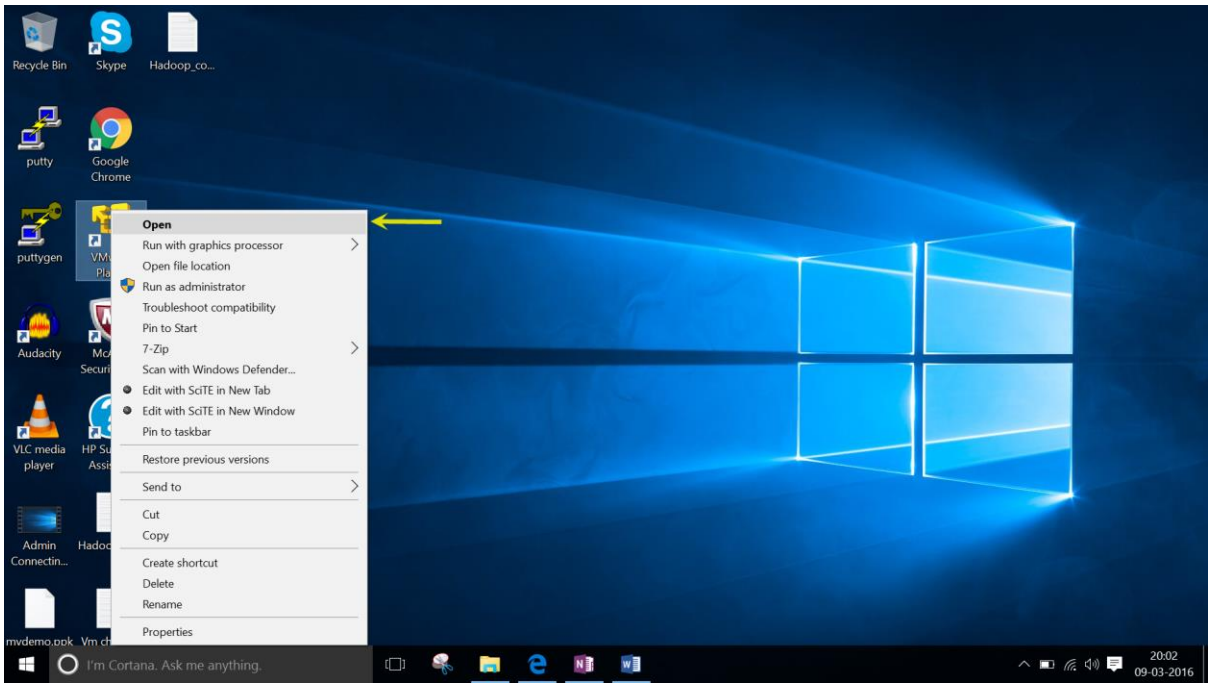
- Under 'Technocrafty_Training_Bundle_For_Cloudera_Developer_Track' folder, navigate to "Additional Software" folder.
- Find the executable file named 'VmWarePlayer.exe' and complete the installation.

**Step2**: Extracting the VMware Image

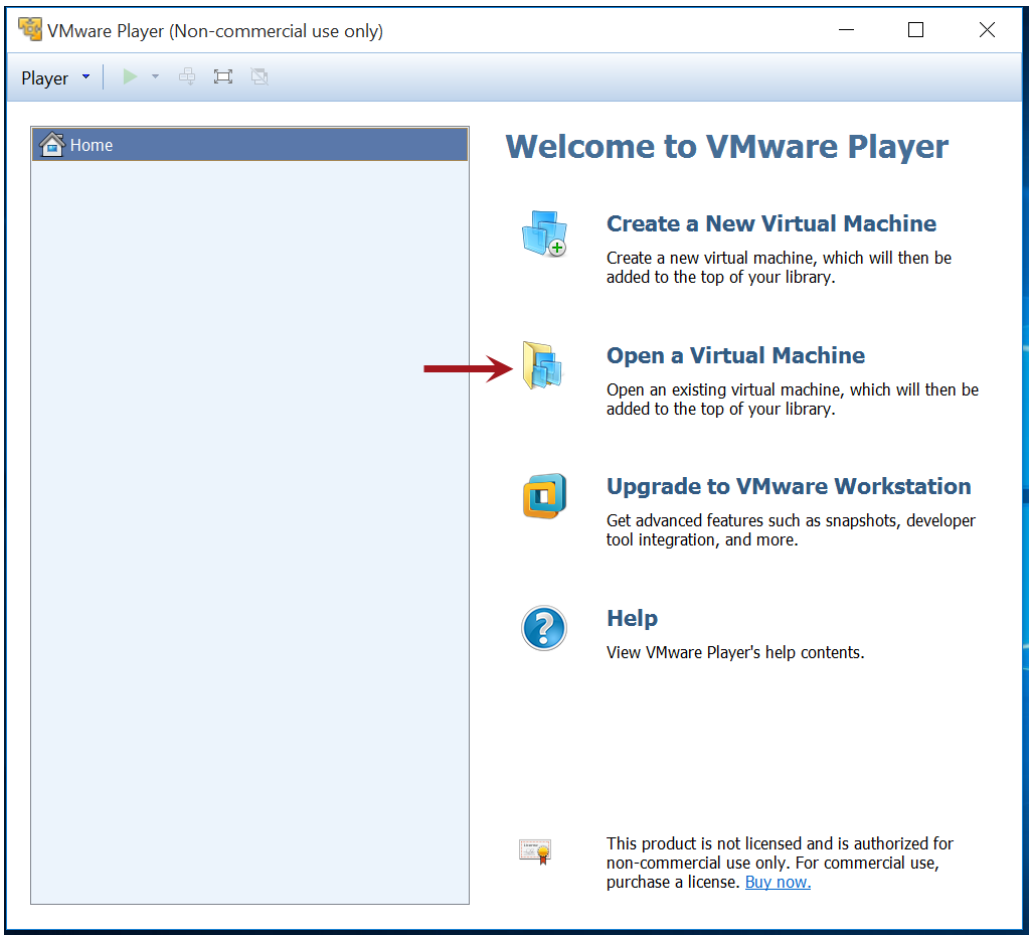
- From the same training bundle, locate the folder named VM image> 'technocrafty-quickstart-vm-5.5.0-0-vmware'
- Unzip/Extract the files and using the VMware player browse to this location to get start with virtual machine.

**Step3**: Loading the VMware Image

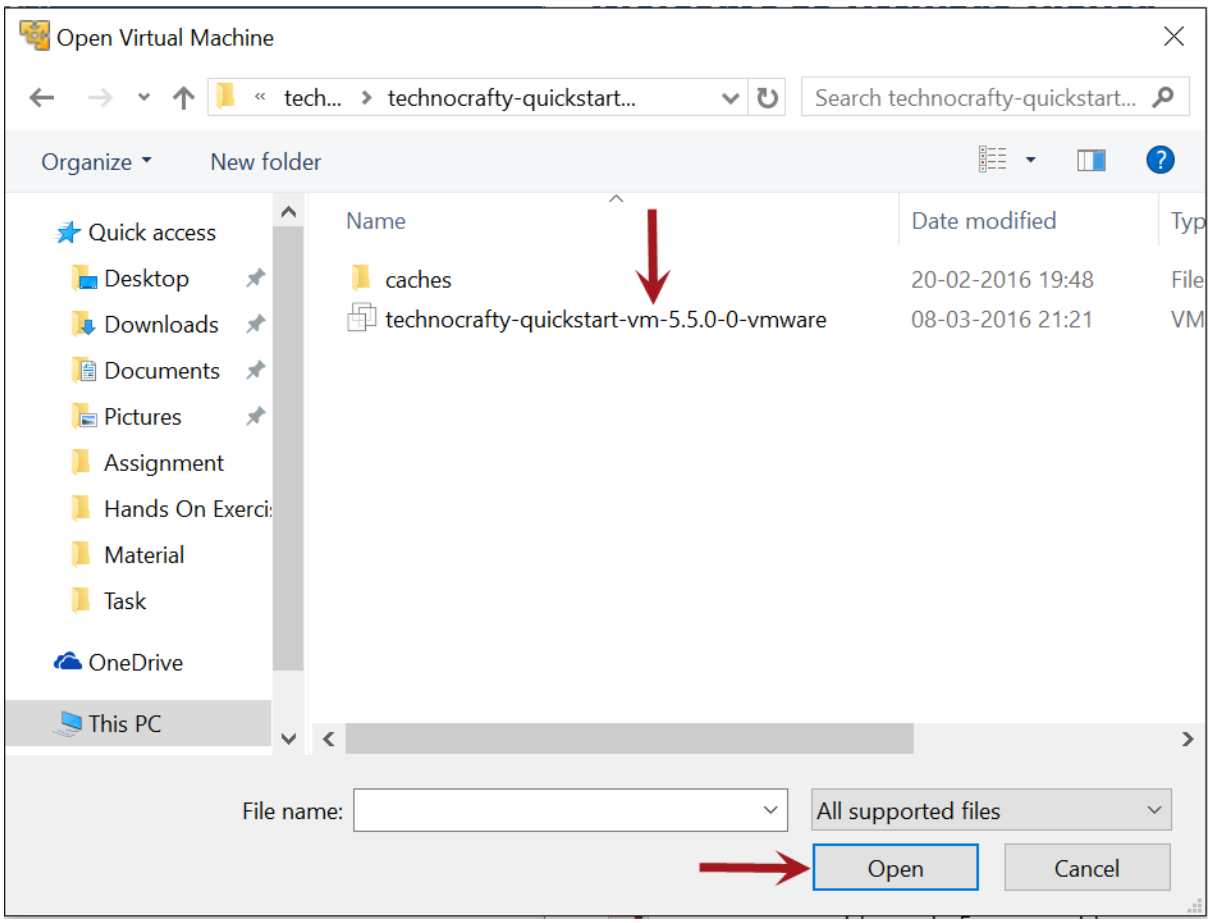
- Right click and select Open or Double click on 'VMware Player' and launch the same.



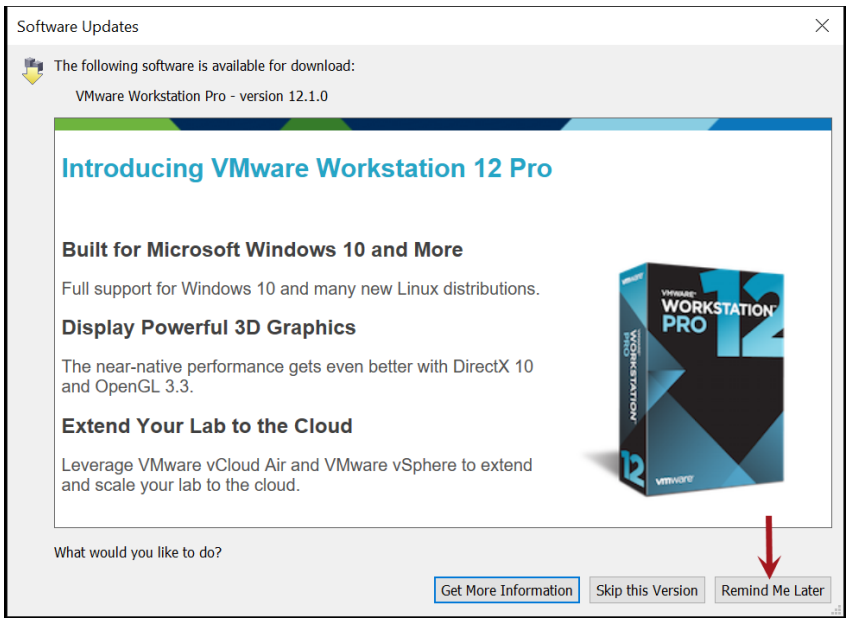
- Select the option as “Open a Virtual Machine”

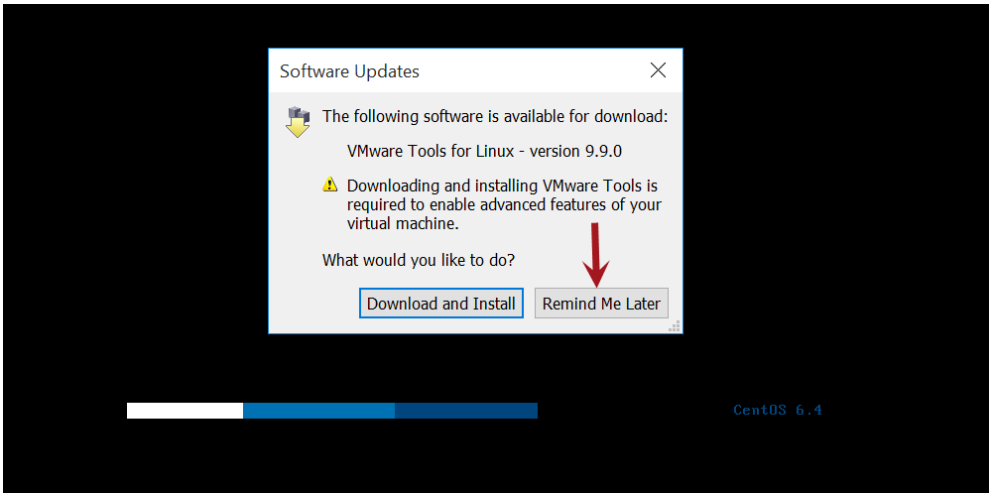


- Navigate to the location where you have extracted the VMware Image in earlier step.




- This will start the Virtual Machine.






- VM is set with following user credentials

  **Username**: technocrafty
  **Password**: technocrafty

Note This Virtual Machine is running on CentOS Linux distribution, Hadoop is configured to run in Pseudo-Distributed Mode i.e. all Hadoop Daemons are running on the same machine with a block replication factor of 1, as there is only one DataNode available.

- Verify all the running services with jps command

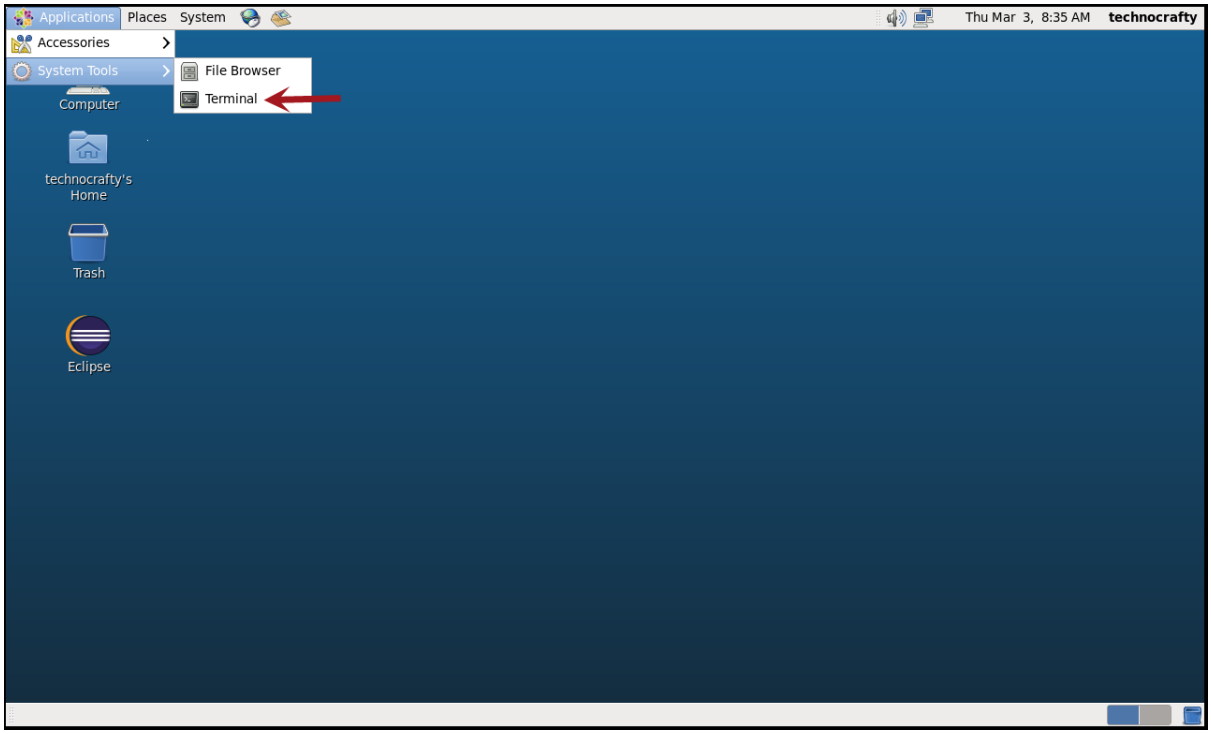
```
[technocrafty@quickstart ~]$ sudo jps
[sudo] password for technocrafty:
2822 ResourceManager
4976
3461 RunJar
2504 Bootstrap
4040 HistoryServer
4921
3201 ThriftServer
2049 DataNode
2557 JobHistoryServer
4182 HRegionServer
4630 Bootstrap
3083 RESTServer
2970 HMaster
2327 SecondaryNameNode
4018 Bootstrap
6085 Jps
1995 QuorumPeerMain
2125 JournalNode
2634 NodeManager
2208 NameNode
4905 Bootstrap
3275 RunJar
```

technocrafty

# Exercise 2: Practicing HDFS Commands

## Task 1: Using Hadoop Command Line Interface

Navigate to Application > System Tools > Terminal




- Operation on Files and directories

To check the content of root directory in HDFS

```
hdfs dfs -ls /
```

```
[technocrafty@quickstart ~]$ hdfs dfs -ls /
Found 5 items
drwxrwxrwx   - hdfs  supergroup          0 2015-11-18 13:03 /benchmarks
drwxr-xr-x   - hbase supergroup          0 2016-04-22 09:05 /hbase
drwxrwxrwt   - hdfs  supergroup          0 2016-03-28 07:52 /tmp
drwxr-xr-x   - hdfs  supergroup          0 2016-02-29 07:54 /user
drwxr-xr-x   - hdfs  supergroup          0 2015-11-18 13:06 /var
[technocrafty@quickstart ~]$
```

View the content of /user directory

```
hdfs dfs -ls /user
```

```
[technocrafty@quickstart ~]$ hdfs dfs -ls /user
Found 10 items
drwxr-xr-x   - cloudera     cloudera               0 2015-11-18 13:01 /user/cloudera
drwxr-xr-x   - hdfs         supergroup             0 2016-02-22 07:51 /user/hdfs
drwxr-xr-x   - mapred       hadoop                 0 2015-11-18 13:03 /user/history
drwxrwxrwx   - hive         supergroup             0 2015-11-18 13:06 /user/hive
drwxrwxrwx   - hue          supergroup             0 2016-03-05 06:46 /user/hue
drwxrwxrwx   - jenkins      supergroup             0 2015-11-18 13:03 /user/jenkins
drwxrwxrwx   - oozie        supergroup             0 2015-11-18 13:04 /user/oozie
drwxrwxrwx   - root         supergroup             0 2015-11-18 13:04 /user/root
drwxr-xr-x   - hdfs         supergroup             0 2015-11-18 13:06 /user/spark
drwxr-xr-x   - technocrafty technocrafty           0 2016-03-28 07:52 /user/technocrafty
[technocrafty@quickstart ~]$
```


For an empty directory the prompt does not show any error while querying whereas if the directory doesn't exist it will throw an error.

Example: View the content of /user/technocrafty directory which is empty and compare with the output of querying a non-existing directory say /music

```
hdfs dfs -ls /user/technocrafty

hdfs dfs -ls /music
```

```
[technocrafty@quickstart ~]$ hdfs dfs -ls /music
ls: `/music': No such file or directory
[technocrafty@quickstart ~]$
```


The directory structure of Hadoop filesystem is different from local filesystem i.e. Linux filesystem. To know the difference, list the files on your local filesystem.

Create a new directory named "hadoop" in HDFS

```
hdfs dfs -mkdir hadoop
```

Create a sample.txt file on local and upload the file into newly created directory

```
hdfs dfs -put sample.txt hadoop/sample.txt
```

View the content of new file

```
hdfs dfs -cat hadoop/sample.txt
```


In HDFS, any non-absolute path is considered relative to your home directory i.e. /user/technocrafty. Unlike Linux filesystem concept of "current" or "present working directory".

Download a file from HDFS to your local filesystem, specify HDFS path and local path to achieve the same.

```
hdfs dfs -get /user/technocrafty/sample.text /home/technocrafty
```

Remove the directory along with its content (perform this step after completing Task 2)

```
hdfs dfs -rm -r hadoop
```

List all the shell commands supported by Hadoop

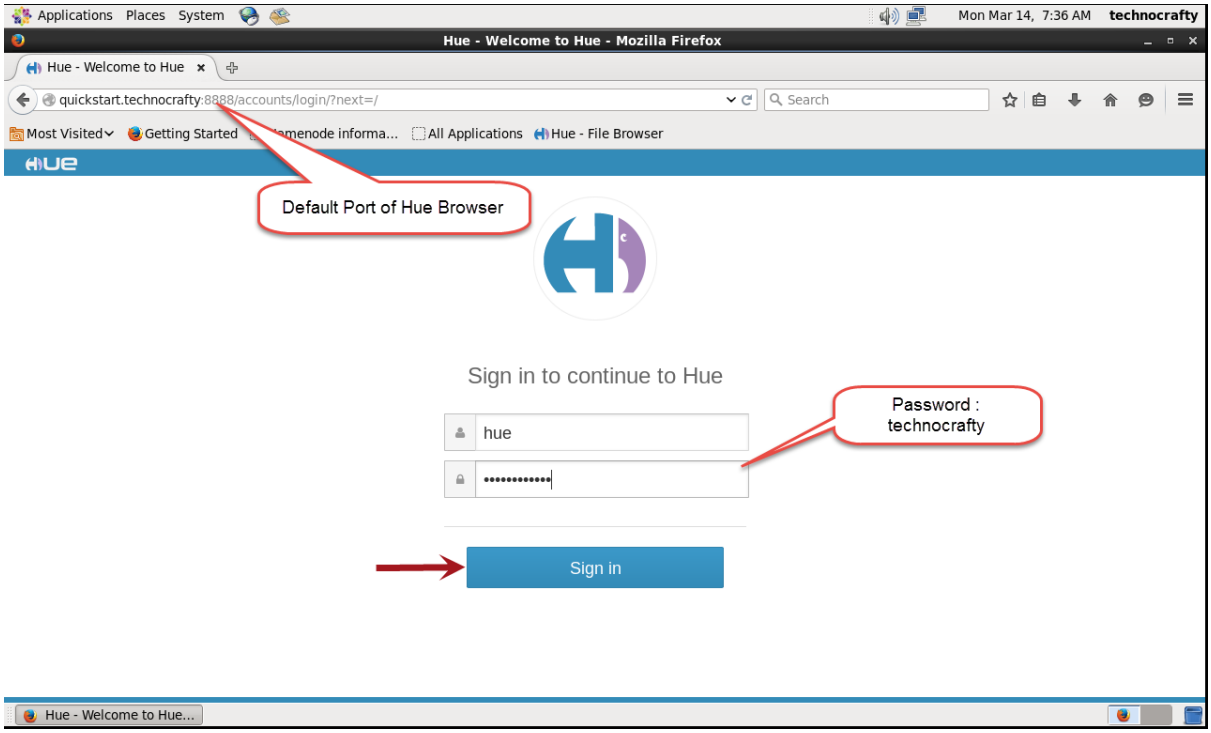
```
hdfs dfs
```

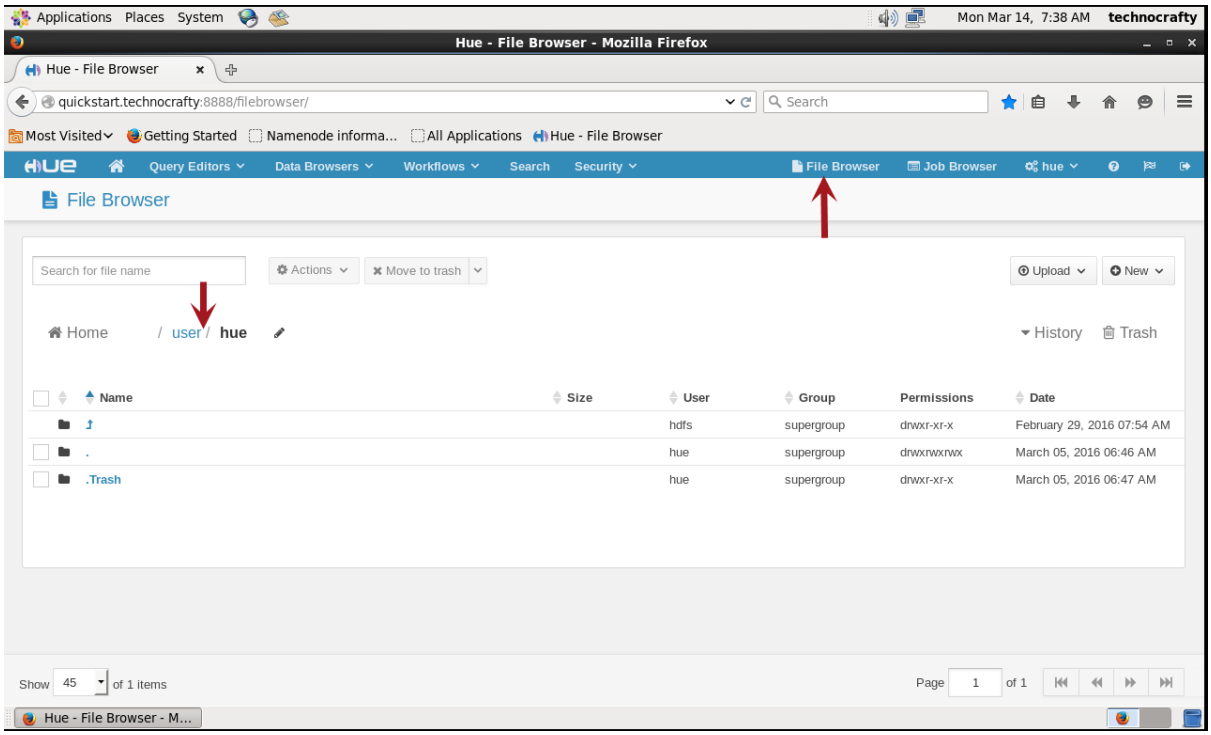
## Task2: Using HUE File browser

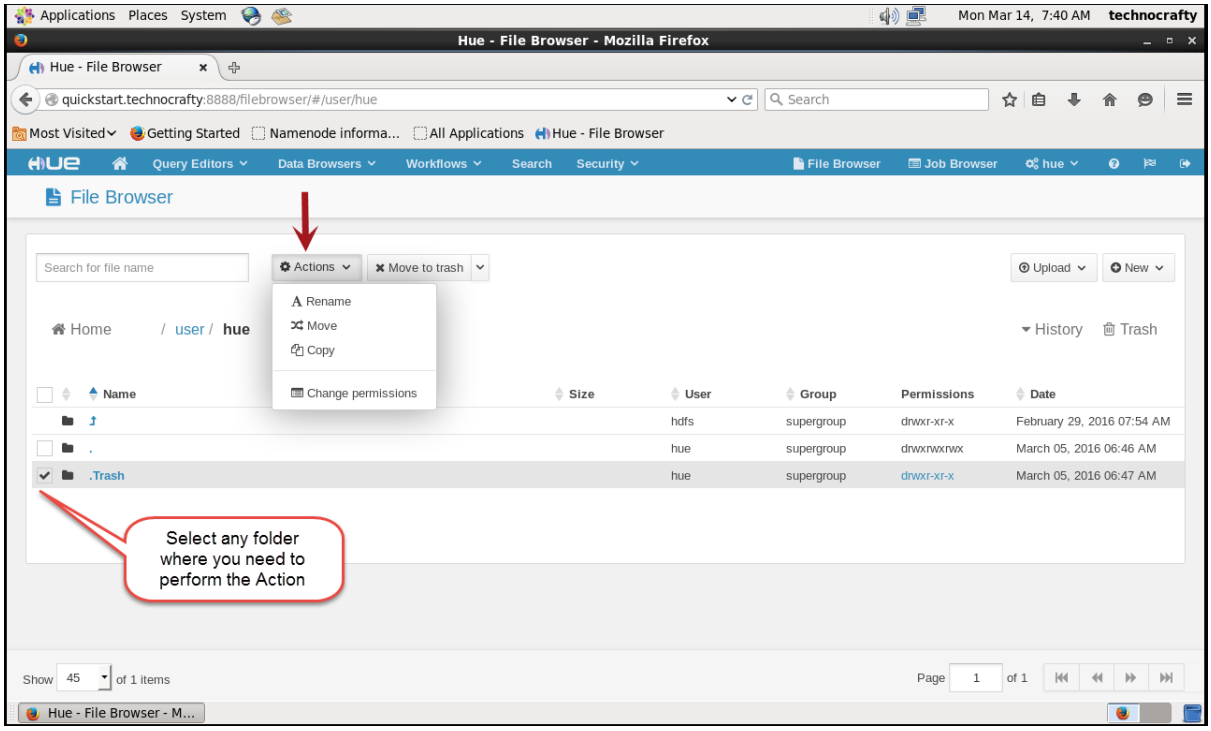
1. Open a web browser on your VM, and navigate to bookmark tab and select HUE. This can be alternatively launched by specifying http://quickstart.technocrafty:8888
2. Enter username as 'hue' and password 'technocrafty'.
3. To access HDFS, click on File Browser in the HUE menu bar
4. By default, the content of home directory i.e. /user/hue will be visible. Expand the slash (/) to view the content of root directory.
5. Point out to the directory named "hadoop" created above and view the file 'sample.txt'
6. Click on View file location in the Action panel on the left
7. Click on the up-arrow to return to the previous directory
8. Click on Upload button and select the appropriate format of the file to be uploaded i.e. plain file or zipped file

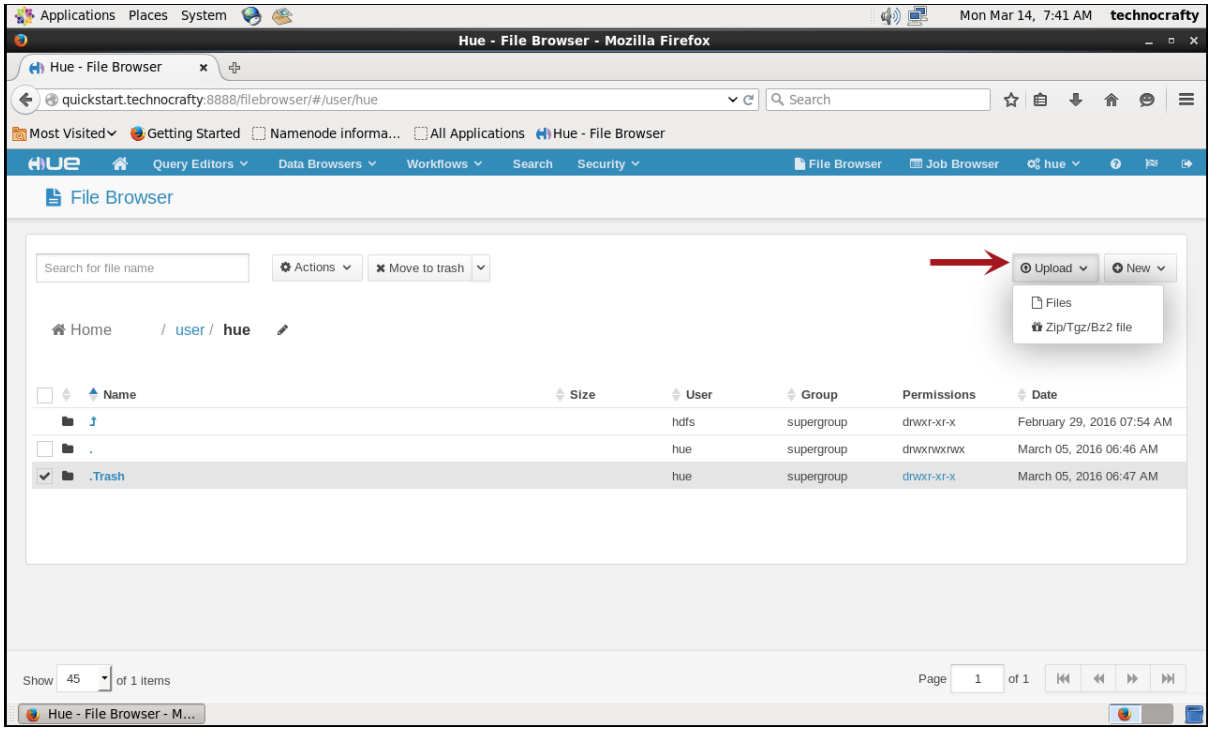

The zipped file will be automatically unzipped after upload.

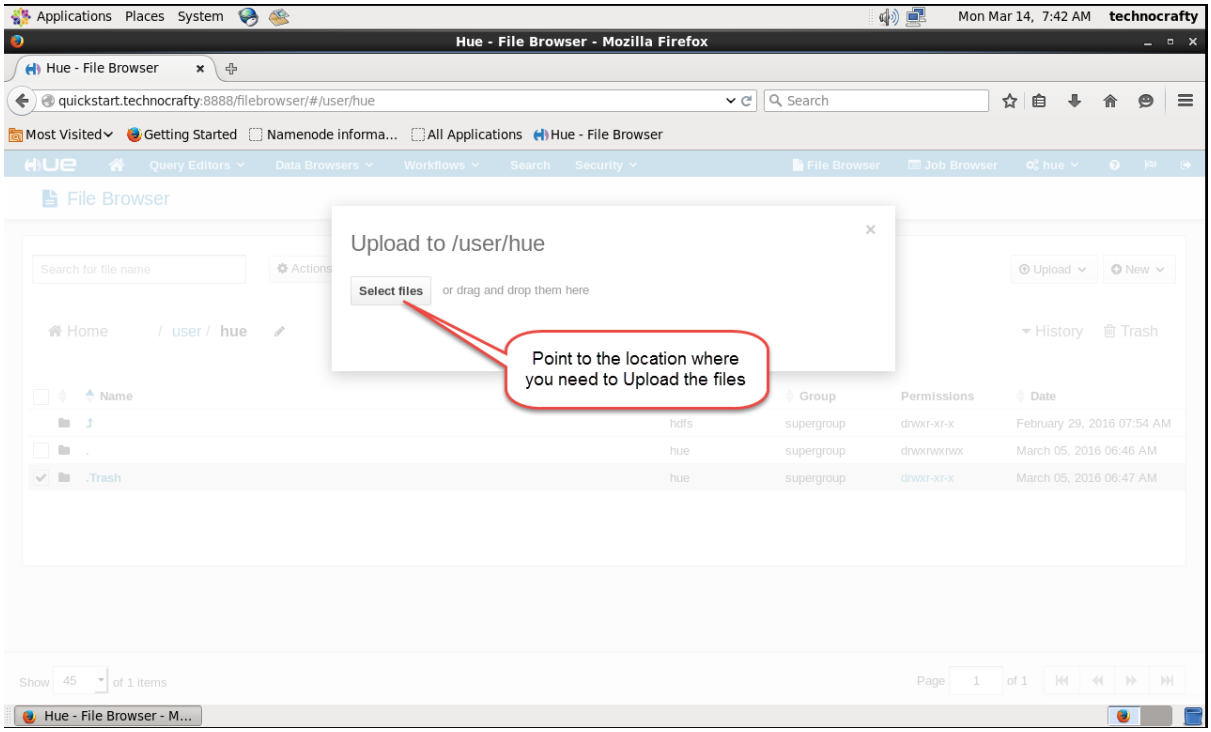
9. Select Files> Select Files, and browser to location of data file on your local filesystem
10. Choose the file and click the Open button.
11. The selected file will be displayed in the directory /user/technocrafty/
12. Click on the checkbox next to file's icon and then tab on Actions button to know the possible actions that can be performed on the file.
13. Once you have practised above steps, you can remove the files by clicking on "Move to Trash" button.


Applications Places System
Mon Mar 14, 7:36 AM technocrafty
Hue - Welcome to Hue - Mozilla Firefox
Hue - Welcome to Hue
quickstart.technocrafty:8888/accounts/login/?next=/
Search
Most Visited
Getting Started
All Applications
Hue - File Browser
Default Port of Hue Browser
Sign in to continue to Hue
hue
Password : technocrafty
Sign in
Hue - Welcome to Hue...


Applications Places System
Mon Mar 14, 7:38 AM technocrafty
Hue - File Browser - Mozilla Firefox
Hue - File Browser
quickstart.technocrafty:8888/filebrowser/
Search
Most Visited
Getting Started
Namenode informa...
All Applications
Hue - File Browser
Query Editors
Data Browsers
Workflows
Search
Security
File Browser
Job Browser
hue
File Browser
Search for file name
Actions
Move to trash
Upload
New
Home / user / hue
History
Trash
Name
Size
User
Group
Permissions
Date
hdfs supergroup drwxr-xr-x February 29, 2016 07:54 AM
hue supergroup drwxrwxrwx March 05, 2016 06:46 AM
.Trash
hue supergroup drwxr-xr-x March 05, 2016 06:47 AM
Show 45 of 1 items
Page 1 of 1
Hue - File Browser - M...


Applications Places System
Mon Mar 14, 7:40 AM technocrafty
Hue - File Browser - Mozilla Firefox
Hue - File Browser
quickstart.technocrafty:8888/filebrowser/#/user/hue
Search
Most Visited
Getting Started
Namenode informa...
All Applications
Hue - File Browser
Query Editors
Data Browsers
Workflows
Search
Security
File Browser
Job Browser
hue
File Browser
Search for file name
Actions
Move to trash
Upload
New
Rename
Move
Copy
Change permissions
Home / user / hue
History
Trash
Name
Size
User
Group
Permissions
Date
hdfs supergroup drwxr-xr-x February 29, 2016 07:54 AM
hue supergroup drwxrwxrwx March 05, 2016 06:46 AM
.Trash
hue supergroup drwxr-xr-x March 05, 2016 06:47 AM
Select any folder where you need to perform the Action
Show 45 of 1 items
Page 1 of 1
Hue - File Browser - M...

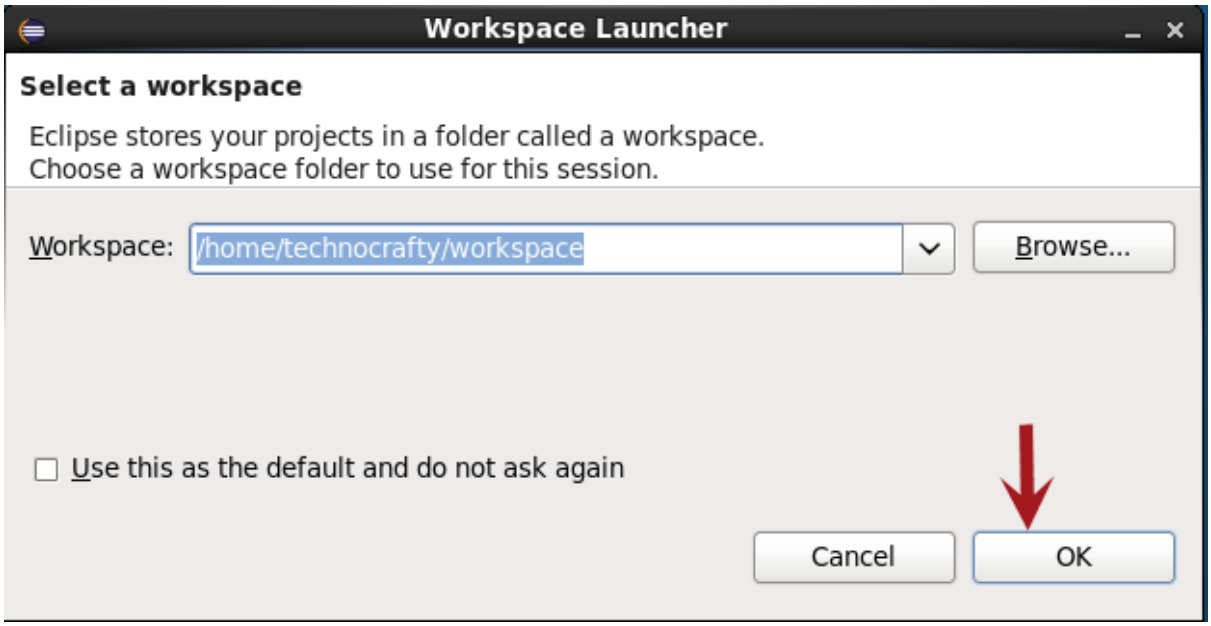

Applications Places System
Mon Mar 14, 7:41 AM technocrafty
Hue - File Browser - Mozilla Firefox
Hue - File Browser
quickstart.technocrafty:8888/filebrowser/#/user/hue
Search
Most Visited
Getting Started
Namenode informa...
All Applications
Hue - File Browser
Query Editors
Data Browsers
Workflows
Search
Security
File Browser
Job Browser
hue
File Browser
Search for file name
Actions
Move to trash
Upload
New
Files
Zip/Tgz/Bz2 file
Home / user / hue
Name
Size
User
Group
Permissions
Date
hdfs supergroup drwxr-xr-x February 29, 2016 07:54 AM
hue supergroup drwxrwxrwx March 05, 2016 06:46 AM
.Trash
hue supergroup drwxr-xr-x March 05, 2016 06:47 AM
Show 45 of 1 items
Page 1 of 1
Hue - File Browser - M...


Applications Places System
Mon Mar 14, 7:42 AM technocrafty
Hue - File Browser - Mozilla Firefox
Hue - File Browser
quickstart.technocrafty:8888/filebrowser/#/user/hue
Search
Most Visited
Getting Started
Namenode informa...
All Applications
Hue - File Browser
Query Editors
Data Browsers
Workflows
Search
Security
File Browser
Job Browser
hue
File Browser
Upload to /user/hue
Select files
or drag and drop them here
Point to the location where you need to Upload the files
Search for file name
Actions
Upload
New
Home / user / hue
History
Trash
Name
Group
Permissions
Date
hdfs supergroup drwxr-xr-x February 29, 2016 07:54 AM
hue supergroup drwxrwxrwx March 05, 2016 06:46 AM
.Trash
hue supergroup drwxr-xr-x March 05, 2016 06:47 AM
Show 45 of 1 items
Page 1 of 1
Hue - File Browser - M...

