**Assignment 1:**

**Performance Analysis Using Hadoop MapReduce**

**Problem Statement:**

Considering five different size text files with .csv or .txt extension to analyze the CPU time to execute the Word count program and generate the graph of time taken in milliseconds(ms) versus file size in MB.

**Description on MapReduce:**

**MapReduce** is a programming model and an associated implementation for processing and generating big data sets with a parallel distributed algorithm on a cluster.

A MapReduce program is composed of a **Map** procedure (or method), which performs filtering and sorting of the data where individual elements are broken down into tuples (key/value pairs). and **Reduce** method which performs summary operation which takes the output from a map as an input and combines those data tuples into a smaller set of tuples any logical operations, calculations, etc. The “MapReduce System” orchestrates the processing by marshalling the distributed servers, running the various task in parallel, managing all communications and data transfers between the various parts of the system and providing for redundancy and fault tolerance. the reduce task is always performed after the map job. It is the specialization of split-apply-combine strategy for data analysis. Processing can occur on data stored either in a filesystem (unstructured) or in a database   
(structured). It takes the advantage of the locality of data, processing it near the place it is stored to minimize communication overhead.

**MapReduce** can also be defined as:

* Prepare the Map() input.
* Run the user provided Map() code.
* “Shuffle” the map output to the reduce processors.
* Runt the user provided Reduce() code.
* Produce the final output.
* **Commands used in the assignment:**

1. To start all the daemons in hadoop file system: **start all.sh**
2. To check whether the daemons are active :

**jps**

1. To load the text files in Hadoop file:

Syntax: Hadoop fs -put filename.txt /

**hadoop fs –put file50.txt /**

**hadoop fs –put file150.txt /**

**hadoop fs –put file289.txt /**

**hadoop fs –put file100.txt /**

**hadoop fs –put file200.txt /**

1. To Check whether the file is loaded into Hadoop file system:

**hadoop fs -ls /**

1. To execute text file using WordCount jar file:

**hadoop jar wc1.jar WordCount file50.txt /out50**

**hadoop jar wc1.jar WordCount file100.txt /out100**

**hadoop jar wc1.jar WordCount file150.txt /out150**

**hadoop jar wc1.jar WordCount file200.txt /out200**

**hadoop jar wc1.jar WordCount file289.txt /out289**

|  |  |  |
| --- | --- | --- |
| No. | Text file (MB) | CPU Time (in ms ) Nodes |
| 1. | 50 | 78120 Standalone |
| 2. | 150 | 231130 Standalone |
| 3. | 289 | 580740 Standalone |

