CMPE274 Project (Spring Semester, 2014)

Music Data Analysis

- Submitted to : Dr. Weider Yu
- Submitted by : Team 9

S.No	Project Member	SJSU ID	Email ID	Phone No
1	Sarvagya Jain	009269377	sarvagya.jain@sjsu.edu	408-326-9743
2	Swapnil Pancholi	009303684	pancholi.swapnil@gmail.com	408-831-8162
3	Arun Malik	009304607	malik.mgm@gmail.com	408-834-6060
4	Pooja Kasu	009305829	pooja.kasu@gmail.com	479-200-1056
5	Gaurav Kesarwani	009278815	gaurav.kesarwani2@gmail.com	408-442-8559

Introduction

- This project is doing an analysis of trends in the music industry.
- The analysis is performed on song data from the dataset which contains data on a million popular music tracks
- Target Users of the project :
  i. Companies like Pandora can use the analysis to suggest songs to listeners based on their preferences.
  ii. Record labels can have a better understanding of what kinds of music will sell.

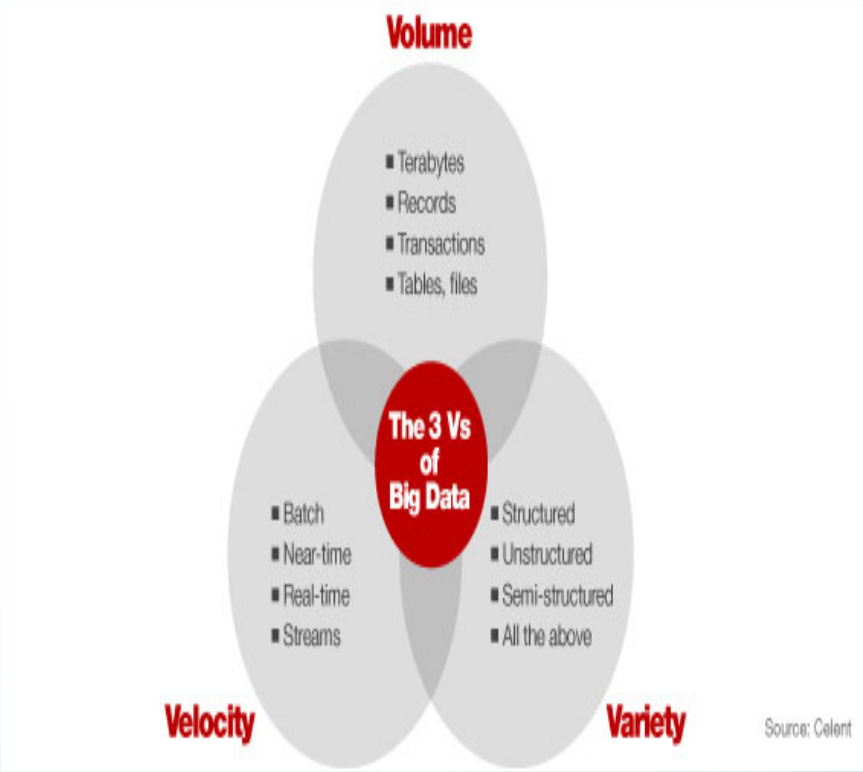
Problem Statement

- To analyze big data of songs which contains data on a million popular music tracks with attributes such as tempo, density, and general loudness, etc.
- To generate a trend out of that analysis which can be of use.

Big Data
and
Project Descriptions

Big Data

- What is Big Data?
  - Volume – Scale of Data
    - Terabytes to Exabyte's of existing data
  - Velocity – Analysis of Streaming Data
  - Variety – Different Forms of Data
    - Structured and Unstructured
  - Veracity – Uncertainty of Data
    - Poor Data Quality
    - Not sure of amount of inaccurate data



BigData in the Project

- With thousands of songs composed every year, there are more than a millions songs.
- In this project we are analyzing about 1 million songs composed from the year 1920s to present.
- Finding the songs of common tempo would be great challenge without any proper analysis.