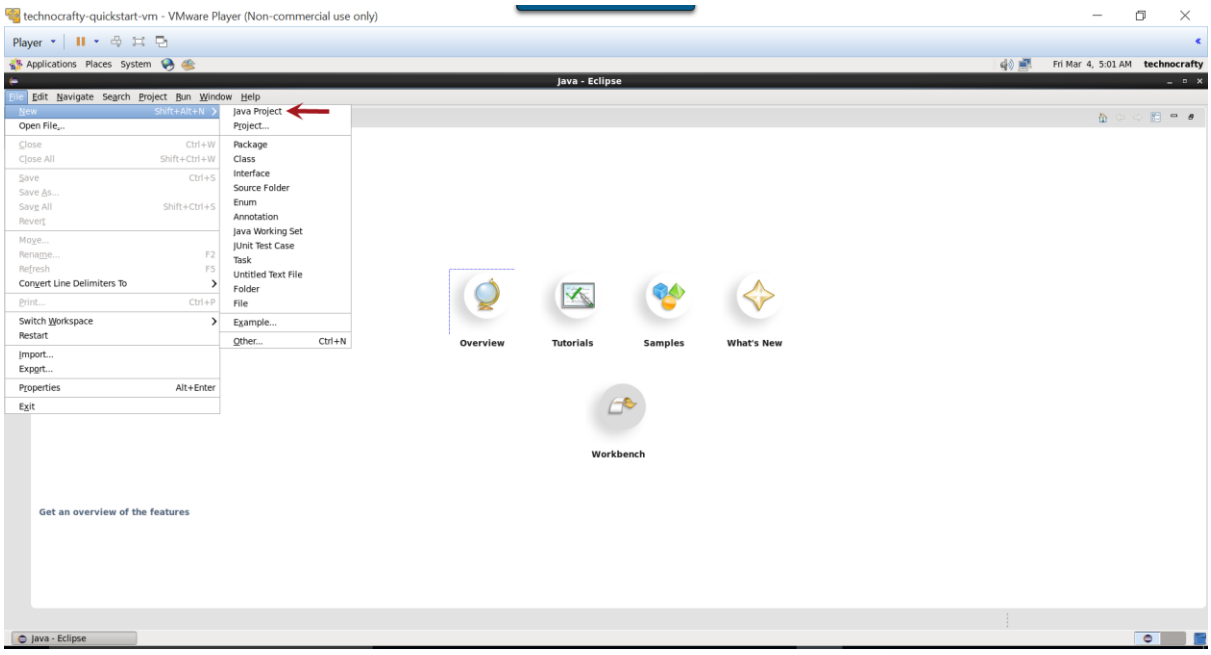
# Exercise 3: Running MapReduce Job

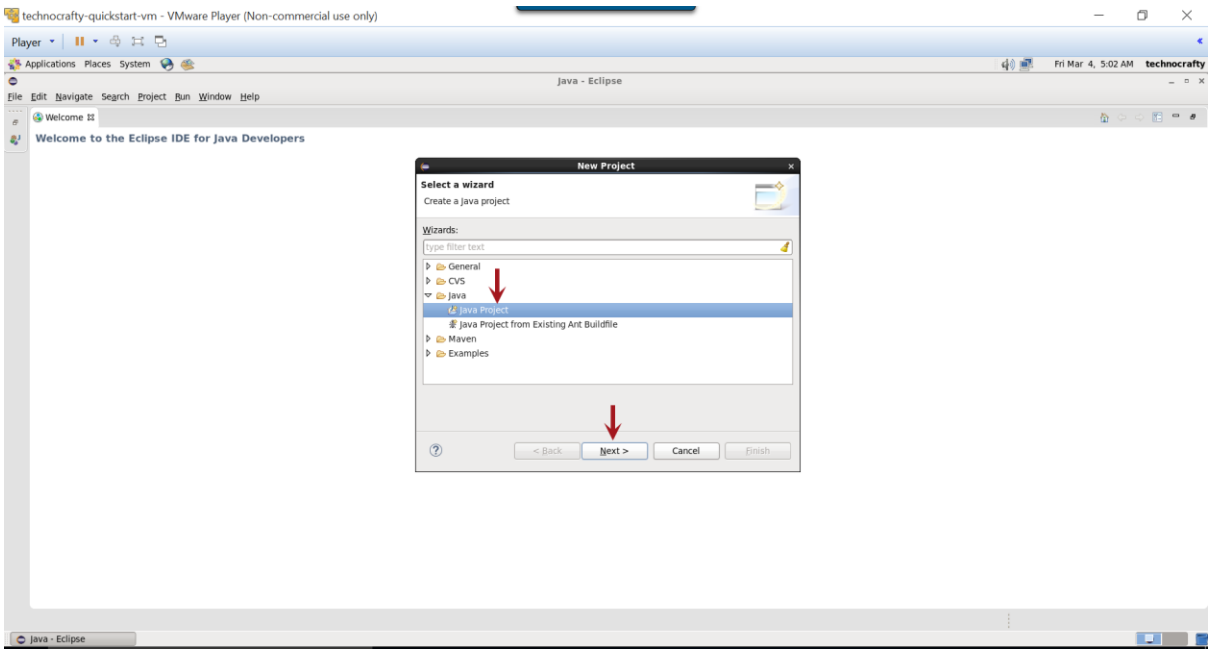
**Prerequisites**: Create an Eclipse WordCount Project

**Step1**: Select the workspace location, and click on "OK".

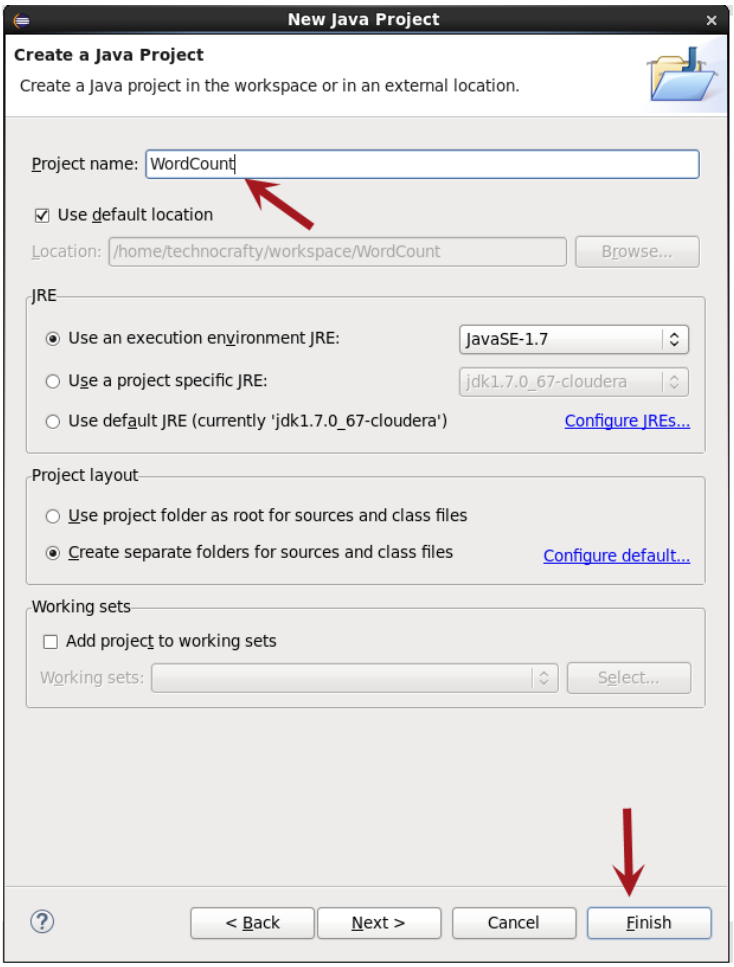



**Step2**: Select File > New > Java Project > Next

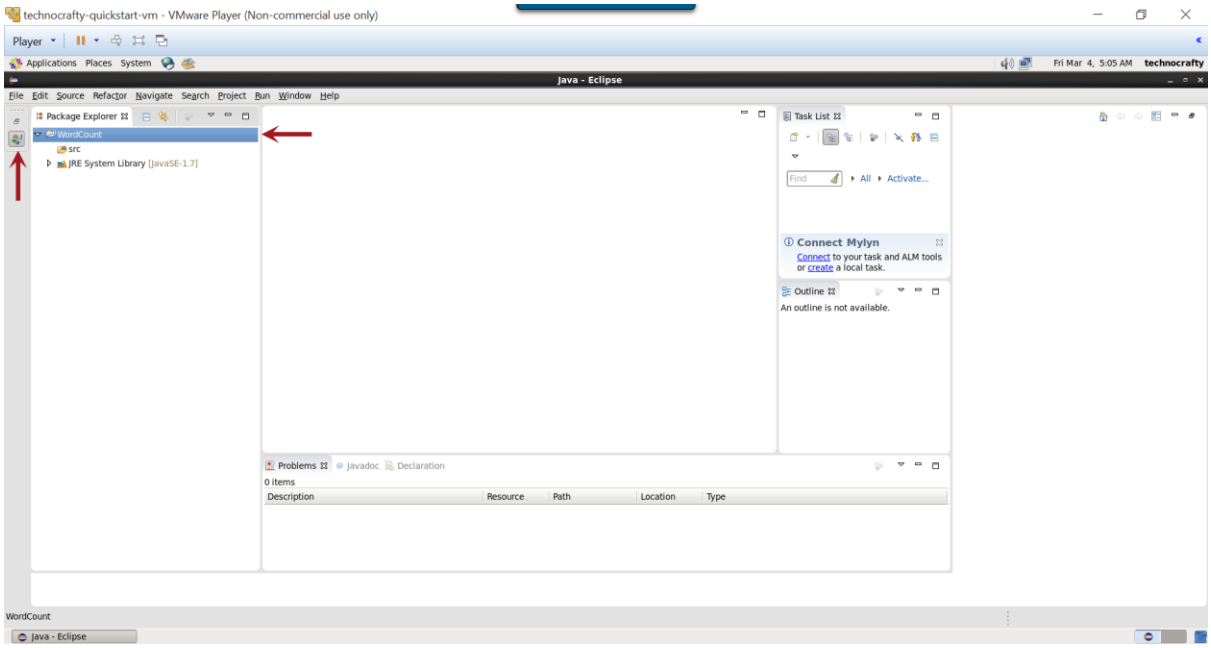




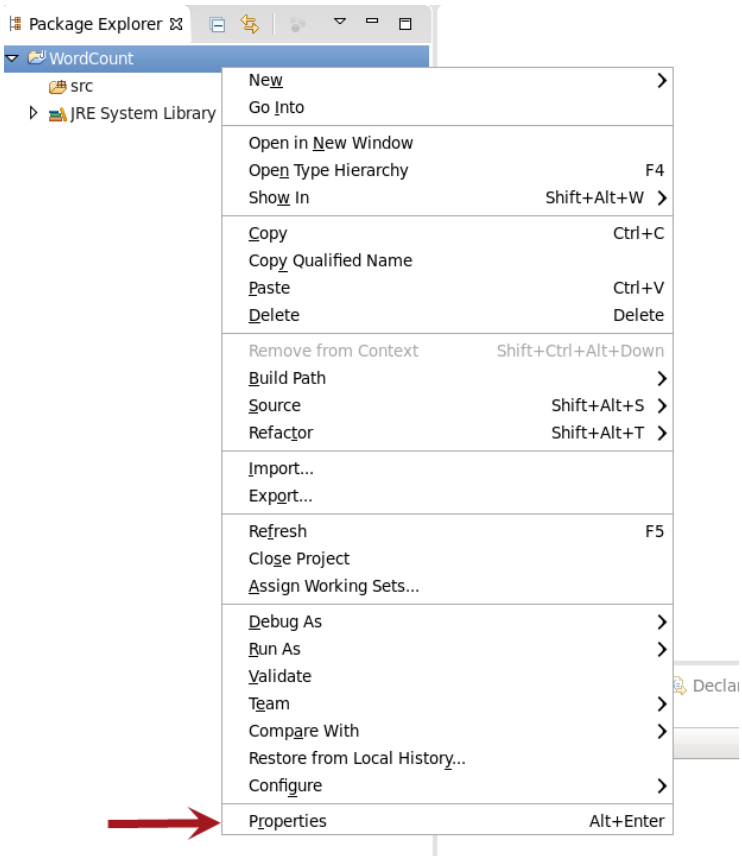

**Step3**: Name the project as "WordCount" and click "Finish"



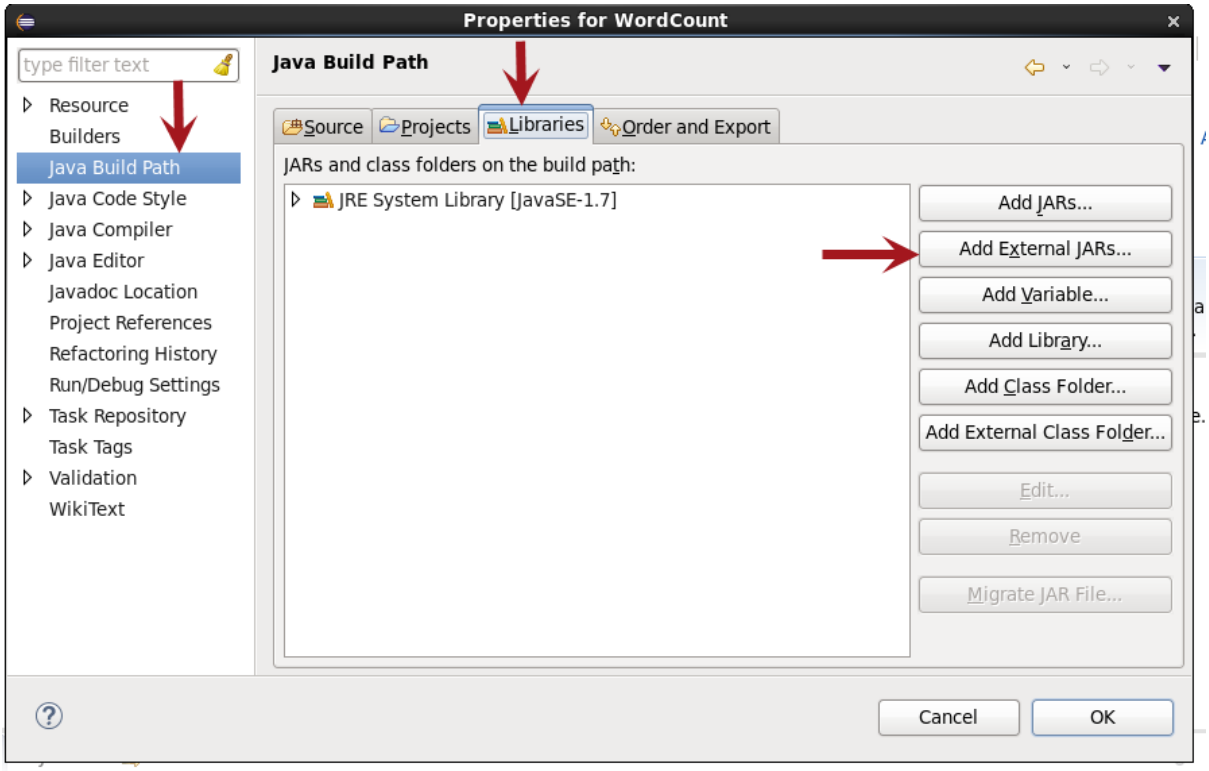

**Step4**: Expand the “Package Explorer” tab and locate your new project i.e. “WordCount”.



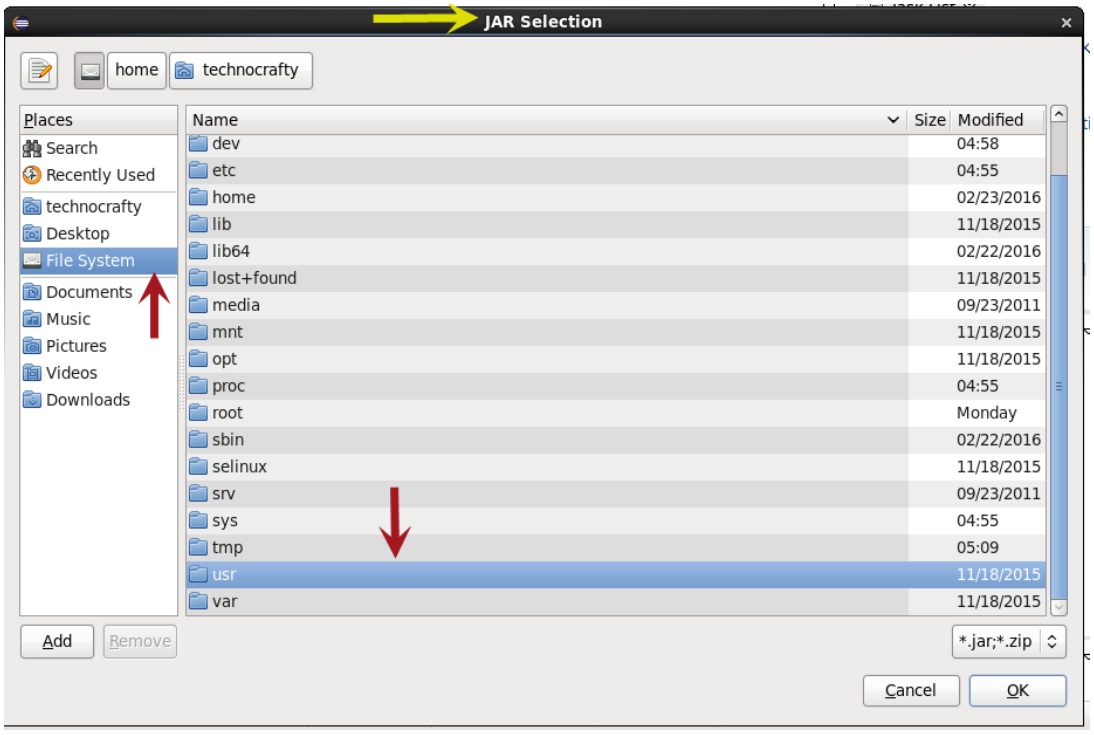

**Step5**: Export Hadoop Libraries by Right Click on WordCount project and selecting Properties

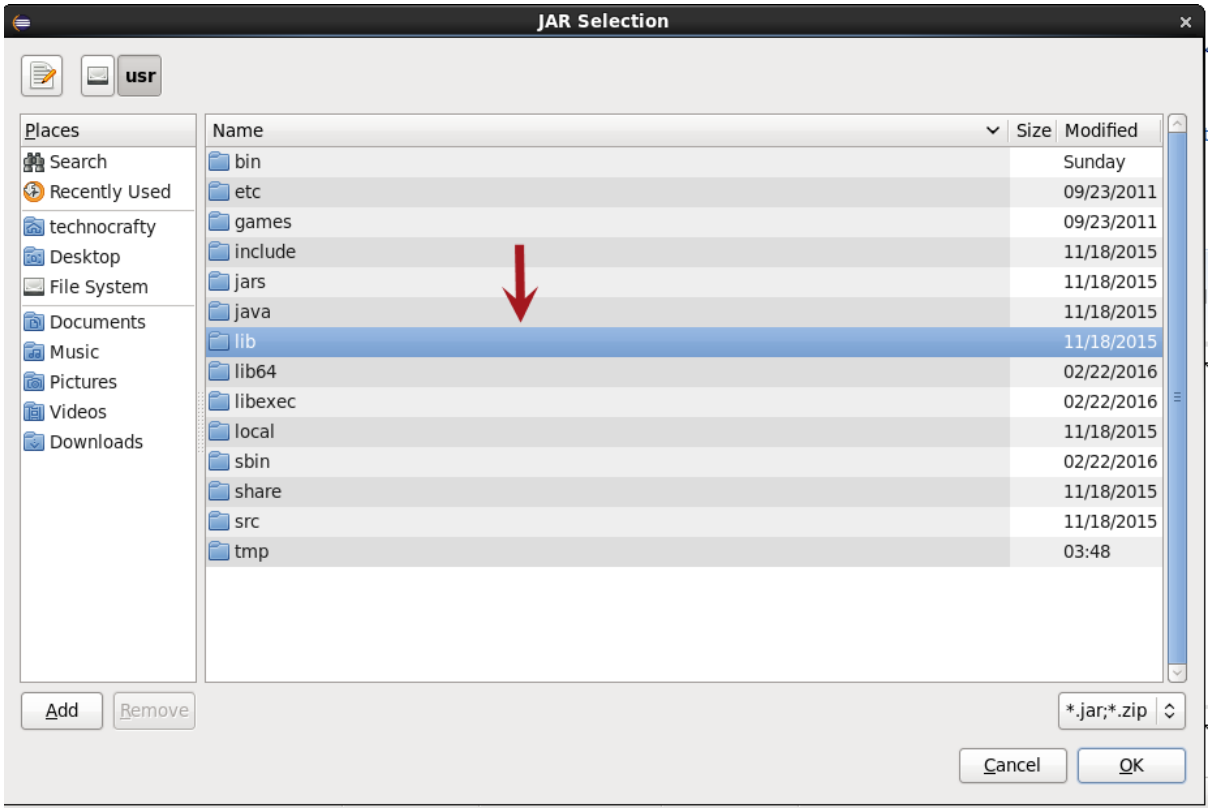


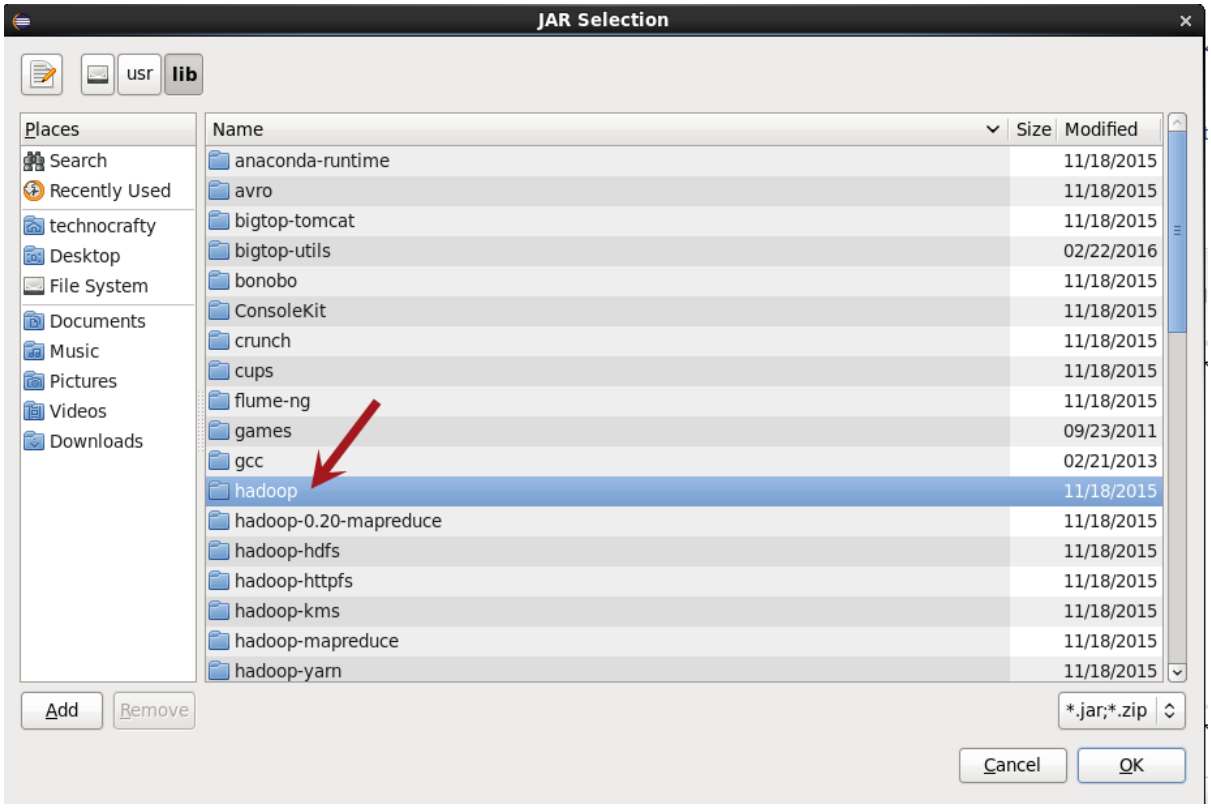
**Step6**: Click on Java Build Path from left panel and select "Libraries" tab from right panel view



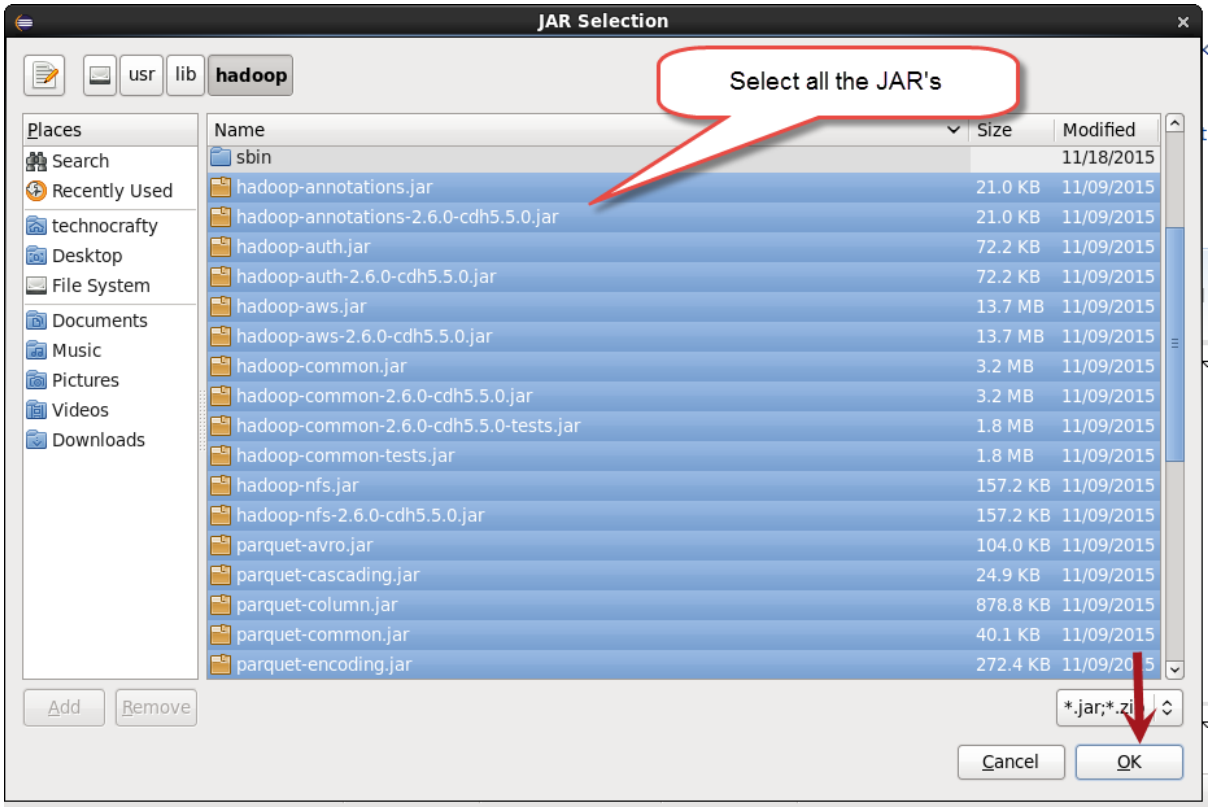

**Step7**: Select "Add External JAR…", then navigate to File System > usr > lib > Hadoop



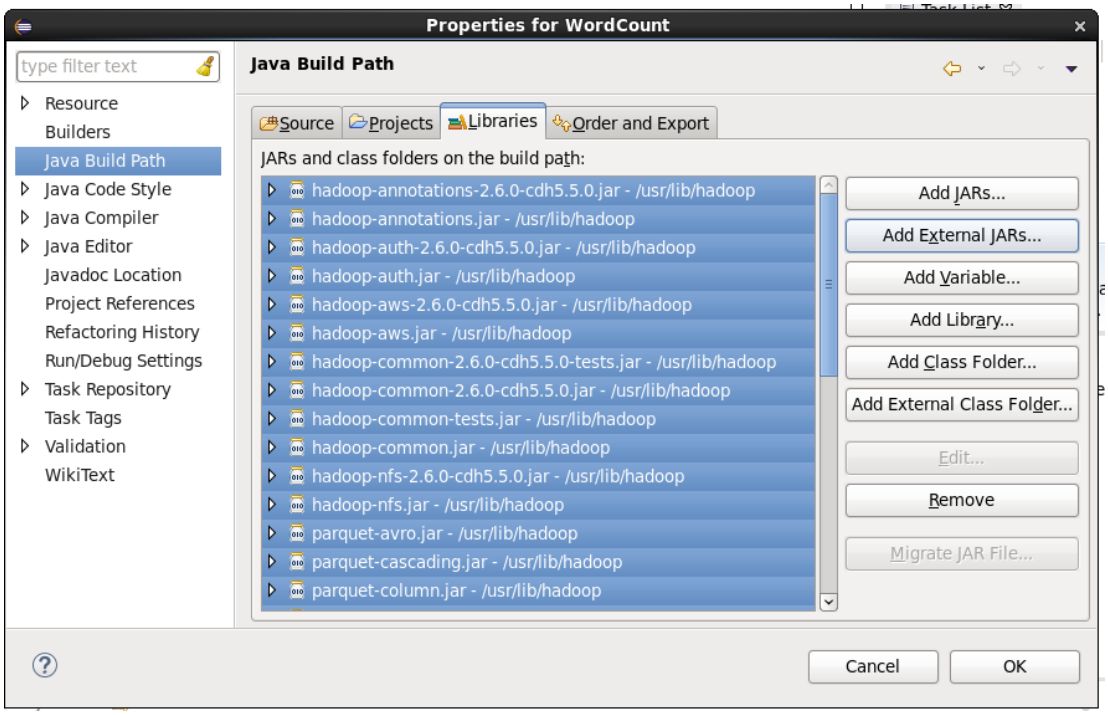



**Step8**: Select all the JAR available under this folder

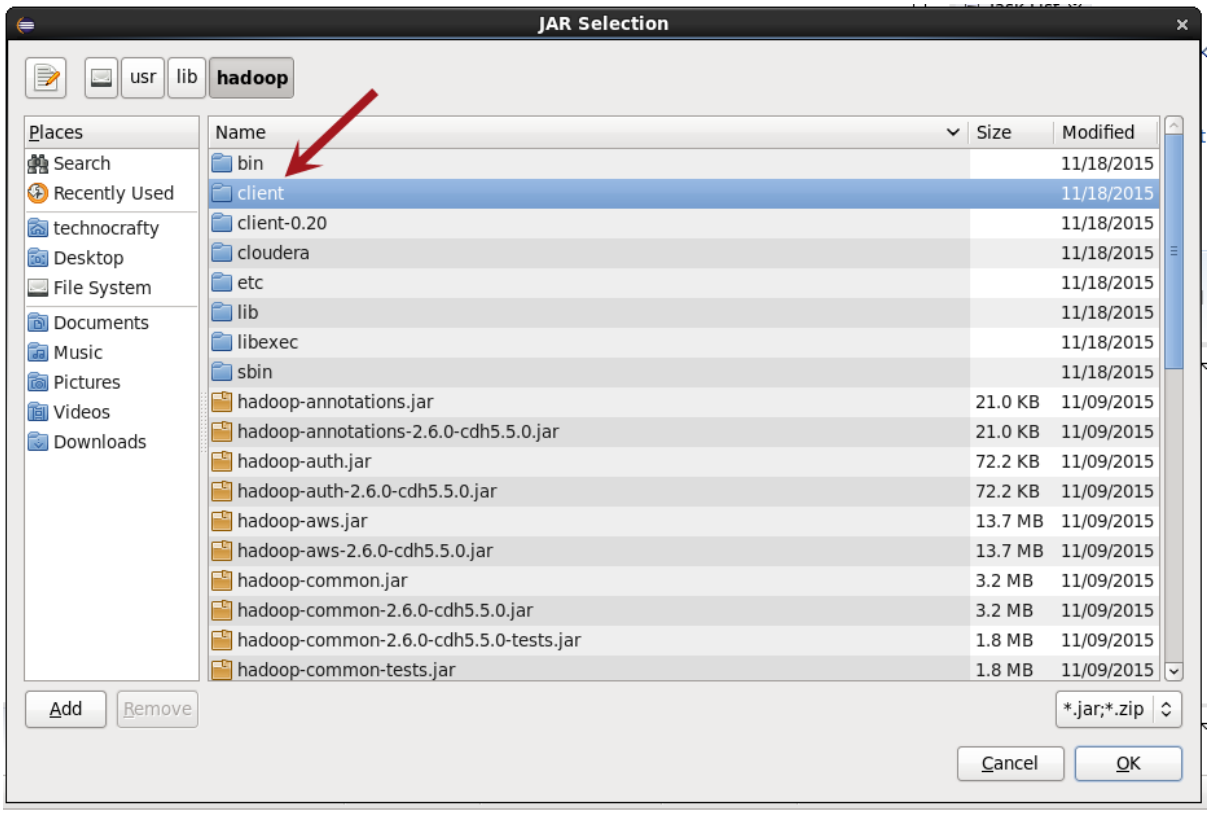



Click on "OK"

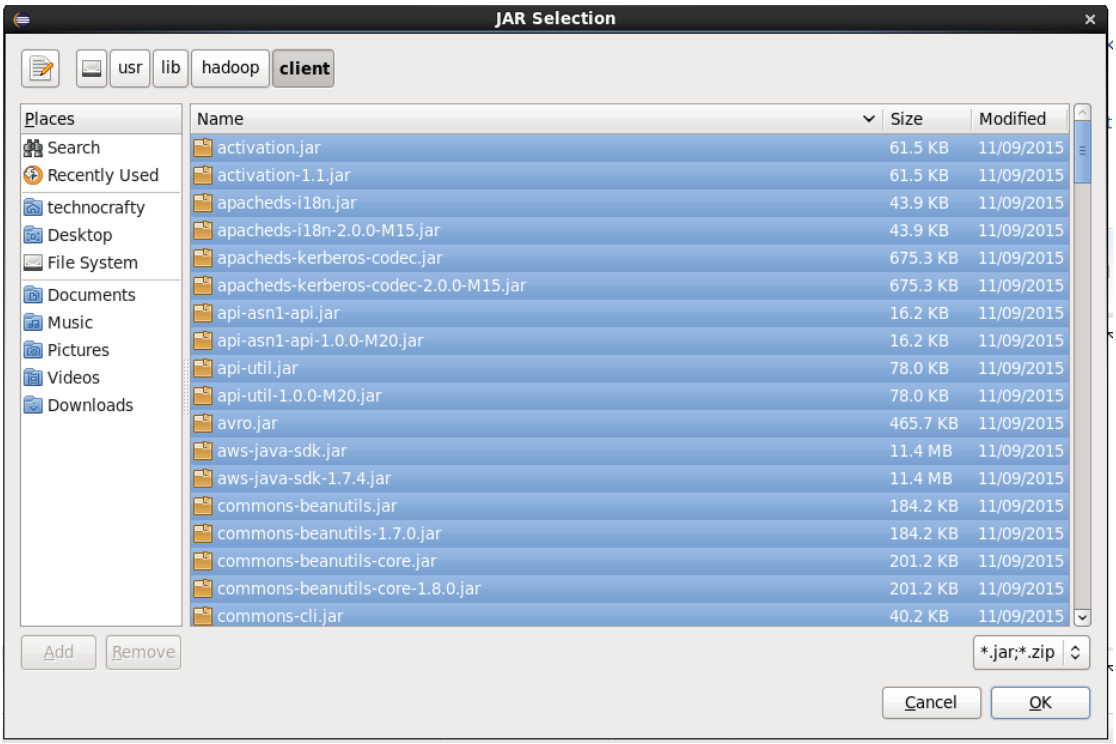


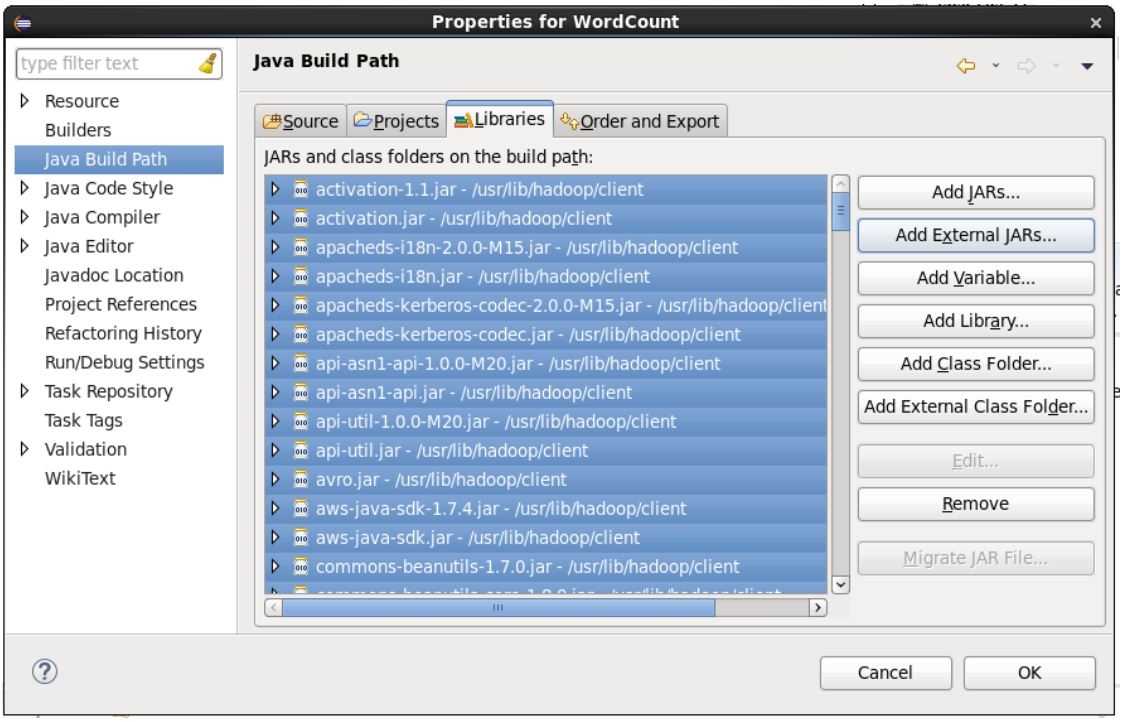

**Step 9**: Perform same steps for /usr/lib/hadoop-mapreduce and /usr/lib/Hadoop-yarn

**Step10**: Now grab all the library JAR files under "Client", by following the same method.

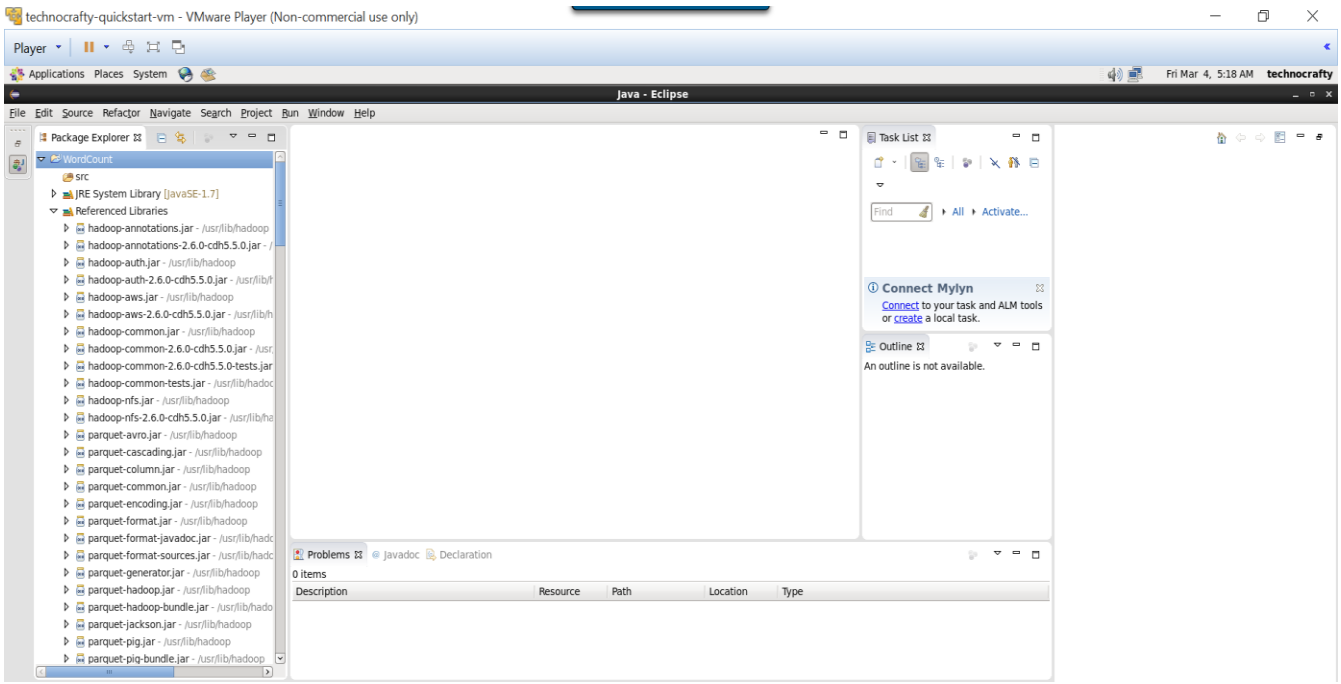



Select the files and click OK

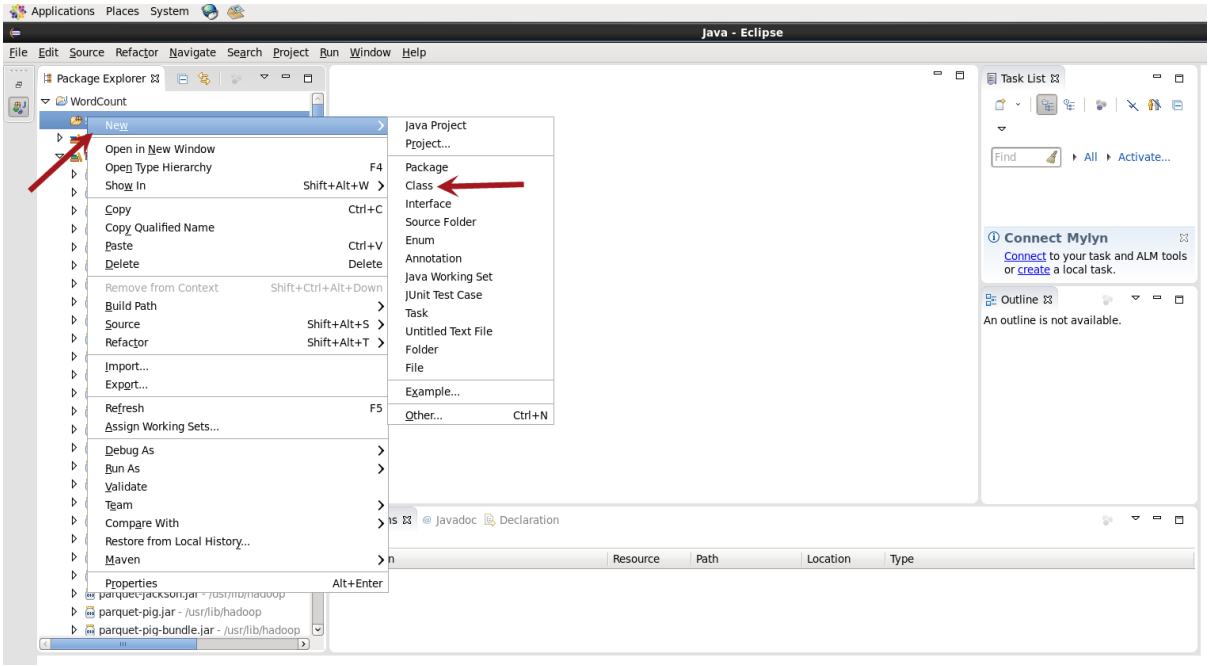




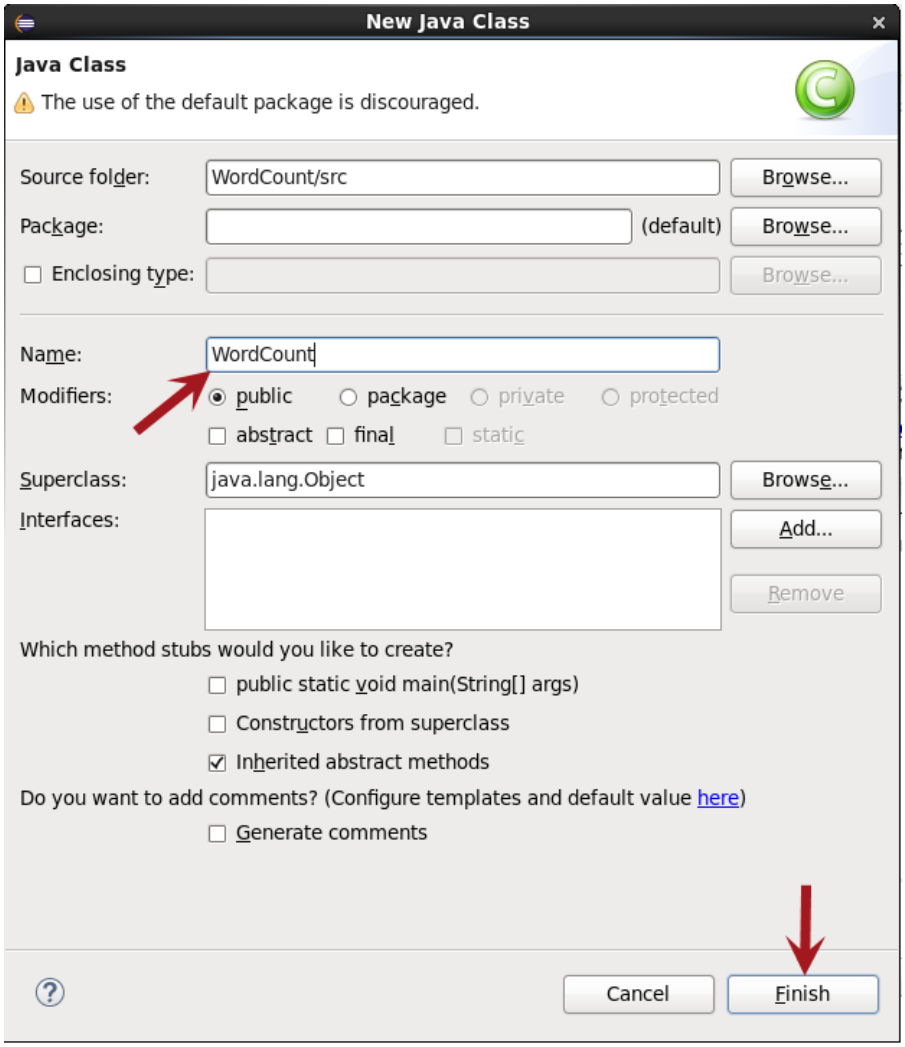

**Step11**: Finally, you will find all the JAR files under "Referenced Libraries"



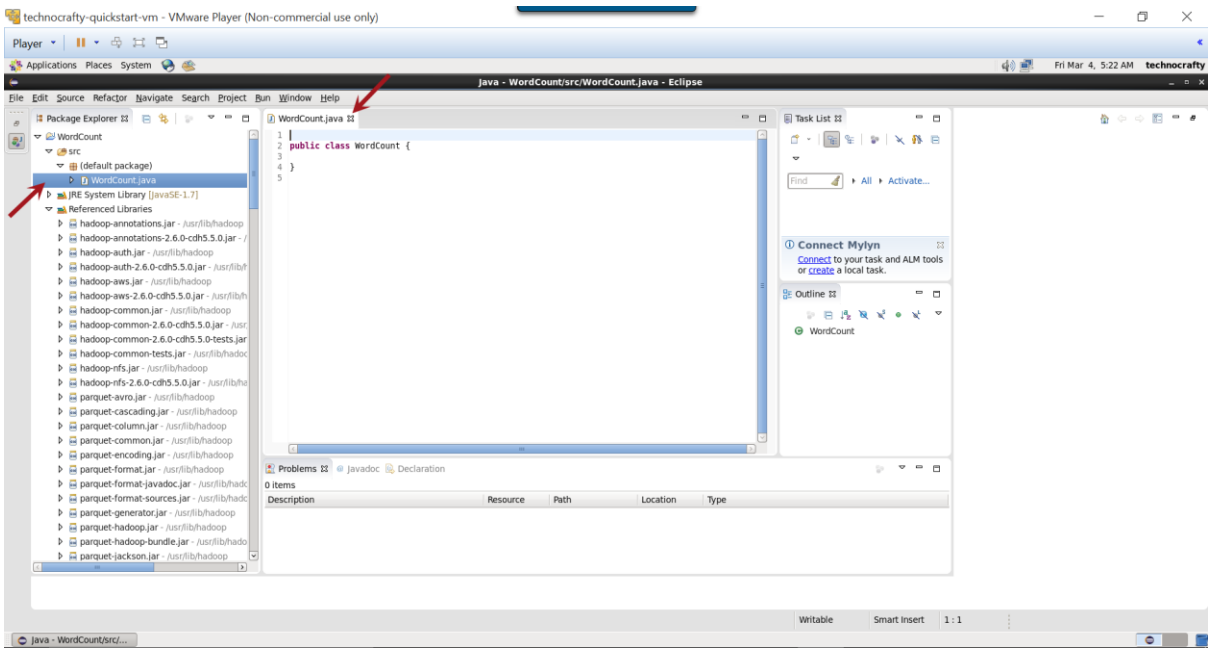
**Step12**: Creating the class files, Right click on src under WordCount and select New > Class