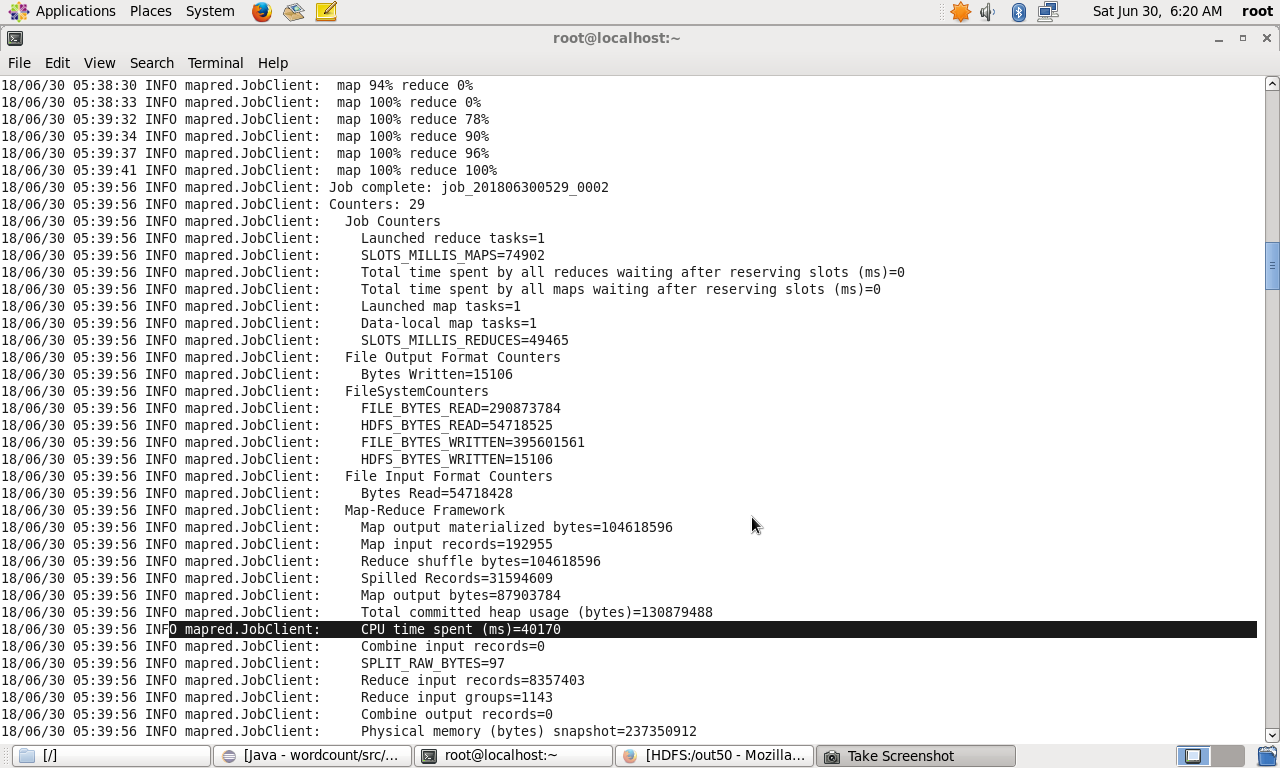
**Specifications:**

**Processor:** Intel(R) Core (TM) i5-5250U CPU @ 1.60GHz 1.60GHz

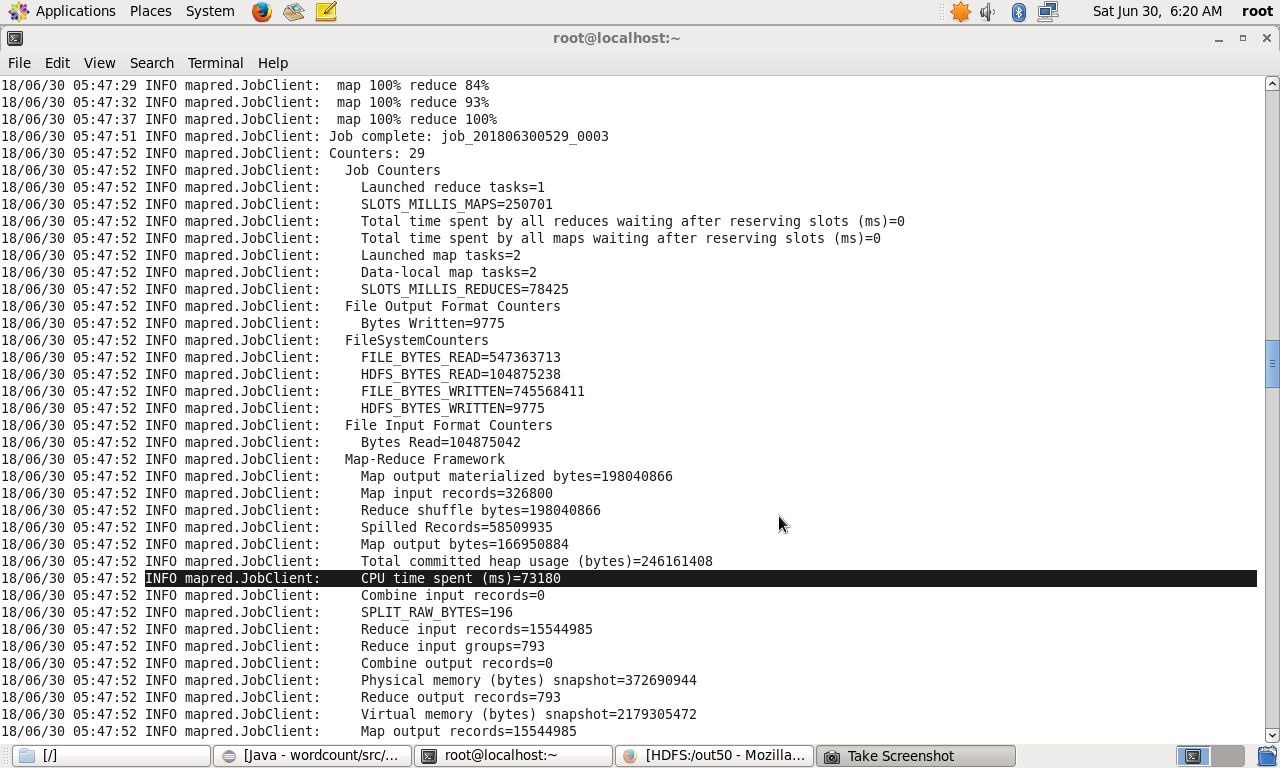