BI ‘s Importance

- Why is BI Important
  - With BI's tools it is easy to use the raw data and transform them into meaningful and useful presentations and image files.
  - Output of the BI’s tool helps the organization in
    - making knowledgeable decisions.
      - To get a insight on the data helping in better decisions
    - Analyzing the data.
      - For data mining, predictive analytics
    - Measurements (Calculate Performance Metrics)
    - For Generating Reports
    - Collaboration platform
      - Making the inside and outside data into one.

BI on BigData

- Tradition BI not applicable for Big Data
  - Traditional BI vendors struggle with Big Data.
  - Traditional BI tools were not designed to work through massive data.

- Current Trend of BI tools
  - Latest BI tools like Microsoft SQL server 2012, Tableau, are creating solutions which make it easy to explore data and analyze them efficiently.
  - They have two sections to analyze the data,
    - Data Discovery – finds interesting data from huge amount of data
    - Analyzes – Analyze the interesting data to form meaningful and useful data.

Analysis in the Project

- Our aim is to analyze the data and help in development of personalized music applications which includes advanced music play listing, personalized radio capabilities, taste profiling, etc.
- We propose an analysis on the dataset on tempo of songs in different eras.
- We are using the analysis to aggregate the similar tracks at one place. This will help in recommending tracks to listeners based on their preferences.

Why this analysis / Benefits

- Music Industry's major challenge - catering to audience tastes.
- No more than 5% of all records released by major labels become gold or platinum hits
- The analysis will help Record Labels to find correlation between song tempo and popularity.
- Aspiring artists often find it difficult to assess what kind of music they should create to satisfy audiences and, therefore, be picked up by labels.
- Furthermore, music recommenders such as Pandora and iHeartRadio, Rdio, SiriusXM, Spotify could use our findings to recommend songs to their listeners based on their previous liking and disliking.

Hadoop

- Apache Hadoop is an open-source software framework for storage and large-scale processing of data-sets on clusters of commodity hardware.
  - Open Source Apache project.
  - Written in Java
- The Apache Hadoop framework is composed of the following modules:
  - Hadoop Common – contains libraries and utilities needed by other Hadoop modules
  - Hadoop Distributed File System (HDFS) – a distributed file-system that stores data on commodity machines, providing very high aggregate bandwidth across the cluster.
  - Hadoop YARN – a resource-management platform responsible for managing compute resources in clusters and using them for scheduling of users' applications.
  - Hadoop MapReduce – a programming model for large scale data processing.