**Step13**: Enter the name of Java Class as "WordCount" and click on Finish.



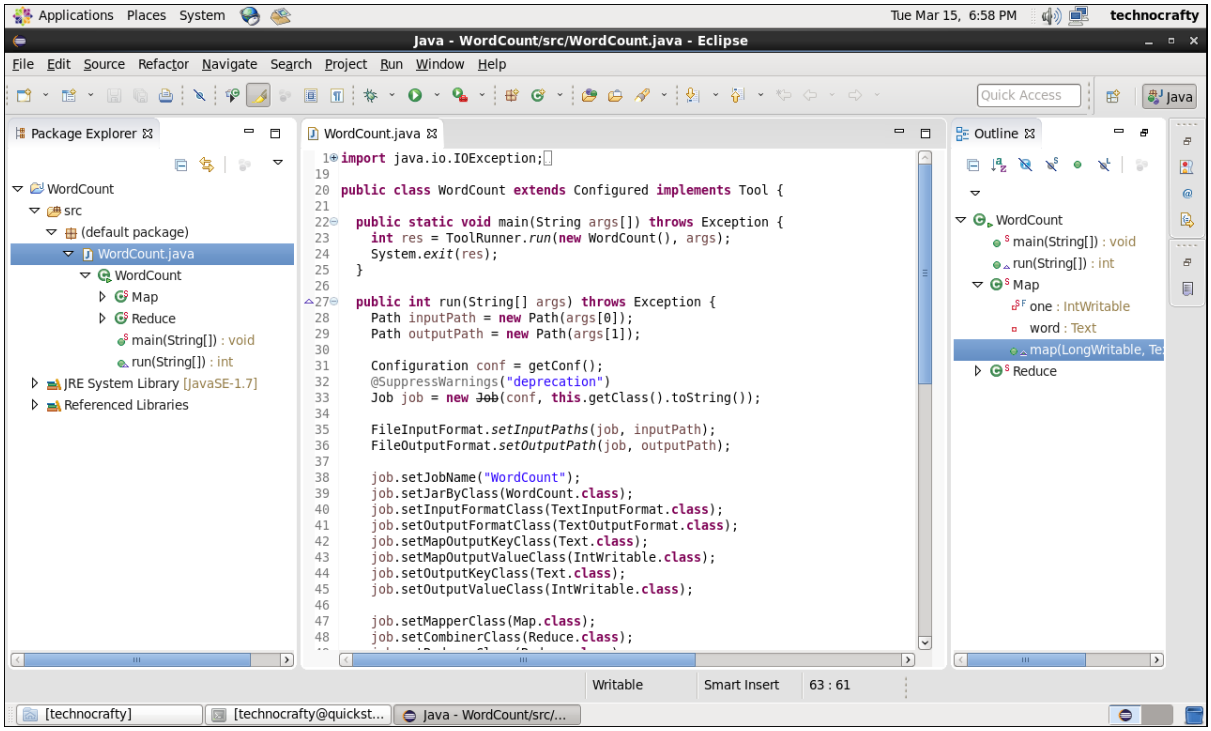

**Step14**: Now it's ready to take your input Mapreduce program



**Step15**: MapReduce Wordcount Program

WordCount.txt

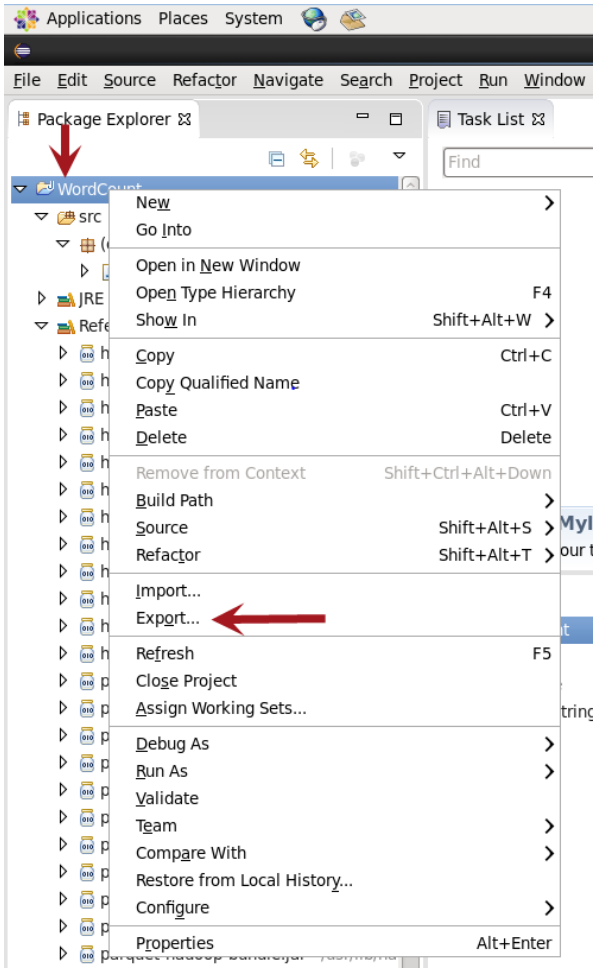
**Step 16**: Save the program



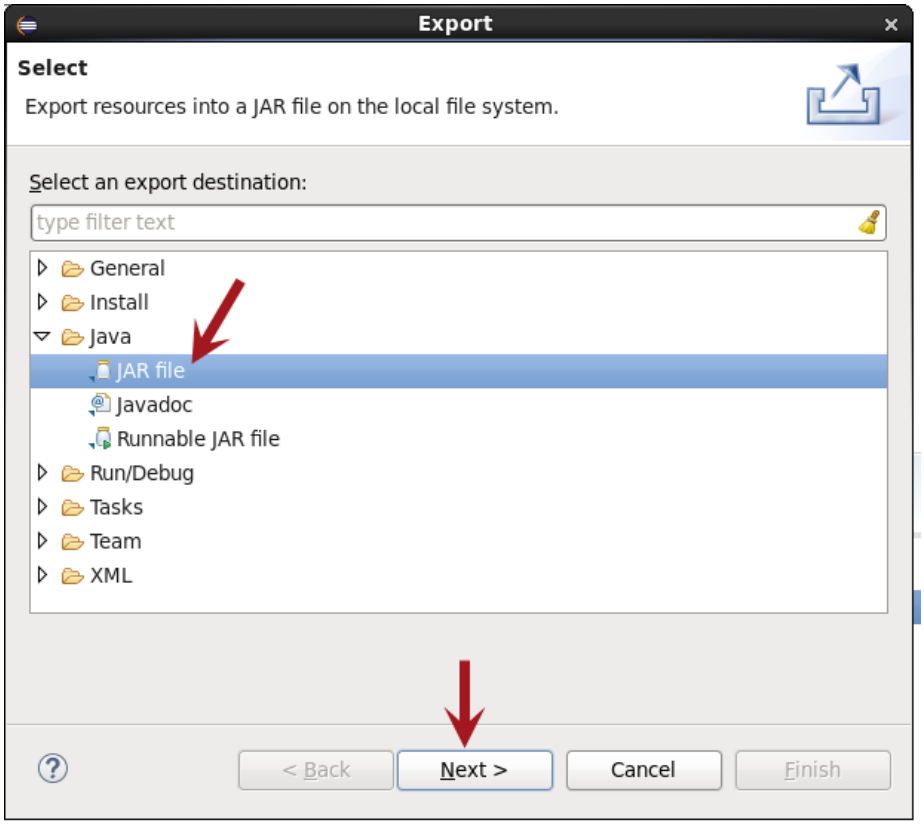


Ensure that there is no compilation error, before exporting the JAR file.

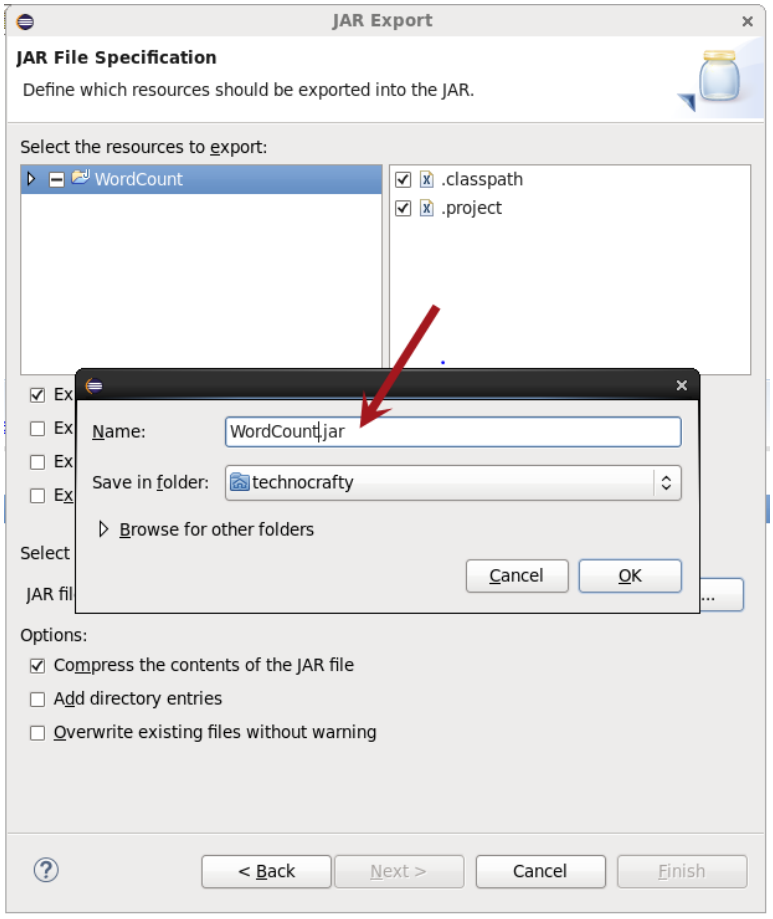
**Step17**: Exporting the JAR file, for that Right Click on WordCount project and select "Export"



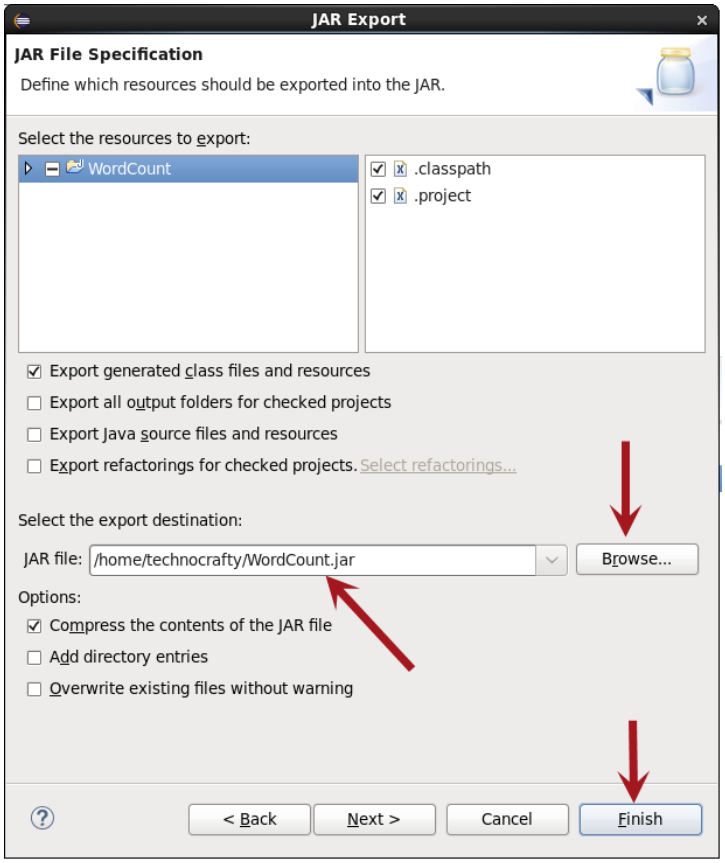

**Step 18**: In the next panel, expand Java and select JAR file




**Step19**: Name the JAR as "WordCount.jar" and select OK.



**Step20**: Browse to the location where you want to save the JAR and select FINISH.




**Step21**: Verify the file at the target location from the terminal.

```
[technocrafty@quickstart ~]$ ls -ltr
total 52
drwxrwsr-x 9 technocrafty technocrafty 4096 Feb 19  2015 eclipse
drwxr-xr-x 2 technocrafty technocrafty 4096 Feb 23 10:16 Videos
drwxr-xr-x 2 technocrafty technocrafty 4096 Feb 23 10:16 Templates
drwxr-xr-x 2 technocrafty technocrafty 4096 Feb 23 10:16 Public
drwxr-xr-x 2 technocrafty technocrafty 4096 Feb 23 10:16 Pictures
drwxr-xr-x 2 technocrafty technocrafty 4096 Feb 23 10:16 Music
drwxr-xr-x 2 technocrafty technocrafty 4096 Feb 23 10:16 Downloads
drwxr-xr-x 2 technocrafty technocrafty 4096 Feb 28 07:35 Desktop
drwxrwxr-x 4 technocrafty technocrafty 4096 Mar  4 05:04 workspace
drwxrwxr-x 2 technocrafty technocrafty 4096 Mar 11 07:17 Datasets
-rw-rw-r-- 1 technocrafty technocrafty 6004 Mar 14 08:02 WordCount.jar
drwxr-xr-x 3 technocrafty technocrafty 4096 Mar 15 18:46 Documents
[technocrafty@quickstart ~]$
```

- **Let's run a simple MapReduce WordCount program.**

**Step1**: Import the data from local filesystem to HDFS

Under home directory of local filesystem, locate a folder named "Datasets" followed by a file named Joyce.txt

```
[technocrafty@quickstart ~]$ cd Datasets/
[technocrafty@quickstart Datasets]$ pwd
/home/technocrafty/Datasets
[technocrafty@quickstart Datasets]$ ls -ltr
total 1540
-rw-rw-r-- 1 technocrafty technocrafty 1573079 Dec  9  2014 Joyce.txt
[technocrafty@quickstart Datasets]$
```

**Step2**: Create a directory named "input" in the HDFS and import "Joyce.txt" file.

```
[technocrafty@quickstart ~]$ hdfs dfs -mkdir input
[technocrafty@quickstart ~]$ hdfs dfs -ls input
[technocrafty@quickstart ~]$ pwd
/home/technocrafty
[technocrafty@quickstart ~]$ cd Datasets/
[technocrafty@quickstart Datasets]$ ls
Joyce.txt
[technocrafty@quickstart Datasets]$ hdfs dfs -put Joyce.txt input
[technocrafty@quickstart Datasets]$ hdfs dfs -ls input
Found 1 items
-rw-r--r--   1 technocrafty technocrafty    1573079 2016-03-15 19:17 input/Joyce.txt
[technocrafty@quickstart Datasets]$
```

**Step3:** Now run the program by using the WordCount.jar created in previous steps

```
[technocrafty@quickstart ~]$ hadoop jar WordCount.jar WordCount input/Joyce.txt output
```

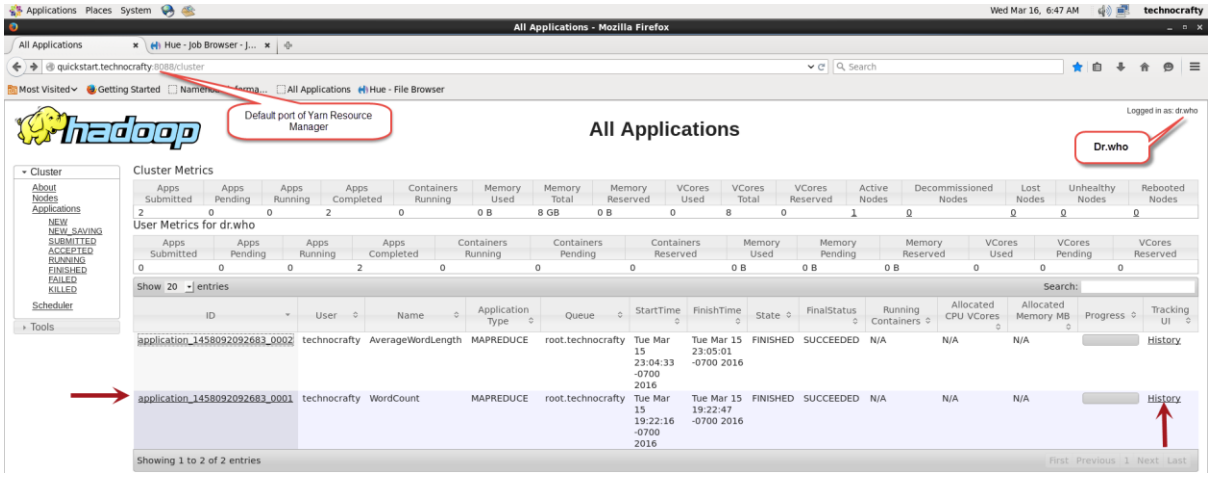
```
        File System Counters
                FILE: Number of bytes read=725062
                FILE: Number of bytes written=1675035
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=1573213
                HDFS: Number of bytes written=527547
                HDFS: Number of read operations=6
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=7395
                Total time spent by all reduces in occupied slots (ms)=8454
                Total time spent by all map tasks (ms)=7395
                Total time spent by all reduce tasks (ms)=8454
                Total vcore-seconds taken by all map tasks=7395
```

```
        Total vcore-seconds taken by all reduce tasks=8454
        Total megabyte-seconds taken by all map tasks=7572480
        Total megabyte-seconds taken by all reduce tasks=8656896
    Map-Reduce Framework
        Map input records=33056
        Map output records=267980
        Map output bytes=2601827
        Map output materialized bytes=725062
        Input split bytes=134
        Combine input records=267980
        Combine output records=50094
        Reduce input groups=50094
        Reduce shuffle bytes=725062
        Reduce input records=50094
        Reduce output records=50094
        Spilled Records=100188
        Shuffled Maps =1
        Failed Shuffles=0
        Merged Map outputs=1
        GC time elapsed (ms)=169
        CPU time spent (ms)=4370
        Physical memory (bytes) snapshot=344633344
        Virtual memory (bytes) snapshot=3008765952
        Total committed heap usage (bytes)=226562048
    Shuffle Errors
        BAD_ID=0
        CONNECTION=0
        IO_ERROR=0
        WRONG_LENGTH=0
        WRONG_MAP=0
        WRONG_REDUCE=0
    File Input Format Counters
        Bytes Read=1573079
    File Output Format Counters
        Bytes Written=527547
```

Check the output file:

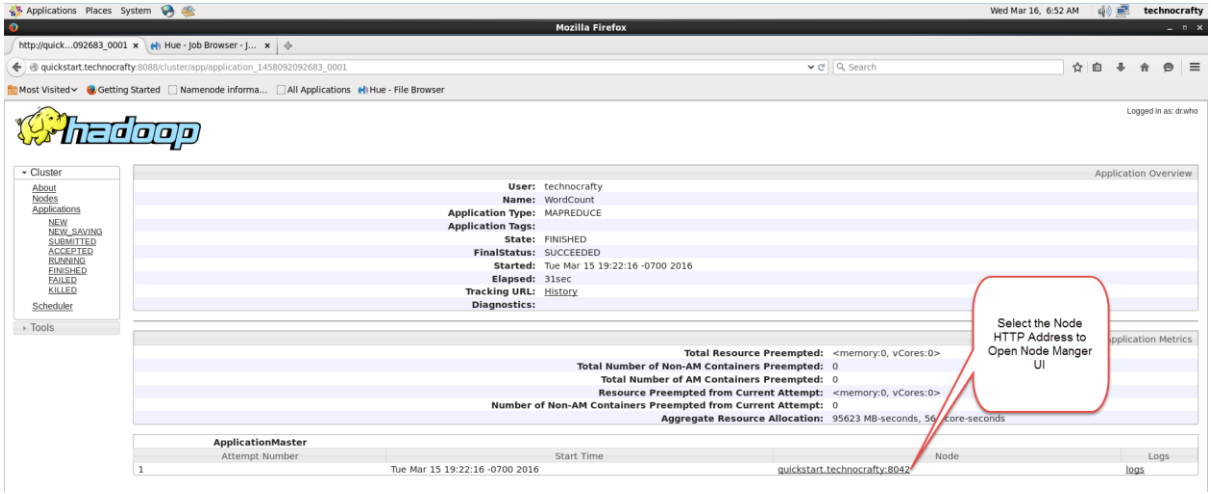
```
[technocrafty@quickstart ~]$ hdfs dfs -ls output
Found 2 items
-rw-r--r--   1 technocrafty technocrafty          0 2016-03-15 19:22 output/_SUCCESS
-rw-r--r--   1 technocrafty technocrafty     527547 2016-03-15 19:22 output/part-r-00000
[technocrafty@quickstart ~]$
```

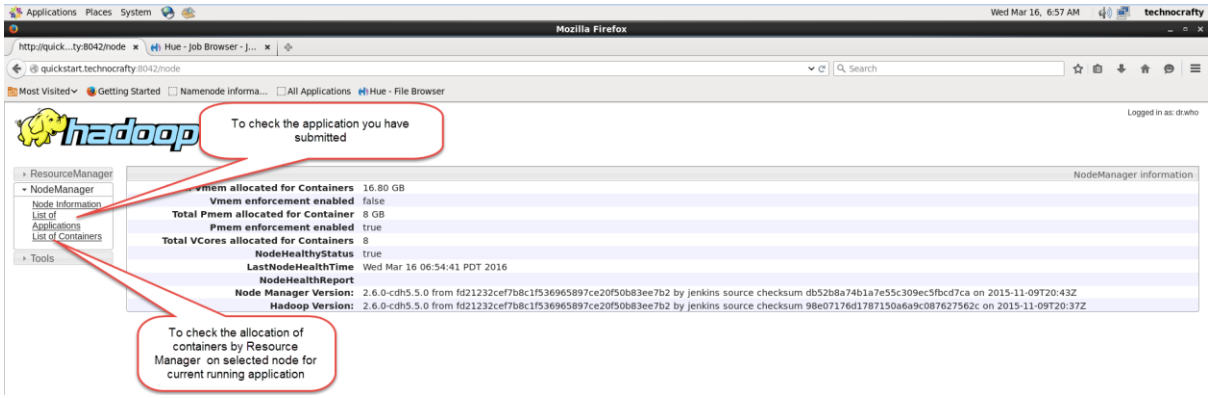
The same counters and job status can be checked from Resource Manager URL
http://quickstart.technocrafty:8088/cluster





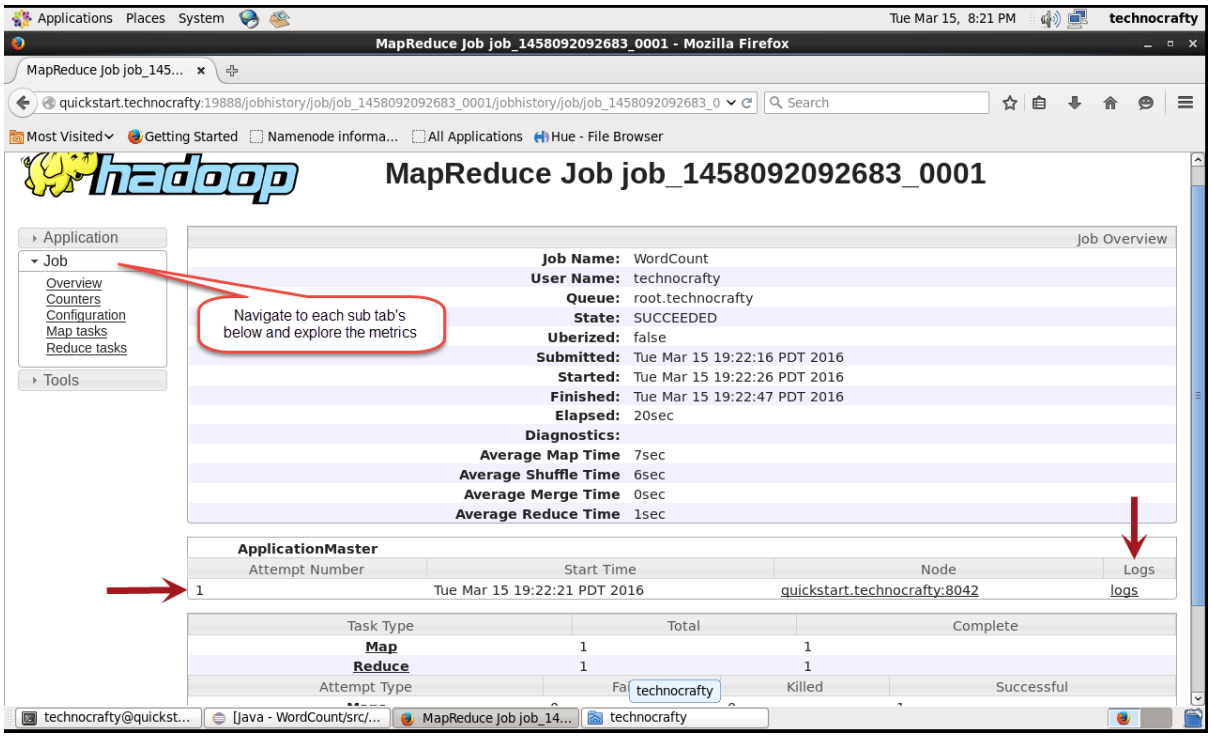
You may notice that YARN says you are logged in as dr.who. This is what is displayed when user authentication is disabled for the cluster. If user authentication were enabled, you would have to log in as a valid user to view the YARN UI, and your actual user name would be displayed, together with user metrics such as how many applications you had run, how much system resources your applications used and so on.







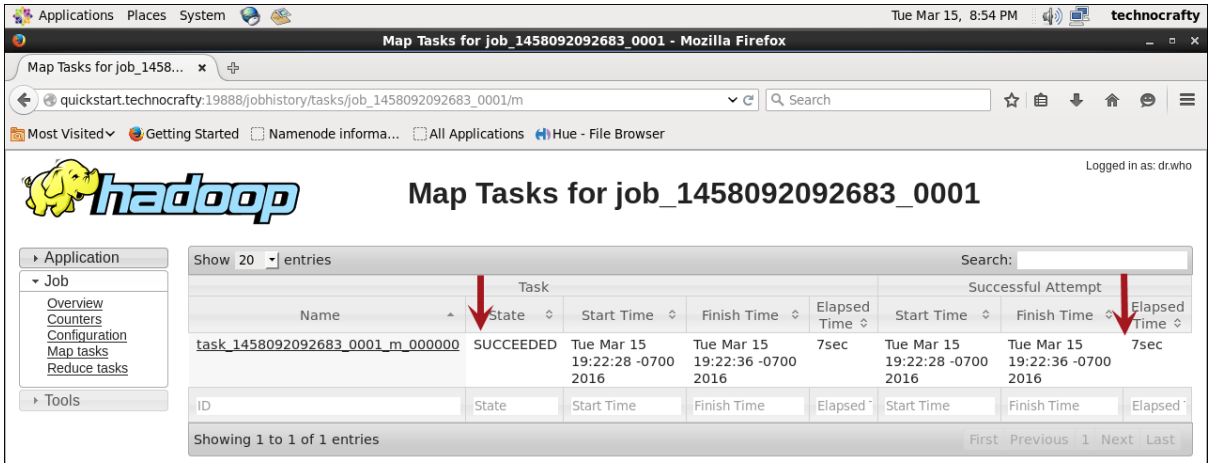
No container information will be available for stopped or completed application. Application must be active or running to view the allocation of containers by Resource Manager.

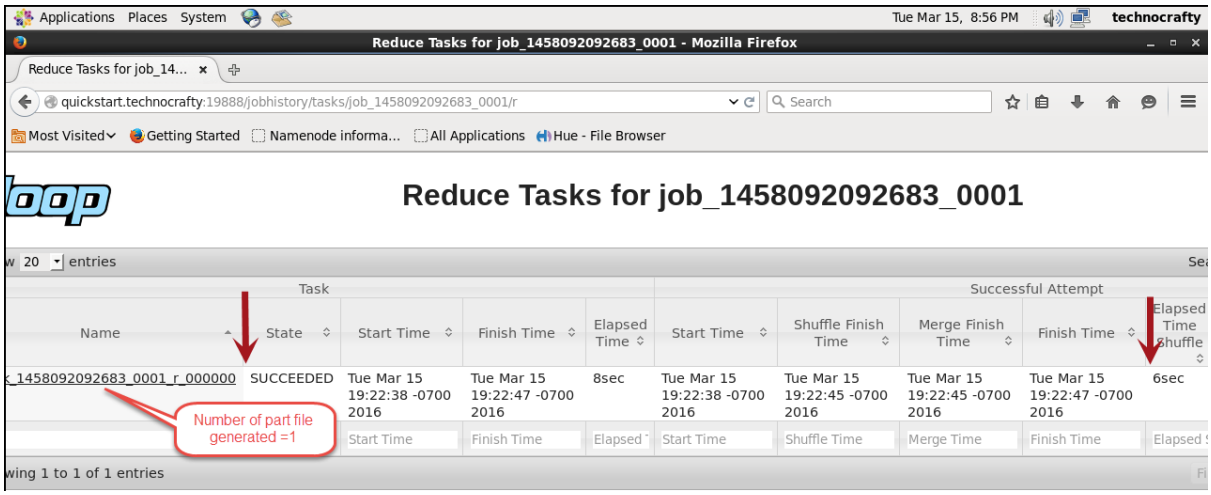


| Counter Group | Counters | | | |
|---|---|---|---|---|
| File System Counters | Name | Map | Reduce | Total |
| | FILE: Number of bytes read | 0 | 725062 | 725062 |
| | FILE: Number of bytes written | 837545 | 837490 | 1675035 |
| | FILE: Number of large read operations | 0 | 0 | 0 |
| | FILE: Number of read operations | 0 | 0 | 0 |
| | FILE: Number of write operations | 0 | 0 | 0 |
| | HDFS: Number of bytes read | 1573213 | 0 | 1573213 |
| | HDFS: Number of bytes written | 0 | 527547 | 527547 |
| | HDFS: Number of large read operations | 0 | 0 | 0 |
| | HDFS: Number of read operations | 3 | 3 | 6 |
| | HDFS: Number of write operations | 0 | 2 | 2 |
| Job Counters | Name | Map | Reduce | Total |
| | Data-local map tasks | 0 | 0 | 1 |
| | Launched map tasks | 0 | 0 | 1 |
| | Launched reduce tasks | 0 | 0 | 1 |
| | Total megabyte-seconds taken by all map tasks | 0 | 0 | 7572480 |
| | Total megabyte-seconds taken by all reduce tasks | 0 | 0 | 8656896 |
| | Total time spent by all map tasks (ms) | 0 | 0 | 7395 |
| | Total time spent by all maps in occupied slots (ms) | 0 | 0 | 7395 |
| | Total time spent by all reduce tasks (ms) | 0 | 0 | 8454 |
| | Total time spent by all reduces in occupied slots (ms) | 0 | 0 | 8454 |
| | Total vcore-seconds taken by all map tasks | 0 | 0 | 7395 |
| | Total vcore-seconds taken by all reduce tasks | 0 | 0 | 8454 |

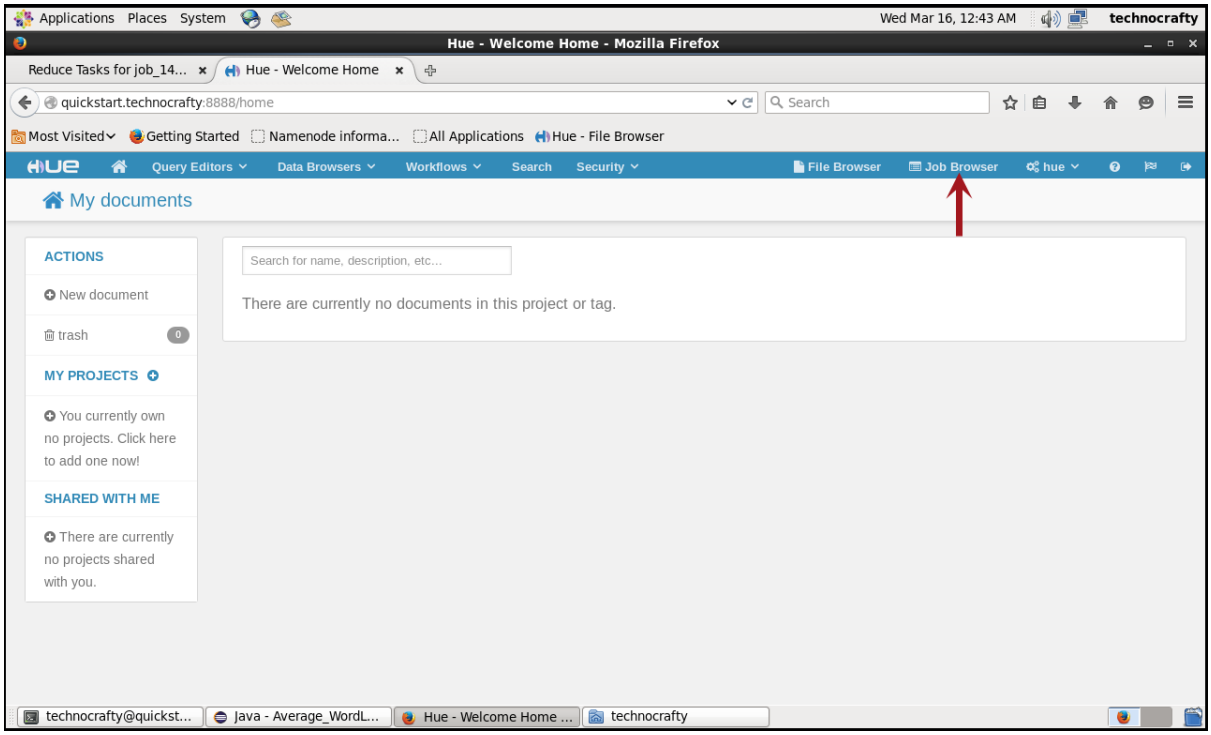
| | Name | Map | Reduce | Total |
|---|---|---|---|---|
| Map-Reduce Framework | Combine input records | 267980 | 0 | 267980 |
| | Combine output records | 50094 | 0 | 50094 |
| | CPU time spent (ms) | 2350 | 2020 | 4370 |
| | Failed Shuffles | 0 | 0 | 0 |
| | GC time elapsed (ms) | 90 | 79 | 169 |
| | Input split bytes | 134 | 0 | 134 |
| | Map input records | 33056 | 0 | 33056 |
| | Map output bytes | 2601827 | 0 | 2601827 |
| | Map output materialized bytes | 725062 | 0 | 725062 |
| | Map output records | 267980 | 0 | 267980 |
| | Merged Map outputs | 0 | 1 | 1 |
| | Physical memory (bytes) snapshot | 223735808 | 120897536 | 344633344 |
| | Reduce input groups | 0 | 50094 | 50094 |
| | Reduce input records | 0 | 50094 | 50094 |
| | Reduce output records | 0 | 50094 | 50094 |
| | Reduce shuffle bytes | 0 | 725062 | 725062 |
| | Shuffled Maps | 0 | 1 | 1 |
| | Spilled Records | 50094 | 50094 | 100188 |
| | Total committed heap usage (bytes) | 165744640 | 60817408 | 226562048 |
| | Virtual memory (bytes) snapshot | 1501155328 | 1507610624 | 3008765952 |

| | Name | Map | Reduce | Total |
|---|---|---|---|---|
| Shuffle Errors | BAD_ID | 0 | 0 | 0 |
| | CONNECTION | 0 | 0 | 0 |
| | IO_ERROR | 0 | 0 | 0 |
| | WRONG_LENGTH | 0 | 0 | 0 |
| | WRONG_MAP | 0 | 0 | 0 |
| | WRONG_REDUCE | 0 | 0 | 0 |
| File Input Format Counters | Name | Map | Reduce | Total |
| | Bytes Read | 1573079 | 0 | 1573079 |
| File Output Format Counters | Name | Map | Reduce | Total |
| | Bytes Written | 0 | 527547 | 527547 |

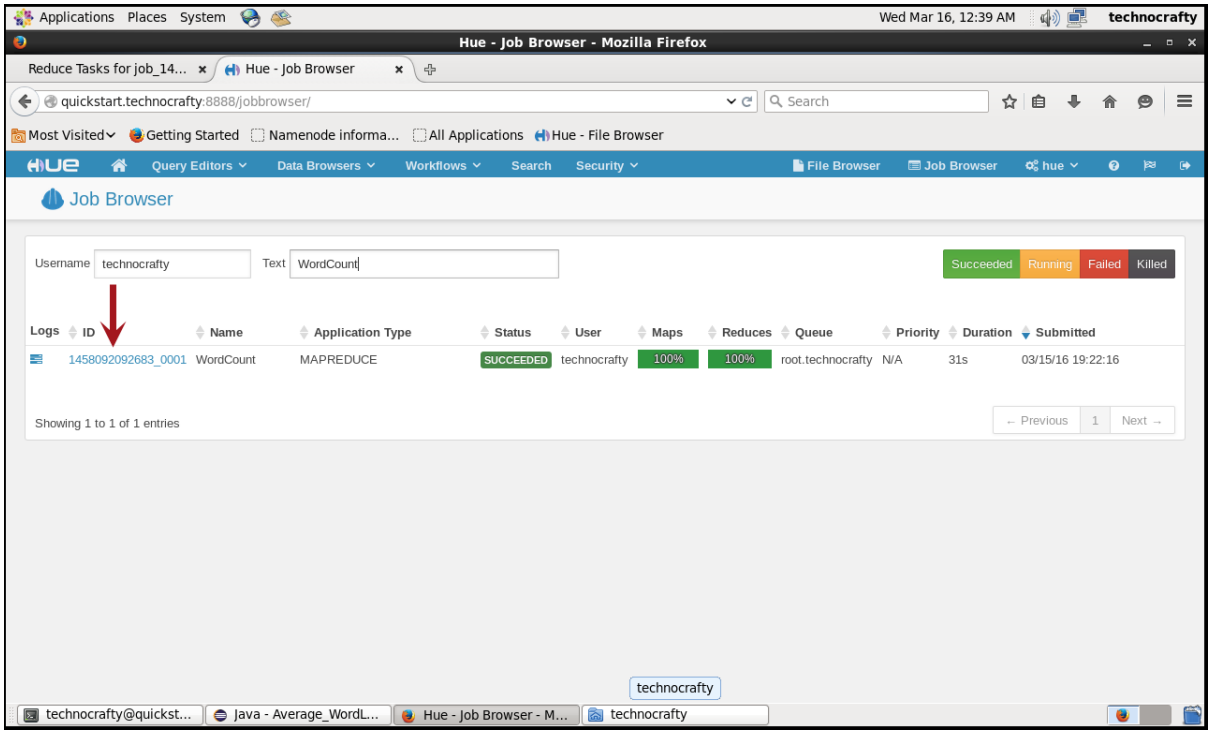


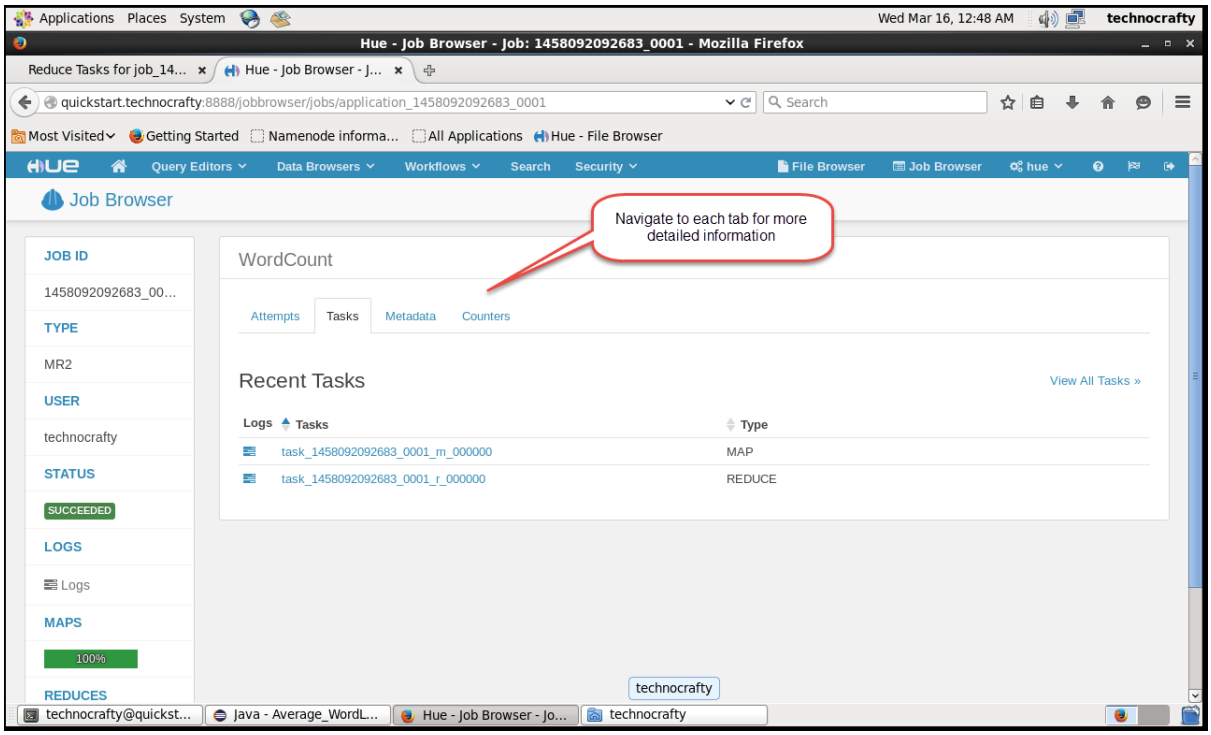




- On HUE browser, navigate to Job Browser



Change the user to technocrafty and you will find job details, click on the Job ID to get the details of counter metrics




- Let's run the same job with multiple reducers

```
[technocrafty@quickstart ~]$ hadoop jar WordCount.jar WordCount -Dmapred.reduce.tasks=4 input/Joyce.txt output_multiple_reducers
```

```
[technocrafty@quickstart ~]$ hdfs dfs -ls output_multiple_reducers
Found 5 items
-rw-r--r--   1 technocrafty technocrafty          0 2016-03-14 08:30 output_multiple_reducers/_SUCCESS
-rw-r--r--   1 technocrafty technocrafty     132044 2016-03-14 08:30 output_multiple_reducers/part-r-00000
-rw-r--r--   1 technocrafty technocrafty     132387 2016-03-14 08:30 output_multiple_reducers/part-r-00001
-rw-r--r--   1 technocrafty technocrafty     131875 2016-03-14 08:30 output_multiple_reducers/part-r-00002
-rw-r--r--   1 technocrafty technocrafty     131241 2016-03-14 08:30 output_multiple_reducers/part-r-00003
```


The number of output files generated is equal to the number of reducers.