**Installed Memory (RAM)**: 8GB

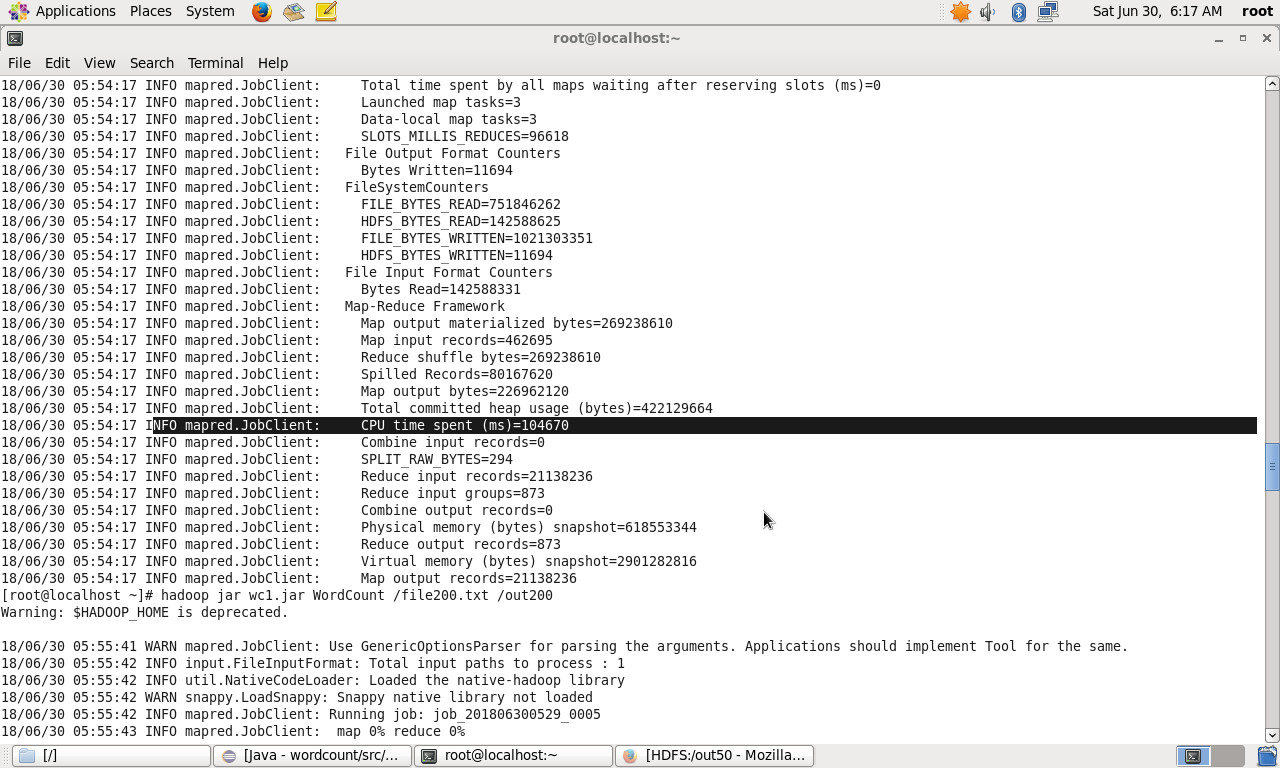
**System Type:** 64-bit Operating System, x64-based processor