Compare the metrics count while running the same Mapreduce program with four reducers with earlier output.

## Follow-Up Assignment:

1. Write a MapReduce program to calculate the Average Wordlength for Joyce.txt dataset.

2. Write a MapReduce program for inverted index (is an index data structure storing a mapping from content, such as words or numbers, to its locations in a database file, or in a document or a set of documents)

   **Hint:** Create multiple files with content and load the files in HDFS input directory, run the program with all the files as input and observe the output.

**Example: Input**

File1 > have fun

File2 > fun full day

**Expected Output:**

day | File2

full | File2

fun | File2 → File1

have | File1

# Exercise 4: Running MapReduce job with Custom Partitioner

- The partitioning phase takes place after the map phase and before the reduce phase.
- The number of partitions is equal to the number of reducers. The data gets partitioned across the reducers according to the partitioning function.
- The default partitioning function is the hash partitioning function where the hashing is done on the key.

However, it might be useful to partition the data according to some other function of the key or the value. Hence let's understand this with one example.

Consider below Employee dataset to perform this task, save this data as empsalary.txt on local filesystem.

(input format will be tab separated data i.e.
Id<tab>Name<tab>Age<tab>Gender<tab>Salary)

| Id | Name | Age | Gender | Salary |
|---|---|---|---|---|
| 1201 | Gopal | 45 | Male | 50,000 |
| 1202 | manisha | 40 | Female | 50,000 |
| 1203 | Khalil | 34 | Male | 30,000 |
| 1204 | prasanth | 30 | Male | 30,000 |
| 1205 | Kiran | 20 | Male | 40,000 |
| 1206 | Laxmi | 25 | Female | 35,000 |
| 1207 | Bhavya | 20 | Female | 15,000 |
| 1208 | Reshma | 19 | Female | 15,000 |

| | | | | |
|---|---|---|---|---|
| 1209 | Kranthi | 22 | Male | 22,000 |
| 1210 | Satish | 24 | Male | 25,000 |
| 1211 | Krishna | 25 | Male | 25,000 |
| 1212 | Arshad | 28 | Male | 20,000 |
| 1213 | Lavanya | 18 | Female | 8,000 |

**Objective**: To write a MapReduce program to process above dataset to find highest salaried employee by gender in different age groups

1. Below 20
2. Between 20 to 30
3. Above 30

Step1: Load the file empsalary.txt present under Datasets directory into HDFS

```
hdfs dfs -put empsalary.txt input
```

Step2: Write MapReduce program without partitioner and create a JAR file.

Step3: Write another program with partitioner logic and run against same dataset, differentiate the output of both the programs.

Running MapReduce job without Partitioner

```
[technocrafty@quickstart ~]$ hadoop jar HighestSalWithout.jar HighestSalWithout  input/empsalary.txt output_empsalary_without
16/03/26 08:03:53 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
16/03/26 08:03:54 INFO input.FileInputFormat: Total input paths to process : 1
16/03/26 08:03:54 INFO mapreduce.JobSubmitter: number of splits:1
16/03/26 08:03:55 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1458825941100_0006
16/03/26 08:03:55 INFO impl.YarnClientImpl: Submitted application application_1458825941100_0006
16/03/26 08:03:56 INFO mapreduce.Job: The url to track the job: http://quickstart.technocrafty:8088/proxy/application_1458825941100_0006/
16/03/26 08:03:56 INFO mapreduce.Job: Running job: job_1458825941100_0006
16/03/26 08:04:07 INFO mapreduce.Job: Job job_1458825941100_0006 running in uber mode : false
16/03/26 08:04:07 INFO mapreduce.Job:  map 0% reduce 0%
16/03/26 08:04:18 INFO mapreduce.Job:  map 100% reduce 0%
16/03/26 08:04:50 INFO mapreduce.Job:  map 100% reduce 67%
16/03/26 08:04:51 INFO mapreduce.Job:  map 100% reduce 100%
16/03/26 08:04:53 INFO mapreduce.Job: Job job_1458825941100_0006 completed successfully
```

Output of MapReduce without Partitioner

```
[technocrafty@quickstart ~]$ hdfs dfs -ls /user/technocrafty/output_empsalary_without
Found 4 items
-rw-r--r--   1 technocrafty technocrafty          0 2016-03-26 08:04 /user/technocrafty/output_empsalary_without/_SUCCESS
-rw-r--r--   1 technocrafty technocrafty         24 2016-03-26 08:04 /user/technocrafty/output_empsalary_without/part-r-00000
-rw-r--r--   1 technocrafty technocrafty          0 2016-03-26 08:04 /user/technocrafty/output_empsalary_without/part-r-00001
-rw-r--r--   1 technocrafty technocrafty          0 2016-03-26 08:04 /user/technocrafty/output_empsalary_without/part-r-00002
[technocrafty@quickstart ~]$ hdfs dfs -cat /user/technocrafty/output_empsalary_without/part-r-00000
Female  51000
Male    50000
[technocrafty@quickstart ~]$ hdfs dfs -cat /user/technocrafty/output_empsalary_without/part-r-00001
[technocrafty@quickstart ~]$ hdfs dfs -cat /user/technocrafty/output_empsalary_without/part-r-00002
[technocrafty@quickstart ~]$
```

## Running MapReduce job with Partitioner

```
[technocrafty@quickstart ~]$ hadoop jar HighestSalary.jar HighestSalary  input/empsalary.txt output_empsalary
16/03/26 07:38:44 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
16/03/26 07:38:45 INFO input.FileInputFormat: Total input paths to process : 1
16/03/26 07:38:46 INFO mapreduce.JobSubmitter: number of splits:1
16/03/26 07:38:46 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1458825941100_0005
16/03/26 07:38:47 INFO impl.YarnClientImpl: Submitted application application_1458825941100_0005
16/03/26 07:38:47 INFO mapreduce.Job: The url to track the job: http://quickstart.technocrafty:8088/proxy/application_1458825941100_0005/
16/03/26 07:38:47 INFO mapreduce.Job: Running job: job_1458825941100_0005
16/03/26 07:38:59 INFO mapreduce.Job: Job job_1458825941100_0005 running in uber mode : false
16/03/26 07:38:59 INFO mapreduce.Job:  map 0% reduce 0%
16/03/26 07:39:07 INFO mapreduce.Job:  map 100% reduce 0%
16/03/26 07:39:44 INFO mapreduce.Job:  map 100% reduce 67%
16/03/26 07:39:45 INFO mapreduce.Job:  map 100% reduce 100%
16/03/26 07:39:50 INFO mapreduce.Job: Job job_1458825941100_0005 completed successfully
```

## Output of MapReduce with Partitioner

```
[technocrafty@quickstart ~]$ hdfs dfs -ls /user/technocrafty/output_empsalary
Found 4 items
-rw-r--r--   1 technocrafty technocrafty          0 2016-03-26 07:39 /user/technocrafty/output_empsalary/_SUCCESS
-rw-r--r--   1 technocrafty technocrafty         24 2016-03-26 07:39 /user/technocrafty/output_empsalary/part-r-00000
-rw-r--r--   1 technocrafty technocrafty         24 2016-03-26 07:39 /user/technocrafty/output_empsalary/part-r-00001
-rw-r--r--   1 technocrafty technocrafty         24 2016-03-26 07:39 /user/technocrafty/output_empsalary/part-r-00002
[technocrafty@quickstart ~]$ hdfs dfs -cat /user/technocrafty/output_empsalary/part-r-00000
Female  15000
Male    40000
[technocrafty@quickstart ~]$ hdfs dfs -cat /user/technocrafty/output_empsalary/part-r-00001
Female  35000
Male    31000
[technocrafty@quickstart ~]$ hdfs dfs -cat /user/technocrafty/output_empsalary/part-r-00002
Female  51000
Male    50000
```

# Exercise 5: Running a MapReduce job with Combiner

**Let's run a WordMean program to check the amount of data passed to reducer in following case**

1. Without Combiner
2. With Combiner

Load the data into HDFS and write a MapReduce program to compare the output in both the cases

**Without Combiner:**

```
[technocrafty@quickstart ~]$ hadoop jar WordMeanWithout.jar WordMeanWithout input/story.txt output_wordmean_without
16/03/27 08:20:03 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
16/03/27 08:20:04 INFO input.FileInputFormat: Total input paths to process : 1
16/03/27 08:20:05 INFO mapreduce.JobSubmitter: number of splits:1
16/03/27 08:20:05 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1459089896960_0003
16/03/27 08:20:05 INFO impl.YarnClientImpl: Submitted application application_1459089896960_0003
16/03/27 08:20:05 INFO mapreduce.Job: The url to track the job: http://quickstart.technocrafty:8088/proxy/application_14590898
96960_0003/
16/03/27 08:20:05 INFO mapreduce.Job: Running job: job_1459089896960_0003
16/03/27 08:20:17 INFO mapreduce.Job: Job job_1459089896960_0003 running in uber mode : false
16/03/27 08:20:17 INFO mapreduce.Job:  map 0% reduce 0%
16/03/27 08:20:40 INFO mapreduce.Job:  map 100% reduce 0%
16/03/27 08:20:53 INFO mapreduce.Job:  map 100% reduce 100%
16/03/27 08:20:53 INFO mapreduce.Job: Job job_1459089896960_0003 completed successfully
```

```
        Map-Reduce Framework
                Map input records=8
                Map output records=304
                Map output bytes=4408
                Map output materialized bytes=5022
                Input split bytes=134
                Combine input records=0
                Combine output records=0
                Reduce input groups=2
                Reduce shuffle bytes=5022
                Reduce input records=304
                Reduce output records=2
                Spilled Records=608
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=156
                CPU time spent (ms)=1750
                Physical memory (bytes) snapshot=337780736
                Virtual memory (bytes) snapshot=3008466944
                Total committed heap usage (bytes)=226562048

        File Input Format Counters
                Bytes Read=796
        File Output Format Counters
                Bytes Written=21
The mean is: 4.223684210526316
[technocrafty@quickstart ~]$
```

```
[technocrafty@quickstart ~]$ hdfs dfs -ls output_wordmean_without
Found 2 items
-rw-r--r--   1 technocrafty technocrafty          0 2016-03-27 08:20 output_wordmean_without/_SUCCESS
-rw-r--r--   1 technocrafty technocrafty         21 2016-03-27 08:20 output_wordmean_without/part-r-00000
[technocrafty@quickstart ~]$ hdfs dfs -cat output_wordmean_without/part-r-00000
count   152
length  642
[technocrafty@quickstart ~]$
```

## With Combiner:

```
[technocrafty@quickstart ~]$ hadoop jar WordMean.jar WordMean input/story.txt output_wordmean
16/03/27 08:01:10 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
16/03/27 08:01:11 INFO input.FileInputFormat: Total input paths to process : 1
16/03/27 08:01:11 INFO mapreduce.JobSubmitter: number of splits:1
16/03/27 08:01:12 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1459089896960_0002
16/03/27 08:01:13 INFO impl.YarnClientImpl: Submitted application application_1459089896960_0002
16/03/27 08:01:13 INFO mapreduce.Job: The url to track the job: http://quickstart.technocrafty:8088/proxy/application_14590898
96960_0002/
16/03/27 08:01:13 INFO mapreduce.Job: Running job: job_1459089896960_0002
16/03/27 08:01:29 INFO mapreduce.Job: Job job_1459089896960_0002 running in uber mode : false
16/03/27 08:01:29 INFO mapreduce.Job:  map 0% reduce 0%
16/03/27 08:01:43 INFO mapreduce.Job:  map 100% reduce 0%
16/03/27 08:01:58 INFO mapreduce.Job:  map 100% reduce 100%
16/03/27 08:01:58 INFO mapreduce.Job: Job job_1459089896960_0002 completed successfully
```

```
        Map-Reduce Framework
                Map input records=8
                Map output records=304
                Map output bytes=4408
                Map output materialized bytes=39
                Input split bytes=134
                Combine input records=304
                Combine output records=2
                Reduce input groups=2
                Reduce shuffle bytes=39
                Reduce input records=2
                Reduce output records=2
                Spilled Records=4
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=206
                CPU time spent (ms)=1910
                Physical memory (bytes) snapshot=353296384
                Virtual memory (bytes) snapshot=3008466944
                Total committed heap usage (bytes)=226562048
```

```
        File Input Format Counters
                Bytes Read=796
        File Output Format Counters
                Bytes Written=21
The mean is: 4.223684210526316
```

```
[technocrafty@quickstart ~]$ hdfs dfs -ls output_wordmean
Found 2 items
-rw-r--r--   1 technocrafty technocrafty          0 2016-03-27 08:01 output_wordmean/_SUCCESS
-rw-r--r--   1 technocrafty technocrafty         21 2016-03-27 08:01 output_wordmean/part-r-00000
[technocrafty@quickstart ~]$ hdfs dfs -cat output_wordmean/part-r-00000
count   152
length  642
[technocrafty@quickstart ~]$
```

# Exercise 6: Sqoop

1. **Importing table from MySQL to HDFS**

**Step1**: Review the database and tables present in MySQL

- Connect to MySQL with user technocrafty and password technocrafty

```
[technocrafty@quickstart ~]$ mysql -u technocrafty -p
```

```
[technocrafty@quickstart ~]$ mysql -u technocrafty -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 26
Server version: 5.1.66 Source distribution

Copyright (c) 2000, 2012, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

- Follow below command to view database list, changing the database and to list the tables.

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| retail_db          |
+--------------------+
2 rows in set (0.00 sec)

mysql> use retail_db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+---------------------+
| Tables_in_retail_db |
+---------------------+
| categories          |
| customers           |
| departments         |
| order_items         |
| orders              |
| products            |
+---------------------+
6 rows in set (0.00 sec)

mysql>
```

**Step2**: Before importing the data, review the structure of tables.

For example, let's take departments table

```
mysql> Describe departments;
+-----------------+-------------+------+-----+---------+----------------+
| Field           | Type        | Null | Key | Default | Extra          |
+-----------------+-------------+------+-----+---------+----------------+
| department_id   | int(11)     | NO   | PRI | NULL    | auto_increment |
| department_name | varchar(45) | NO   |     | NULL    |                |
+-----------------+-------------+------+-----+---------+----------------+
2 rows in set (0.00 sec)
```

**Step3**: To know the content of departments table

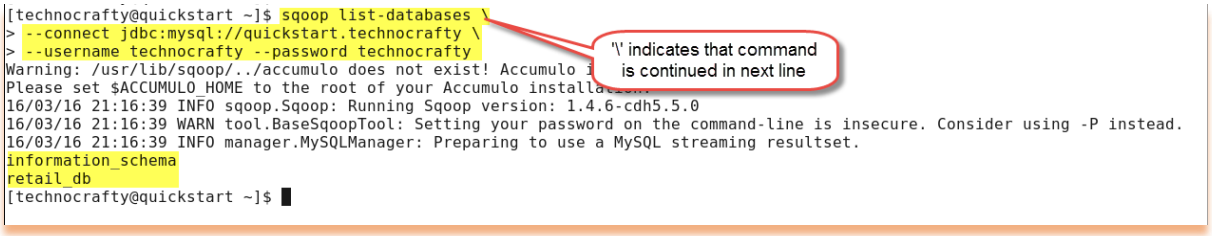
```
mysql> select * from departments;
+---------------+-----------------+
| department_id | department_name |
+---------------+-----------------+
|             2 | Fitness         |
|             3 | Footwear        |
|             4 | Apparel         |
|             5 | Golf            |
|             6 | Outdoors        |
|             7 | Fan Shop        |
+---------------+-----------------+
6 rows in set (0.00 sec)

mysql>
```

- To know various options run sqoop help on the terminal prompt.

```
[technocrafty@quickstart ~]$ sqoop help
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
16/03/16 08:51:13 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.5.0
usage: sqoop COMMAND [ARGS]

Available commands:
  codegen            Generate code to interact with database records
  create-hive-table  Import a table definition into Hive
  eval               Evaluate a SQL statement and display the results
  export             Export an HDFS directory to a database table
  help               List available commands
  import             Import a table from a database to HDFS
  import-all-tables  Import tables from a database to HDFS
  import-mainframe   Import datasets from a mainframe server to HDFS
  job                Work with saved jobs
  list-databases     List available databases on a server
  list-tables        List available tables in a database
  merge              Merge results of incremental imports
  metastore          Run a standalone Sqoop metastore
  version            Display version information

See 'sqoop help COMMAND' for information on a specific command.
```

- To list the databases available in your database server

```
$ sqoop list-databases --connect jdbc:mysql://quickstart.technocrafty --username technocrafty --password technocrafty
```

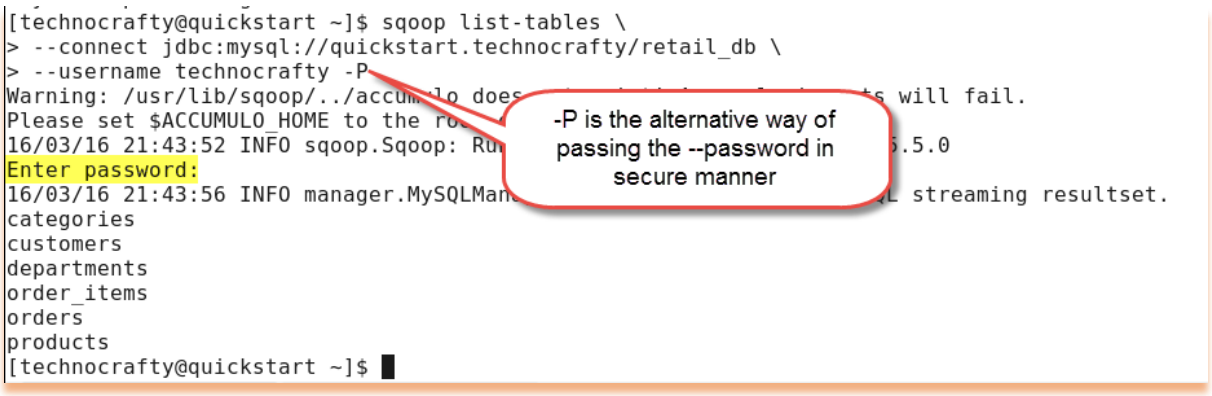




To complete the command or query in single line, remove the slash from the end '\'

- To list the tables in retail_db database

```
$ sqoop list-tables --connect jdbc:mysql://quickstart.technocrafty/retail_db --username technocrafty --password technocrafty
```



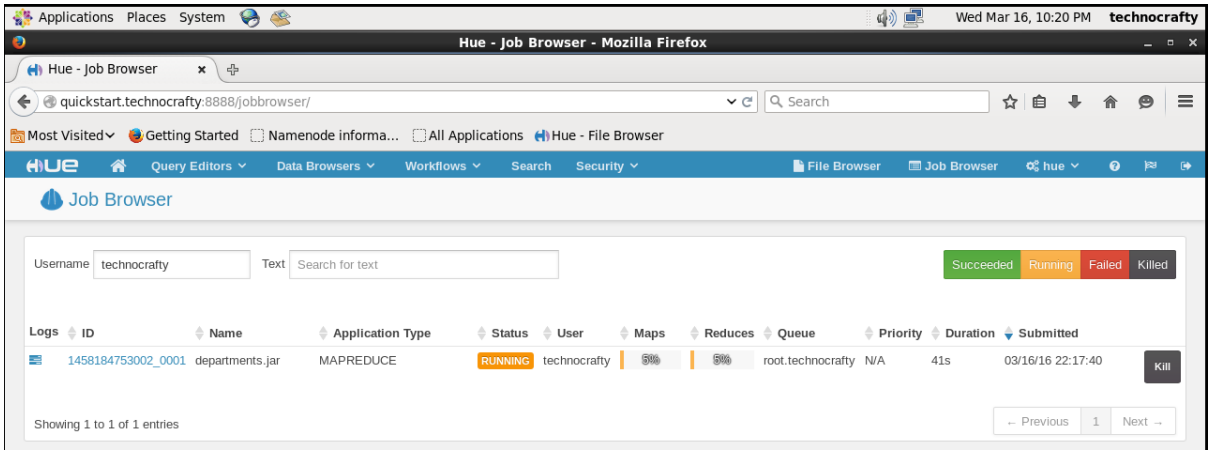

- Run **sqoop import -- help** to know various options
- Import 'departments' table from 'retail_db' database to HDFS

```
[technocrafty@quickstart ~]$ sqoop import --connect jdbc:mysql://quickstart.technocrafty/retail_db --username technocrafty --
```

password technocrafty --table deaprtments --target-dir /user/technocrafty/dept_table --fields-terminated-by '\t'

```
[technocrafty@quickstart ~]$ sqoop import \
> --connect jdbc:mysql://quickstart.technocrafty/retail_db \
> --username technocrafty --password technocrafty \
> --table departments \
> --target-dir /user/technocrafty/dept_table \
> --fields-terminated-by '\t'
```

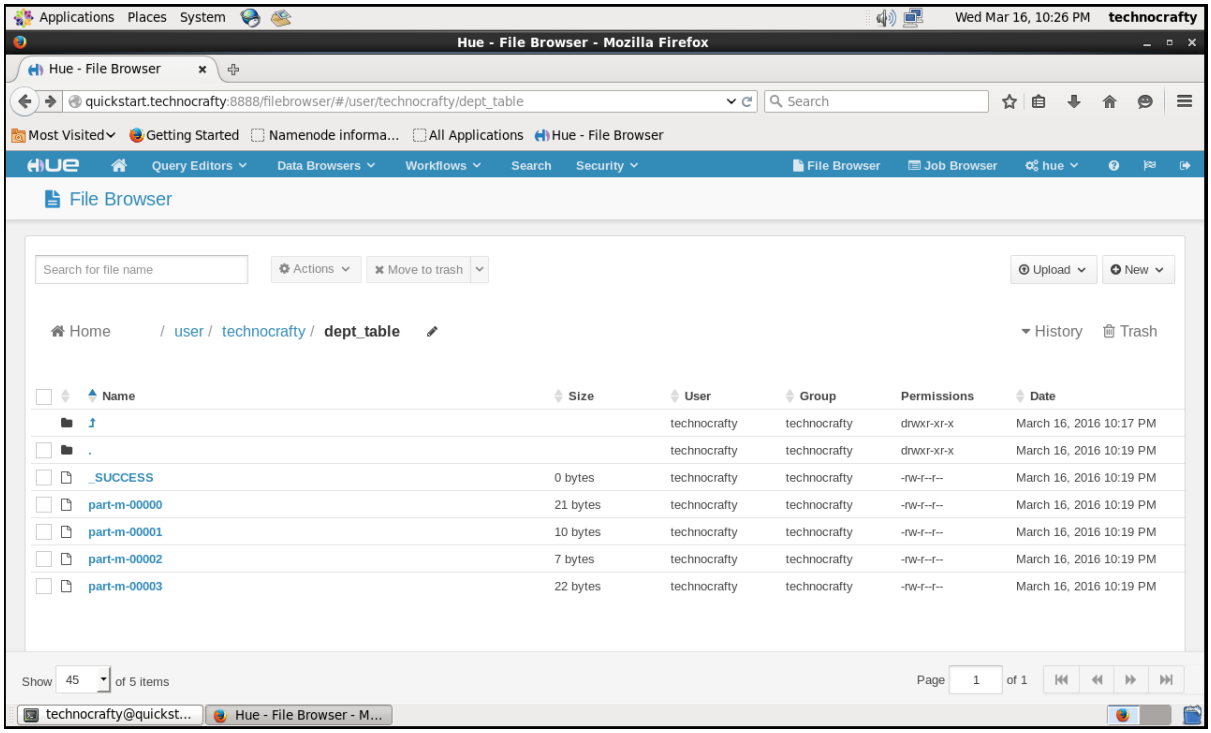
Status of above MapReduce job can be checked in HUE Job Browser.




Verify the output at the target directory in HDFS

```
[technocrafty@quickstart ~]$ hdfs dfs -ls dept_table
Found 5 items
-rw-r--r--   1 technocrafty technocrafty          0 2016-03-16 22:19 dept_table/_SUCCESS
-rw-r--r--   1 technocrafty technocrafty         21 2016-03-16 22:19 dept_table/part-m-00000
-rw-r--r--   1 technocrafty technocrafty         10 2016-03-16 22:19 dept_table/part-m-00001
-rw-r--r--   1 technocrafty technocrafty          7 2016-03-16 22:19 dept_table/part-m-00002
-rw-r--r--   1 technocrafty technocrafty         22 2016-03-16 22:19 dept_table/part-m-00003
```

View the content of part files on HUE File Browser




Alternatively, you can also verify from command line terminal

```
[technocrafty@quickstart ~]$ hdfs dfs -cat dept_table/part-m-00000
2       Fitness
3       Footwear
[technocrafty@quickstart ~]$ hdfs dfs -cat dept_table/part-m-00001
4       Apparel
[technocrafty@quickstart ~]$ hdfs dfs -cat dept_table/part-m-00002
5       Golf
[technocrafty@quickstart ~]$ hdfs dfs -cat dept_table/part-m-00003
6       Outdoors
7       Fan Shop
```

## 2. Sqoop Import using Compression

Let us now import another table say 'orders' using compression

```
$ sqoop import --connect jdbc:mysql://quickstart.technocrafty/retail_db --username technocrafty --password technocrafty --compression-codec=snappy --target-dir=/user/technocrafty/orders_table --table orders
```

Verify the files at target directory location.

```
[technocrafty@quickstart ~]$ hdfs dfs -ls orders_table
Found 5 items
-rw-r--r--   1 technocrafty technocrafty          0 2016-03-16 23:03 orders_table/_SUCCESS
-rw-r--r--   1 technocrafty technocrafty     218873 2016-03-16 23:03 orders_table/part-m-00000.snappy
-rw-r--r--   1 technocrafty technocrafty     218610 2016-03-16 23:03 orders_table/part-m-00001.snappy
-rw-r--r--   1 technocrafty technocrafty     220683 2016-03-16 23:03 orders_table/part-m-00002.snappy
-rw-r--r--   1 technocrafty technocrafty     233412 2016-03-16 23:03 orders_table/part-m-00003.snappy
[technocrafty@quickstart ~]$
```

## Follow Up Assignment:

1. Import customers table and change the delimiter to comma.
2. Import products table with parameter --target-dir
3. Import products table with parameter --warehouse-dir


The difference between --target-dir and --warehouse-dir is that, in case of --target-dir if the directory is already present it will throw an error, whereas in case of --warehouse-dir it will treat the existing directory as parent directory and create a sub-directory with table name.

4. Using Sqoop, Export data from HDFS to MySQL.

   a. Create 'store' table with ID and name field in 'retail_db' database.
   b. Create a file in HDFS with 'store' information
   c. Perform the export using sqoop from HDFS to MySQL

5. Add another entry in 'store' table and append the same to HDFS using incremental import.

**Hint**: Incremental import should have additionally below parameters

--incremental append
--check-column id
--last-value

6. Replace the NULL string in the table structure with \\N


The --null-non-string option tells Sqoop to represent null values as \N, which makes the imported data compatible with Hive and Impala.

# Exercise 7: Hive

### Data Processing with Hive

We will use the dataset for this exercise from previous imported data using sqoop i.e. departments data.

- Ensure that you have the data available in the HDFS before proceeding

```
[technocrafty@quickstart ~]$ hdfs dfs -ls dept_table
Found 5 items
-rw-r--r--   1 technocrafty technocrafty          0 2016-03-16 22:19 dept_table/_SUCCESS
-rw-r--r--   1 technocrafty technocrafty         21 2016-03-16 22:19 dept_table/part-m-00000
-rw-r--r--   1 technocrafty technocrafty         10 2016-03-16 22:19 dept_table/part-m-00001
-rw-r--r--   1 technocrafty technocrafty          7 2016-03-16 22:19 dept_table/part-m-00002
-rw-r--r--   1 technocrafty technocrafty         22 2016-03-16 22:19 dept_table/part-m-00003
```

- Prepare the data for Hive by creating external table on departments dataset, without making any changes to data.

**Step1**: Invoke the HIVE shell

```
[technocrafty@quickstart ~]$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.properties
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.
hive>
```

**Step2**: Create an External table name department

```
CREATE EXTERNAL TABLE department
(department_id INT, department_name string)
 ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
LOCATION '/user/technocrafty/dept_table';
```

```
hive> CREATE EXTERNAL TABLE department (department_id INT, department_name string) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' LOCATION '/user/technocrafty/dept_table';
OK
Time taken: 4.245 seconds
```

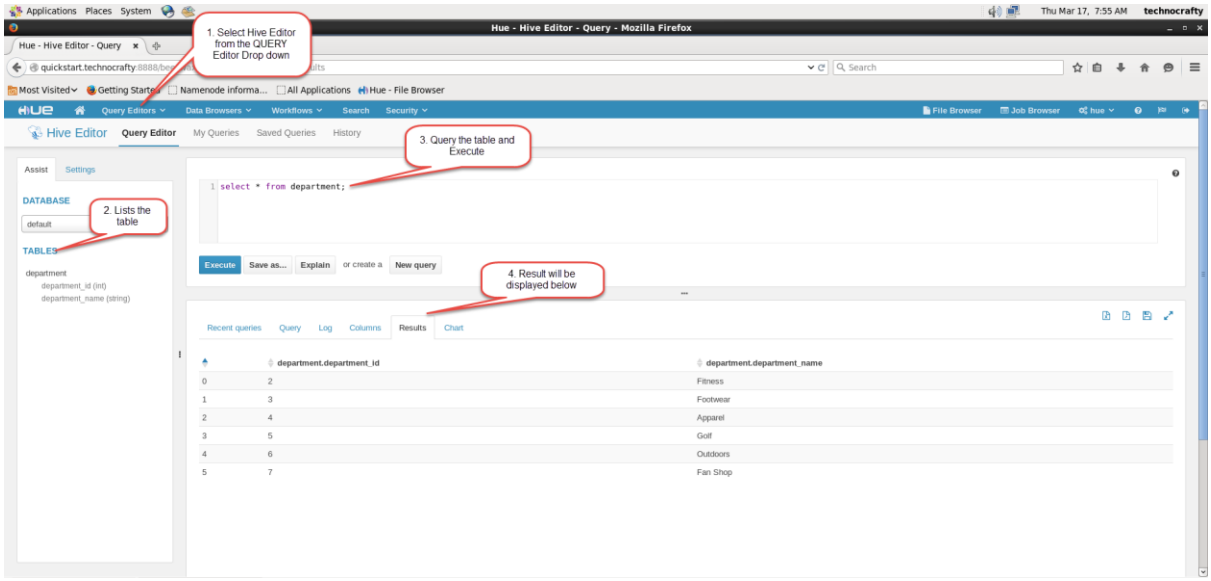
**Step3**: List and view the structure of department table from the command line

```
hive> show tables;
OK
department
Time taken: 0.814 seconds, Fetched: 1 row(s)
hive> describe department;
OK
department_id           int
department_name         string
Time taken: 0.437 seconds, Fetched: 2 row(s)
```

**Step4**: Query the table from command line and quit the hive shell

```
hive> select * from department;
OK
2       Fitness
3       Footwear
4       Apparel
5       Golf
6       Outdoors
7       Fan Shop
Time taken: 0.678 seconds, Fetched: 6 row(s)
hive> quit
```

**Step5**: All the above operation can be performed from HUE browser > Query Editor > Hive Editor



## Hive Partitioning

**Step1:** Let's take employee data (emp.txt) present under Datasets directory on local filesystem and load the same into HDFS

- Create a directory named employee and move emp.txt file to HDFS.

**Step2:** Objective is to partition this dataset into three department namely A,B,C

**Step3:** Create an external table to store this data

```
create external table employee (EmployeeID Int,FirstName string,Designation string,Salary Int,Department string) row format delimited fields terminated by "," location '/user/technocrafty/employee';
```

```
hive> select * from employee;
OK
1       Anne    Admin   50000   A
2       Gokul   Admin   50000   B
3       Janet   Sales   60000   A
4       Hari    Admin   50000   C
5       Sanker  Admin   50000   C
6       Margaret        Tech    12000   A
7       Nirmal  Tech    12000   B
8       jinju   Engineer        45000   B
9       Nancy   Admin   50000   A
10      Andrew  Manager 40000   A
11      Arun    Manager 40000   B
12      Harish  Sales   60000   B
13      Robert  Manager 40000   A
14      Laura   Engineer        45000   A
15      Anju    Ceo     100000  B
16      Aarathi Manager 40000   B
17      Parvathy        Engineer        45000   B
18      Gopika  Admin   50000   B
19      Steven  Engineer        45000   A
20      Michael Ceo     100000  A
Time taken: 0.478 seconds, Fetched: 20 row(s)
```

**Step4:** Create another hive table with partition on Department

```
create table employee_part (EmployeeID Int,FirstName String,Designation String,Salary Int) PARTITIONED BY (Department String) row format delimited fields terminated by ",";
```

**Step5:** Insert data into Partitioned table 'employee_part' by using the data from employee table.

- **Static Partition**

Partition the data of department A by using the select command

```
> insert into table employee_part PARTITION(department='A')
> SELECT EmployeeID, FirstName,Designation,Salary FROM employee where department='A';
```

```
hive> insert into table employee_part PARTITION(department='A')
    > SELECT EmployeeID, FirstName,Designation,Salary FROM employee where department='A';
Query ID = technocrafty_20160317094848_0771101c-e535-41ec-a55b-c0bca85cc62c
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1458184753002_0007, Tracking URL = http://quickstart.technocrafty:8088/proxy/application_1458184753002_0007/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1458184753002_0007
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2016-03-17 09:48:56,051 Stage-1 map = 0%,  reduce = 0%
2016-03-17 09:49:06,307 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.06 sec
MapReduce Total cumulative CPU time: 2 seconds 60 msec
Ended Job = job_1458184753002_0007
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to: hdfs://quickstart.technocrafty:8020/user/hive/warehouse/employee_part/department=A/.hive-staging_hive_2016-03-17_09-48-37_090_859438638290764996-1/-ext-10000
Loading data to table default.employee_part partition (department=A)
Partition default.employee_part{department=A} stats: [numFiles=1, numRows=9, totalSize=200, rawDataSize=191]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 2.06 sec   HDFS Read: 4687 HDFS Write: 291 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 60 msec
OK
Time taken: 32.623 seconds
```

```
hive> select * from employee_part;
OK
1       Anne    Admin   50000   A
3       Janet   Sales   60000   A
6       Margaret        Tech    12000   A
9       Nancy   Admin   50000   A
10      Andrew  Manager 40000   A
13      Robert  Manager 40000   A
14      Laura   Engineer        45000   A
19      Steven  Engineer        45000   A
20      Michael Ceo     100000  A
Time taken: 0.088 seconds, Fetched: 9 row(s)
```

Similarly, the data can be partitioned for department B and C in same manner.

## ➢ Dynamic Partition

You need to enable below parameters to run dynamic partitioning

1. set hive.exec.dynamic.partition=true
   This enable dynamic partitions, by default it is false.
2. set hive.exec.dynamic.partition.mode=nonstrict
   We are using the dynamic partition without a static partition (A table can be partitioned based on multiple columns in hive) in such case we have to enable the non-strict mode. In strict mode we can use dynamic partition only with a Static Partition.
3. set hive.exec.max.dynamic.partitions.pernode=3
   The default value is 100, we have to modify the same according to the possible no of partitions

```
hive> set hive.exec.dynamic.partition=true
    > ;
hive> set hive.exec.dynamic.partition.mode=nonstrict;
hive> set hive.exec.max.dynamic.partitions.pernode=3
    > ;
```

```
INSERT OVERWRITE TABLE employee_part PARTITION(department) SELECT EmployeeID, FirstName,Designation,Salary,department FROM employee;
```

```
hive> INSERT OVERWRITE TABLE employee_part PARTITION(department) SELECT EmployeeID, FirstName,Designation,Salary,department FROM employee;
Query ID = technocrafty_20160317095959_bffff790-49ce-4232-853b-d7aaba5f9f81
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1458184753002_0008, Tracking URL = http://quickstart.technocrafty:8088/proxy/application_1458184753002_0008/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1458184753002_0008
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2016-03-17 09:59:56,135 Stage-1 map = 0%,  reduce = 0%
2016-03-17 10:00:06,441 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.66 sec
MapReduce Total cumulative CPU time: 1 seconds 660 msec
Ended Job = job_1458184753002_0008
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to: hdfs://quickstart.technocrafty:8020/user/hive/warehouse/employee_part/.hive-staging_hive_2016-03-17_09-59-43_548_6454283354063553051-1/-ext-10000
Loading data to table default.employee_part partition (department=null)
         Time taken for load dynamic partitions : 904
        Loading partition {department=A}
        Loading partition {department=C}
        Loading partition {department=B}
         Time taken for adding to write entity : 5
Partition default.employee_part{department=A} stats: [numFiles=1, numRows=9, totalSize=200, rawDataSize=191]
Partition default.employee_part{department=B} stats: [numFiles=1, numRows=9, totalSize=200, rawDataSize=191]
Partition default.employee_part{department=C} stats: [numFiles=1, numRows=2, totalSize=40, rawDataSize=38]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 1.66 sec   HDFS Read: 4636 HDFS Write: 638 SUCCESS
Total MapReduce CPU Time Spent: 1 seconds 660 msec
OK
Time taken: 26.218 seconds
```

```
hive> select * from employee_part;
OK
1       Anne    Admin   50000   A
3       Janet   Sales   60000   A
6       Margaret        Tech    12000   A
9       Nancy   Admin   50000   A
10      Andrew  Manager 40000   A
13      Robert  Manager 40000   A
14      Laura   Engineer        45000   A
19      Steven  Engineer        45000   A
20      Michael Ceo     100000  A
2       Gokul   Admin   50000   B
7       Nirmal  Tech    12000   B
8       jinju   Engineer        45000   B
11      Arun    Manager 40000   B
12      Harish  Sales   60000   B
15      Anju    Ceo     100000  B
16      Aarathi Manager 40000   B
17      Parvathy        Engineer        45000   B
18      Gopika  Admin   50000   B
4       Hari    Admin   50000   C
5       Sanker  Admin   50000   C
Time taken: 0.084 seconds, Fetched: 20 row(s)
```

## Follow-Up Assignment

1. **Analyse batting datasets using Hive**

**Source:** http://seanlahman.com/files/database/lahman-csv_2015-01-24.zip

Download and unzip the datasets, basically extract only batting.csv dataset.

Perform following action to get the desired output:

Step1: Create a table to hold the data

Step2: Load the data file batting.csv into the newly created table

Step3: Ensure that data is properly copied over by querying the table

Step4: Create another external table with fields player id, year and runs

Step5: Insert data into the external table from original table.

Step6: Write a SQL to obtain the highest score for each year.

Step7: Join the resultant of previous output to get the player id who scored highest runs.

2. **Direct import from Sqoop to Hive table**

Import all the tables present under MySQL database 'retail_db' into HDFS by fulfilling below criteria

- Enable compression
- Fetch the files in Parquet format
- Directly create table to represent these HDFS files in Hive Metastore with matching schemas

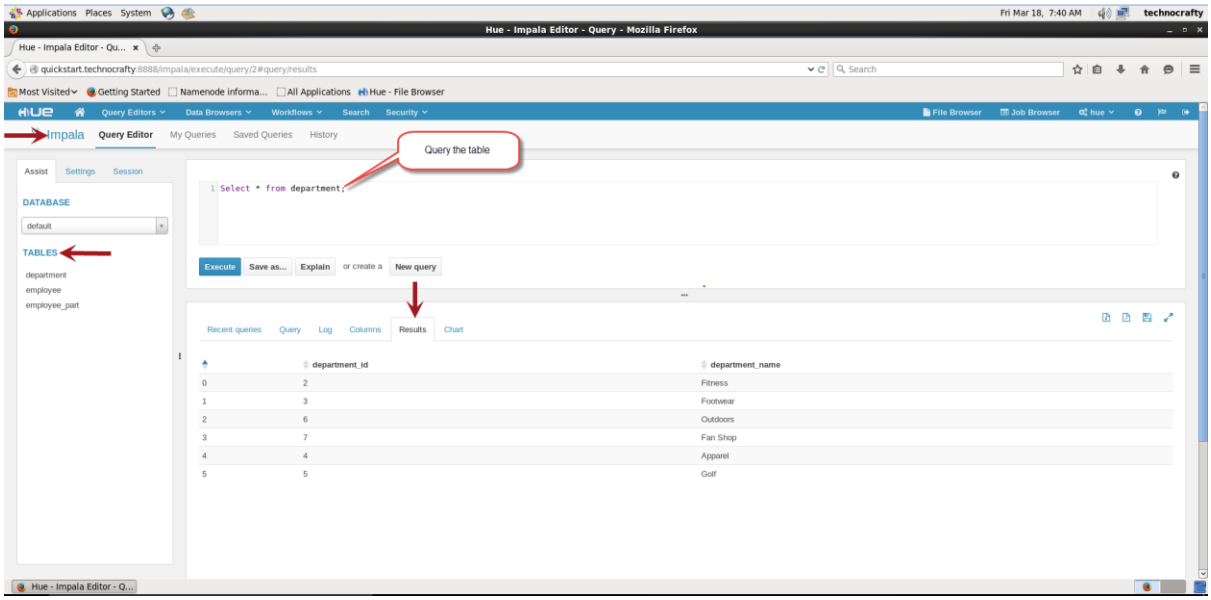
3. **Hive Partitioning**

Use the same batting dataset and create static and dynamic partition to group the runs of player by a particular year of your choice.

# Exercise 8: Impala

**In this section you will query the imported data from previous exercise using Impala**

Go to Hue Web page > Query Editor > Impala




Note

All the tables created while performing Hive exercise will be visible in Impala Editor also.

# Exercise9: Pig

Let's understand data processing with Pig

**Step1**: Pick the dataset for this exercise from local filesystem under Datasets directory. Find the file named BX-Book.txt

```
[technocrafty@quickstart Downloads]$ cd /home/technocrafty/Datasets
[technocrafty@quickstart Datasets]$ ls -ltr
total 73188
-rw-rw-r-- 1 technocrafty technocrafty  1573079 Dec  9  2014 Joyce.txt
-rw-rw-r-- 1 technocrafty technocrafty      480 Mar 17 09:20 emp.txt
-rw-rw-r-- 1 technocrafty technocrafty 73360118 Mar 18 09:03 BX-Books.txt
[technocrafty@quickstart Datasets]$
```

**Step2**: Import the data into HDFS

```
$ hdfs dfs -put BX-Books.txt input/BX-Books.txt
```

```
[technocrafty@quickstart Downloads]$ hdfs dfs -put BX-Books.txt input/BX-Books.txt
[technocrafty@quickstart Downloads]$ hdfs dfs -ls input
Found 3 items
-rw-r--r--   1 technocrafty technocrafty   73360118 2016-03-18 09:05 input/BX-Books.txt
-rw-r--r--   1 technocrafty technocrafty    1573079 2016-03-15 19:17 input/Joyce.txt
-rw-r--r--   1 technocrafty technocrafty         80 2016-03-15 23:03 input/sample.txt
[technocrafty@quickstart Downloads]$
```

**Step3**: Start the Pig console

Invoke GRUNT shell by typing pig on the console

Verify the file BX-Books.txt, should be visible in HDFS

```
cd input

ls BX-Books.txt
```

```
grunt> cd input
grunt> ls BX-Books.txt
hdfs://quickstart.technocrafty:8020/user/technocrafty/input/BX-Books.txt<r 1>   73360118
grunt>
```

**Step4**: Load the data into Pig collection

```
books = LOAD '/user/technocrafty/input/BX-Books.txt'
>> USING PigStorage(';') AS (ISBN:chararray, BookTitle:chararray,
BookAuthor:chararray, YearOfPublication:int, Publisher:chararray);
```

```
grunt> cd input
grunt> books = LOAD '/user/technocrafty/input/BX-Books.txt'
>> USING PigStorage(';') AS (ISBN:chararray, BookTitle:chararray,BookAuthor:chararray, YearOfPublication:int,Publisher:chararray);
```

**Step5**: Verify the loaded data structure

```
DESCRIBE books;
```

```
grunt> DESCRIBE books;
books: {ISBN: chararray,BookTitle: chararray,BookAuthor: chararray,YearOfPublication: int,Publisher: chararray}
grunt>
```

**Step6**: Find the number of books written by year

```
groupByYear = GROUP books BY YearOfPublication;
DESCRIBE groupByYear;
```

```
grunt> groupByYear = GROUP books BY YearOfPublication;
grunt> DESCRIBE groupByYear;
groupByYear: {group: int,books: {(ISBN: chararray,BookTitle: chararray,BookAuthor: chararray,YearOfPublication: int,Publisher: chararray)}}
grunt>
```

groupByYear contains the data with all unique year values, along with the list of books that belongs to that year

**Step7**: Use FOREACH to generate the book count by year

```
countByYear = FOREACH groupByYear
>> GENERATE group AS YearOfPublication, COUNT($1) AS BookCount;
DESCRIBE countByYear;
```

```
grunt> countByYear = FOREACH groupByYear
>> GENERATE group AS YearOfPublication, COUNT($1) AS BookCount;
grunt> DESCRIBE countByYear;
countByYear: {YearOfPublication: int,BookCount: long}
grunt>
```

countByYear set calculates the count of books over groupByYear data set

**Step8**: Store the result back into HDFS using DUMP command

```
DUMP countByYear;
```

```
grunt> DUMP countByYear;
2016-03-18 09:25:10,860 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: GROUP_BY
2016-03-18 09:25:10,975 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimizer - {RULES_ENABLED=[AddForEach, ColumnMapKeyPrune, DuplicateForEachColumnRewrite, GroupByConstParallelSetter, ImplicitSplitInserter, LimitOptimizer, LoadTypeCastInserter, MergeFilter, MergeForEach, NewPartitionFilterOptimizer, PushDownForEachFlatten, PushUpFilter, SplitFilter, StreamTypeCastInserter], RULES_DISABLED=[FilterLogicExpressionSimplifier, PartitionFilterOptimizer]}
```

This will start MapReduce task

```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.6.0-cdh5.5.0  0.12.0-cdh5.5.0 technocrafty    2016-03-18 09:25:12     2016-03-18 09:26:13     GROUP_BY

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTIme      AvgMapTime      MedianMapTime   MaxReduceTime   MinReduceTime   AvgReduceTime   MedianReducetime        Alias   Feature Outputs
job_1458309384404_0001  1       1       15      15      15      15      13      13      13      13      books,countByYear,groupByYear   GROUP_BY,COMBINER       hdfs://quickstart.technocrafty
:8020/tmp/temp-1363236575/tmp-1304444013,

Input(s):
Successfully read 271379 records (73360509 bytes) from: "/user/technocrafty/input/BX-Books.txt"

Output(s):
Successfully stored 116 records (1068 bytes) in: "hdfs://quickstart.technocrafty:8020/tmp/temp-1363236575/tmp-1304444013"

Counters:
Total records written : 116
Total bytes written : 1068
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_1458309384404_0001


2016-03-18 09:26:13,792 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
2016-03-18 09:26:13,821 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
2016-03-18 09:26:13,821 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2016-03-18 09:26:13,821 [main] INFO  org.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple] was not set... will not generate code.
2016-03-18 09:26:13,845 [main] INFO  org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2016-03-18 09:26:13,845 [main] INFO  org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(0,4619)
(1376,1)
(1378,1)
(1806,1)
(1897,1)
(1900,3)
(1901,7)
```

You will observe listing of years along with the number of books for that year, also you may find redundant entry or blank year which we will clear in next steps.

**Step9**: Create a set of all authors and corresponding years in which they wrote books

Let's first clean-up the data and have only positive year of publication

```
books = FILTER books BY YearOfPublication > 0;
```

Now create a set of authors and all the year they wrote books in

```
pivot = FOREACH (GROUP books BY BookAuthor)
>> GENERATE group AS BookAuthor, FLATTEN(books.YearOfPublication) AS Year;
```

```
grunt> books = FILTER books BY YearOfPublication > 0;
2016-03-18 09:31:41,607 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
2016-03-18 09:31:41,607 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
grunt> pivot = FOREACH (GROUP books BY BookAuthor)
>> GENERATE group AS BookAuthor, FLATTEN(books.YearOfPublication) AS Year;
grunt>
```

Group operation creates a collection where each key corresponds to a list of values, FLATTEN "flattens" this list to generate entries for each list value.

**Step10**: Create a set for authors book count by year

```
authorYearGroup = GROUP pivot BY (BookAuthor, Year);
with_count = FOREACH authorYearGroup
>> GENERATE FLATTEN(group), COUNT(pivot) as count;
DESCRIBE with_count;
```

```
grunt> authorYearGroup = GROUP pivot BY (BookAuthor, Year);
grunt> with_count = FOREACH authorYearGroup
>> GENERATE FLATTEN(group), COUNT(pivot) as count;
grunt> DESCRIBE with_count;
with_count: {group::BookAuthor: chararray,group::Year: int,count: long}
grunt>
```

authorYearGroup will find all (author,year) combinations

FLATTEN on this group will expand the BookAuthor and Year, finally take the count of number of books for author/year combination.

**Step11**: Lets group the result by sorting the book count, and then placing the year/count tuples into a collection with the author

```
author_result = FOREACH (GROUP with_count BY BookAuthor) {
>> order_by_count = ORDER with_count BY count DESC;
>> GENERATE group AS Author, order_by_count.(Year, count) AS Books;
>> };
DESCRIBE author_result;
```

```
grunt> author_result = FOREACH (GROUP with_count BY BookAuthor) {
>> order_by_count = ORDER with_count BY count DESC;
>> GENERATE group AS Author, order_by_count.(Year, count) AS Books;
>> };
grunt> DESCRIBE author_result;
author_result: {Author: chararray,Books: {(group::Year: int,count: long)}}
grunt>
```

**Step12**: Extract set of all publishers and author from books collection

```
pub_auth = FOREACH books GENERATE Publisher, BookAuthor;
distinct_authors = FOREACH (GROUP pub_auth BY Publisher) {
>> da = DISTINCT pub_auth.BookAuthor;
>> GENERATE group AS Publisher, da AS Author;
>> };
distinct_flat = FOREACH distinct_authors GENERATE Publisher, FLATTEN(Author) AS Author;
```

```
grunt> pub_auth = FOREACH books GENERATE Publisher, BookAuthor;
grunt> distinct_authors = FOREACH (GROUP pub_auth BY Publisher) {
>> da = DISTINCT pub_auth.BookAuthor;
>> GENERATE group AS Publisher, da AS Author;
>> };
grunt> distinct_flat = FOREACH distinct_authors GENERATE Publisher, FLATTEN(Author) AS Author;
grunt>
```

pub_auth will group the author by publisher, we then extract only the distinct author per publisher.

distinct_authors collection is then FALTTENED into distinct_flat in order to obtain publisher/author list similar to that of author/year list

**Step13**: Join the result sets of step 9 and step 10 i.e. output of authors/year collection with publisher/author collection.

```
joined = JOIN distinct_flat BY Author, author_result BY Author;
filtered = FOREACH joined GENERATE
>> distinct_flat::Publisher AS Publisher,
>> distinct_flat::Author AS Author,
>> author_result::Books AS Books;
```

```
grunt> joined = JOIN distinct_flat BY Author, author_result BY Author;
grunt> filtered = FOREACH joined GENERATE
>> distinct_flat::Publisher AS Publisher,
>> distinct_flat::Author AS Author,
>> author_result::Books AS Books;
grunt>
```

Here we are joining the author/year results with publisher/author result. The ": :" is a dereference operator which is used when you need unambiguously declare a column.

**Step14**: Finally generate the result of publisher/authors/books and Store the data back into HDFS

```
result = FOREACH (GROUP filtered BY Publisher) {
>> order_by_pub = ORDER filtered BY Publisher ASC;
>> GENERATE group AS Publisher, order_by_pub.(Author, Books);
};
```

```
grunt> result = FOREACH (GROUP filtered BY Publisher) {
>> order_by_pub = ORDER filtered BY Publisher ASC;
>> GENERATE group AS Publisher, order_by_pub.(Author, Books);
>> };
grunt>
```

We are sorting the publishers, and then generating a collection with publishers/authors/books

Alternatively, you can also DUMP the result but let's also save the result in HDFS.

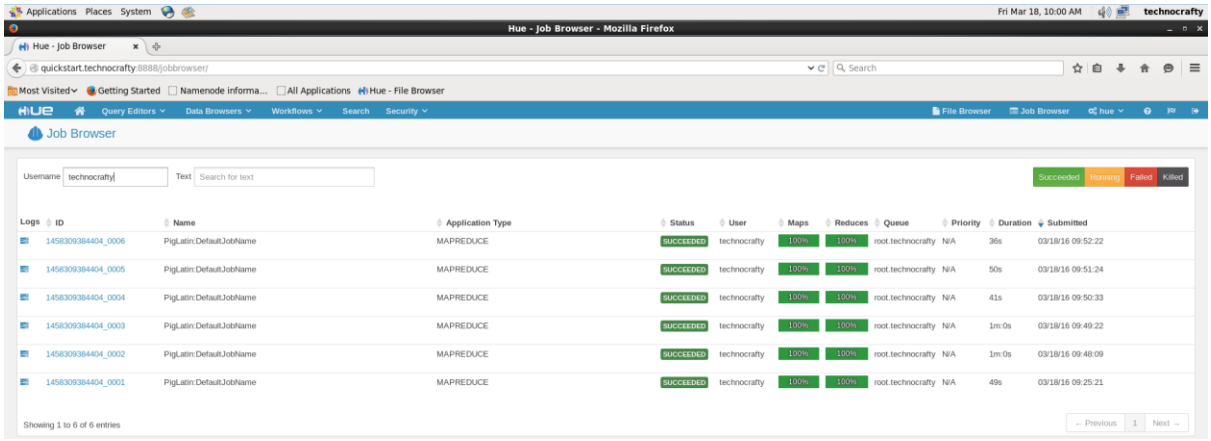
```
STORE result INTO '/user/technocrafty/books_publisher';
```

```
grunt> STORE result INTO '/user/technocrafty/books_publisher';
2016-03-18 09:48:03,674 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: HASH_JOIN,GROUP_BY,FILTER
2016-03-18 09:48:03,678 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimizer - {RULES_ENABLED=[AddForEach, ColumnMapKeyPrune, DuplicateForEachColumnRewrite, GroupByConstParallelSetter, ImplicitSplitInserter, LimitOptimizer, LoadTypeCastInserter, MergeFilter, MergeForEach, NewPartitionFilterOptimizer, PushDownForEachFlatten, PushUpFilter, SplitFilter, StreamTypeCastInserter], RULES_DISABLED=[FilterLogicExpressionSimplifier, PartitionFilterOptimizer]}
```

This will start the MapReduce task

```
2016-03-18 09:53:02,149 [main] INFO  org.apache.pig.tools.pigstats.SimplePigStats - Script Statistics:

HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.6.0-cdh5.5.0  0.12.0-cdh5.5.0 technocrafty    2016-03-18 09:48:03     2016-03-18 09:53:02     HASH_JOIN,GROUP_BY,FILTER

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTIme      AvgMapTime      MedianMapTime   MaxReduceTime   MinReduceTime   AvgReduceTime   MedianReducetime        Alias   Feature Outputs
job_1458309384404_0002  1       1       17      17      17      17      25      25      25      25      1-108,1-110,1-111,books,da,distinct_authors,distinct_flat,pivot,pub_auth        MULTI_QUERY
job_1458309384404_0003  1       1       21      21      21      21      16      16      16      16      authorYearGroup,with_count      GROUP_BY,COMBINER
job_1458309384404_0004  1       1       13      13      13      13      12      12      12      12      1-109,author_result     GROUP_BY
job_1458309384404_0005  2       1       21      21      21      21      14      14      14      14      filtered,joined HASH_JOIN
job_1458309384404_0006  1       1       10      10      10      10      12      12      12      12      1-112,result    GROUP_BY        /user/technocrafty/books_publisher,

Input(s):
Successfully read 271379 records (73360509 bytes) from: "/user/technocrafty/input/BX-Books.txt"

Output(s):
Successfully stored 16397 records (11646343 bytes) in: "/user/technocrafty/books_publisher"

Counters:
Total records written : 16397
Total bytes written : 11646343
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_1458309384404_0002  ->      job_1458309384404_0003,
job_1458309384404_0003  ->      job_1458309384404_0004,
job_1458309384404_0004  ->      job_1458309384404_0005,
job_1458309384404_0005  ->      job_1458309384404_0006,
job_1458309384404_0006
```

To view the status of MapReduce job, launch HUE Job Browser




**Step15**: List the file generated at the target location in HDFS

```
[technocrafty@quickstart ~]$ hdfs dfs -ls books_publisher
Found 2 items
-rw-r--r--   1 technocrafty technocrafty          0 2016-03-18 09:52 books_publisher/_SUCCESS
-rw-r--r--   1 technocrafty technocrafty   11646343 2016-03-18 09:52 books_publisher/part-r-00000
```

```
[technocrafty@quickstart ~]$ hdfs dfs -cat books_publisher/part-r-00000 | tail -n 50
Random House Children's Books (A Division of Random House Group)        {(Jane Gardam,{(1997,1),(2002,1),(2003,1),(2001,1),(1981,1),(1988,1),(1995,1),(1996,1)}),(Enid Blyton,{(1996,24),(2001
,17),(1994,13),(1998,13),(2000,13),(1990,11),(1995,10),(2003,8),(1992,8),(1985,6),(1978,6),(2002,5),(1997,5),(1988,3),(1991,3),(1977,3),(1999,2),(1968,2),(1971,1),(1970,1),(1967,1),(1962,1),
(1984,1),(1951,1),(1987,1),(1989,1)}),(Rosemary Sutcliff,{(1994,3),(1989,2),(1987,2),(1990,2),(1981,2),(1993,1),(1991,1),(1984,1),(1968,1),(1971,1),(1973,1),(1979,1),(2000,1)}),(Stan Cullimo
re,{(1993,1),(1999,1)}),(Anthony Masters,{(1990,3),(1994,2),(1984,1)}),(Paul Temple,{(1986,1)}),(Gabrielle Vincent,{(1982,1)}),(Annette Curtis Klause,{(1992,2),(1997,2),(1994,1),(1990,1),(19
95,1)}),(Ruth Elwin Harris,{(2002,4),(1986,1),(1995,1)}),(Barbara Wersba,{(1990,1),(1987,1),(1973,1)}),(Norman Hunter,{(1947,1),(1999,1),(1991,1),(1975,1),(1974,1)}),(Barbara Ireson,{(1970,1
),(1985,1),(1986,1)}),(Erich Kastner,{(1950,1),(1985,1),(1998,1)}),(Bonnie Bryant Hiller,{(1989,3),(1991,3),(1992,1)}),(Jennifer Walsh,{(1990,1)}),(Linda Hoy,{(1983,1)}),(Peggy Woodford,{(19
94,1),(1986,1)}),(Elizabeth Barnard,{(1989,1)}),(Samuel H. Klarreich,{(1989,1)}),(Jonathan Allen,{(1993,1),(1999,1)}),(Helen Bannerman,{(2003,1),(1965,1)}),(Ann Jungman,{(1993,2),(1994,1)}),
(Philip Schofield,{(1988,1)}),(Robin McKinley,{(1986,4),(2000,3),(2002,3),(1984,3),(2001,2),(1993,2),(2003,2),(1989,2),(1987,2),(1998,2),(1985,2),(1982,2),(1992,1),(1991,1),(1983,1),(1994,1)
,(1995,1),(1997,1),(1978,1)}),(Dick King-Smith,{(1999,6),(1997,5),(2000,5),(1995,5),(1998,5),(1996,4),(1994,4),(1990,3),(1992,2),(1988,2),(1982,2),(2001,1),(1993,1),(1987,1),(2003,1),(2005,1
),(2002,1)}),(Francine Pascal,{(1984,45),(1988,30),(1994,26),(1996,26),(1985,25),(1986,23),(1995,22),(1993,21),(1991,20),(1987,19),(2000,17),(1999,17),(1997,16),(1992,16),(1989,16),(1990,14)
,(2001,10),(2002,3),(1983,3),(2003,2),(1982,2)}),(Jane Hissey,{(1997,1),(1999,1),(1987,1),(1992,1),(1986,1),(1996,1)}),(B.B. Hillier,{(1990,1)})}
Straight Arrow Books$$$ [distributed by Quick Fox Inc., New York        {(Hunter S Thompson,{(1973,1),(1997,1)})}
Addison Wesley Longman ELT Division (a Pearson Education company)       {(John Tully,{(1987,1)}),(Roland John,{(1983,1)}),(Jane Homeshaw,{(1980,1),(1987,1)})}
Commissie voor de Collectieve Propaganda van het Nederlandse Boek       {(Bertus Aafjes,{(1976,1),(1979,1),(1973,1)}),(Henri A. A. R Knap,{(1981,1)})}
Stichting voor de Collectieve Propaganda van het Nederlandse Boek       {(Maarten 't Hart,{(2003,2),(1999,2),(1984,1),(2001,1),(2000,1),(1986,1),(1978,1)})}
The Center for the Study of Language and Information Publications       {(Byron Reeves,{(1998,1)})}
Distributed to the book trade in the United States by Harper & Row      {(Van Wolverton,{(1988,2),(1994,1),(1985,1)})}
Alliance for Committed Civic Engagement & Social Solutions (ACCESS)     {(Tawn Holstra,{(1999,1)})}
Distributed by Jews for the Preservation of Firearms Ownership, Inc     {(Richard W Stevens,{(1999,1)})}
```

## Follow-Up Assignment

### 1. Analyse Movies dataset using Pig

**Step1**: Create a file named movies.txt and load into any directory of HDFS

```
1,The Nightmare Before Christmas,1993,3.9,4568
2,The Mummy,1932,3.5,4388
3,Orphans of the Storm,1921,3.2,9062
4,The Object of Beauty,1991,2.8,6150
5,Night Tide,1963,2.8,5126
6,One Magic Christmas,1985,3.8,5333
7,Muriel's Wedding,1994,3.5,6323
8,Mother's Boys,1994,3.4,5733
9,Nosferatu: Original Version,1929,3.5,5651
10,Nick of Time,1995,3.4,5333
```

**Step2**: Find out how many movies were rated above 3.5?

**Step3**: List the movies that were released between 1960 and 1995

**Step4**: List the movies starting with alphabet N.

**Step5**: List the movie names with its duration in minutes.

### 2. PIG for ETL processing

**Source**: http://seanlahman.com/files/database/lahman-csv_2015-01-24.zip

**Step1**: Unzip the file into a directory and upload batting.csv file

**Step2**: Write a pig script fulfilling below criteria

What is the highest run scored by a player for each year?

What is the player ID of the highest run scorer?

### 3. Process Songs Dataset using Pig

**Source**: http://static.echonest.com/millionsongsubset_full.tar.gz

**Step1**: Load the input data and filter out unpopular songs or songs without latitude/longitude localization.

**Step2**: Produce all pairs of different songs and calculate distance between their artist's localization.

**Step3**: For each song, calculate the average distance between the songs artist's localization and all other artist's localization.

**Step4**: Find the most popular song for a given location

**Step5**: Find the most isolated song which are recorded by artists who live far away from other artists.

**Step6**: Store the result in such a format that can be visualized in Google Maps.

4. **Analyse New York Stock Exchange dataset**

**Source**: infochimps_dataset_4778_download_16677-csv.zip

Locate the files named NYSE_daily_prices_A.csv and NYSE_dividends_A.csv

**Step1**: Upload the files into HDFS

**Step2**: Load the file NYSE_daily_prices_A.csv using PigStorage with a relation say STOCK_A

**Step3**: Define a schema for relation STOCK_A

**Step4**: Create another relation to limit the collection of stocks for 100 entries.

**Step5**: View the data of newly created relation.

**Step6**: Create another relation with field symbol, date and close details from previous relation.

**Step7**: Store the relationship data into HDFS file

**Step8**: Load the file NYSE_dividends_A.csv using PigStorage with relation named DIV_A

**Step9**: Group dividends price by symbol of your choice for example AZZ

**Step10**: Perform a join on relation STOCK_A and DIV_A by symbol and date field.

**Step11**: Store or Dump the resultant back to the HDFS.

# Exercise 10: Oozie

To Run a MapReduce JAR using Oozie workflow

1. We will use WordCount.jar for this example
2. Navigate to workspace directory on local filesystem and list the class files associated with WordCount

```
[technocrafty@quickstart workspace]$ cd WordCount
[technocrafty@quickstart WordCount]$ ls
bin  src
[technocrafty@quickstart WordCount]$ cd bin
[technocrafty@quickstart bin]$ ls
WordCount.class  WordCount$Map.class  WordCount$Reduce.class
[technocrafty@quickstart bin]$
```

3. Create a job.properties file

The parameters for the job must be provided in a file, either a Java Properties file (.properties) or a Hadoop XML Configuration file (.xml). This file must be specified with the -config option.

```
nameNode=hdfs://quickstart.technocrafty:8020
jobTracker=localhost:8021
queueName=default
examplesRoot=examplesoozie
oozie.wf.application.path=${nameNode}/user/technocrafty/examplesoozie/map-reduce
outputDir=map-reduce
```

```
[technocrafty@quickstart ~]$ cat job.properties
nameNode=hdfs://quickstart.technocrafty:8020
jobTracker=localhost:8021
queueName=default
examplesRoot=examplesoozie
oozie.wf.application.path=${nameNode}/user/technocrafty/examplesoozie/map-reduce
outputDir=map-reduce
```

4. Create a workflow.xml, which defines a set of actions to be performed as a sequence or in Control Dependency DAG (Direct Acyclic Graph).

Note: "control dependency" means that the second action cannot run until the first action has been completed.

```
<workflow-app name="map-reduce-wf" xmlns="uri:oozie:workflow:0.1">
 <start to="mr-node"/>
 <action name="mr-node">
    <map-reduce>
      <job-tracker>${jobTracker}</job-tracker>
      <name-node>${nameNode}</name-node>
      <prepare>
        <delete path="${nameNode}/user/technocrafty/${outputDir}"/>
      </prepare>

   <configuration>
     <property>
       <name>mapred.mapper.new-api</name>
       <value>true</value>
     </property>
     <property>
       <name>mapred.reducer.new-api</name>
       <value>true</value>
     </property>
     <property>
       <name>mapred.job.queue.name</name>
       <value>${queueName}</value>
     </property>
     <property>
       <name>mapreduce.map.class</name>
       <value>WordCount.Map.WordCount$Map.class</value>
     </property>
     <property>
       <name>mapreduce.reduce.class</name>
       <value>WordCount.Reduce.WordCount$Reduce.class</value>
     </property>
     <property>
       <name>mapreduce.combine.class</name>
       <value>WordCount.Reduce.WordCount$Reduce.class</value>
     </property>
     <property>
       <name>mapred.output.key.class</name>
       <value>org.apache.hadoop.io.Text</value>
     </property>
     <property>
       <name>mapred.output.value.class</name>
       <value>org.apache.hadoop.io.IntWritable</value>
     </property>
     <property>
       <name>mapred.input.dir</name>
       <value>/user/technocrafty/examplesoozie/input-data/text</value>
     </property>
     <property>
       <name>mapred.output.dir</name>
       <value>/user/technocrafty/${outputDir}</value>
     </property>
   </configuration>
```

```
  </map-reduce>
  <ok to="end"/>
  <error to="fail"/>
 </action>
   <kill name="fail">
   <message>Map/Reduce failed, error
message[${wf:errorMessage(wf:lastErrorNode())}]</message>
   </kill>
   <end name="end"/>
</workflow-app>
```

5. Validate the workflow.xml file

```
oozie validate -oozie http://quickstart.technocrafty:11000/oozie workflow.xml
```

```
[technocrafty@quickstart ~]$ oozie validate -oozie http://quickstart.technocrafty:11000/oozie workflow.xml
Valid workflow-app
[technocrafty@quickstart ~]$
```

6. Create a directory on HDFS under which all the files related to the Oozie job will be kept. In this directory, push the workflow.xml created in the previous step.

```
$ hdfs dfs -mkdir -p /user/technocrafty/examplesoozie/map-reduce
$ hdfs dfs -put workflow.xml /user/technocrafty/examplesoozie/map-reduce/workflow.xml
```

```
[technocrafty@quickstart ~]$ hdfs dfs -mkdir -p /user/technocrafty/examplesoozie/map-reduce
[technocrafty@quickstart ~]$ hdfs dfs -put workflow.xml /user/technocrafty/examplesoozie/map-reduce/workf
low.xml
[technocrafty@quickstart ~]$ hdfs dfs -ls /user/technocrafty/examplesoozie/map-reduce
Found 1 items
-rw-r--r--   1 technocrafty technocrafty       9406 2016-04-23 07:54 /user/technocrafty/examplesoozie/map
-reduce/workflow.xml
[technocrafty@quickstart ~]$
```

7. Now under the directory created for the Oozie job, create a folder named lib in which the required library / jar files are kept.

```
$ hdfs dfs -mkdir -p /user/technocrafty/examplesoozie/map-reduce/lib
```

```
[technocrafty@quickstart ~]$ hdfs dfs -mkdir -p /user/technocrafty/examplesoozie/map-reduce/lib
[technocrafty@quickstart ~]$ hdfs dfs -ls /user/technocrafty/examplesoozie/map-reduce
Found 2 items
drwxr-xr-x   - technocrafty technocrafty          0 2016-04-23 07:55 /user/technocrafty/examplesoozie/map
-reduce/lib
-rw-r--r--   1 technocrafty technocrafty       9406 2016-04-23 07:54 /user/technocrafty/examplesoozie/map
-reduce/workflow.xml
[technocrafty@quickstart ~]$
```

8. Once the directory is created, copy Hadoop MapReduce examples jar under this directory.

```
$ hdfs dfs -put /home/technocrafty/WordCount.jar
/user/technocrafty/examplesoozie/map-reduce/lib/WordCount.jar
```

```
[technocrafty@quickstart ~]$ hdfs dfs -put WordCount.jar /user/technocrafty/examplesoozie/map-reduce/lib/
WordCount.jar
[technocrafty@quickstart ~]$ hdfs dfs -ls /user/technocrafty/examplesoozie/map-reduce/lib
Found 1 items
-rw-r--r--   1 technocrafty technocrafty       6004 2016-04-23 07:57 /user/technocrafty/examplesoozie/map
-reduce/lib/WordCount.jar
[technocrafty@quickstart ~]$
```

9. Now you can execute the workflow created, and use it to run Hadoop MapReduce program for WordCount

```
$ oozie job -oozie http://quickstart.technocrafty:11000/oozie -config job.properties -run
```

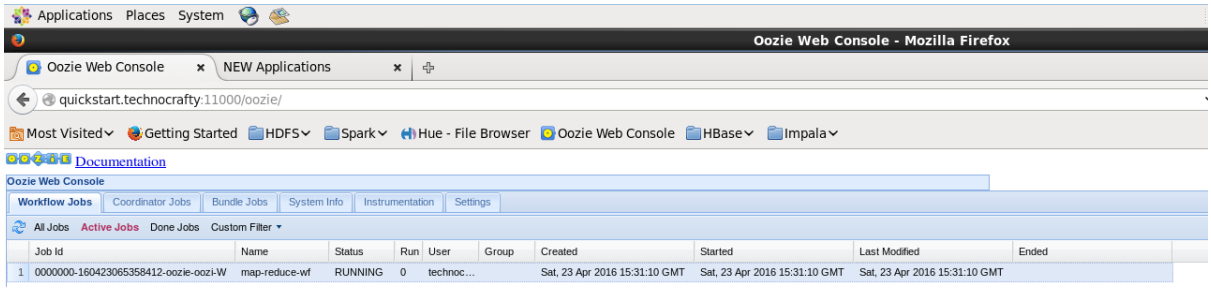
```
[technocrafty@quickstart ~]$ oozie job -oozie http://quickstart.technocrafty:11000/oozie -config job.prop
erties -run
job: 0000000-160423065358412-oozie-oozi-W
[technocrafty@quickstart ~]$
```

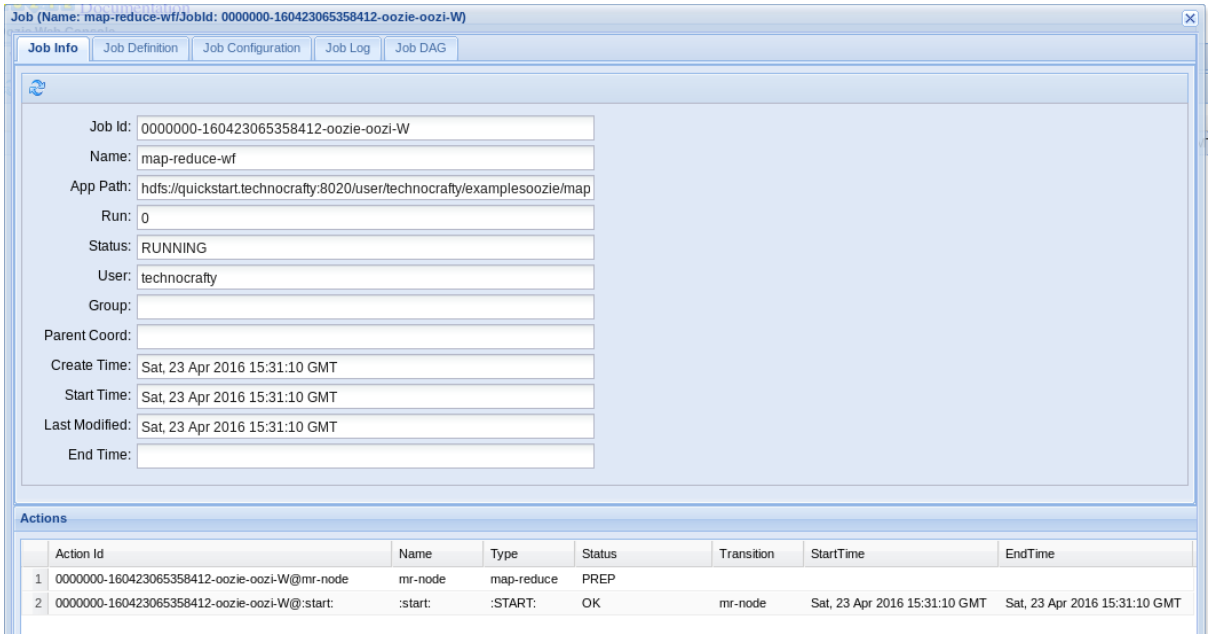
10. You can view the status of the job as shown below:

```
$ oozie job -oozie http://quickstart.technocrafty:11000/oozie -info 0000000-
160423065358412-oozie-oozi-W
```

```
[technocrafty@quickstart ~]$ oozie job -oozie http://quickstart.technocrafty:11000/oozie -info 0000000-16
0423065358412-oozie-oozi-W
Job ID : 0000000-160423065358412-oozie-oozi-W
------------------------------------------------------------------------------------------------------------
--------------------------
Workflow Name : map-reduce-wf
App Path      : hdfs://quickstart.technocrafty:8020/user/technocrafty/examplesoozie/map-reduce
Status        : RUNNING
Run           : 0
User          : technocrafty
Group         : -
Created       : 2016-04-23 15:31 GMT
Started       : 2016-04-23 15:31 GMT
Last Modified : 2016-04-23 15:31 GMT
Ended         : -
CoordAction ID: -

Actions
------------------------------------------------------------------------------------------------------------
--------------------------
ID                                                                            Status    Ext ID
      Ext Status Err Code
------------------------------------------------------------------------------------------------------------
--------------------------
0000000-160423065358412-oozie-oozi-W@mr-node                                  PREP      -
      -          -
------------------------------------------------------------------------------------------------------------
--------------------------
0000000-160423065358412-oozie-oozi-W@:start:                                  OK        -
      OK         -
------------------------------------------------------------------------------------------------------------
--------------------------
```

From the Oozie console:




11. Once the job is completed, verify its output.

# Exercise 11: Spark

- Spark shell can be launched in two ways i.e. Scala and Python.
- To launch Scala spark shell, follow below steps

```
$ spark-shell
```

```
[technocrafty@quickstart Datasets]$ spark-shell
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/zookeeper/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLo
ggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/jars/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.cl
ass]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
16/08/20 20:10:13 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... u
sing builtin-java classes where applicable
16/08/20 20:10:13 INFO spark.SecurityManager: Changing view acls to: technocrafty
16/08/20 20:10:13 INFO spark.SecurityManager: Changing modify acls to: technocrafty
16/08/20 20:10:13 INFO spark.SecurityManager: SecurityManager: authentication disabled; ui acls disabled;
 users with view permissions: Set(technocrafty); users with modify permissions: Set(technocrafty)
16/08/20 20:10:14 INFO spark.HttpServer: Starting HTTP Server
16/08/20 20:10:14 INFO server.Server: jetty-8.y.z-SNAPSHOT
16/08/20 20:10:14 INFO server.AbstractConnector: Started SocketConnector@0.0.0.0:51259
16/08/20 20:10:14 INFO util.Utils: Successfully started service 'HTTP class server' on port 51259.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 1.5.0-cdh5.5.0
      /_/
```


You will see several INFO and WARNING message on the command prompt after launching spark-shell, which can be disregarded.

Logs will appear as below and finally you will get Scala Prompt

```
SQL context available as sqlContext.

scala>
```

- Spark creates a SparkContext object called sc, verify that the object exists

```
scala> sc
res0: org.apache.spark.SparkContext = org.apache.spark.SparkContext@380f60a3
```

- To know various SparkContext methods which are available, type sc. (sc followed by dot) and then TAB key

```
scala> sc.
accumulable               accumulableCollection     accumulator               addFile                   addJar                    addSparkListener          appName
applicationAttemptId      applicationId             asInstanceOf              binaryFiles               binaryRecords             broadcast                 cancelAllJobs
cancelJobGroup            clearCallSite             clearFiles                clearJars                 clearJobGroup             defaultMinPartitions      defaultMinSplits
defaultParallelism        emptyRDD                  externalBlockStoreFolderName files                  getAllPools               getCheckpointDir          getConf
getExecutorMemoryStatus   getExecutorStorageStatus  getLocalProperty          getPersistentRDDs         getPoolForName            getRDDStorageInfo         getSchedulingMode
hadoopConfiguration       hadoopFile                hadoopRDD                 initLocalProperties       isInstanceOf              isLocal                   jars
killExecutor              killExecutors             makeRDD                   master                    metricsSystem             newAPIHadoopFile          newAPIHadoopRDD
objectFile                parallelize               range                     requestExecutors          runApproximateJob         runJob                    sequenceFile
setCallSite               setCheckpointDir          setJobDescription         setJobGroup               setLocalProperty          setLogLevel               sparkUser
startTime                 statusTracker             stop                      submitJob                 tachyonFolderName         textFile                  toString
union                     version                   wholeTextFiles
```

- Similarly, to launch Python Spark Shell, follow below steps

```
$ pyspark
```

```
16/03/22 08:38:51 INFO storage.BlockManagerMaster: Registered BlockManager
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 1.5.0-cdh5.5.0
      /_/

Using Python version 2.6.6 (r266:84292, Feb 22 2013 00:00:18)
SparkContext available as sc, HiveContext available as sqlContext.
>>>
```

- Type sc to check the SparkContext object

```
>>> sc
<pyspark.context.SparkContext object at 0x110b050>
```

- To exit from shell prompt, press CTRL+D or type exit().

## 1. Basic Commands and Operations

Spark is based on the concept of Resilient Distributed Dataset (RDD), which is fault tolerant collection of elements that can be operated in parallel.

**Two ways to create RDDs:**

- Parallelizing an existing collection
- Referencing a dataset from an External Storage System

**Parallelized collection:**

To create a parallelized collection holding numbers 1 to 6

Example 1

val data = Array (1,2,3,4,5,6)

val distData = sc.parallelize(data)

```
scala> val data = Array (1,2,3,4,5,6)
data: Array[Int] = Array(1, 2, 3, 4, 5, 6)

scala> val distData = sc.parallelize(data)
distData: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[0] at parallelize at <console>:23
```

Once 'distData' dataset is created, we can perform operations such as:

- Finding the sum, mean & variance

distData.sum

distData.mean

distData.variance

```
scala> distData.sum
16/03/24 06:58:14 INFO spark.SparkContext: Starting job: sum at <console>:26
16/03/24 06:58:14 INFO scheduler.DAGScheduler: Got job 0 (sum at <console>:26) with 1 output partitions
16/03/24 06:58:14 INFO scheduler.DAGScheduler: Final stage: ResultStage 0(sum at <console>:26)
16/03/24 06:58:14 INFO scheduler.DAGScheduler: Parents of final stage: List()
16/03/24 06:58:14 INFO scheduler.DAGScheduler: Missing parents: List()
16/03/24 06:58:14 INFO scheduler.DAGScheduler: Submitting ResultStage 0 (MapPartitionsRDD[1] at numericRDDToDoubleRDDFunctions at <console>:26), which has no missing parents
16/03/24 06:58:16 INFO storage.MemoryStore: ensureFreeSpace(2336) called with curMem=0, maxMem=560497950
16/03/24 06:58:16 INFO storage.MemoryStore: Block broadcast_0 stored as values in memory (estimated size 2.3 KB, free 534.5 MB)
16/03/24 06:58:16 INFO storage.MemoryStore: ensureFreeSpace(1449) called with curMem=2336, maxMem=560497950
16/03/24 06:58:16 INFO storage.MemoryStore: Block broadcast_0_piece0 stored as bytes in memory (estimated size 1449.0 B, free 534.5 MB)
16/03/24 06:58:16 INFO storage.BlockManagerInfo: Added broadcast_0_piece0 in memory on localhost:41207 (size: 1449.0 B, free: 534.5 MB)
16/03/24 06:58:17 INFO spark.SparkContext: Created broadcast 0 from broadcast at DAGScheduler.scala:861
16/03/24 06:58:17 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ResultStage 0 (MapPartitionsRDD[1] at numericRDDToDoubleRDDFunctions at <console>:26)
16/03/24 06:58:17 INFO scheduler.TaskSchedulerImpl: Adding task set 0.0 with 1 tasks
16/03/24 06:58:17 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 0.0 (TID 0, localhost, partition 0,PROCESS_LOCAL, 2049 bytes)
16/03/24 06:58:18 INFO executor.Executor: Running task 0.0 in stage 0.0 (TID 0)
16/03/24 06:58:18 INFO executor.Executor: Finished task 0.0 in stage 0.0 (TID 0). 955 bytes result sent to driver
16/03/24 06:58:18 INFO scheduler.DAGScheduler: ResultStage 0 (sum at <console>:26) finished in 1.212 s
16/03/24 06:58:18 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 0.0 (TID 0) in 1039 ms on localhost (1/1)
16/03/24 06:58:18 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 0.0, whose tasks have all completed, from pool
16/03/24 06:58:18 INFO scheduler.DAGScheduler: Job 0 finished: sum at <console>:26, took 4.364338 s
res2: Double = 21.0
```

```
scala> distData.mean
16/03/24 07:01:22 INFO spark.SparkContext: Starting job: mean at <console>:26
16/03/24 07:01:22 INFO scheduler.DAGScheduler: Got job 1 (mean at <console>:26) with 1 output partitions
16/03/24 07:01:22 INFO scheduler.DAGScheduler: Final stage: ResultStage 1(mean at <console>:26)
16/03/24 07:01:22 INFO scheduler.DAGScheduler: Parents of final stage: List()
16/03/24 07:01:22 INFO scheduler.DAGScheduler: Missing parents: List()
16/03/24 07:01:22 INFO scheduler.DAGScheduler: Submitting ResultStage 1 (MapPartitionsRDD[3] at mean at <console>:26), which has no missing parents
16/03/24 07:01:22 INFO storage.MemoryStore: ensureFreeSpace(2496) called with curMem=3785, maxMem=560497950
16/03/24 07:01:22 INFO storage.MemoryStore: Block broadcast_1 stored as values in memory (estimated size 2.4 KB, free 534.5 MB)
16/03/24 07:01:22 INFO storage.MemoryStore: ensureFreeSpace(1531) called with curMem=6281, maxMem=560497950
16/03/24 07:01:22 INFO storage.MemoryStore: Block broadcast_1_piece0 stored as bytes in memory (estimated size 1531.0 B, free 534.5 MB)
16/03/24 07:01:22 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on localhost:41207 (size: 1531.0 B, free: 534.5 MB)
16/03/24 07:01:22 INFO spark.SparkContext: Created broadcast 1 from broadcast at DAGScheduler.scala:861
16/03/24 07:01:22 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ResultStage 1 (MapPartitionsRDD[3] at mean at <console>:26)
16/03/24 07:01:22 INFO scheduler.TaskSchedulerImpl: Adding task set 1.0 with 1 tasks
16/03/24 07:01:22 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 1.0 (TID 1, localhost, partition 0,PROCESS_LOCAL, 2049 bytes)
16/03/24 07:01:22 INFO executor.Executor: Running task 0.0 in stage 1.0 (TID 1)
16/03/24 07:01:23 INFO executor.Executor: Finished task 0.0 in stage 1.0 (TID 1). 1080 bytes result sent to driver
16/03/24 07:01:23 INFO scheduler.DAGScheduler: ResultStage 1 (mean at <console>:26) finished in 0.201 s
16/03/24 07:01:23 INFO scheduler.DAGScheduler: Job 1 finished: mean at <console>:26, took 0.341082 s
16/03/24 07:01:23 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 1.0 (TID 1) in 210 ms on localhost (1/1)
16/03/24 07:01:23 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 1.0, whose tasks have all completed, from pool
res4: Double = 3.5
```

```
scala> distData.variance
16/03/24 07:02:25 INFO spark.SparkContext: Starting job: variance at <console>:26
16/03/24 07:02:25 INFO scheduler.DAGScheduler: Got job 2 (variance at <console>:26) with 1 output partitions
16/03/24 07:02:25 INFO scheduler.DAGScheduler: Final stage: ResultStage 2(variance at <console>:26)
16/03/24 07:02:25 INFO scheduler.DAGScheduler: Parents of final stage: List()
16/03/24 07:02:25 INFO scheduler.DAGScheduler: Missing parents: List()
16/03/24 07:02:25 INFO scheduler.DAGScheduler: Submitting ResultStage 2 (MapPartitionsRDD[5] at variance at <console>:26), which has no missing parents
16/03/24 07:02:25 INFO storage.MemoryStore: ensureFreeSpace(2496) called with curMem=7812, maxMem=560497950
16/03/24 07:02:25 INFO storage.MemoryStore: Block broadcast_2 stored as values in memory (estimated size 2.4 KB, free 534.5 MB)
16/03/24 07:02:25 INFO storage.MemoryStore: ensureFreeSpace(1530) called with curMem=10308, maxMem=560497950
16/03/24 07:02:25 INFO storage.MemoryStore: Block broadcast_2_piece0 stored as bytes in memory (estimated size 1530.0 B, free 534.5 MB)
16/03/24 07:02:25 INFO storage.BlockManagerInfo: Added broadcast_2_piece0 in memory on localhost:41207 (size: 1530.0 B, free: 534.5 MB)
16/03/24 07:02:25 INFO spark.SparkContext: Created broadcast 2 from broadcast at DAGScheduler.scala:861
16/03/24 07:02:25 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ResultStage 2 (MapPartitionsRDD[5] at variance at <console>:26)
16/03/24 07:02:25 INFO scheduler.TaskSchedulerImpl: Adding task set 2.0 with 1 tasks
16/03/24 07:02:25 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 2.0 (TID 2, localhost, partition 0,PROCESS_LOCAL, 2049 bytes)
16/03/24 07:02:25 INFO executor.Executor: Running task 0.0 in stage 2.0 (TID 2)
16/03/24 07:02:25 INFO executor.Executor: Finished task 0.0 in stage 2.0 (TID 2). 1080 bytes result sent to driver
16/03/24 07:02:25 INFO scheduler.DAGScheduler: ResultStage 2 (variance at <console>:26) finished in 0.026 s
16/03/24 07:02:25 INFO scheduler.DAGScheduler: Job 2 finished: variance at <console>:26, took 0.074857 s
res5: Double = 2.9166666666666665
```


Spark sets the number of partition (i.e.to cut the dataset into) automatically based on your cluster, but this can be set manually by passing second parameter to *parallelize* (e.g. sc.parallelize(data,10)

**External Datasets:**

Spark can create distributed datasets from any storage system supported by Hadoop, Spark supports text files, SequenceFiles and any other Hadoop InputFormat.