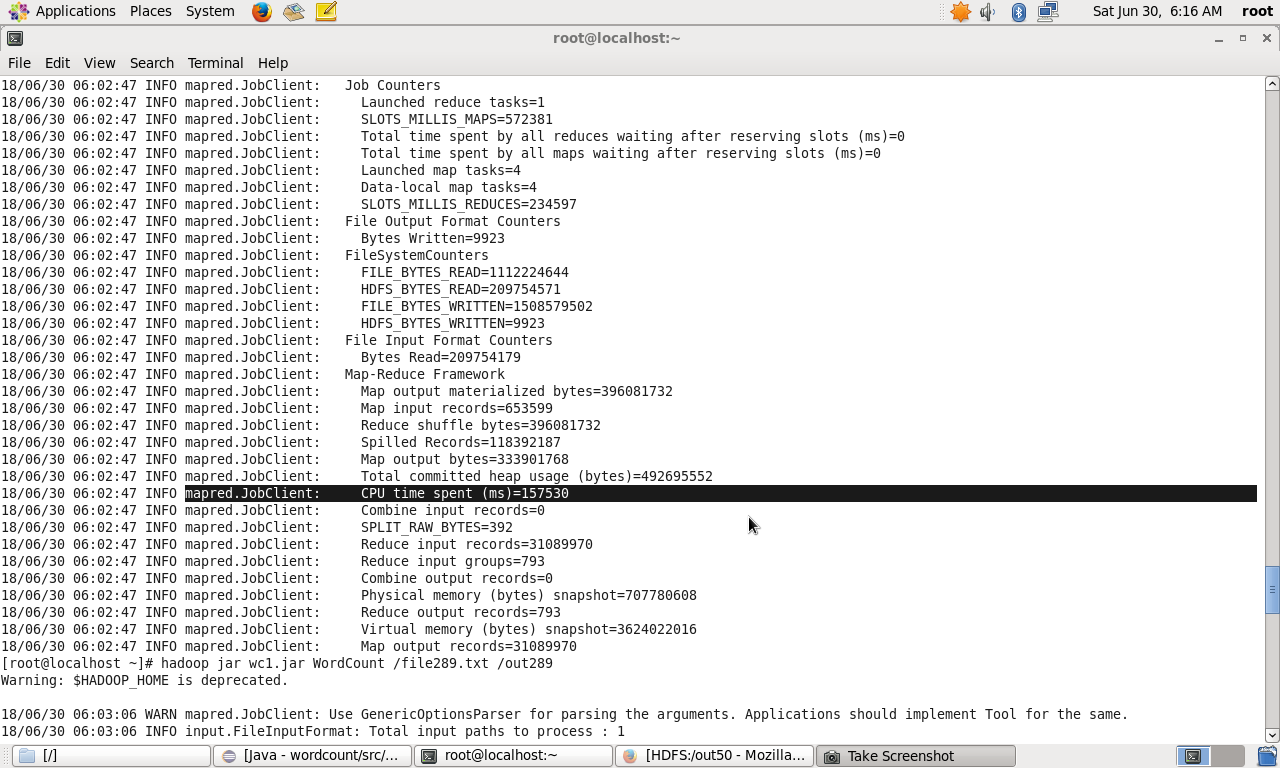
**Screenshot:**

****

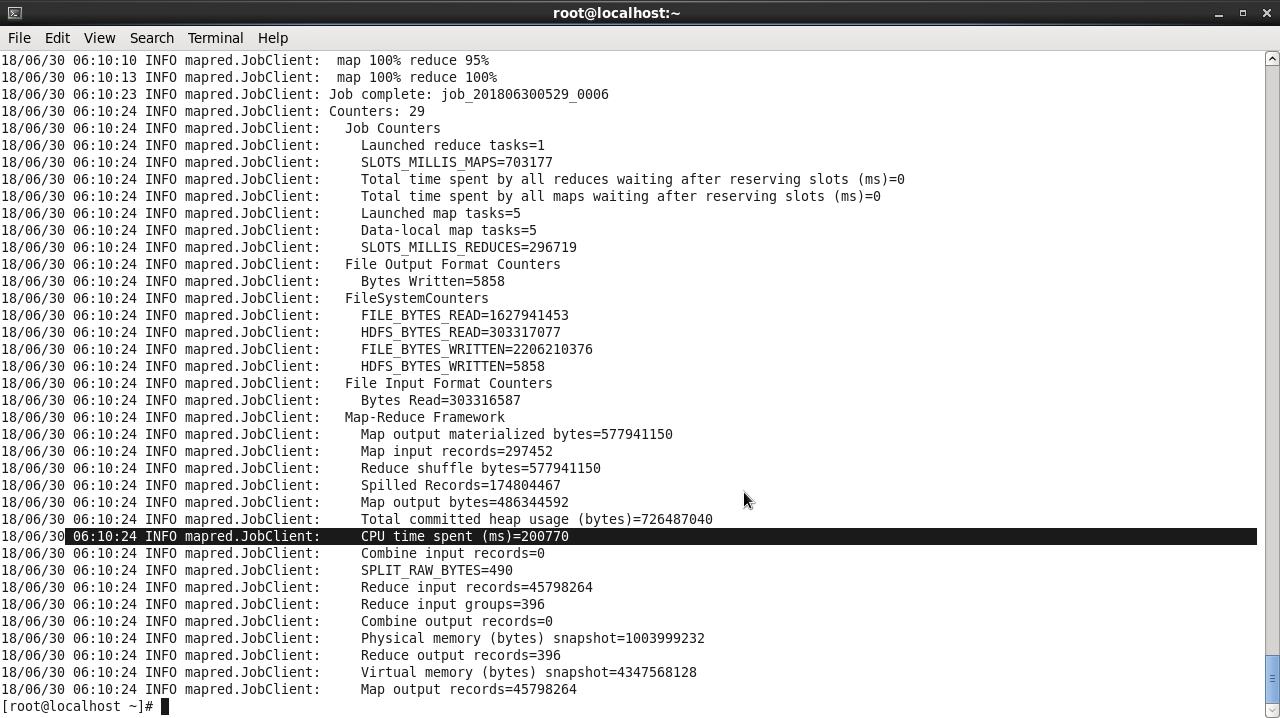
**Fig 1: CPU time (ms) to compute file file50.txt**

**Fig 2: CPU time (ms) to compute file file100.txt**

**Fig 3: CPU time (ms) to compute file file150.txt**

****

**Fig 3: CPU time (ms) to compute file file200.txt**

****

**Fig 3: CPU time (ms) to compute file file289.txt**

**Graph:**

**Fig 4: Graph of Text file Size (MB) vs CPU Time (milliseconds)Final**

**Output:** In web Browser[**http://localhost50070**](http://localhost50070) **select browse files**