To load a file from your Local Filesystem say batting.txt

Example2

val textFile = sc.textFile("file:/home/technocrafty/Datasets/batting.txt")


While using local filesystem, the file must be accessible at the same path on worker nodes.

Operations on RDD will be discussed in next section.


Spark's file-based input methods supports directories, compressed files and wildcards as well. i.e. textFile("/my/directory"), textFile("/my/directory/*.txt"), and textFile("/my/directory/*.gz")

## 2. Operations on RDD

**RDDs supports two types of operations:**

- **Transformations**

Creates a new dataset from an existing one

- **Actions**

Returns a value to the driver program after running a computation on the dataset.

Example: map is a transformation that passes each dataset element through a function and return a new RDD, whereas reduce is an action that aggregates all the elements of RDD using function and returns final result.


All transformations in Spark are lazy i.e. transformations are only computed when an action requires a result to be returned.

textFile dataset was already created in previous step, lets perform below operations

- counting the occurrence of lines having a particular word in it

Example 3: filter using python shell

```
textFile = sc.textFile ("file:/home/technocrafty/Datasets/Joyce.txt")
linesWithJoyce = textFile.filter(lambda line: "Joyce" in    line)
linesWithJoyce.count()
exit()
```

```
>>> textFile = sc.textFile("file:/home/technocrafty/Datasets/Joyce.txt")
16/03/25 06:47:05 INFO storage.MemoryStore: ensureFreeSpace(124088) called with curMem=0, maxMem=560497950
16/03/25 06:47:05 INFO storage.MemoryStore: Block broadcast_0 stored as values in memory (estimated size 121.2 KB, free 534.4 MB)
16/03/25 06:47:05 INFO storage.MemoryStore: ensureFreeSpace(15269) called with curMem=124088, maxMem=560497950
16/03/25 06:47:05 INFO storage.MemoryStore: Block broadcast_0_piece0 stored as bytes in memory (estimated size 14.9 KB, free 534.4 MB)
16/03/25 06:47:05 INFO storage.BlockManagerInfo: Added broadcast_0_piece0 in memory on localhost:40816 (size: 14.9 KB, free: 534.5 MB)
16/03/25 06:47:05 INFO spark.SparkContext: Created broadcast 0 from textFile at NativeMethodAccessorImpl.java:-2
```

```
>>> linesWithJoyce = textFile.filter(lambda line: "Joyce" in    line)
>>> linesWithJoyce.count()
16/03/25 06:48:47 WARN shortcircuit.DomainSocketFactory: The short-circuit local reads feature cannot be used because libhadoop cannot be loaded.
16/03/25 06:48:47 INFO mapred.FileInputFormat: Total input paths to process : 1
16/03/25 06:48:47 INFO spark.SparkContext: Starting job: count at <stdin>:1
16/03/25 06:48:47 INFO scheduler.DAGScheduler: Got job 0 (count at <stdin>:1) with 1 output partitions
16/03/25 06:48:47 INFO scheduler.DAGScheduler: Final stage: ResultStage 0(count at <stdin>:1)
16/03/25 06:48:47 INFO scheduler.DAGScheduler: Parents of final stage: List()
16/03/25 06:48:47 INFO scheduler.DAGScheduler: Missing parents: List()
16/03/25 06:48:47 INFO scheduler.DAGScheduler: Submitting ResultStage 0 (PythonRDD[2] at count at <stdin>:1), which has no missing parents
16/03/25 06:48:47 INFO storage.MemoryStore: ensureFreeSpace(6240) called with curMem=139357, maxMem=560497950
16/03/25 06:48:47 INFO storage.MemoryStore: Block broadcast_1 stored as values in memory (estimated size 6.1 KB, free 534.4 MB)
16/03/25 06:48:47 INFO storage.MemoryStore: ensureFreeSpace(3780) called with curMem=145597, maxMem=560497950
16/03/25 06:48:47 INFO storage.MemoryStore: Block broadcast_1_piece0 stored as bytes in memory (estimated size 3.7 KB, free 534.4 MB)
16/03/25 06:48:47 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on localhost:40816 (size: 3.7 KB, free: 534.5 MB)
16/03/25 06:48:48 INFO spark.SparkContext: Created broadcast 1 from broadcast at DAGScheduler.scala:861
16/03/25 06:48:48 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ResultStage 0 (PythonRDD[2] at count at <stdin>:1)
16/03/25 06:48:48 INFO scheduler.TaskSchedulerImpl: Adding task set 0.0 with 1 tasks
16/03/25 06:48:48 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 0.0 (TID 0, localhost, partition 0,PROCESS_LOCAL, 2151 bytes)
16/03/25 06:48:48 INFO executor.Executor: Running task 0.0 in stage 0.0 (TID 0)
16/03/25 06:48:48 INFO rdd.HadoopRDD: Input split: file:/home/technocrafty/Datasets/Joyce.txt:0+1573079
16/03/25 06:48:48 INFO Configuration.deprecation: mapred.tip.id is deprecated. Instead, use mapreduce.task.id
16/03/25 06:48:48 INFO Configuration.deprecation: mapred.task.id is deprecated. Instead, use mapreduce.task.attempt.id
16/03/25 06:48:48 INFO Configuration.deprecation: mapred.task.is.map is deprecated. Instead, use mapreduce.task.ismap
16/03/25 06:48:48 INFO Configuration.deprecation: mapred.task.partition is deprecated. Instead, use mapreduce.task.partition
16/03/25 06:48:48 INFO Configuration.deprecation: mapred.job.id is deprecated. Instead, use mapreduce.job.id
16/03/25 06:48:49 INFO python.PythonRDD: Times: total = 1003, boot = 277, init = 255, finish = 471
16/03/25 06:48:49 INFO executor.Executor: Finished task 0.0 in stage 0.0 (TID 0). 2124 bytes result sent to driver
16/03/25 06:48:49 INFO scheduler.DAGScheduler: ResultStage 0 (count at <stdin>:1) finished in 1.209 s
16/03/25 06:48:49 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 0.0 (TID 0) in 1191 ms on localhost (1/1)
16/03/25 06:48:49 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 0.0, whose tasks have all completed, from pool
16/03/25 06:48:49 INFO scheduler.DAGScheduler: Job 0 finished: count at <stdin>:1, took 1.465017 s
4
```

Example: filter using scala shell

```
val textFile = sc.textFile("file:/home/technocrafty/Datasets/Joyce.txt")
textFile.count() //number of items in this RDD
textFile.first() //First  item in this RDD
val linesWithJoyce = textFile.filter(line => line.contains("Joyce"))
linesWithJoyce.count() //How many lines contain "Joyce"
```

```
scala> val textFile = sc.textFile("file:/home/technocrafty/Datasets/Joyce.txt")
16/03/24 07:11:21 INFO storage.MemoryStore: ensureFreeSpace(92440) called with curMem=11838, maxMem=560497950
16/03/24 07:11:21 INFO storage.MemoryStore: Block broadcast_3 stored as values in memory (estimated size 90.3 KB, free 534.4 MB)
16/03/24 07:11:22 INFO storage.MemoryStore: ensureFreeSpace(21233) called with curMem=104278, maxMem=560497950
16/03/24 07:11:22 INFO storage.MemoryStore: Block broadcast_3_piece0 stored as bytes in memory (estimated size 20.7 KB, free 534.4 MB)
16/03/24 07:11:22 INFO storage.BlockManagerInfo: Added broadcast_3_piece0 in memory on localhost:41207 (size: 20.7 KB, free: 534.5 MB)
16/03/24 07:11:22 INFO spark.SparkContext: Created broadcast 3 from textFile at <console>:21
textFile: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[7] at textFile at <console>:21
```

```
scala> textFile.count()
16/03/24 07:12:24 INFO mapred.FileInputFormat: Total input paths to process : 1
16/03/24 07:12:24 INFO spark.SparkContext: Starting job: count at <console>:24
16/03/24 07:12:24 INFO scheduler.DAGScheduler: Got job 3 (count at <console>:24) with 1 output partitions
16/03/24 07:12:24 INFO scheduler.DAGScheduler: Final stage: ResultStage 3(count at <console>:24)
16/03/24 07:12:24 INFO scheduler.DAGScheduler: Parents of final stage: List()
16/03/24 07:12:24 INFO scheduler.DAGScheduler: Missing parents: List()
16/03/24 07:12:24 INFO scheduler.DAGScheduler: Submitting ResultStage 3 (MapPartitionsRDD[7] at textFile at <console>:21), which has no missing parents
16/03/24 07:12:24 INFO storage.MemoryStore: ensureFreeSpace(3000) called with curMem=125511, maxMem=560497950
16/03/24 07:12:24 INFO storage.MemoryStore: Block broadcast_4 stored as values in memory (estimated size 2.9 KB, free 534.4 MB)
16/03/24 07:12:24 INFO storage.MemoryStore: ensureFreeSpace(1789) called with curMem=128511, maxMem=560497950
16/03/24 07:12:24 INFO storage.MemoryStore: Block broadcast_4_piece0 stored as bytes in memory (estimated size 1789.0 B, free 534.4 MB)
16/03/24 07:12:24 INFO storage.BlockManagerInfo: Added broadcast_4_piece0 in memory on localhost:41207 (size: 1789.0 B, free: 534.5 MB)
16/03/24 07:12:24 INFO spark.SparkContext: Created broadcast 4 from broadcast at DAGScheduler.scala:861
16/03/24 07:12:24 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ResultStage 3 (MapPartitionsRDD[7] at textFile at <console>:21)
16/03/24 07:12:24 INFO scheduler.TaskSchedulerImpl: Adding task set 3.0 with 1 tasks
16/03/24 07:12:24 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 3.0 (TID 3, localhost, partition 0,PROCESS_LOCAL, 2151 bytes)
16/03/24 07:12:24 INFO executor.Executor: Running task 0.0 in stage 3.0 (TID 3)
16/03/24 07:12:24 INFO rdd.HadoopRDD: Input split: file:/home/technocrafty/Datasets/Joyce.txt:0+1573079
16/03/24 07:12:25 INFO Configuration.deprecation: mapred.tip.id is deprecated. Instead, use mapreduce.task.id
16/03/24 07:12:25 INFO Configuration.deprecation: mapred.task.id is deprecated. Instead, use mapreduce.task.attempt.id
16/03/24 07:12:25 INFO Configuration.deprecation: mapred.task.is.map is deprecated. Instead, use mapreduce.task.ismap
16/03/24 07:12:25 INFO Configuration.deprecation: mapred.task.partition is deprecated. Instead, use mapreduce.task.partition
16/03/24 07:12:25 INFO Configuration.deprecation: mapred.job.id is deprecated. Instead, use mapreduce.job.id
16/03/24 07:12:25 INFO executor.Executor: Finished task 0.0 in stage 3.0 (TID 3). 2082 bytes result sent to driver
16/03/24 07:12:25 INFO scheduler.DAGScheduler: ResultStage 3 (count at <console>:24) finished in 0.977 s
16/03/24 07:12:25 INFO scheduler.DAGScheduler: Job 3 finished: count at <console>:24, took 1.172793 s
res6: Long = 33056
```

```
scala> textFile.first()
16/03/24 07:14:35 INFO spark.SparkContext: Starting job: first at <console>:24
16/03/24 07:14:35 INFO scheduler.DAGScheduler: Got job 4 (first at <console>:24) with 1 output partitions
16/03/24 07:14:35 INFO scheduler.DAGScheduler: Final stage: ResultStage 4(first at <console>:24)
16/03/24 07:14:35 INFO scheduler.DAGScheduler: Parents of final stage: List()
16/03/24 07:14:35 INFO scheduler.DAGScheduler: Missing parents: List()
16/03/24 07:14:35 INFO scheduler.DAGScheduler: Submitting ResultStage 4 (MapPartitionsRDD[7] at textFile at <console>:21), which has no missing parents
16/03/24 07:14:35 INFO storage.MemoryStore: ensureFreeSpace(3168) called with curMem=130300, maxMem=560497950
16/03/24 07:14:35 INFO storage.MemoryStore: Block broadcast_5 stored as values in memory (estimated size 3.1 KB, free 534.4 MB)
16/03/24 07:14:35 INFO storage.MemoryStore: ensureFreeSpace(1849) called with curMem=133468, maxMem=560497950
16/03/24 07:14:35 INFO storage.MemoryStore: Block broadcast_5_piece0 stored as bytes in memory (estimated size 1849.0 B, free 534.4 MB)
16/03/24 07:14:35 INFO storage.BlockManagerInfo: Added broadcast_5_piece0 in memory on localhost:41207 (size: 1849.0 B, free: 534.5 MB)
16/03/24 07:14:35 INFO spark.SparkContext: Created broadcast 5 from broadcast at DAGScheduler.scala:861
16/03/24 07:14:35 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ResultStage 4 (MapPartitionsRDD[7] at textFile at <console>:21)
16/03/24 07:14:35 INFO scheduler.TaskSchedulerImpl: Adding task set 4.0 with 1 tasks
16/03/24 07:14:35 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 4.0 (TID 4, localhost, partition 0,PROCESS_LOCAL, 2151 bytes)
16/03/24 07:14:35 INFO executor.Executor: Running task 0.0 in stage 4.0 (TID 4)
16/03/24 07:14:36 INFO rdd.HadoopRDD: Input split: file:/home/technocrafty/Datasets/Joyce.txt:0+1573079
16/03/24 07:14:36 INFO executor.Executor: Finished task 0.0 in stage 4.0 (TID 4). 2101 bytes result sent to driver
16/03/24 07:14:36 INFO scheduler.DAGScheduler: ResultStage 4 (first at <console>:24) finished in 0.066 s
16/03/24 07:14:36 INFO scheduler.DAGScheduler: Job 4 finished: first at <console>:24, took 0.147247 s
res7: String = The Project Gutenberg EBook of Ulysses, by James Joyce
```

```
scala> val linesWithJoyce = textFile.filter(line => line.contains("Joyce"))
linesWithJoyce: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[8] at filter at <console>:23
```

```
scala> linesWithJoyce.count()
16/03/24 07:19:43 INFO spark.SparkContext: Starting job: count at <console>:26
16/03/24 07:19:43 INFO scheduler.DAGScheduler: Got job 5 (count at <console>:26) with 1 output partitions
16/03/24 07:19:43 INFO scheduler.DAGScheduler: Final stage: ResultStage 5(count at <console>:26)
16/03/24 07:19:43 INFO scheduler.DAGScheduler: Parents of final stage: List()
16/03/24 07:19:43 INFO scheduler.DAGScheduler: Missing parents: List()
16/03/24 07:19:43 INFO scheduler.DAGScheduler: Submitting ResultStage 5 (MapPartitionsRDD[8] at filter at <console>:23), which has no missing parents
16/03/24 07:19:43 INFO storage.MemoryStore: ensureFreeSpace(3232) called with curMem=135317, maxMem=560497950
16/03/24 07:19:43 INFO storage.MemoryStore: Block broadcast_6 stored as values in memory (estimated size 3.2 KB, free 534.4 MB)
16/03/24 07:19:43 INFO storage.MemoryStore: ensureFreeSpace(1894) called with curMem=138549, maxMem=560497950
16/03/24 07:19:43 INFO storage.MemoryStore: Block broadcast_6_piece0 stored as bytes in memory (estimated size 1894.0 B, free 534.4 MB)
16/03/24 07:19:43 INFO storage.BlockManagerInfo: Added broadcast_6_piece0 in memory on localhost:41207 (size: 1894.0 B, free: 534.5 MB)
16/03/24 07:19:43 INFO spark.SparkContext: Created broadcast 6 from broadcast at DAGScheduler.scala:861
16/03/24 07:19:43 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ResultStage 5 (MapPartitionsRDD[8] at filter at <console>:23)
16/03/24 07:19:43 INFO scheduler.TaskSchedulerImpl: Adding task set 5.0 with 1 tasks
16/03/24 07:19:43 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 5.0 (TID 5, localhost, partition 0,PROCESS_LOCAL, 2151 bytes)
16/03/24 07:19:43 INFO executor.Executor: Running task 0.0 in stage 5.0 (TID 5)
16/03/24 07:19:43 INFO rdd.HadoopRDD: Input split: file:/home/technocrafty/Datasets/Joyce.txt:0+1573079
16/03/24 07:19:44 INFO executor.Executor: Finished task 0.0 in stage 5.0 (TID 5). 2082 bytes result sent to driver
16/03/24 07:19:44 INFO scheduler.DAGScheduler: ResultStage 5 (count at <console>:26) finished in 0.656 s
16/03/24 07:19:44 INFO scheduler.DAGScheduler: Job 5 finished: count at <console>:26, took 0.751108 s
res8: Long = 4
```

- Add up the sizes of all the lines

```
val lineLength = textFile.map(s => s.length)

val totalLength = lineLength.reduce((a, b) => a + b)
```

```
scala> val lineLength = textFile.map(s => s.length)
lineLength: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[9] at map at <console>:23
```

```
scala> val totalLength = lineLength.reduce((a,b) => a+b)
16/03/24 07:35:43 INFO spark.SparkContext: Starting job: reduce at <console>:25
16/03/24 07:35:43 INFO scheduler.DAGScheduler: Got job 6 (reduce at <console>:25) with 1 output partitions
16/03/24 07:35:43 INFO scheduler.DAGScheduler: Final stage: ResultStage 6(reduce at <console>:25)
16/03/24 07:35:43 INFO scheduler.DAGScheduler: Parents of final stage: List()
16/03/24 07:35:43 INFO scheduler.DAGScheduler: Missing parents: List()
16/03/24 07:35:43 INFO scheduler.DAGScheduler: Submitting ResultStage 6 (MapPartitionsRDD[9] at map at <console>:23), which has no missing parents
16/03/24 07:35:43 INFO storage.MemoryStore: ensureFreeSpace(3400) called with curMem=140443, maxMem=560497950
16/03/24 07:35:43 INFO storage.MemoryStore: Block broadcast_7 stored as values in memory (estimated size 3.3 KB, free 534.4 MB)
16/03/24 07:35:43 INFO storage.MemoryStore: ensureFreeSpace(1977) called with curMem=143843, maxMem=560497950
16/03/24 07:35:43 INFO storage.MemoryStore: Block broadcast_7_piece0 stored as bytes in memory (estimated size 1977.0 B, free 534.4 MB)
16/03/24 07:35:43 INFO storage.BlockManagerInfo: Added broadcast_7_piece0 in memory on localhost:41207 (size: 1977.0 B, free: 534.5 MB)
16/03/24 07:35:43 INFO spark.SparkContext: Created broadcast 7 from broadcast at DAGScheduler.scala:861
16/03/24 07:35:43 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ResultStage 6 (MapPartitionsRDD[9] at map at <console>:23)
16/03/24 07:35:43 INFO scheduler.TaskSchedulerImpl: Adding task set 6.0 with 1 tasks
16/03/24 07:35:43 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 6.0 (TID 6, localhost, partition 0,PROCESS_LOCAL, 2151 bytes)
16/03/24 07:35:43 INFO executor.Executor: Running task 0.0 in stage 6.0 (TID 6)
16/03/24 07:35:43 INFO rdd.HadoopRDD: Input split: file:/home/technocrafty/Datasets/Joyce.txt:0+1573079
16/03/24 07:35:43 INFO executor.Executor: Finished task 0.0 in stage 6.0 (TID 6). 2160 bytes result sent to driver
16/03/24 07:35:43 INFO scheduler.DAGScheduler: ResultStage 6 (reduce at <console>:25) finished in 0.077 s
16/03/24 07:35:43 INFO scheduler.DAGScheduler: Job 6 finished: reduce at <console>:25, took 0.128379 s
totalLength: Int = 1506967
```

- Line with maximum words

Example:

textFile.map(line => line.split(" ").size).reduce((a,b) => if(a>b)  a else b)

```
scala> textFile.map(line => line.split(" ").size).reduce((a,b) => if(a>b) a else b)
16/03/24 07:39:58 INFO spark.SparkContext: Starting job: reduce at <console>:24
16/03/24 07:39:58 INFO scheduler.DAGScheduler: Got job 7 (reduce at <console>:24) with 1 output partitions
16/03/24 07:39:58 INFO scheduler.DAGScheduler: Final stage: ResultStage 7(reduce at <console>:24)
16/03/24 07:39:58 INFO scheduler.DAGScheduler: Parents of final stage: List()
16/03/24 07:39:58 INFO scheduler.DAGScheduler: Missing parents: List()
16/03/24 07:39:58 INFO scheduler.DAGScheduler: Submitting ResultStage 7 (MapPartitionsRDD[10] at map at <console>:24), which has no missing parents
16/03/24 07:39:58 INFO storage.MemoryStore: ensureFreeSpace(3400) called with curMem=145820, maxMem=560497950
16/03/24 07:39:58 INFO storage.MemoryStore: Block broadcast_8 stored as values in memory (estimated size 3.3 KB, free 534.4 MB)
16/03/24 07:39:58 INFO storage.MemoryStore: ensureFreeSpace(1986) called with curMem=149220, maxMem=560497950
16/03/24 07:39:58 INFO storage.MemoryStore: Block broadcast_8_piece0 stored as bytes in memory (estimated size 1986.0 B, free 534.4 MB)
16/03/24 07:39:58 INFO storage.BlockManagerInfo: Added broadcast_8_piece0 in memory on localhost:41207 (size: 1986.0 B, free: 534.5 MB)
16/03/24 07:39:58 INFO spark.SparkContext: Created broadcast 8 from broadcast at DAGScheduler.scala:861
16/03/24 07:39:58 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ResultStage 7 (MapPartitionsRDD[10] at map at <console>:24)
16/03/24 07:39:58 INFO scheduler.TaskSchedulerImpl: Adding task set 7.0 with 1 tasks
16/03/24 07:39:58 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 7.0 (TID 7, localhost, partition 0,PROCESS_LOCAL, 2151 bytes)
16/03/24 07:39:58 INFO executor.Executor: Running task 0.0 in stage 7.0 (TID 7)
16/03/24 07:39:59 INFO rdd.HadoopRDD: Input split: file:/home/technocrafty/Datasets/Joyce.txt:0+1573079
16/03/24 07:39:59 INFO executor.Executor: Finished task 0.0 in stage 7.0 (TID 7). 2160 bytes result sent to driver
16/03/24 07:39:59 INFO scheduler.DAGScheduler: ResultStage 7 (reduce at <console>:24) finished in 0.452 s
16/03/24 07:39:59 INFO scheduler.DAGScheduler: Job 7 finished: reduce at <console>:24, took 0.512457 s
res9: Int = 22
```

Example4: filter

```
val sampleData = 1 to 5000
val totData =  sc.parallelize(sampleData)
val result = totData.filter(_ %2==0)
result.collect()
```

with number of partition = 2

```
val totDataPar = sc.parallelize(sampleData,2)
val resultPar  = totDataPar.filter(_%2==0)
resultPar.collect()
```

```
scala> val sampleData = 1 to 5000
sampleData: scala.collection.immutable.Range.Inclusive = Range(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82
, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123,
124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161,
162, 163, 164, 165, 166, 167, 168, 16...
scala> val totData = sc.parallelize(sampleData)
totData: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[11] at parallelize at <console>:23

scala> val result = totData.filter(_%2 == 0)
result: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[12] at filter at <console>:25
```

```
scala> result.collect()
16/03/24 07:46:27 INFO storage.BlockManagerInfo: Removed broadcast_8_piece0 on localhost:41207 in memory (size: 1986.0 B, free: 534.5 MB)
16/03/24 07:46:27 INFO spark.ContextCleaner: Cleaned accumulator 8
16/03/24 07:46:27 INFO spark.SparkContext: Starting job: collect at <console>:28
16/03/24 07:46:27 INFO scheduler.DAGScheduler: Got job 8 (collect at <console>:28) with 1 output partitions
16/03/24 07:46:27 INFO scheduler.DAGScheduler: Final stage: ResultStage 8(collect at <console>:28)
16/03/24 07:46:27 INFO scheduler.DAGScheduler: Parents of final stage: List()
16/03/24 07:46:27 INFO scheduler.DAGScheduler: Missing parents: List()
16/03/24 07:46:27 INFO scheduler.DAGScheduler: Submitting ResultStage 8 (MapPartitionsRDD[12] at filter at <console>:25), which has no missing parents
16/03/24 07:46:27 INFO storage.MemoryStore: ensureFreeSpace(1952) called with curMem=113673, maxMem=560497950
16/03/24 07:46:27 INFO storage.MemoryStore: Block broadcast_9 stored as values in memory (estimated size 1952.0 B, free 534.4 MB)
16/03/24 07:46:27 INFO storage.MemoryStore: ensureFreeSpace(1209) called with curMem=115625, maxMem=560497950
16/03/24 07:46:27 INFO storage.MemoryStore: Block broadcast_9_piece0 stored as bytes in memory (estimated size 1209.0 B, free 534.4 MB)
16/03/24 07:46:27 INFO storage.BlockManagerInfo: Added broadcast_9_piece0 in memory on localhost:41207 (size: 1209.0 B, free: 534.5 MB)
16/03/24 07:46:27 INFO spark.SparkContext: Created broadcast 9 from broadcast at DAGScheduler.scala:861
16/03/24 07:46:27 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ResultStage 8 (MapPartitionsRDD[12] at filter at <console>:25)
16/03/24 07:46:27 INFO scheduler.TaskSchedulerImpl: Adding task set 8.0 with 1 tasks
16/03/24 07:46:27 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 8.0 (TID 8, localhost, partition 0,PROCESS_LOCAL, 2142 bytes)
16/03/24 07:46:27 INFO executor.Executor: Running task 0.0 in stage 8.0 (TID 8)
16/03/24 07:46:27 INFO executor.Executor: Finished task 0.0 in stage 8.0 (TID 8). 10946 bytes result sent to driver
16/03/24 07:46:27 INFO scheduler.DAGScheduler: ResultStage 8 (collect at <console>:28) finished in 0.078 s
16/03/24 07:46:27 INFO scheduler.DAGScheduler: Job 8 finished: collect at <console>:28, took 0.099621 s
16/03/24 07:46:27 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 8.0 (TID 8) in 82 ms on localhost (1/1)
16/03/24 07:46:27 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 8.0, whose tasks have all completed, from pool
res10: Array[Int] = Array(2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84,
86, 88, 90, 92, 94, 96, 98, 100, 102, 104, 106, 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128, 130, 132, 134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 16
4, 166, 168, 170, 172, 174, 176, 178, 180, 182, 184, 186, 188, 190, 192, 194, 196, 198, 200, 202, 204, 206, 208, 210, 212, 214, 216, 218, 220, 222, 224, 226, 228, 230, 232, 234, 236, 238, 24
0, 242, 244, 246, 248, 250, 252, 254, 256, 258, 260, 262, 264, 266, 268, 270, 272, 274, 276, 278, 280, 282, 284, 286, 288, 290, 292, 294, 296, 298, 300, 302, 304, 306, 308, 310, 312, 314, 31
6, 318, 320, 322, 324, 326, 328, 330,...
scala>
```

```
scala> val totDataPar = sc.parallelize(sampleData,2)
totDataPar: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[13] at parallelize at <console>:23

scala> val resultPar = totDataPar.filter(_%2 == 0)
resultPar: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[14] at filter at <console>:25

scala> resultPar.collect()
16/03/24 07:48:41 INFO spark.SparkContext: Starting job: collect at <console>:28
16/03/24 07:48:41 INFO scheduler.DAGScheduler: Got job 9 (collect at <console>:28) with 2 output partitions
16/03/24 07:48:41 INFO scheduler.DAGScheduler: Final stage: ResultStage 9(collect at <console>:28)
16/03/24 07:48:41 INFO scheduler.DAGScheduler: Parents of final stage: List()
16/03/24 07:48:41 INFO scheduler.DAGScheduler: Missing parents: List()
16/03/24 07:48:41 INFO scheduler.DAGScheduler: Submitting ResultStage 9 (MapPartitionsRDD[14] at filter at <console>:25), which has no missing parents
16/03/24 07:48:41 INFO storage.MemoryStore: ensureFreeSpace(1952) called with curMem=116834, maxMem=560497950
16/03/24 07:48:41 INFO storage.MemoryStore: Block broadcast_10 stored as values in memory (estimated size 1952.0 B, free 534.4 MB)
16/03/24 07:48:41 INFO storage.MemoryStore: ensureFreeSpace(1209) called with curMem=118786, maxMem=560497950
16/03/24 07:48:41 INFO storage.MemoryStore: Block broadcast_10_piece0 stored as bytes in memory (estimated size 1209.0 B, free 534.4 MB)
16/03/24 07:48:41 INFO storage.BlockManagerInfo: Added broadcast_10_piece0 in memory on localhost:41207 (size: 1209.0 B, free: 534.5 MB)
16/03/24 07:48:41 INFO spark.SparkContext: Created broadcast 10 from broadcast at DAGScheduler.scala:861
16/03/24 07:48:41 INFO scheduler.DAGScheduler: Submitting 2 missing tasks from ResultStage 9 (MapPartitionsRDD[14] at filter at <console>:25)
16/03/24 07:48:41 INFO scheduler.TaskSchedulerImpl: Adding task set 9.0 with 2 tasks
16/03/24 07:48:41 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 9.0 (TID 9, localhost, partition 0,PROCESS_LOCAL, 2085 bytes)
16/03/24 07:48:41 INFO executor.Executor: Running task 0.0 in stage 9.0 (TID 9)
16/03/24 07:48:41 INFO executor.Executor: Finished task 0.0 in stage 9.0 (TID 9). 5921 bytes result sent to driver
16/03/24 07:48:41 INFO scheduler.TaskSetManager: Starting task 1.0 in stage 9.0 (TID 10, localhost, partition 1,PROCESS_LOCAL, 2142 bytes)
16/03/24 07:48:41 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 9.0 (TID 9) in 14 ms on localhost (1/2)
16/03/24 07:48:41 INFO executor.Executor: Running task 1.0 in stage 9.0 (TID 10)
16/03/24 07:48:41 INFO executor.Executor: Finished task 1.0 in stage 9.0 (TID 10). 5921 bytes result sent to driver
16/03/24 07:48:41 INFO scheduler.DAGScheduler: ResultStage 9 (collect at <console>:28) finished in 0.035 s
16/03/24 07:48:41 INFO scheduler.DAGScheduler: Job 9 finished: collect at <console>:28, took 0.044997 s
16/03/24 07:48:41 INFO scheduler.TaskSetManager: Finished task 1.0 in stage 9.0 (TID 10) in 23 ms on localhost (2/2)
16/03/24 07:48:41 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 9.0, whose tasks have all completed, from pool
res11: Array[Int] = Array(2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84,
86, 88, 90, 92, 94, 96, 98, 100, 102, 104, 106, 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128, 130, 132, 134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 16
4, 166, 168, 170, 172, 174, 176, 178, 180, 182, 184, 186, 188, 190, 192, 194, 196, 198, 200, 202, 204, 206, 208, 210, 212, 214, 216, 218, 220, 222, 224, 226, 228, 230, 232, 234, 236, 238, 24
0, 242, 244, 246, 248, 250, 252, 254, 256, 258, 260, 262, 264, 266, 268, 270, 272, 274, 276, 278, 280, 282, 284, 286, 288, 290, 292, 294, 296, 298, 300, 302, 304, 306, 308, 310, 312, 314, 31
6, 318, 320, 322, 324, 326, 328, 330,...
scala>
```

Example 5: python example of map

```
nums = sc.parallelize([1, 2, 3, 4])

squared = nums.map(lambda x: x * x).collect()

for num in squared: print "%i " % (num)
```

```
>>> nums = sc.parallelize([1, 2, 3, 4])
>>> squared = nums.map(lambda x: x * x).collect()
16/03/25 06:51:20 INFO spark.SparkContext: Starting job: collect at <stdin>:1
16/03/25 06:51:20 INFO scheduler.DAGScheduler: Got job 1 (collect at <stdin>:1) with 1 output partitions
16/03/25 06:51:20 INFO scheduler.DAGScheduler: Final stage: ResultStage 1(collect at <stdin>:1)
16/03/25 06:51:20 INFO scheduler.DAGScheduler: Parents of final stage: List()
16/03/25 06:51:20 INFO scheduler.DAGScheduler: Missing parents: List()
16/03/25 06:51:20 INFO scheduler.DAGScheduler: Submitting ResultStage 1 (PythonRDD[4] at collect at <stdin>:1), which has no missing parents
16/03/25 06:51:20 INFO storage.MemoryStore: ensureFreeSpace(3568) called with curMem=149377, maxMem=560497950
16/03/25 06:51:20 INFO storage.MemoryStore: Block broadcast_2 stored as values in memory (estimated size 3.5 KB, free 534.4 MB)
16/03/25 06:51:20 INFO storage.MemoryStore: ensureFreeSpace(2213) called with curMem=152945, maxMem=560497950
16/03/25 06:51:20 INFO storage.MemoryStore: Block broadcast_2_piece0 stored as bytes in memory (estimated size 2.2 KB, free 534.4 MB)
16/03/25 06:51:20 INFO storage.BlockManagerInfo: Added broadcast_2_piece0 in memory on localhost:40816 (size: 2.2 KB, free: 534.5 MB)
16/03/25 06:51:20 INFO spark.SparkContext: Created broadcast 2 from broadcast at DAGScheduler.scala:861
16/03/25 06:51:20 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ResultStage 1 (PythonRDD[4] at collect at <stdin>:1)
16/03/25 06:51:20 INFO scheduler.TaskSchedulerImpl: Adding task set 1.0 with 1 tasks
16/03/25 06:51:20 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 1.0 (TID 1, localhost, partition 0,PROCESS_LOCAL, 2097 bytes)
16/03/25 06:51:20 INFO executor.Executor: Running task 0.0 in stage 1.0 (TID 1)
16/03/25 06:51:20 INFO python.PythonRDD: Times: total = 12, boot = 7, init = 5, finish = 0
16/03/25 06:51:20 INFO executor.Executor: Finished task 0.0 in stage 1.0 (TID 1). 1036 bytes result sent to driver
16/03/25 06:51:20 INFO scheduler.DAGScheduler: ResultStage 1 (collect at <stdin>:1) finished in 0.074 s
16/03/25 06:51:20 INFO scheduler.DAGScheduler: Job 1 finished: collect at <stdin>:1, took 0.103998 s
>>> 16/03/25 06:51:20 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 1.0 (TID 1) in 74 ms on localhost (1/1)
16/03/25 06:51:20 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 1.0, whose tasks have all completed, from pool

>>> for num in squared: print "%i " % (num)
...
1
4
9
16
>>>
```

Example: Scala example of map

```
val input = sc.parallelize(List(1, 2, 3, 4))

val result = input.map(x => x * x)

println(result.collect().mkString(","))
```

```
scala> val input = sc.parallelize(List(1,2,3,4))
input: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[15] at parallelize at <console>:21

scala> val result = input.map(x => x*x)
result: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[16] at map at <console>:23

scala> println(result.collect().mkString(","))
16/03/24 07:54:59 INFO spark.SparkContext: Starting job: collect at <console>:26
16/03/24 07:54:59 INFO scheduler.DAGScheduler: Got job 10 (collect at <console>:26) with 1 output partitions
16/03/24 07:54:59 INFO scheduler.DAGScheduler: Final stage: ResultStage 10(collect at <console>:26)
16/03/24 07:54:59 INFO scheduler.DAGScheduler: Parents of final stage: List()
16/03/24 07:54:59 INFO scheduler.DAGScheduler: Missing parents: List()
16/03/24 07:54:59 INFO scheduler.DAGScheduler: Submitting ResultStage 10 (MapPartitionsRDD[16] at map at <console>:23), which has no missing parents
16/03/24 07:54:59 INFO storage.MemoryStore: ensureFreeSpace(1952) called with curMem=119995, maxMem=560497950
16/03/24 07:54:59 INFO storage.MemoryStore: Block broadcast_11 stored as values in memory (estimated size 1952.0 B, free 534.4 MB)
16/03/24 07:54:59 INFO storage.MemoryStore: ensureFreeSpace(1212) called with curMem=121947, maxMem=560497950
16/03/24 07:54:59 INFO storage.MemoryStore: Block broadcast_11_piece0 stored as bytes in memory (estimated size 1212.0 B, free 534.4 MB)
16/03/24 07:54:59 INFO storage.BlockManagerInfo: Added broadcast_11_piece0 in memory on localhost:41207 (size: 1212.0 B, free: 534.5 MB)
16/03/24 07:54:59 INFO spark.SparkContext: Created broadcast 11 from broadcast at DAGScheduler.scala:861
16/03/24 07:54:59 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ResultStage 10 (MapPartitionsRDD[16] at map at <console>:23)
16/03/24 07:54:59 INFO scheduler.TaskSchedulerImpl: Adding task set 10.0 with 1 tasks
16/03/24 07:54:59 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 10.0 (TID 11, localhost, partition 0,PROCESS_LOCAL, 2041 bytes)
16/03/24 07:54:59 INFO executor.Executor: Running task 0.0 in stage 10.0 (TID 11)
16/03/24 07:54:59 INFO executor.Executor: Finished task 0.0 in stage 10.0 (TID 11). 914 bytes result sent to driver
16/03/24 07:54:59 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 10.0 (TID 11) in 9 ms on localhost (1/1)
16/03/24 07:54:59 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 10.0, whose tasks have all completed, from pool
16/03/24 07:54:59 INFO scheduler.DAGScheduler: ResultStage 10 (collect at <console>:26) finished in 0.037 s
16/03/24 07:54:59 INFO scheduler.DAGScheduler: Job 10 finished: collect at <console>:26, took 0.049508 s
1,4,9,16
```

Example 6: union

```
val seta = sc.parallelize(1 to 10)
val setb = sc.parallelize(5 to 15)
(seta union setb).collect
```

```
scala> val seta = sc.parallelize(1 to 10)
seta: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[17] at parallelize at <console>:21

scala> val setb = sc.parallelize(5 to 15)
setb: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[18] at parallelize at <console>:21

scala> (seta union setb).collect
16/03/24 07:58:52 INFO spark.SparkContext: Starting job: collect at <console>:26
16/03/24 07:58:52 INFO scheduler.DAGScheduler: Got job 11 (collect at <console>:26) with 2 output partitions
16/03/24 07:58:52 INFO scheduler.DAGScheduler: Final stage: ResultStage 11(collect at <console>:26)
16/03/24 07:58:52 INFO scheduler.DAGScheduler: Parents of final stage: List()
16/03/24 07:58:52 INFO scheduler.DAGScheduler: Missing parents: List()
16/03/24 07:58:52 INFO scheduler.DAGScheduler: Submitting ResultStage 11 (UnionRDD[19] at union at <console>:26), which has no missing parents
16/03/24 07:58:52 INFO storage.MemoryStore: ensureFreeSpace(1984) called with curMem=123159, maxMem=560497950
16/03/24 07:58:52 INFO storage.MemoryStore: Block broadcast_12 stored as values in memory (estimated size 1984.0 B, free 534.4 MB)
16/03/24 07:58:52 INFO storage.MemoryStore: ensureFreeSpace(1306) called with curMem=125143, maxMem=560497950
16/03/24 07:58:52 INFO storage.MemoryStore: Block broadcast_12_piece0 stored as bytes in memory (estimated size 1306.0 B, free 534.4 MB)
16/03/24 07:58:52 INFO storage.BlockManagerInfo: Added broadcast_12_piece0 in memory on localhost:41207 (size: 1306.0 B, free: 534.5 MB)
16/03/24 07:58:52 INFO spark.SparkContext: Created broadcast 12 from broadcast at DAGScheduler.scala:861
16/03/24 07:58:52 INFO scheduler.DAGScheduler: Submitting 2 missing tasks from ResultStage 11 (UnionRDD[19] at union at <console>:26)
16/03/24 07:58:52 INFO scheduler.TaskSchedulerImpl: Adding task set 11.0 with 2 tasks
16/03/24 07:58:52 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 11.0 (TID 12, localhost, partition 0,PROCESS_LOCAL, 2251 bytes)
16/03/24 07:58:52 INFO executor.Executor: Running task 0.0 in stage 11.0 (TID 12)
16/03/24 07:58:52 INFO executor.Executor: Finished task 0.0 in stage 11.0 (TID 12). 938 bytes result sent to driver
16/03/24 07:58:52 INFO scheduler.TaskSetManager: Starting task 1.0 in stage 11.0 (TID 13, localhost, partition 1,PROCESS_LOCAL, 2251 bytes)
16/03/24 07:58:52 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 11.0 (TID 12) in 11 ms on localhost (1/2)
16/03/24 07:58:52 INFO executor.Executor: Running task 1.0 in stage 11.0 (TID 13)
16/03/24 07:58:52 INFO executor.Executor: Finished task 1.0 in stage 11.0 (TID 13). 942 bytes result sent to driver
16/03/24 07:58:52 INFO scheduler.TaskSetManager: Finished task 1.0 in stage 11.0 (TID 13) in 8 ms on localhost (2/2)
16/03/24 07:58:52 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 11.0, whose tasks have all completed, from pool
16/03/24 07:58:52 INFO scheduler.DAGScheduler: ResultStage 11 (collect at <console>:26) finished in 0.015 s
16/03/24 07:58:52 INFO scheduler.DAGScheduler: Job 11 finished: collect at <console>:26, took 0.057182 s
res13: Array[Int] = Array(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15)
```

Example 7: flatMap() using scala shell

Splitting lines into multiple words

```
val lines = sc.parallelize(List("hello world", "hi"))
val words = lines.flatMap(line => line.split(" "))
words.first()  // returns "hello"
```

```
scala> val lines = sc.parallelize(List("hello world", "hi"))
lines: org.apache.spark.rdd.RDD[String] = ParallelCollectionRDD[20] at parallelize at <console>:21

scala> val words = lines.flatMap(line => line.split(" "))
words: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[21] at flatMap at <console>:23

scala> words.first()
16/03/24 08:00:51 INFO spark.SparkContext: Starting job: first at <console>:26
16/03/24 08:00:51 INFO scheduler.DAGScheduler: Got job 12 (first at <console>:26) with 1 output partitions
16/03/24 08:00:51 INFO scheduler.DAGScheduler: Final stage: ResultStage 12(first at <console>:26)
16/03/24 08:00:51 INFO scheduler.DAGScheduler: Parents of final stage: List()
16/03/24 08:00:51 INFO scheduler.DAGScheduler: Missing parents: List()
16/03/24 08:00:51 INFO scheduler.DAGScheduler: Submitting ResultStage 12 (MapPartitionsRDD[21] at flatMap at <console>:23), which has no missing parents
16/03/24 08:00:51 INFO storage.MemoryStore: ensureFreeSpace(1944) called with curMem=126449, maxMem=560497950
16/03/24 08:00:51 INFO storage.MemoryStore: Block broadcast_13 stored as values in memory (estimated size 1944.0 B, free 534.4 MB)
16/03/24 08:00:51 INFO storage.MemoryStore: ensureFreeSpace(1206) called with curMem=128393, maxMem=560497950
16/03/24 08:00:51 INFO storage.MemoryStore: Block broadcast_13_piece0 stored as bytes in memory (estimated size 1206.0 B, free 534.4 MB)
16/03/24 08:00:51 INFO storage.BlockManagerInfo: Added broadcast_13_piece0 in memory on localhost:41207 (size: 1206.0 B, free: 534.5 MB)
16/03/24 08:00:51 INFO spark.SparkContext: Created broadcast 13 from broadcast at DAGScheduler.scala:861
16/03/24 08:00:51 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ResultStage 12 (MapPartitionsRDD[21] at flatMap at <console>:23)
16/03/24 08:00:51 INFO scheduler.TaskSchedulerImpl: Adding task set 12.0 with 1 tasks
16/03/24 08:00:51 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 12.0 (TID 14, localhost, partition 0,PROCESS_LOCAL, 2106 bytes)
16/03/24 08:00:51 INFO executor.Executor: Running task 0.0 in stage 12.0 (TID 14)
16/03/24 08:00:51 INFO executor.Executor: Finished task 0.0 in stage 12.0 (TID 14). 923 bytes result sent to driver
16/03/24 08:00:51 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 12.0 (TID 14) in 45 ms on localhost (1/1)
16/03/24 08:00:51 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 12.0, whose tasks have all completed, from pool
16/03/24 08:00:51 INFO scheduler.DAGScheduler: ResultStage 12 (first at <console>:26) finished in 0.045 s
16/03/24 08:00:51 INFO scheduler.DAGScheduler: Job 12 finished: first at <console>:26, took 0.084712 s
res14: String = hello
```

Example: flatMap() using python shell

```
lines = sc.parallelize(["hello world", "hi"])
words = lines.flatMap(lambda line: line.split(" "))
words.first()  # returns "hello"
```

```
>>> lines = sc.parallelize(["hello world", "hi"])
>>> words = lines.flatMap(lambda line: line.split(" "))
>>> words.first()
16/03/25 06:54:22 INFO spark.SparkContext: Starting job: runJob at PythonRDD.scala:361
16/03/25 06:54:22 INFO scheduler.DAGScheduler: Got job 2 (runJob at PythonRDD.scala:361) with 1 output partitions
16/03/25 06:54:22 INFO scheduler.DAGScheduler: Final stage: ResultStage 2(runJob at PythonRDD.scala:361)
16/03/25 06:54:22 INFO scheduler.DAGScheduler: Parents of final stage: List()
16/03/25 06:54:22 INFO scheduler.DAGScheduler: Missing parents: List()
16/03/25 06:54:22 INFO scheduler.DAGScheduler: Submitting ResultStage 2 (PythonRDD[6] at RDD at PythonRDD.scala:43), which has no missing parents
16/03/25 06:54:22 INFO storage.MemoryStore: ensureFreeSpace(4048) called with curMem=155158, maxMem=560497950
16/03/25 06:54:22 INFO storage.MemoryStore: Block broadcast_3 stored as values in memory (estimated size 4.0 KB, free 534.4 MB)
16/03/25 06:54:22 INFO storage.MemoryStore: ensureFreeSpace(2633) called with curMem=159206, maxMem=560497950
16/03/25 06:54:22 INFO storage.MemoryStore: Block broadcast_3_piece0 stored as bytes in memory (estimated size 2.6 KB, free 534.4 MB)
16/03/25 06:54:22 INFO storage.BlockManagerInfo: Added broadcast_3_piece0 in memory on localhost:40816 (size: 2.6 KB, free: 534.5 MB)
16/03/25 06:54:22 INFO spark.SparkContext: Created broadcast 3 from broadcast at DAGScheduler.scala:861
16/03/25 06:54:22 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ResultStage 2 (PythonRDD[6] at RDD at PythonRDD.scala:43)
16/03/25 06:54:22 INFO scheduler.TaskSchedulerImpl: Adding task set 2.0 with 1 tasks
16/03/25 06:54:22 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 2.0 (TID 2, localhost, partition 0,PROCESS_LOCAL, 2110 bytes)
16/03/25 06:54:22 INFO executor.Executor: Running task 0.0 in stage 2.0 (TID 2)
16/03/25 06:54:22 INFO python.PythonRDD: Times: total = 16, boot = 8, init = 8, finish = 0
16/03/25 06:54:22 INFO executor.Executor: Finished task 0.0 in stage 2.0 (TID 2). 1002 bytes result sent to driver
16/03/25 06:54:22 INFO scheduler.DAGScheduler: ResultStage 2 (runJob at PythonRDD.scala:361) finished in 0.049 s
16/03/25 06:54:22 INFO scheduler.DAGScheduler: Job 2 finished: runJob at PythonRDD.scala:361, took 0.087697 s
16/03/25 06:54:22 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 2.0 (TID 2) in 56 ms on localhost (1/1)
16/03/25 06:54:22 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 2.0, whose tasks have all completed, from pool
'hello'
>>>
```

### 3. Paired RDD

- Spark provides special operations on RDDs containing key/value pairs. These RDDs are called pair RDD.
- Pair RDD allows you to act on each key in parallel or regroup data across the network.

Example 8: foldByKey

```
val a = sc.parallelize(List("kim","kumar","muthu","tim","lak","vamsi"),2)
val b = a.map(x => (x.length,x))

b.collect
b.foldByKey("")(_+_).collect
```

```
scala> val a = sc.parallelize(List("kim","kumar","muthu","tim","lak","vamsi"),2)
16/03/24 08:02:22 INFO spark.ContextCleaner: Cleaned accumulator 9
16/03/24 08:02:22 INFO storage.BlockManagerInfo: Removed broadcast_9_piece0 on localhost:41207 in memory (size: 1209.0 B, free: 534.5 MB)
a: org.apache.spark.rdd.RDD[String] = ParallelCollectionRDD[22] at parallelize at <console>:21
```

```
scala> val b = a.map(x => (x.length,x))
b: org.apache.spark.rdd.RDD[(Int, String)] = MapPartitionsRDD[23] at map at <console>:23
```

```
scala> b.collect
16/03/24 08:03:16 INFO spark.SparkContext: Starting job: collect at <console>:26
16/03/24 08:03:16 INFO scheduler.DAGScheduler: Got job 13 (collect at <console>:26) with 2 output partitions
16/03/24 08:03:16 INFO scheduler.DAGScheduler: Final stage: ResultStage 13(collect at <console>:26)
16/03/24 08:03:16 INFO scheduler.DAGScheduler: Parents of final stage: List()
16/03/24 08:03:16 INFO scheduler.DAGScheduler: Missing parents: List()
16/03/24 08:03:16 INFO scheduler.DAGScheduler: Submitting ResultStage 13 (MapPartitionsRDD[23] at map at <console>:23), which has no missing parents
16/03/24 08:03:16 INFO storage.MemoryStore: ensureFreeSpace(1928) called with curMem=113673, maxMem=560497950
16/03/24 08:03:16 INFO storage.MemoryStore: Block broadcast_14 stored as values in memory (estimated size 1928.0 B, free 534.4 MB)
16/03/24 08:03:16 INFO storage.MemoryStore: ensureFreeSpace(1185) called with curMem=115601, maxMem=560497950
16/03/24 08:03:16 INFO storage.MemoryStore: Block broadcast_14_piece0 stored as bytes in memory (estimated size 1185.0 B, free 534.4 MB)
16/03/24 08:03:16 INFO storage.BlockManagerInfo: Added broadcast_14_piece0 in memory on localhost:41207 (size: 1185.0 B, free: 534.5 MB)
16/03/24 08:03:16 INFO spark.SparkContext: Created broadcast 14 from broadcast at DAGScheduler.scala:861
16/03/24 08:03:16 INFO scheduler.DAGScheduler: Submitting 2 missing tasks from ResultStage 13 (MapPartitionsRDD[23] at map at <console>:23)
16/03/24 08:03:16 INFO scheduler.TaskSchedulerImpl: Adding task set 13.0 with 2 tasks
16/03/24 08:03:16 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 13.0 (TID 15, localhost, partition 0,PROCESS_LOCAL, 2109 bytes)
16/03/24 08:03:16 INFO executor.Executor: Running task 0.0 in stage 13.0 (TID 15)
16/03/24 08:03:16 INFO executor.Executor: Finished task 0.0 in stage 13.0 (TID 15). 1102 bytes result sent to driver
16/03/24 08:03:16 INFO scheduler.TaskSetManager: Starting task 1.0 in stage 13.0 (TID 16, localhost, partition 1,PROCESS_LOCAL, 2107 bytes)
16/03/24 08:03:16 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 13.0 (TID 15) in 10 ms on localhost (1/2)
16/03/24 08:03:16 INFO executor.Executor: Running task 1.0 in stage 13.0 (TID 16)
16/03/24 08:03:16 INFO executor.Executor: Finished task 1.0 in stage 13.0 (TID 16). 1100 bytes result sent to driver
16/03/24 08:03:16 INFO scheduler.TaskSetManager: Finished task 1.0 in stage 13.0 (TID 16) in 9 ms on localhost (2/2)
16/03/24 08:03:16 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 13.0, whose tasks have all completed, from pool
16/03/24 08:03:16 INFO scheduler.DAGScheduler: ResultStage 13 (collect at <console>:26) finished in 0.017 s
16/03/24 08:03:16 INFO scheduler.DAGScheduler: Job 13 finished: collect at <console>:26, took 0.025213 s
res15: Array[(Int, String)] = Array((3,kim), (5,kumar), (5,muthu), (3,tim), (3,lak), (5,vamsi))
```

```
scala> b.foldByKey("")(_+_).collect
16/03/24 08:03:36 INFO spark.SparkContext: Starting job: collect at <console>:26
16/03/24 08:03:36 INFO scheduler.DAGScheduler: Registering RDD 23 (map at <console>:23)
16/03/24 08:03:36 INFO scheduler.DAGScheduler: Got job 14 (collect at <console>:26) with 2 output partitions
16/03/24 08:03:36 INFO scheduler.DAGScheduler: Final stage: ResultStage 15(collect at <console>:26)
16/03/24 08:03:36 INFO scheduler.DAGScheduler: Parents of final stage: List(ShuffleMapStage 14)
16/03/24 08:03:36 INFO scheduler.DAGScheduler: Missing parents: List(ShuffleMapStage 14)
16/03/24 08:03:36 INFO scheduler.DAGScheduler: Submitting ShuffleMapStage 14 (MapPartitionsRDD[23] at map at <console>:23), which has no missing parents
16/03/24 08:03:36 INFO storage.MemoryStore: ensureFreeSpace(3736) called with curMem=116786, maxMem=560497950
16/03/24 08:03:36 INFO storage.MemoryStore: Block broadcast_15 stored as values in memory (estimated size 3.6 KB, free 534.4 MB)
16/03/24 08:03:36 INFO storage.MemoryStore: ensureFreeSpace(2102) called with curMem=120522, maxMem=560497950
16/03/24 08:03:36 INFO storage.MemoryStore: Block broadcast_15_piece0 stored as bytes in memory (estimated size 2.1 KB, free 534.4 MB)
16/03/24 08:03:36 INFO storage.BlockManagerInfo: Added broadcast_15_piece0 in memory on localhost:41207 (size: 2.1 KB, free: 534.5 MB)
16/03/24 08:03:36 INFO spark.SparkContext: Created broadcast 15 from broadcast at DAGScheduler.scala:861
16/03/24 08:03:36 INFO scheduler.DAGScheduler: Submitting 2 missing tasks from ShuffleMapStage 14 (MapPartitionsRDD[23] at map at <console>:23)
16/03/24 08:03:36 INFO scheduler.TaskSchedulerImpl: Adding task set 14.0 with 2 tasks
16/03/24 08:03:36 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 14.0 (TID 17, localhost, partition 0,PROCESS_LOCAL, 2098 bytes)
16/03/24 08:03:36 INFO executor.Executor: Running task 0.0 in stage 14.0 (TID 17)
16/03/24 08:03:36 INFO executor.Executor: Finished task 0.0 in stage 14.0 (TID 17). 1159 bytes result sent to driver
16/03/24 08:03:36 INFO scheduler.TaskSetManager: Starting task 1.0 in stage 14.0 (TID 18, localhost, partition 1,PROCESS_LOCAL, 2096 bytes)
16/03/24 08:03:36 INFO executor.Executor: Running task 1.0 in stage 14.0 (TID 18)
16/03/24 08:03:36 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 14.0 (TID 17) in 191 ms on localhost (1/2)
16/03/24 08:03:36 INFO executor.Executor: Finished task 1.0 in stage 14.0 (TID 18). 1159 bytes result sent to driver
16/03/24 08:03:36 INFO scheduler.TaskSetManager: Finished task 1.0 in stage 14.0 (TID 18) in 19 ms on localhost (2/2)
16/03/24 08:03:36 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 14.0, whose tasks have all completed, from pool
16/03/24 08:03:36 INFO scheduler.DAGScheduler: ShuffleMapStage 14 (map at <console>:23) finished in 0.199 s
16/03/24 08:03:36 INFO scheduler.DAGScheduler: looking for newly runnable stages
16/03/24 08:03:36 INFO scheduler.DAGScheduler: running: Set()
16/03/24 08:03:36 INFO scheduler.DAGScheduler: waiting: Set(ResultStage 15)
16/03/24 08:03:36 INFO scheduler.DAGScheduler: failed: Set()
16/03/24 08:03:36 INFO scheduler.DAGScheduler: Missing parents for ResultStage 15: List()
16/03/24 08:03:36 INFO scheduler.DAGScheduler: Submitting ResultStage 15 (ShuffledRDD[24] at foldByKey at <console>:26), which is now runnable
16/03/24 08:03:36 INFO storage.MemoryStore: ensureFreeSpace(4264) called with curMem=122624, maxMem=560497950
16/03/24 08:03:36 INFO storage.MemoryStore: Block broadcast_16 stored as values in memory (estimated size 4.2 KB, free 534.4 MB)
16/03/24 08:03:36 INFO storage.MemoryStore: ensureFreeSpace(2302) called with curMem=126888, maxMem=560497950
16/03/24 08:03:36 INFO storage.MemoryStore: Block broadcast_16_piece0 stored as bytes in memory (estimated size 2.2 KB, free 534.4 MB)
16/03/24 08:03:36 INFO storage.BlockManagerInfo: Added broadcast_16_piece0 in memory on localhost:41207 (size: 2.2 KB, free: 534.5 MB)
16/03/24 08:03:36 INFO spark.SparkContext: Created broadcast 16 from broadcast at DAGScheduler.scala:861
16/03/24 08:03:36 INFO scheduler.DAGScheduler: Submitting 2 missing tasks from ResultStage 15 (ShuffledRDD[24] at foldByKey at <console>:26)
16/03/24 08:03:36 INFO scheduler.TaskSchedulerImpl: Adding task set 15.0 with 2 tasks
16/03/24 08:03:36 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 15.0 (TID 19, localhost, partition 0,PROCESS_LOCAL, 1901 bytes)
16/03/24 08:03:36 INFO executor.Executor: Running task 0.0 in stage 15.0 (TID 19)
16/03/24 08:03:36 INFO storage.ShuffleBlockFetcherIterator: Getting 0 non-empty blocks out of 2 blocks
```

```
16/03/24 08:03:36 INFO storage.ShuffleBlockFetcherIterator: Started 0 remote fetches in 10 ms
16/03/24 08:03:36 INFO executor.Executor: Finished task 0.0 in stage 15.0 (TID 19). 1161 bytes result sent to driver
16/03/24 08:03:36 INFO scheduler.TaskSetManager: Starting task 1.0 in stage 15.0 (TID 20, localhost, partition 1,PROCESS_LOCAL, 1901 bytes)
16/03/24 08:03:36 INFO executor.Executor: Running task 1.0 in stage 15.0 (TID 20)
16/03/24 08:03:36 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 15.0 (TID 19) in 90 ms on localhost (1/2)
16/03/24 08:03:36 INFO storage.ShuffleBlockFetcherIterator: Getting 2 non-empty blocks out of 2 blocks
16/03/24 08:03:36 INFO storage.ShuffleBlockFetcherIterator: Started 0 remote fetches in 0 ms
16/03/24 08:03:37 INFO executor.Executor: Finished task 1.0 in stage 15.0 (TID 20). 1349 bytes result sent to driver
16/03/24 08:03:37 INFO scheduler.DAGScheduler: ResultStage 15 (collect at <console>:26) finished in 0.241 s
16/03/24 08:03:37 INFO scheduler.DAGScheduler: Job 14 finished: collect at <console>:26, took 0.588536 s
res16: Array[(Int, String)] = Array((3,kimtimlak), (5,kumarmuthuvamsi))
```

Example 9: foldByKey

```
val deptEmployees =
  List
  (
   ("dept1",("kumar1",1000.0)),
   ("dept1",("kumar2",1200.0)),
   ("dept2",("kumar3",2200.0)),
   ("dept2",("kumar4",1400.0)),
   ("dept2",("kumar5",1000.0)),
   ("dept2",("kumar6",800.0)),
   ("dept1",("kumar7",2000.0)),
   ("dept1",("kumar8",1000.0)),
```

```
  ("dept1",("kumar9",500.0))
  )


val employeeRDD = sc.makeRDD(deptEmployees)
val maxByDept = employeeRDD.foldByKey(("dummy",Double.MinValue))((acc,element)

=> if(acc._2 > element._2)acc else element)

println("Maximum salaries in each dept" + maxByDept.collect().toList)
```

```
deptEmployees: List[(String, (String, Double))] = List((dept1,(kumar1,1000.0)), (dept1,(kumar2,1200.0)), (dept2,(kumar3,2200.0)), (dept2,(kumar4,1400.0)), (dept2,(kumar5,1000.0)), (dept2,(ku
mar6,800.0)), (dept1,(kumar7,2000.0)), (dept1,(kumar8,1000.0)), (dept1,(kumar9,500.0)))

scala> val employeeRDD = sc.makeRDD(deptEmployees)
employeeRDD: org.apache.spark.rdd.RDD[(String, (String, Double))] = ParallelCollectionRDD[25] at makeRDD at <console>:23

scala> val maxByDept = employeeRDD.foldByKey(("dummy",Double.MinValue))((acc,element)=> if(acc._2 >element._2)acc else element)
maxByDept: org.apache.spark.rdd.RDD[(String, (String, Double))] = ShuffledRDD[26] at foldByKey at <console>:25
```

```
scala> println("Maximum salaries in eachdept" + maxByDept.collect().toList)
16/03/24 08:21:34 INFO spark.SparkContext: Starting job: collect at <console>:28
16/03/24 08:21:34 INFO scheduler.DAGScheduler: Registering RDD 25 (makeRDD at <console>:23)
16/03/24 08:21:34 INFO scheduler.DAGScheduler: Got job 15 (collect at <console>:28) with 1 output partitions
16/03/24 08:21:34 INFO scheduler.DAGScheduler: Final stage: ResultStage 17(collect at <console>:28)
16/03/24 08:21:34 INFO scheduler.DAGScheduler: Parents of final stage: List(ShuffleMapStage 16)
16/03/24 08:21:34 INFO scheduler.DAGScheduler: Missing parents: List(ShuffleMapStage 16)
16/03/24 08:21:34 INFO scheduler.DAGScheduler: Submitting ShuffleMapStage 16 (ParallelCollectionRDD[25] at makeRDD at <console>:23), which has no missing parents
16/03/24 08:21:34 INFO storage.MemoryStore: ensureFreeSpace(3192) called with curMem=113673, maxMem=560497950
16/03/24 08:21:34 INFO storage.MemoryStore: Block broadcast_17 stored as values in memory (estimated size 3.1 KB, free 534.4 MB)
16/03/24 08:21:34 INFO storage.MemoryStore: ensureFreeSpace(1743) called with curMem=116865, maxMem=560497950
16/03/24 08:21:34 INFO storage.MemoryStore: Block broadcast_17_piece0 stored as bytes in memory (estimated size 1743.0 B, free 534.4 MB)
16/03/24 08:21:34 INFO storage.BlockManagerInfo: Added broadcast_17_piece0 in memory on localhost:41207 (size: 1743.0 B, free: 534.5 MB)
16/03/24 08:21:34 INFO spark.SparkContext: Created broadcast 17 from broadcast at DAGScheduler.scala:861
16/03/24 08:21:34 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ShuffleMapStage 16 (ParallelCollectionRDD[25] at makeRDD at <console>:23)
16/03/24 08:21:34 INFO scheduler.TaskSchedulerImpl: Adding task set 16.0 with 1 tasks
16/03/24 08:21:34 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 16.0 (TID 21, localhost, partition 0,PROCESS_LOCAL, 2520 bytes)
16/03/24 08:21:34 INFO executor.Executor: Running task 0.0 in stage 16.0 (TID 21)
16/03/24 08:21:35 INFO executor.Executor: Finished task 0.0 in stage 16.0 (TID 21). 1158 bytes result sent to driver
16/03/24 08:21:35 INFO scheduler.DAGScheduler: ShuffleMapStage 16 (makeRDD at <console>:23) finished in 0.257 s
16/03/24 08:21:35 INFO scheduler.DAGScheduler: looking for newly runnable stages
16/03/24 08:21:35 INFO scheduler.DAGScheduler: running: Set()
16/03/24 08:21:35 INFO scheduler.DAGScheduler: waiting: Set(ResultStage 17)
16/03/24 08:21:35 INFO scheduler.DAGScheduler: failed: Set()
16/03/24 08:21:35 INFO scheduler.DAGScheduler: Missing parents for ResultStage 17: List()
16/03/24 08:21:35 INFO scheduler.DAGScheduler: Submitting ResultStage 17 (ShuffledRDD[26] at foldByKey at <console>:25), which is now runnable
16/03/24 08:21:35 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 16.0 (TID 21) in 255 ms on localhost (1/1)
16/03/24 08:21:35 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 16.0, whose tasks have all completed, from pool
16/03/24 08:21:35 INFO storage.MemoryStore: ensureFreeSpace(3920) called with curMem=118608, maxMem=560497950
16/03/24 08:21:35 INFO storage.MemoryStore: Block broadcast_18 stored as values in memory (estimated size 3.8 KB, free 534.4 MB)
16/03/24 08:21:35 INFO storage.MemoryStore: ensureFreeSpace(2056) called with curMem=122528, maxMem=560497950
16/03/24 08:21:35 INFO storage.MemoryStore: Block broadcast_18_piece0 stored as bytes in memory (estimated size 2.0 KB, free 534.4 MB)
16/03/24 08:21:35 INFO storage.BlockManagerInfo: Added broadcast_18_piece0 in memory on localhost:41207 (size: 2.0 KB, free: 534.5 MB)
16/03/24 08:21:35 INFO spark.SparkContext: Created broadcast 18 from broadcast at DAGScheduler.scala:861
16/03/24 08:21:35 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ResultStage 17 (ShuffledRDD[26] at foldByKey at <console>:25)
16/03/24 08:21:35 INFO scheduler.TaskSchedulerImpl: Adding task set 17.0 with 1 tasks
16/03/24 08:21:35 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 17.0 (TID 22, localhost, partition 0,PROCESS_LOCAL, 1901 bytes)
16/03/24 08:21:35 INFO executor.Executor: Running task 0.0 in stage 17.0 (TID 22)
16/03/24 08:21:35 INFO storage.ShuffleBlockFetcherIterator: Getting 1 non-empty blocks out of 1 blocks
16/03/24 08:21:35 INFO storage.ShuffleBlockFetcherIterator: Started 0 remote fetches in 0 ms
16/03/24 08:21:35 INFO executor.Executor: Finished task 0.0 in stage 17.0 (TID 22). 1375 bytes result sent to driver
16/03/24 08:21:35 INFO scheduler.DAGScheduler: ResultStage 17 (collect at <console>:28) finished in 0.046 s
```

```
16/03/24 08:21:35 INFO scheduler.DAGScheduler: Job 15 finished: collect at <console>:28, took 0.494997 s
Maximum salaries in eachdeptList((dept2,(kumar3,2200.0)), (dept1,(kumar7,2000.0)))
```

Example 10: reduceByKey

```
val textFile = sc.textFile("file:/home/technocrafty/Datasets/Joyce.txt ")

val counts = textFile.flatMap(line => line.split(" "))
counts.collect
val counts = textFile.flatMap(line => line.split(" ")).map(word => (word,1))
counts.collect
val counts = textFile.flatMap(line => line.split(" ")).map(word =>
(word,1)).reduceByKey(_+_)
counts.collect.foreach(println)
```

```
scala> textFile
res18: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[7] at textFile at <console>:21

scala> val counts = textFile.flatMap(line => line.split(" "))
counts: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[27] at flatMap at <console>:23
```

```
scala> counts.collect
16/03/24 08:38:10 INFO spark.SparkContext: Starting job: collect at <console>:26
16/03/24 08:38:10 INFO scheduler.DAGScheduler: Got job 16 (collect at <console>:26) with 1 output partitions
16/03/24 08:38:10 INFO scheduler.DAGScheduler: Final stage: ResultStage 18(collect at <console>:26)
16/03/24 08:38:10 INFO scheduler.DAGScheduler: Parents of final stage: List()
16/03/24 08:38:10 INFO scheduler.DAGScheduler: Missing parents: List()
16/03/24 08:38:10 INFO scheduler.DAGScheduler: Submitting ResultStage 18 (MapPartitionsRDD[27] at flatMap at <console>:23), which has no missing parents
16/03/24 08:38:10 INFO storage.MemoryStore: ensureFreeSpace(3384) called with curMem=124584, maxMem=560497950
16/03/24 08:38:10 INFO storage.MemoryStore: Block broadcast_19 stored as values in memory (estimated size 3.3 KB, free 534.4 MB)
16/03/24 08:38:10 INFO storage.MemoryStore: ensureFreeSpace(1928) called with curMem=127968, maxMem=560497950
16/03/24 08:38:10 INFO storage.MemoryStore: Block broadcast_19_piece0 stored as bytes in memory (estimated size 1928.0 B, free 534.4 MB)
16/03/24 08:38:10 INFO storage.BlockManagerInfo: Added broadcast_19_piece0 in memory on localhost:41207 (size: 1928.0 B, free: 534.5 MB)
16/03/24 08:38:10 INFO spark.SparkContext: Created broadcast 19 from broadcast at DAGScheduler.scala:861
16/03/24 08:38:10 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ResultStage 18 (MapPartitionsRDD[27] at flatMap at <console>:23)
16/03/24 08:38:10 INFO scheduler.TaskSchedulerImpl: Adding task set 18.0 with 1 tasks
16/03/24 08:38:10 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 18.0 (TID 23, localhost, partition 0,PROCESS_LOCAL, 2151 bytes)
16/03/24 08:38:10 INFO executor.Executor: Running task 0.0 in stage 18.0 (TID 23)
16/03/24 08:38:10 INFO rdd.HadoopRDD: Input split: file:/home/technocrafty/Datasets/Joyce.txt:0+1573079
16/03/24 08:38:10 INFO executor.Executor: Finished task 0.0 in stage 18.0 (TID 23). 2108494 bytes result sent to driver
16/03/24 08:38:10 INFO spark.ContextCleaner: Cleaned accumulator 17
16/03/24 08:38:10 INFO scheduler.DAGScheduler: ResultStage 18 (collect at <console>:26) finished in 0.845 s
16/03/24 08:38:10 INFO scheduler.DAGScheduler: Job 16 finished: collect at <console>:26, took 0.914905 s
16/03/24 08:38:10 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 18.0 (TID 23) in 848 ms on localhost (1/1)
16/03/24 08:38:10 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 18.0, whose tasks have all completed, from pool
16/03/24 08:38:11 INFO storage.BlockManagerInfo: Removed broadcast_17_piece0 on localhost:41207 in memory (size: 1743.0 B, free: 534.5 MB)
16/03/24 08:38:11 INFO spark.ContextCleaner: Cleaned accumulator 18
16/03/24 08:38:11 INFO storage.BlockManagerInfo: Removed broadcast_18_piece0 on localhost:41207 in memory (size: 2.0 KB, free: 534.5 MB)
res19: Array[String] = Array(The, Project, Gutenberg, EBook, of, Ulysses,, by, James, Joyce, "", This, eBook, is, for, the, use, of, anyone, anywhere, at, no, cost, and, with, almost, no, re
strictions, whatsoever., "", You, may, copy, it,, give, it, away, or, re-use, it, under, the, terms, of, the, Project, Gutenberg, License, included, with, this, eBook, or, online, at, www.gu
tenberg.org, "", "", Title:, Ulysses, "", Author:, James, Joyce, "", Posting, Date:, August, 1,, 2008, [EBook, #4300], Release, Date:, July,, 2003, [Last, updated:, November, 17,, 2011], "",
 Language:, English, "", Character, set, encoding:, ASCII, "", ***, START, OF, THIS, PROJECT, GUTENBERG, EBOOK, ULYSSES, ***, "", "", "", "", Produced, by, Col, Choat, "", "", "", "", "", UL
YSSES, "", by, James, Joyce, "", "", ...
```

```
scala> val counts = textFile.flatMap(line => line.split(" ")).map(word16/03/24 08:38:42 INFO storage.BlockManagerInfo: Removed broadcast_19_piece0 on localhost:41207 in memory (size: 1928.0
B, free: 534.5 MB)
16/03/24 08:38:42 INFO spark.ContextCleaner: Cleaned accumulator 19
<console>:1: error: unclosed string literal
"))
^
 => (word,1))
counts: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[29] at map at <console>:23
```

```
scala> counts.collect
16/03/24 08:39:19 INFO spark.SparkContext: Starting job: collect at <console>:26
16/03/24 08:39:19 INFO scheduler.DAGScheduler: Got job 17 (collect at <console>:26) with 1 output partitions
16/03/24 08:39:19 INFO scheduler.DAGScheduler: Final stage: ResultStage 19(collect at <console>:26)
16/03/24 08:39:19 INFO scheduler.DAGScheduler: Parents of final stage: List()
16/03/24 08:39:19 INFO scheduler.DAGScheduler: Missing parents: List()
16/03/24 08:39:19 INFO scheduler.DAGScheduler: Submitting ResultStage 19 (MapPartitionsRDD[29] at map at <console>:23), which has no missing parents
16/03/24 08:39:19 INFO storage.MemoryStore: ensureFreeSpace(3536) called with curMem=113673, maxMem=560497950
16/03/24 08:39:19 INFO storage.MemoryStore: Block broadcast_20 stored as values in memory (estimated size 3.5 KB, free 534.4 MB)
16/03/24 08:39:19 INFO storage.MemoryStore: ensureFreeSpace(1965) called with curMem=117209, maxMem=560497950
16/03/24 08:39:19 INFO storage.MemoryStore: Block broadcast_20_piece0 stored as bytes in memory (estimated size 1965.0 B, free 534.4 MB)
16/03/24 08:39:19 INFO storage.BlockManagerInfo: Added broadcast_20_piece0 in memory on localhost:41207 (size: 1965.0 B, free: 534.5 MB)
16/03/24 08:39:19 INFO spark.SparkContext: Created broadcast 20 from broadcast at DAGScheduler.scala:861
16/03/24 08:39:19 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ResultStage 19 (MapPartitionsRDD[29] at map at <console>:23)
16/03/24 08:39:19 INFO scheduler.TaskSchedulerImpl: Adding task set 19.0 with 1 tasks
16/03/24 08:39:19 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 19.0 (TID 24, localhost, partition 0,PROCESS_LOCAL, 2151 bytes)
16/03/24 08:39:19 INFO executor.Executor: Running task 0.0 in stage 19.0 (TID 24)
16/03/24 08:39:19 INFO rdd.HadoopRDD: Input split: file:/home/technocrafty/Datasets/Joyce.txt:0+1573079
16/03/24 08:39:19 INFO executor.Executor: Finished task 0.0 in stage 19.0 (TID 24). 5182615 bytes result sent to driver
16/03/24 08:39:20 INFO scheduler.DAGScheduler: ResultStage 19 (collect at <console>:26) finished in 0.883 s
16/03/24 08:39:20 INFO scheduler.DAGScheduler: Job 17 finished: collect at <console>:26, took 1.069404 s
16/03/24 08:39:20 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 19.0 (TID 24) in 887 ms on localhost (1/1)
16/03/24 08:39:20 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 19.0, whose tasks have all completed, from pool
res20: Array[(String, Int)] = Array((The,1), (Project,1), (Gutenberg,1), (EBook,1), (of,1), (Ulysses,,1), (by,1), (James,1), (Joyce,1), ("",1), (This,1), (eBook,1), (is,1), (for,1), (the,1),
 (use,1), (of,1), (anyone,1), (anywhere,1), (at,1), (no,1), (cost,1), (and,1), (with,1), (almost,1), (no,1), (restrictions,1), (whatsoever.,1), ("",1), (You,1), (may,1), (copy,1), (it,,1), (
give,1), (it,1), (away,1), (or,1), (re-use,1), (it,1), (under,1), (the,1), (terms,1), (of,1), (the,1), (Project,1), (Gutenberg,1), (License,1), (included,1), (with,1), (this,1), (eBook,1), (
or,1), (online,1), (at,1), (www.gutenberg.org,1), ("",1), ("",1), (Title:,1), (Ulysses,1), ("",1), (Author:,1), (James,1), (Joyce,1), ("",1), (Posting,1), (Date:,1), (August,1), (1,,1), (200
8,1), ([EBook,1), (#4300],1), (Releas...
```

```
scala> val counts = textFile.flatMap(line => line.split(" ")).map(word => (word,1)).reduceByKey(_+_)
16/03/24 08:40:10 INFO spark.ContextCleaner: Cleaned accumulator 20
16/03/24 08:40:10 INFO storage.BlockManagerInfo: Removed broadcast_20_piece0 on localhost:41207 in memory (size: 1965.0 B, free: 534.5 MB)
counts: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[32] at reduceByKey at <console>:23
```

Final print output logs are too big to display here, but the content will look as below

```
(Pooles,1)
(issued,4)
(hunted,1)
(calm:,2)
(papa,,2)
(Fein.,1)
(Tramway,1)
(gravediggers,6)
(pipespills.,1)
(undesirables,1)
(crater,1)
(unhearing,1)
(veering,1)
(untaught,1)
(positive,4)
(rustbearded,,1)
(Obvious,1)
(Doubles,1)
(posters,,1)
(barometer,2)
(purse._,1)
(luxury,,1)
(Highland,3)
(everyman,1)
(speaking.,3)
(apartment,,1)
```

Example 11: reduceByKey

```
val myRDD = sc.parallelize(Seq((1,"A"),(2,"B"),(2,"D"),(3,"C"),(3,"A"),(3,"B"),
(3,"A")),1)
val resultRDD = myRDD.reduceByKey((x, y) => {println("x"+x+"::"+"y"+y);x+y})
resultRDD.foreach(println)
```

```
scala> val myRDD =sc.parallelize(Seq((1,"A"),(2,"B"),(2,"D"),(3,"C"),(3,"A"),(3,"B"),(3,"A")),1)
myRDD: org.apache.spark.rdd.RDD[(Int, String)] = ParallelCollectionRDD[33] at parallelize at <console>:21

scala> val resultRDD = myRDD.reduceByKey((x,y) => {println("x"+x+"::"+"y"+y);x+y})
resultRDD: org.apache.spark.rdd.RDD[(Int, String)] = ShuffledRDD[34] at reduceByKey at <console>:23
```

```
scala> resultRDD.foreach(println)
16/03/24 08:47:27 INFO spark.ContextCleaner: Cleaned accumulator 21
16/03/24 08:47:27 INFO storage.BlockManagerInfo: Removed broadcast_21_piece0 on localhost:41207 in memory (size: 2.3 KB, free: 534.5 MB)
16/03/24 08:47:27 INFO spark.ContextCleaner: Cleaned accumulator 22
16/03/24 08:47:27 INFO storage.BlockManagerInfo: Removed broadcast_22_piece0 on localhost:41207 in memory (size: 1377.0 B, free: 534.5 MB)
16/03/24 08:47:31 INFO spark.SparkContext: Starting job: foreach at <console>:26
16/03/24 08:47:31 INFO scheduler.DAGScheduler: Registering RDD 33 (parallelize at <console>:21)
16/03/24 08:47:31 INFO scheduler.DAGScheduler: Got job 19 (foreach at <console>:26) with 1 output partitions
16/03/24 08:47:31 INFO scheduler.DAGScheduler: Final stage: ResultStage 23(foreach at <console>:26)
16/03/24 08:47:31 INFO scheduler.DAGScheduler: Parents of final stage: List(ShuffleMapStage 22)
16/03/24 08:47:31 INFO scheduler.DAGScheduler: Missing parents: List(ShuffleMapStage 22)
16/03/24 08:47:31 INFO scheduler.DAGScheduler: Submitting ShuffleMapStage 22 (ParallelCollectionRDD[33] at parallelize at <console>:21), which has no missing parents
16/03/24 08:47:31 INFO storage.MemoryStore: ensureFreeSpace(1896) called with curMem=113673, maxMem=560497950
16/03/24 08:47:31 INFO storage.MemoryStore: Block broadcast_23 stored as values in memory (estimated size 1896.0 B, free 534.4 MB)
16/03/24 08:47:31 INFO storage.MemoryStore: ensureFreeSpace(1201) called with curMem=115569, maxMem=560497950
16/03/24 08:47:31 INFO storage.MemoryStore: Block broadcast_23_piece0 stored as bytes in memory (estimated size 1201.0 B, free 534.4 MB)
16/03/24 08:47:32 INFO storage.BlockManagerInfo: Added broadcast_23_piece0 in memory on localhost:41207 (size: 1201.0 B, free: 534.5 MB)
16/03/24 08:47:32 INFO spark.SparkContext: Created broadcast 23 from broadcast at DAGScheduler.scala:861
16/03/24 08:47:32 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ShuffleMapStage 22 (ParallelCollectionRDD[33] at parallelize at <console>:21)
16/03/24 08:47:32 INFO scheduler.TaskSchedulerImpl: Adding task set 22.0 with 1 tasks
16/03/24 08:47:32 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 22.0 (TID 27, localhost, partition 0,PROCESS_LOCAL, 2278 bytes)
16/03/24 08:47:32 INFO executor.Executor: Running task 0.0 in stage 22.0 (TID 27)
xB::yD
xC::yA
xCA::yB
xCAB::yA
16/03/24 08:47:32 INFO executor.Executor: Finished task 0.0 in stage 22.0 (TID 27). 1158 bytes result sent to driver
16/03/24 08:47:32 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 22.0 (TID 27) in 639 ms on localhost (1/1)
16/03/24 08:47:32 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 22.0, whose tasks have all completed, from pool
16/03/24 08:47:32 INFO scheduler.DAGScheduler: ShuffleMapStage 22 (parallelize at <console>:21) finished in 0.638 s
16/03/24 08:47:32 INFO scheduler.DAGScheduler: looking for newly runnable stages
16/03/24 08:47:32 INFO scheduler.DAGScheduler: running: Set()
16/03/24 08:47:32 INFO scheduler.DAGScheduler: waiting: Set(ResultStage 23)
16/03/24 08:47:32 INFO scheduler.DAGScheduler: failed: Set()
16/03/24 08:47:32 INFO scheduler.DAGScheduler: Missing parents for ResultStage 23: List()
16/03/24 08:47:32 INFO scheduler.DAGScheduler: Submitting ResultStage 23 (ShuffledRDD[34] at reduceByKey at <console>:23), which is now runnable
16/03/24 08:47:32 INFO storage.MemoryStore: ensureFreeSpace(2272) called with curMem=116770, maxMem=560497950
16/03/24 08:47:32 INFO storage.MemoryStore: Block broadcast_24 stored as values in memory (estimated size 2.2 KB, free 534.4 MB)
16/03/24 08:47:32 INFO storage.MemoryStore: ensureFreeSpace(1357) called with curMem=119042, maxMem=560497950
16/03/24 08:47:32 INFO storage.MemoryStore: Block broadcast_24_piece0 stored as bytes in memory (estimated size 1357.0 B, free 534.4 MB)
16/03/24 08:47:32 INFO storage.BlockManagerInfo: Added broadcast_24_piece0 in memory on localhost:41207 (size: 1357.0 B, free: 534.5 MB)
16/03/24 08:47:32 INFO spark.SparkContext: Created broadcast 24 from broadcast at DAGScheduler.scala:861
16/03/24 08:47:32 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ResultStage 23 (ShuffledRDD[34] at reduceByKey at <console>:23)
```

```
16/03/24 08:47:32 INFO scheduler.TaskSchedulerImpl: Adding task set 23.0 with 1 tasks
16/03/24 08:47:32 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 23.0 (TID 28, localhost, partition 0,PROCESS_LOCAL, 1901 bytes)
16/03/24 08:47:32 INFO executor.Executor: Running task 0.0 in stage 23.0 (TID 28)
16/03/24 08:47:32 INFO storage.ShuffleBlockFetcherIterator: Getting 1 non-empty blocks out of 1 blocks
16/03/24 08:47:32 INFO storage.ShuffleBlockFetcherIterator: Started 0 remote fetches in 0 ms
(1,A)
(3,CABA)
(2,BD)
16/03/24 08:47:32 INFO executor.Executor: Finished task 0.0 in stage 23.0 (TID 28). 1165 bytes result sent to driver
16/03/24 08:47:32 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 23.0 (TID 28) in 101 ms on localhost (1/1)
16/03/24 08:47:32 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 23.0, whose tasks have all completed, from pool
16/03/24 08:47:32 INFO scheduler.DAGScheduler: ResultStage 23 (foreach at <console>:26) finished in 0.096 s
16/03/24 08:47:32 INFO scheduler.DAGScheduler: Job 19 finished: foreach at <console>:26, took 1.580848 s
```

Example 12: collectAsMap

val myRDD = sc.parallelize(Seq((1,"A"),(2,"B"),(2,"D"),(3,"C"),(3,"A"),(3,"B"),

(3,"A")))

myRDD.collectAsMap()

```
scala> val myRDD = sc.parallelize(Seq((1,"A"),(2,"B"),(2,"D"),(3,"C"),(3,"A"),(3,"B"),(3,"A")))
myRDD: org.apache.spark.rdd.RDD[(Int, String)] = ParallelCollectionRDD[35] at parallelize at <console>:21

scala> myRDD.collectAsMap()
16/03/24 08:50:49 INFO spark.SparkContext: Starting job: collectAsMap at <console>:24
16/03/24 08:50:49 INFO scheduler.DAGScheduler: Got job 20 (collectAsMap at <console>:24) with 1 output partitions
16/03/24 08:50:49 INFO scheduler.DAGScheduler: Final stage: ResultStage 24(collectAsMap at <console>:24)
16/03/24 08:50:49 INFO scheduler.DAGScheduler: Parents of final stage: List()
16/03/24 08:50:49 INFO scheduler.DAGScheduler: Missing parents: List()
16/03/24 08:50:49 INFO scheduler.DAGScheduler: Submitting ResultStage 24 (ParallelCollectionRDD[35] at parallelize at <console>:21), which has no missing parents
16/03/24 08:50:49 INFO storage.MemoryStore: ensureFreeSpace(1208) called with curMem=120399, maxMem=560497950
16/03/24 08:50:49 INFO storage.MemoryStore: Block broadcast_25 stored as values in memory (estimated size 1208.0 B, free 534.4 MB)
16/03/24 08:50:49 INFO storage.MemoryStore: ensureFreeSpace(776) called with curMem=121607, maxMem=560497950
16/03/24 08:50:49 INFO storage.MemoryStore: Block broadcast_25_piece0 stored as bytes in memory (estimated size 776.0 B, free 534.4 MB)
16/03/24 08:50:49 INFO storage.BlockManagerInfo: Added broadcast_25_piece0 in memory on localhost:41207 (size: 776.0 B, free: 534.5 MB)
16/03/24 08:50:49 INFO spark.SparkContext: Created broadcast 25 from broadcast at DAGScheduler.scala:861
16/03/24 08:50:49 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ResultStage 24 (ParallelCollectionRDD[35] at parallelize at <console>:21)
16/03/24 08:50:49 INFO scheduler.TaskSchedulerImpl: Adding task set 24.0 with 1 tasks
16/03/24 08:50:49 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 24.0 (TID 29, localhost, partition 0,PROCESS_LOCAL, 2289 bytes)
16/03/24 08:50:49 INFO executor.Executor: Running task 0.0 in stage 24.0 (TID 29)
16/03/24 08:50:49 INFO executor.Executor: Finished task 0.0 in stage 24.0 (TID 29). 1163 bytes result sent to driver
16/03/24 08:50:49 INFO scheduler.DAGScheduler: ResultStage 24 (collectAsMap at <console>:24) finished in 0.012 s
16/03/24 08:50:49 INFO scheduler.DAGScheduler: Job 20 finished: collectAsMap at <console>:24, took 0.094938 s
16/03/24 08:50:49 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 24.0 (TID 29) in 13 ms on localhost (1/1)
16/03/24 08:50:49 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 24.0, whose tasks have all completed, from pool
res23: scala.collection.Map[Int,String] = Map(2 -> D, 1 -> A, 3 -> A)
```

Example 13: countByKey

myRDD.countByKey()

```
16/03/24 08:51:30 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 25.0 (TID 30) in 140 ms on localhost (1/1)
16/03/24 08:51:30 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 25.0, whose tasks have all completed, from pool
16/03/24 08:51:30 INFO scheduler.DAGScheduler: Missing parents for ResultStage 26: List()
16/03/24 08:51:30 INFO scheduler.DAGScheduler: Submitting ResultStage 26 (ShuffledRDD[37] at countByKey at <console>:24), which is now runnable
16/03/24 08:51:30 INFO storage.MemoryStore: ensureFreeSpace(2336) called with curMem=126529, maxMem=560497950
16/03/24 08:51:30 INFO storage.MemoryStore: Block broadcast_27 stored as values in memory (estimated size 2.3 KB, free 534.4 MB)
16/03/24 08:51:30 INFO storage.MemoryStore: ensureFreeSpace(1371) called with curMem=128865, maxMem=560497950
16/03/24 08:51:30 INFO storage.MemoryStore: Block broadcast_27_piece0 stored as bytes in memory (estimated size 1371.0 B, free 534.4 MB)
16/03/24 08:51:30 INFO storage.BlockManagerInfo: Added broadcast_27_piece0 in memory on localhost:41207 (size: 1371.0 B, free: 534.5 MB)
16/03/24 08:51:30 INFO spark.SparkContext: Created broadcast 27 from broadcast at DAGScheduler.scala:861
16/03/24 08:51:30 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ResultStage 26 (ShuffledRDD[37] at countByKey at <console>:24)
16/03/24 08:51:30 INFO scheduler.TaskSchedulerImpl: Adding task set 26.0 with 1 tasks
16/03/24 08:51:30 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 26.0 (TID 31, localhost, partition 0,PROCESS_LOCAL, 1901 bytes)
16/03/24 08:51:30 INFO executor.Executor: Running task 0.0 in stage 26.0 (TID 31)
16/03/24 08:51:30 INFO storage.ShuffleBlockFetcherIterator: Getting 1 non-empty blocks out of 1 blocks
16/03/24 08:51:30 INFO storage.ShuffleBlockFetcherIterator: Started 0 remote fetches in 0 ms
16/03/24 08:51:30 INFO executor.Executor: Finished task 0.0 in stage 26.0 (TID 31). 1417 bytes result sent to driver
16/03/24 08:51:30 INFO scheduler.DAGScheduler: ResultStage 26 (countByKey at <console>:24) finished in 0.024 s
16/03/24 08:51:30 INFO scheduler.DAGScheduler: Job 21 finished: countByKey at <console>:24, took 0.298763 s
16/03/24 08:51:30 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 26.0 (TID 31) in 28 ms on localhost (1/1)
16/03/24 08:51:30 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 26.0, whose tasks have all completed, from pool
res24: scala.collection.Map[Int,Long] = Map(1 -> 1, 3 -> 4, 2 -> 2)
```

Example 14: groupBy

```
val a = sc.parallelize(1 to 15)
a.groupBy(x => {if (x % 2 == 0) "even" else "odd"}).collect
```

```
scala> val a = sc.parallelize(1 to 15)
a: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[38] at parallelize at <console>:21

scala> a.groupBy(x => {if (x % 2 == 0) "even" else"odd"}).collect
16/03/24 08:52:50 INFO spark.SparkContext: Starting job: collect at <console>:24
16/03/24 08:52:50 INFO scheduler.DAGScheduler: Registering RDD 39 (groupBy at <console>:24)
16/03/24 08:52:50 INFO scheduler.DAGScheduler: Got job 22 (collect at <console>:24) with 1 output partitions
16/03/24 08:52:50 INFO scheduler.DAGScheduler: Final stage: ResultStage 28(collect at <console>:24)
16/03/24 08:52:50 INFO scheduler.DAGScheduler: Parents of final stage: List(ShuffleMapStage 27)
16/03/24 08:52:50 INFO scheduler.DAGScheduler: Missing parents: List(ShuffleMapStage 27)
16/03/24 08:52:50 INFO scheduler.DAGScheduler: Submitting ShuffleMapStage 27 (MapPartitionsRDD[39] at groupBy at <console>:24), which has no missing parents
16/03/24 08:52:50 INFO storage.MemoryStore: ensureFreeSpace(3424) called with curMem=130236, maxMem=560497950
16/03/24 08:52:50 INFO storage.MemoryStore: Block broadcast_28 stored as values in memory (estimated size 3.3 KB, free 534.4 MB)
16/03/24 08:52:51 INFO storage.MemoryStore: ensureFreeSpace(1868) called with curMem=133660, maxMem=560497950
16/03/24 08:52:51 INFO storage.MemoryStore: Block broadcast_28_piece0 stored as bytes in memory (estimated size 1868.0 B, free 534.4 MB)
16/03/24 08:52:51 INFO storage.BlockManagerInfo: Added broadcast_28_piece0 in memory on localhost:41207 (size: 1868.0 B, free: 534.5 MB)
16/03/24 08:52:51 INFO spark.SparkContext: Created broadcast 28 from broadcast at DAGScheduler.scala:861
16/03/24 08:52:51 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ShuffleMapStage 27 (MapPartitionsRDD[39] at groupBy at <console>:24)
16/03/24 08:52:51 INFO scheduler.TaskSchedulerImpl: Adding task set 27.0 with 1 tasks
16/03/24 08:52:51 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 27.0 (TID 32, localhost, partition 0,PROCESS_LOCAL, 2131 bytes)
16/03/24 08:52:51 INFO executor.Executor: Running task 0.0 in stage 27.0 (TID 32)
16/03/24 08:52:51 INFO executor.Executor: Finished task 0.0 in stage 27.0 (TID 32). 1158 bytes result sent to driver
16/03/24 08:52:51 INFO scheduler.DAGScheduler: ShuffleMapStage 27 (groupBy at <console>:24) finished in 0.667 s
```

```
16/03/24 08:52:51 INFO executor.Executor: Finished task 0.0 in stage 28.0 (TID 33). 1835 bytes result sent to driver
16/03/24 08:52:51 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 28.0 (TID 33) in 95 ms on localhost (1/1)
16/03/24 08:52:51 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 28.0, whose tasks have all completed, from pool
16/03/24 08:52:51 INFO scheduler.DAGScheduler: ResultStage 28 (collect at <console>:24) finished in 0.091 s
16/03/24 08:52:51 INFO scheduler.DAGScheduler: Job 22 finished: collect at <console>:24, took 0.976306 s
res25: Array[(String, Iterable[Int])] = Array((even,CompactBuffer(2, 4, 6, 8, 10, 12, 14)), (odd,CompactBuffer(1, 3, 5, 7, 9, 11, 13, 15)))
```

Example 15: groupByKey

```
val name = sc.parallelize(List("kim","kumar","muthu","tim","lak","vams"))
val namekey = name.keyBy(_.length)
val mycounter = namekey.map(x => (x._1,1))
mycounter.collect
```

```
scala> val name = sc.parallelize(List("kim","kumar","muthu","tim","lak","vams"))
name: org.apache.spark.rdd.RDD[String] = ParallelCollectionRDD[41] at parallelize at <console>:21

scala> val namekey = name.keyBy(_.length)
16/03/24 08:55:40 INFO spark.ContextCleaner: Cleaned accumulator 23
16/03/24 08:55:40 INFO storage.BlockManagerInfo: Removed broadcast_23_piece0 on localhost:41207 in memory (size: 1201.0 B, free: 534.5 MB)
16/03/24 08:55:40 INFO spark.ContextCleaner: Cleaned accumulator 24
16/03/24 08:55:40 INFO storage.BlockManagerInfo: Removed broadcast_24_piece0 on localhost:41207 in memory (size: 1357.0 B, free: 534.5 MB)
16/03/24 08:55:40 INFO spark.ContextCleaner: Cleaned accumulator 25
16/03/24 08:55:40 INFO storage.BlockManagerInfo: Removed broadcast_25_piece0 on localhost:41207 in memory (size: 776.0 B, free: 534.5 MB)
16/03/24 08:55:40 INFO spark.ContextCleaner: Cleaned shuffle 4
16/03/24 08:55:40 INFO spark.ContextCleaner: Cleaned accumulator 26
16/03/24 08:55:40 INFO storage.BlockManagerInfo: Removed broadcast_26_piece0 on localhost:41207 in memory (size: 1530.0 B, free: 534.5 MB)
16/03/24 08:55:40 INFO spark.ContextCleaner: Cleaned accumulator 27
16/03/24 08:55:40 INFO storage.BlockManagerInfo: Removed broadcast_27_piece0 on localhost:41207 in memory (size: 1371.0 B, free: 534.5 MB)
16/03/24 08:55:40 INFO spark.ContextCleaner: Cleaned shuffle 5
16/03/24 08:55:40 INFO spark.ContextCleaner: Cleaned accumulator 28
16/03/24 08:55:40 INFO storage.BlockManagerInfo: Removed broadcast_28_piece0 on localhost:41207 in memory (size: 1868.0 B, free: 534.5 MB)
16/03/24 08:55:40 INFO spark.ContextCleaner: Cleaned accumulator 29
16/03/24 08:55:40 INFO storage.BlockManagerInfo: Removed broadcast_29_piece0 on localhost:41207 in memory (size: 2.0 KB, free: 534.5 MB)
namekey: org.apache.spark.rdd.RDD[(Int, String)] = MapPartitionsRDD[42] at keyBy at <console>:23

scala> val mycounter = namekey.map(x => (x._1,1))
mycounter: org.apache.spark.rdd.RDD[(Int, Int)] = MapPartitionsRDD[43] at map at <console>:25
```

```
scala> mycounter.collect
16/03/24 08:56:46 INFO spark.SparkContext: Starting job: collect at <console>:28
16/03/24 08:56:46 INFO scheduler.DAGScheduler: Got job 23 (collect at <console>:28) with 1 output partitions
16/03/24 08:56:46 INFO scheduler.DAGScheduler: Final stage: ResultStage 29(collect at <console>:28)
16/03/24 08:56:46 INFO scheduler.DAGScheduler: Parents of final stage: List()
16/03/24 08:56:46 INFO scheduler.DAGScheduler: Missing parents: List()
16/03/24 08:56:46 INFO scheduler.DAGScheduler: Submitting ResultStage 29 (MapPartitionsRDD[43] at map at <console>:25), which has no missing parents
16/03/24 08:56:46 INFO storage.MemoryStore: ensureFreeSpace(2168) called with curMem=113673, maxMem=560497950
16/03/24 08:56:46 INFO storage.MemoryStore: Block broadcast_30 stored as values in memory (estimated size 2.1 KB, free 534.4 MB)
16/03/24 08:56:46 INFO storage.MemoryStore: ensureFreeSpace(1269) called with curMem=115841, maxMem=560497950
16/03/24 08:56:46 INFO storage.MemoryStore: Block broadcast_30_piece0 stored as bytes in memory (estimated size 1269.0 B, free 534.4 MB)
16/03/24 08:56:46 INFO storage.BlockManagerInfo: Added broadcast_30_piece0 in memory on localhost:41207 (size: 1269.0 B, free: 534.5 MB)
16/03/24 08:56:46 INFO spark.SparkContext: Created broadcast 30 from broadcast at DAGScheduler.scala:861
16/03/24 08:56:46 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ResultStage 29 (MapPartitionsRDD[43] at map at <console>:25)
16/03/24 08:56:46 INFO scheduler.TaskSchedulerImpl: Adding task set 29.0 with 1 tasks
16/03/24 08:56:46 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 29.0 (TID 34, localhost, partition 0,PROCESS_LOCAL, 2128 bytes)
16/03/24 08:56:46 INFO executor.Executor: Running task 0.0 in stage 29.0 (TID 34)
16/03/24 08:56:46 INFO executor.Executor: Finished task 0.0 in stage 29.0 (TID 34). 1128 bytes result sent to driver
16/03/24 08:56:46 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 29.0 (TID 34) in 23 ms on localhost (1/1)
16/03/24 08:56:46 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 29.0, whose tasks have all completed, from pool
16/03/24 08:56:46 INFO scheduler.DAGScheduler: ResultStage 29 (collect at <console>:28) finished in 0.025 s
16/03/24 08:56:46 INFO scheduler.DAGScheduler: Job 23 finished: collect at <console>:28, took 0.096540 s
res26: Array[(Int, Int)] = Array((3,1), (5,1), (5,1), (3,1), (3,1), (4,1))
```

Example 16: Join

```
val name = sc.parallelize(List("kim","kumar","muthu","tim","lak","vams"))
val namekey = name.keyBy(_.length)

namekey.collect

val sub = sc.parallelize(Lis t("English","Maths","Tamil","Science"))

val subkey = sub.keyBy(_.length)

subkey.collect

namekey.join(subkey).collect
```

```
scala> namekey.collect
16/03/24 08:58:33 INFO spark.SparkContext: Starting job: collect at <console>:26
16/03/24 08:58:33 INFO scheduler.DAGScheduler: Got job 24 (collect at <console>:26) with 1 output partitions
16/03/24 08:58:33 INFO scheduler.DAGScheduler: Final stage: ResultStage 30(collect at <console>:26)
16/03/24 08:58:33 INFO scheduler.DAGScheduler: Parents of final stage: List()
16/03/24 08:58:33 INFO scheduler.DAGScheduler: Missing parents: List()
16/03/24 08:58:33 INFO scheduler.DAGScheduler: Submitting ResultStage 30 (MapPartitionsRDD[42] at keyBy at <console>:23), which has no missing parents
16/03/24 08:58:33 INFO storage.MemoryStore: ensureFreeSpace(2024) called with curMem=117110, maxMem=560497950
16/03/24 08:58:33 INFO storage.MemoryStore: Block broadcast_31 stored as values in memory (estimated size 2024.0 B, free 534.4 MB)
16/03/24 08:58:33 INFO storage.MemoryStore: ensureFreeSpace(1219) called with curMem=119134, maxMem=560497950
16/03/24 08:58:33 INFO storage.MemoryStore: Block broadcast_31_piece0 stored as bytes in memory (estimated size 1219.0 B, free 534.4 MB)
16/03/24 08:58:33 INFO storage.BlockManagerInfo: Added broadcast_31_piece0 in memory on localhost:41207 (size: 1219.0 B, free: 534.5 MB)
16/03/24 08:58:33 INFO spark.SparkContext: Created broadcast 31 from broadcast at DAGScheduler.scala:861
16/03/24 08:58:33 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ResultStage 30 (MapPartitionsRDD[42] at keyBy at <console>:23)
16/03/24 08:58:33 INFO scheduler.TaskSchedulerImpl: Adding task set 30.0 with 1 tasks
16/03/24 08:58:33 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 30.0 (TID 35, localhost, partition 0,PROCESS_LOCAL, 2128 bytes)
16/03/24 08:58:33 INFO executor.Executor: Running task 0.0 in stage 30.0 (TID 35)
16/03/24 08:58:33 INFO executor.Executor: Finished task 0.0 in stage 30.0 (TID 35). 1162 bytes result sent to driver
16/03/24 08:58:33 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 30.0 (TID 35) in 14 ms on localhost (1/1)
16/03/24 08:58:33 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 30.0, whose tasks have all completed, from pool
16/03/24 08:58:33 INFO scheduler.DAGScheduler: ResultStage 30 (collect at <console>:26) finished in 0.012 s
16/03/24 08:58:33 INFO scheduler.DAGScheduler: Job 24 finished: collect at <console>:26, took 0.092957 s
res27: Array[(Int, String)] = Array((3,kim), (5,kumar), (5,muthu), (3,tim), (3,lak), (4,vams))
```

```
scala> val sub = sc.parallelize(List("English","Maths","Tamil","Science"))
sub: org.apache.spark.rdd.RDD[String] = ParallelCollectionRDD[44] at parallelize at <console>:21

scala> val subkey = sub.keyBy(_.length)
subkey: org.apache.spark.rdd.RDD[(Int, String)] = MapPartitionsRDD[45] at keyBy at <console>:23

scala> subkey.collect
16/03/24 09:04:49 INFO spark.SparkContext: Starting job: collect at <console>:26
16/03/24 09:04:49 INFO scheduler.DAGScheduler: Got job 25 (collect at <console>:26) with 1 output partitions
16/03/24 09:04:49 INFO scheduler.DAGScheduler: Final stage: ResultStage 31(collect at <console>:26)
16/03/24 09:04:49 INFO scheduler.DAGScheduler: Parents of final stage: List()
16/03/24 09:04:49 INFO scheduler.DAGScheduler: Missing parents: List()
16/03/24 09:04:49 INFO scheduler.DAGScheduler: Submitting ResultStage 31 (MapPartitionsRDD[45] at keyBy at <console>:23), which has no missing parents
16/03/24 09:04:49 INFO storage.MemoryStore: ensureFreeSpace(2024) called with curMem=120353, maxMem=560497950
16/03/24 09:04:49 INFO storage.MemoryStore: Block broadcast_32 stored as values in memory (estimated size 2024.0 B, free 534.4 MB)
16/03/24 09:04:49 INFO storage.MemoryStore: ensureFreeSpace(1219) called with curMem=122377, maxMem=560497950
16/03/24 09:04:49 INFO storage.MemoryStore: Block broadcast_32_piece0 stored as bytes in memory (estimated size 1219.0 B, free 534.4 MB)
16/03/24 09:04:49 INFO storage.BlockManagerInfo: Added broadcast_32_piece0 in memory on localhost:41207 (size: 1219.0 B, free: 534.5 MB)
16/03/24 09:04:49 INFO spark.SparkContext: Created broadcast 32 from broadcast at DAGScheduler.scala:861
16/03/24 09:04:49 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ResultStage 31 (MapPartitionsRDD[45] at keyBy at <console>:23)
16/03/24 09:04:49 INFO scheduler.TaskSchedulerImpl: Adding task set 31.0 with 1 tasks
16/03/24 09:04:49 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 31.0 (TID 36, localhost, partition 0,PROCESS_LOCAL, 2123 bytes)
16/03/24 09:04:49 INFO executor.Executor: Running task 0.0 in stage 31.0 (TID 36)
16/03/24 09:04:49 INFO executor.Executor: Finished task 0.0 in stage 31.0 (TID 36). 1130 bytes result sent to driver
16/03/24 09:04:49 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 31.0 (TID 36) in 10 ms on localhost (1/1)
16/03/24 09:04:49 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 31.0, whose tasks have all completed, from pool
16/03/24 09:04:49 INFO scheduler.DAGScheduler: ResultStage 31 (collect at <console>:26) finished in 0.010 s
16/03/24 09:04:49 INFO scheduler.DAGScheduler: Job 25 finished: collect at <console>:26, took 0.075700 s
res28: Array[(Int, String)] = Array((7,English), (5,Maths), (5,Tamil), (7,Science))
```

```
scala> namekey.join(subkey).collect
16/03/24 09:05:59 INFO spark.SparkContext: Starting job: collect at <console>:30
16/03/24 09:05:59 INFO scheduler.DAGScheduler: Registering RDD 42 (keyBy at <console>:23)
16/03/24 09:05:59 INFO scheduler.DAGScheduler: Registering RDD 45 (keyBy at <console>:23)
16/03/24 09:05:59 INFO scheduler.DAGScheduler: Got job 26 (collect at <console>:30) with 1 output partitions
16/03/24 09:05:59 INFO scheduler.DAGScheduler: Final stage: ResultStage 34(collect at <console>:30)
16/03/24 09:05:59 INFO scheduler.DAGScheduler: Parents of final stage: List(ShuffleMapStage 33, ShuffleMapStage 32)
16/03/24 09:05:59 INFO scheduler.DAGScheduler: Missing parents: List(ShuffleMapStage 33, ShuffleMapStage 32)
16/03/24 09:05:59 INFO scheduler.DAGScheduler: Submitting ShuffleMapStage 32 (MapPartitionsRDD[42] at keyBy at <console>:23), which has no missing parents
16/03/24 09:05:59 INFO storage.MemoryStore: ensureFreeSpace(2336) called with curMem=123596, maxMem=560497950
16/03/24 09:05:59 INFO storage.MemoryStore: Block broadcast_33 stored as values in memory (estimated size 2.3 KB, free 534.4 MB)
16/03/24 09:05:59 INFO storage.MemoryStore: ensureFreeSpace(1417) called with curMem=125932, maxMem=560497950
16/03/24 09:05:59 INFO storage.MemoryStore: Block broadcast_33_piece0 stored as bytes in memory (estimated size 1417.0 B, free 534.4 MB)
16/03/24 09:05:59 INFO storage.BlockManagerInfo: Added broadcast_33_piece0 in memory on localhost:41207 (size: 1417.0 B, free: 534.5 MB)
16/03/24 09:05:59 INFO spark.SparkContext: Created broadcast 33 from broadcast at DAGScheduler.scala:861
16/03/24 09:05:59 INFO scheduler.DAGScheduler: Submitting 1 missing tasks from ShuffleMapStage 32 (MapPartitionsRDD[42] at keyBy at <console>:23)
16/03/24 09:05:59 INFO scheduler.TaskSchedulerImpl: Adding task set 32.0 with 1 tasks
16/03/24 09:05:59 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 32.0 (TID 37, localhost, partition 0,PROCESS_LOCAL, 2117 bytes)
```

```
16/03/24 09:05:59 INFO executor.Executor: Running task 0.0 in stage 34.0 (TID 39)
16/03/24 09:05:59 INFO storage.ShuffleBlockFetcherIterator: Getting 1 non-empty blocks out of 1 blocks
16/03/24 09:05:59 INFO storage.ShuffleBlockFetcherIterator: Started 0 remote fetches in 0 ms
16/03/24 09:06:00 INFO storage.ShuffleBlockFetcherIterator: Getting 1 non-empty blocks out of 1 blocks
16/03/24 09:06:00 INFO storage.ShuffleBlockFetcherIterator: Started 0 remote fetches in 1 ms
16/03/24 09:06:00 INFO executor.Executor: Finished task 0.0 in stage 34.0 (TID 39). 1415 bytes result sent to driver
16/03/24 09:06:00 INFO scheduler.DAGScheduler: ResultStage 34 (collect at <console>:30) finished in 0.636 s
16/03/24 09:06:00 INFO scheduler.DAGScheduler: Job 26 finished: collect at <console>:30, took 0.969522 s
res29: Array[(Int, (String, String))] = Array((5,(kumar,Maths)), (5,(kumar,Tamil)), (5,(muthu,Maths)), (5,(muthu,Tamil)))
16/03/24 09:06:00 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 34.0 (TID 39) in 640 ms on localhost (1/1)
16/03/24 09:06:00 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 34.0, whose tasks have all completed, from pool
```

Example 17: Join

Inner Join

val names1 = sc.parallelize(List(“apple”, “mango”, “grapes”)).map(a => (a,1))

val names2 = sc.parallelize(List(“grapes”, “litchi”, “pears” )).map(a => (a,1))

names1.join(names2).collect

leftOuterJoin

names1.leftOuterJoin(names2).collect

rightOuterJoin

names1.rightOuterJoin(names2).collect

```
scala> val names1 = sc.parallelize(List("apple", "mango", "grapes")).map(a => (a,1))
names1: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[8] at map at <console>:21

scala> val names2 = sc.parallelize(List("grapes", "litchi", "pears")).map(a => (a,1))
names2: org.apache.spark.rdd.RDD[(String, Int)] = MapPartitionsRDD[10] at map at <console>:21

scala>
```

## Inner Join

```
16/03/25 06:38:37 INFO executor.Executor: Running task 0.0 in stage 5.0 (TID 5)
16/03/25 06:38:37 INFO storage.ShuffleBlockFetcherIterator: Getting 1 non-empty blocks out of 1 blocks
16/03/25 06:38:37 INFO storage.ShuffleBlockFetcherIterator: Started 0 remote fetches in 0 ms
16/03/25 06:38:37 INFO storage.ShuffleBlockFetcherIterator: Getting 1 non-empty blocks out of 1 blocks
16/03/25 06:38:37 INFO storage.ShuffleBlockFetcherIterator: Started 0 remote fetches in 0 ms
16/03/25 06:38:37 INFO executor.Executor: Finished task 0.0 in stage 5.0 (TID 5). 1328 bytes result sent to driver
16/03/25 06:38:37 INFO scheduler.DAGScheduler: ResultStage 5 (collect at <console>:26) finished in 0.019 s
16/03/25 06:38:37 INFO scheduler.DAGScheduler: Job 1 finished: collect at <console>:26, took 0.085547 s
res1: Array[(String, (Int, Int))] = Array((grapes,(1,1)))

scala> 16/03/25 06:38:37 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 5.0 (TID 5) in 27 ms on localhost (1/1)
16/03/25 06:38:37 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 5.0, whose tasks have all completed, from pool
```

## leftOuterJoin

```
16/03/25 06:40:10 INFO storage.ShuffleBlockFetcherIterator: Getting 1 non-empty blocks out of 1 blocks
16/03/25 06:40:10 INFO storage.ShuffleBlockFetcherIterator: Started 0 remote fetches in 0 ms
16/03/25 06:40:10 INFO storage.ShuffleBlockFetcherIterator: Getting 1 non-empty blocks out of 1 blocks
16/03/25 06:40:10 INFO storage.ShuffleBlockFetcherIterator: Started 0 remote fetches in 0 ms
16/03/25 06:40:10 INFO executor.Executor: Finished task 0.0 in stage 8.0 (TID 8). 1481 bytes result sent to driver
16/03/25 06:40:10 INFO scheduler.DAGScheduler: ResultStage 8 (collect at <console>:26) finished in 0.019 s
16/03/25 06:40:10 INFO scheduler.DAGScheduler: Job 2 finished: collect at <console>:26, took 0.126941 s
16/03/25 06:40:10 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 8.0 (TID 8) in 23 ms on localhost (1/1)
16/03/25 06:40:10 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 8.0, whose tasks have all completed, from pool
res2: Array[(String, (Int, Option[Int]))] = Array((apple,(1,None)), (grapes,(1,Some(1))), (mango,(1,None)))
```

## rightOuterJoin

```
16/03/25 06:42:06 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 11.0 (TID 11, localhost, partition 0,PROCESS_LOCAL, 1974 bytes)
16/03/25 06:42:06 INFO executor.Executor: Running task 0.0 in stage 11.0 (TID 11)
16/03/25 06:42:06 INFO storage.ShuffleBlockFetcherIterator: Getting 1 non-empty blocks out of 1 blocks
16/03/25 06:42:06 INFO storage.ShuffleBlockFetcherIterator: Started 0 remote fetches in 0 ms
16/03/25 06:42:06 INFO storage.ShuffleBlockFetcherIterator: Getting 1 non-empty blocks out of 1 blocks
16/03/25 06:42:06 INFO storage.ShuffleBlockFetcherIterator: Started 0 remote fetches in 1 ms
16/03/25 06:42:06 INFO executor.Executor: Finished task 0.0 in stage 11.0 (TID 11). 1482 bytes result sent to driver
16/03/25 06:42:06 INFO scheduler.DAGScheduler: ResultStage 11 (collect at <console>:26) finished in 0.027 s
16/03/25 06:42:06 INFO scheduler.DAGScheduler: Job 3 finished: collect at <console>:26, took 0.130906 s
res3: Array[(String, (Option[Int], Int))] = Array((litchi,(None,1)), (pears,(None,1)), (grapes,(Some(1),1)))

scala> 16/03/25 06:42:06 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 11.0 (TID 11) in 33 ms on localhost (1/1)
16/03/25 06:42:06 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 11.0, whose tasks have all completed, from pool
```

# Exercise 12: Hbase

1. Start Hbase service as below from /etc/init.d, if they are not running

```
[technocrafty@quickstart init.d]$ sudo service hbase-master start
starting master, logging to /var/log/hbase/hbase-hbase-master-quickstart.technocrafty.out
Started HBase master daemon (hbase-master):                [  OK  ]
[technocrafty@quickstart init.d]$ sudo service hbase-regionserver status
hbase-regionserver is not running.
[technocrafty@quickstart init.d]$ sudo service hbase-regionserver start
Starting Hadoop HBase regionserver daemon: starting regionserver, logging to /var/log/hbase/hbase-hbase-regionserver-quickstart.technocrafty.out
hbase-regionserver.
[technocrafty@quickstart init.d]$ sudo service hbase-regionserver status
hbase-regionserver is running
[technocrafty@quickstart init.d]$ sudo service hbase-master status
HBase master daemon is running                             [  OK  ]
```

2. To invoke Hbase shell

```
[technocrafty@quickstart ~]$ hbase shell
2016-03-25 07:48:50,811 INFO  [main] Configuration.deprecation: hadoop.native.lib is deprecated. Instead, use io.native.lib.available
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.0.0-cdh5.5.0, rUnknown, Mon Nov  9 12:40:01 PST 2015

hbase(main):001:0>
```

3. Basic Commands

List – displays list of tables in Hbase

```
hbase(main):001:0> list
TABLE
0 row(s) in 0.3400 seconds

=> []
hbase(main):002:0>
```

Version- returns the version of Hbase

```
hbase(main):002:0> version
1.0.0-cdh5.5.0, rUnknown, Mon Nov  9 12:40:01 PST 2015
```

Table_help- instructs how to use table-referenced commands

```
hbase(main):005:0> table_help
Help for table-reference commands.

You can either create a table via 'create' and then manipulate the table via commands like 'put', 'get', etc.
See the standard help information for how to use each of these commands.

However, as of 0.96, you can also get a reference to a table, on which you can invoke commands.
For instance, you can get create a table and keep around a reference to it via:

   hbase> t = create 't', 'cf'

Or, if you have already created the table, you can get a reference to it:

   hbase> t = get_table 't'

You can do things like call 'put' on the table:

  hbase> t.put 'r', 'cf:q', 'v'

which puts a row 'r' with column family 'cf', qualifier 'q' and value 'v' into table t.

To read the data out, you can scan the table:

  hbase> t.scan
```

whoami- user details of Hbase

```
hbase(main):004:0> whoami
technocrafty (auth:SIMPLE)
```

4. Create table using Hbase shell

Syntax: create '<table name>','<column family>'

```
hbase(main):004:0> create 'emp','personal data','professional data'
0 row(s) in 1.4700 seconds

=> Hbase::Table - emp
```

```
hbase(main):005:0> list
TABLE
emp
1 row(s) in 0.0110 seconds

=> ["emp"]
```

5. Disabling a table

```
hbase(main):011:0> disable 'emp'
0 row(s) in 0.0350 seconds

hbase(main):012:0> list
TABLE
emp
1 row(s) in 0.0080 seconds

=> ["emp"]
hbase(main):013:0> scan 'emp'
ROW                                                    COLUMN+CELL

ERROR: emp is disabled.
```

After disabling the table, you can still see the table under list but scan will throw an error.

To verify, table is disabled

```
hbase(main):014:0> is_disabled 'emp'
true
0 row(s) in 0.0510 seconds
```

6. Enabling a table

```
hbase(main):015:0> enable 'emp'
0 row(s) in 0.2910 seconds

hbase(main):016:0> is_enabled 'emp'
true
0 row(s) in 0.0170 seconds
```

7. Description of table

```
hbase(main):018:0> describe 'emp'
Table emp is ENABLED
emp
COLUMN FAMILIES DESCRIPTION
{NAME => 'personal data', DATA_BLOCK_ENCODING => 'NONE', BLOOMFILTER => 'ROW', REPLICATION_SCOPE => '0', VERSIONS => '1', COMPRESSION => 'NONE', MIN_VERSIONS => '0', TTL => 'FOREVER', KEEP_DELETED_CELLS => 'FALSE', BLOCKSIZE => '65536',
IN_MEMORY => 'false', BLOCKCACHE => 'true'}
{NAME => 'professional data', DATA_BLOCK_ENCODING => 'NONE', BLOOMFILTER => 'ROW', REPLICATION_SCOPE => '0', VERSIONS => '1', COMPRESSION => 'NONE', MIN_VERSIONS => '0', TTL => 'FOREVER', KEEP_DELETED_CELLS => 'FALSE', BLOCKSIZE => '6553
6', IN_MEMORY => 'false', BLOCKCACHE => 'true'}
2 row(s) in 0.0370 seconds
```

8. Alter table

To change maximum number of cells of a column family

```
hbase(main):019:0> alter 'emp', NAME ⇒ 'personal data', VERSIONS ⇒ 5
Updating all regions with the new schema...
0/1 regions updated.
1/1 regions updated.
Done.
0 row(s) in 2.4220 seconds
```

9. Inserting Data

```
hbase(main):002:0> put 'emp','1','personal data:name','raju'
0 row(s) in 0.1550 seconds

hbase(main):003:0> put 'emp','1','personal data:city','hyderabad'
0 row(s) in 0.0150 seconds

hbase(main):004:0> put 'emp','1','professional data:designation','manager'
0 row(s) in 0.0110 seconds

hbase(main):005:0> put 'emp','1','professional data:salary','50000'
0 row(s) in 0.0190 seconds
```

Verify the data

```
hbase(main):006:0> scan 'emp'
ROW                                                   COLUMN+CELL
 1                                                    column=personal data:city, timestamp=1458920476770, value=hyderabad
 1                                                    column=personal data:name, timestamp=1458920463555, value=raju
 1                                                    column=professional data:designation, timestamp=1458920521112, value=manager
 1                                                    column=professional data:salary, timestamp=1458920531635, value=50000
1 row(s) in 0.1580 seconds
```

10. Reading Data

Syntax: get '<table name>', 'row1'

```
hbase(main):010:0> get 'emp','1'
COLUMN                                                CELL
 personal data:city                                   timestamp=1458920476770, value=hyderabad
 personal data:name                                   timestamp=1458920463555, value=raju
 professional data:designation                        timestamp=1458920521112, value=manager
 professional data:salary                             timestamp=1458920531635, value=50000
4 row(s) in 0.0400 seconds
```

11. Truncate table

```
hbase(main):012:0> truncate 'emp'
Truncating 'emp' table (it may take a while):
 - Disabling table...
 - Truncating table...
0 row(s) in 1.6620 seconds
```

```
hbase(main):013:0> scan 'emp'
ROW                                                   COLUMN+CELL
0 row(s) in 0.3430 seconds
```

12. Dropping a table

```
hbase(main):014:0> disable 'emp'
0 row(s) in 1.3530 seconds

hbase(main):015:0> drop 'emp'
0 row(s) in 0.3510 seconds

hbase(main):016:0> list
TABLE
0 row(s) in 0.0080 seconds
```

## Managing an Hbase Table with Hive

1. Create a table 'hbaseUserRatings' in Hbase with a single family column 'cf1'

```
create 'hbaseUserRatings','cf1'
```

2. Put rows into the table as below and scan the table

```
put 'hbaseUserRatings','10000','cf1:moveid','42'
put 'hbaseUserRatings','10000','cf1:ratings','5'
put 'hbaseUserRatings','11000','cf1:moveid','21'
put 'hbaseUserRatings','11000','cf1:ratings','4'
scan 'hbaseUserRatings'
```

```
hbase(main):017:0> create 'hbaseUserRatings','cf1'
0 row(s) in 0.6930 seconds

=> Hbase::Table - hbaseUserRatings
hbase(main):018:0> put 'hbaseUserRatings','10000','cf1:moveid','42'
0 row(s) in 0.0180 seconds

hbase(main):019:0> put 'hbaseUserRatings','10000','cf1:ratings','5'
0 row(s) in 0.0080 seconds

hbase(main):020:0> put 'hbaseUserRatings','11000','cf1:moveid','21'
0 row(s) in 0.0060 seconds

hbase(main):021:0> put 'hbaseUserRatings','11000','cf1:ratings','4'
0 row(s) in 0.0060 seconds
```

```
hbase(main):022:0> scan 'hbaseUserRatings'
ROW                                                   COLUMN+CELL
 10000                                                column=cf1:moveid, timestamp=1458922386514, value=42
 10000                                                column=cf1:ratings, timestamp=1458922416218, value=5
 11000                                                column=cf1:moveid, timestamp=1458922444712, value=21
 11000                                                column=cf1:ratings, timestamp=1458922461084, value=4
2 row(s) in 0.0300 seconds
```

3. Exit from Hbase shell and start Hive by invoking hive shell

```
[technocrafty@quickstart ~]$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.properties
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.
hive>
```

4. Add Hive's hbase handler and the Hbase jar

```
add jar /usr/lib/hive/lib/hive-hbase-handler-1.1.0-cdh5.5.0.jar;

add jar /usr/lib/hive/lib/hive-hbase-handler.jar;
```

```
[technocrafty@quickstart init.d]$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.properties
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.
hive> add jar /usr/lib/hive/lib/hive-hbase-handler-1.1.0-cdh5.5.0.jar;
Added [/usr/lib/hive/lib/hive-hbase-handler-1.1.0-cdh5.5.0.jar] to class path
Added resources: [/usr/lib/hive/lib/hive-hbase-handler-1.1.0-cdh5.5.0.jar]
hive> add jar /usr/lib/hive/lib/hive-hbase-handler.jar
    > ;
Added [/usr/lib/hive/lib/hive-hbase-handler.jar] to class path
Added resources: [/usr/lib/hive/lib/hive-hbase-handler.jar]
hive>
```

5. Create an external table and execute select to check the content of hbaseUserRatings table

```
create external table hbaseUserRatings (userid int,moveid int,ratings int) stored by
'org.apache.hadoop.hive.hbase.HBaseStorageHandler' with

SERDEPROPERTIES ("hbase.columns.mapping" = ":key,cf1:moveid,cf1:ratings")

TBLPROPERTIES ("hbase.table.name" = "hbaseUserRatings");
```

```
hive> create external table hbaseUserRatings (userid int,moveid int,ratings int) stored by 'org.apache.hadoop.hive.hbase.HBaseStorageHandler' with
    > SERDEPROPERTIES ("hbase.columns.mapping" = ":key,cf1:moveid,cf1:ratings")
    > TBLPROPERTIES ("hbase.table.name" = "hbaseUserRatings");
OK
Time taken: 0.247 seconds
```

```
select * from hbaseuserratings;
```

```
hive> select * from hbaseuserratings;
OK
10000   42      5
11000   21      4
Time taken: 0.215 seconds, Fetched: 2 row(s)
hive>
```

## Bulk Import

In this excerise we will start with creating a new HBase table for loading, prepare the file with importtsv tool and then will load the prepared files into an HBase table with completebulkload tool.

**Step1**: Create a new HBase table for loading

```
create 'test','cf'
```

```
hbase(main):007:0> create 'test','cf'
0 row(s) in 0.8750 seconds

=> Hbase::Table - test
hbase(main):008:0>
```

**Step2**: Upload test.txt file from local filesystem to HDFS

```
hdfs dfs -put test.txt input
```

To check the content of test.txt file on HDFS

```
[technocrafty@quickstart Datasets]$ hdfs dfs -cat input/test.txt
1,tom
2,sam
3,jerry
4,marry
5,john
6,mickey
7,mouse
8,donald
9,duck
10,bear
[technocrafty@quickstart Datasets]$
```

**Step3**: Prepare the files to be imported using importtsv tool

- ImportTsv is a utility that will load data in TSV format into HBase.
- It has two distinct usages: loading data from TSV format in HDFS into HBase via Puts, and preparing StoreFiles to be loaded via the completebulkload.

```
hbase org.apache.hadoop.hbase.mapreduce.ImportTsv -Dimporttsv.separator="," -Dimporttsv.columns=HBASE_ROW_KEY,cf test /user/technocrafty/input/test.txt
```

```
[technocrafty@quickstart Datasets]$ hbase org.apache.hadoop.hbase.mapreduce.ImportTsv -Dimporttsv.separator=”,” \
> -Dimporttsv.columns=HBASE_ROW_KEY,cf test /user/technocrafty/input/test.txt
```

This will start the MapReduce job

```
2016-03-28 07:41:02,825 INFO  [main] mapreduce.Job:  map 100% reduce 0%
2016-03-28 07:41:02,838 INFO  [main] mapreduce.Job: Job job_1459174501329_0002 completed successfully
2016-03-28 07:41:03,037 INFO  [main] mapreduce.Job: Counters: 31
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=142966
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=209
                HDFS: Number of bytes written=0
                HDFS: Number of read operations=2
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=0
        Job Counters
                Launched map tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=7450
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=7450
                Total vcore-seconds taken by all map tasks=7450
                Total megabyte-seconds taken by all map tasks=7628800
        Map-Reduce Framework
                Map input records=10
                Map output records=10
                Input split bytes=133
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=138
                CPU time spent (ms)=1550
                Physical memory (bytes) snapshot=127057920
                Virtual memory (bytes) snapshot=1506975744
                Total committed heap usage (bytes)=60882944
        ImportTsv
```

Verify the data loaded into HBase table

```
hbase(main):009:0> scan 'test'
ROW                                   COLUMN+CELL
 1                                    column=cf:, timestamp=1459176035874, value=tom
 10                                   column=cf:, timestamp=1459176035874, value=bear
 2                                    column=cf:, timestamp=1459176035874, value=sam
 3                                    column=cf:, timestamp=1459176035874, value=jerry
 4                                    column=cf:, timestamp=1459176035874, value=marry
 5                                    column=cf:, timestamp=1459176035874, value=john
 6                                    column=cf:, timestamp=1459176035874, value=mickey
 7                                    column=cf:, timestamp=1459176035874, value=mouse
 8                                    column=cf:, timestamp=1459176035874, value=donald
 9                                    column=cf:, timestamp=1459176035874, value=duck
10 row(s) in 0.0700 seconds
```

**Step4**: Load the files into an HBase table using completebulkload tool

- Create another table in HBase

```
hbase(main):010:0> create 'test1','cf'
0 row(s) in 0.4540 seconds

=> Hbase::Table - test1
hbase(main):011:0>
```

- Use ImportTsv to generate HFile for the text file in HDFS

```
[technocrafty@quickstart ~]$ hbase org.apache.hadoop.hbase.mapreduce.ImportTsv -Dimporttsv.separator="," \
> -Dimporttsv.bulk.output=/user/technocrafty/test-hfile \
> -Dimporttsv.columns=HBASE_ROW_KEY,cf test1 /user/technocrafty/input/test.txt
```

This command will be executed by MapReduce job

```
2016-03-28 07:52:48,109 INFO  [main] mapreduce.Job: Running job: job_1459174501329_0003
2016-03-28 07:52:57,512 INFO  [main] mapreduce.Job: Job job_1459174501329_0003 running in uber mode : false
2016-03-28 07:52:57,516 INFO  [main] mapreduce.Job:  map 0% reduce 0%
2016-03-28 07:53:06,661 INFO  [main] mapreduce.Job:  map 100% reduce 0%
2016-03-28 07:53:16,755 INFO  [main] mapreduce.Job:  map 100% reduce 100%
2016-03-28 07:53:17,801 INFO  [main] mapreduce.Job: Job job_1459174501329_0003 completed successfully
2016-03-28 07:53:18,022 INFO  [main] mapreduce.Job: Counters: 50
        File System Counters
                FILE: Number of bytes read=493
                FILE: Number of bytes written=292141
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=209
                HDFS: Number of bytes written=1340
                HDFS: Number of read operations=8
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=3
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=6795
                Total time spent by all reduces in occupied slots (ms)=7357
                Total time spent by all map tasks (ms)=6795
                Total time spent by all reduce tasks (ms)=7357
                Total vcore-seconds taken by all map tasks=6795
                Total vcore-seconds taken by all reduce tasks=7357
                Total megabyte-seconds taken by all map tasks=6958080
                Total megabyte-seconds taken by all reduce tasks=7533568
```

- The Hfile is generated as

```
[technocrafty@quickstart ~]$ hdfs dfs -ls /user/technocrafty/test-hfile
Found 2 items
-rw-r--r--   1 technocrafty technocrafty          0 2016-03-28 07:53 /user/technocrafty/test-hfile/_SUCCESS
drwxr-xr-x   - technocrafty technocrafty          0 2016-03-28 07:53 /user/technocrafty/test-hfile/cf
[technocrafty@quickstart ~]$ hdfs dfs -ls /user/technocrafty/test-hfile/cf
Found 1 items
-rw-r--r--   1 technocrafty technocrafty       1340 2016-03-28 07:53 /user/technocrafty/test-hfile/cf/79390d15d1794485b1cc54e1
44807bf7
[technocrafty@quickstart ~]$
```

- Note that data is not loaded in HBase table

```
hbase(main):011:0> scan 'test1'
ROW                                   COLUMN+CELL
0 row(s) in 0.0170 seconds

hbase(main):012:0>
```

**Step5**: Use completebulkload to load the Hfile to HBase

completebulkload utility will move generated StoreFiles into an HBase table

There are two ways to invoke this utility:

- with explicit classname and
- via the driver

```
[technocrafty@quickstart conf]$ hbase org.apache.hadoop.hbase.mapreduce.LoadIncrementalHFiles \
> /user/technocrafty/test-hfile test1
```

```
hbase(main):013:0> scan 'test1'
ROW                                COLUMN+CELL
 1                                 column=cf:, timestamp=1459176761555, value=tom
 10                                column=cf:, timestamp=1459176761555, value=bear
 2                                 column=cf:, timestamp=1459176761555, value=sam
 3                                 column=cf:, timestamp=1459176761555, value=jerry
 4                                 column=cf:, timestamp=1459176761555, value=marry
 5                                 column=cf:, timestamp=1459176761555, value=john
 6                                 column=cf:, timestamp=1459176761555, value=mickey
 7                                 column=cf:, timestamp=1459176761555, value=mouse
 8                                 column=cf:, timestamp=1459176761555, value=donald
 9                                 column=cf:, timestamp=1459176761555, value=duck
10 row(s) in 0.0780 seconds
```