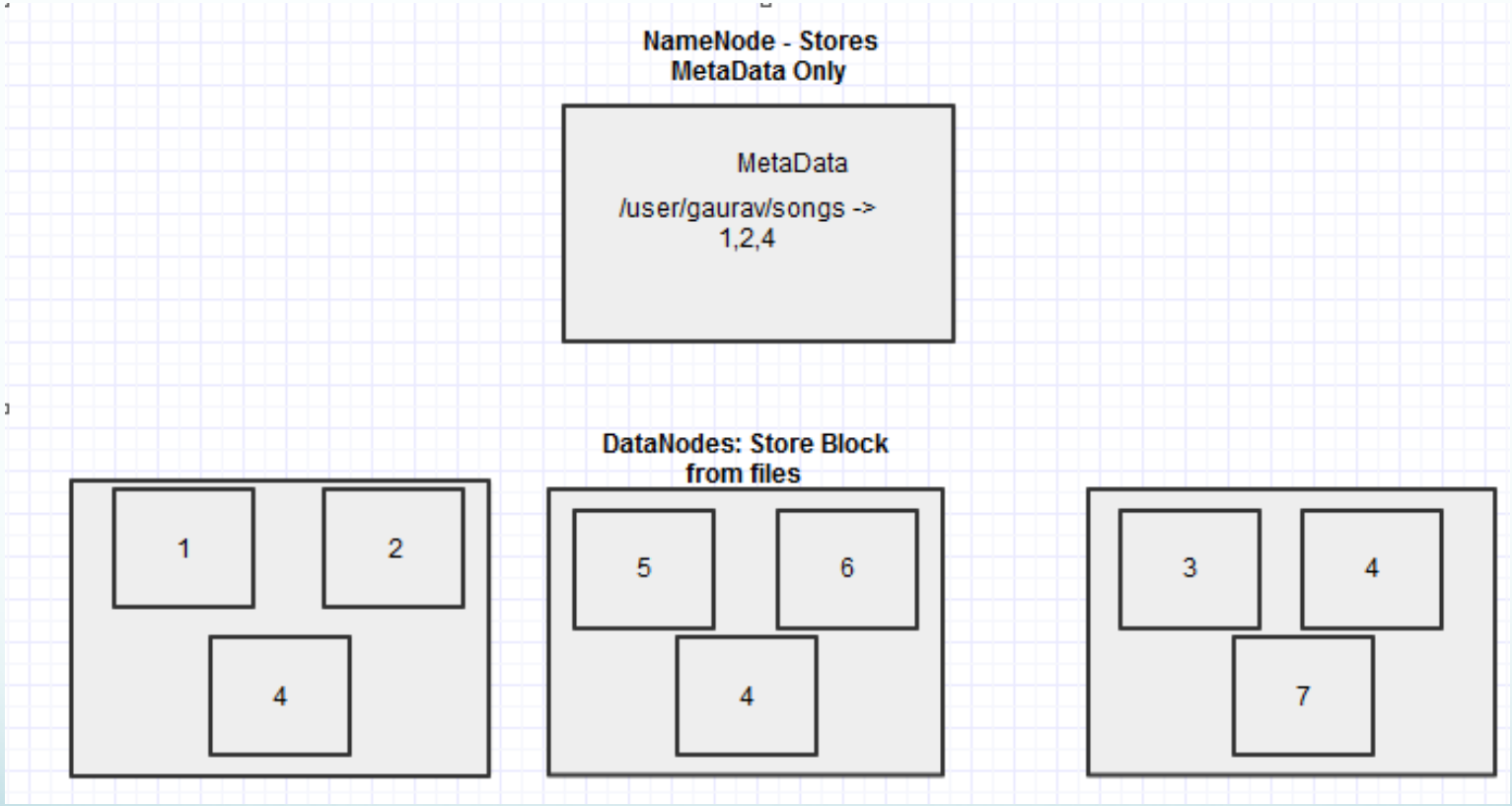
Hadoop Distributed File System

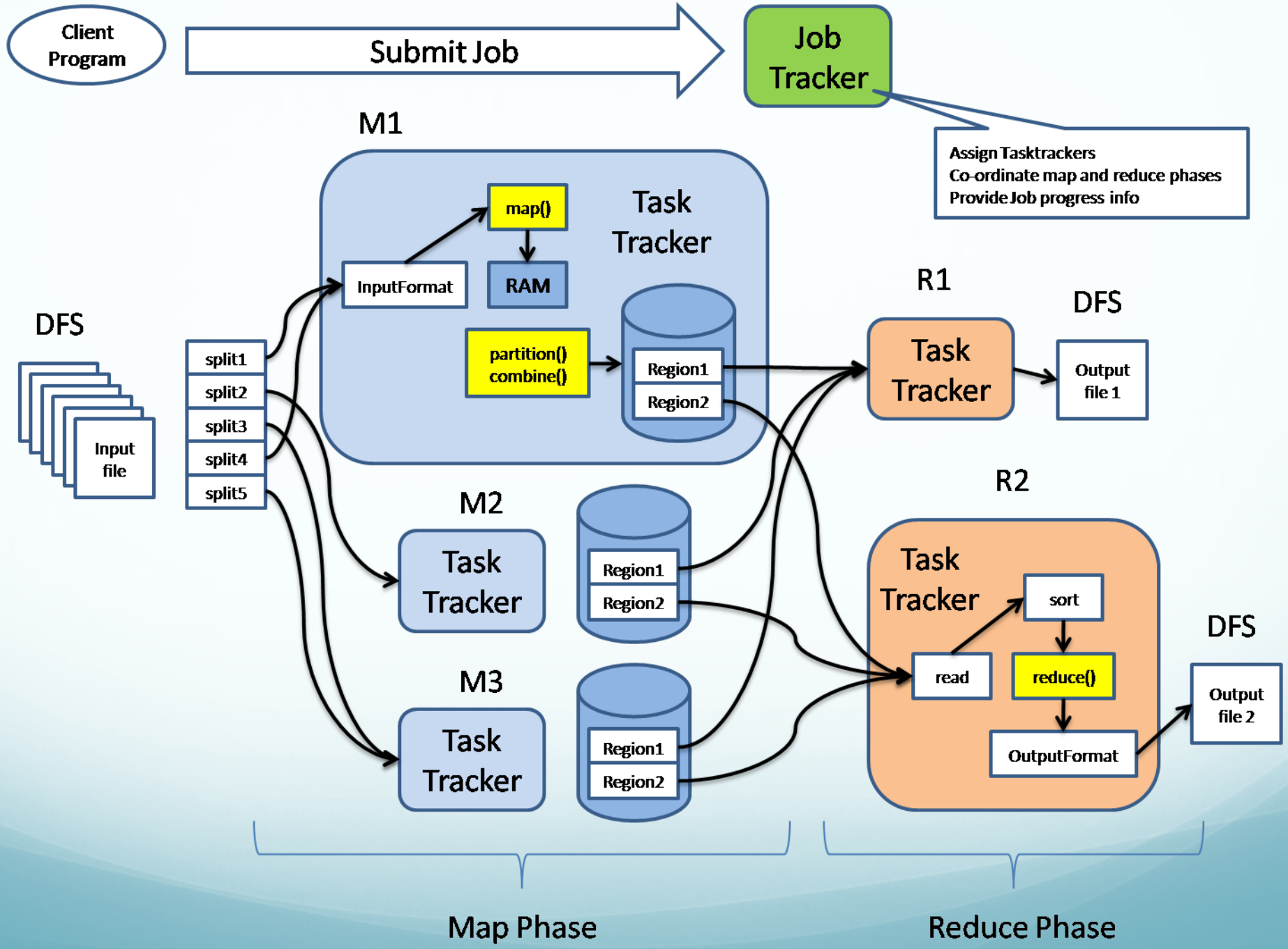
- Distributed file system designed to hold very large amounts of data (terabytes or even petabytes). This requires spreading the data across a large number of machines.
- Provide high-throughput access to this information.
- Files are stored in a redundant fashion across multiple machines to ensure their durability to failure and high availability to very parallel applications.
- HDFS should provide fast, scalable access to this information. It should be possible to serve a larger number of clients by simply adding more machines to the cluster.

HDFS - Details

- HDFS is a block structured file system. Individual files are broken down into blocks of fixed size.
- HDFS files are not a part of ordinary file system. HDFS comes with its own utilities for file management.
- A small Hadoop cluster includes a single master and multiple worker nodes.
- The master node consists of a JobTracker, TaskTracker, NameNode and DataNode.
- A slave or worker node acts as both a DataNode and TaskTracker, though it is possible to have data-only worker nodes and compute-only worker nodes
- NameNode – Namenode stores all the metadata for the file system
- DataNodes – Individual machines in the cluster used to store file blocks.
- HDFS replicates each block across a number of machines (3 by default).

HDFS Architecture




Client Program
Submit Job
Job Tracker
Assign Tasktrackers
Co-ordinate map and reduce phases
Provide Job progress info
M1
map()
Task Tracker
InputFormat
RAM
partition()
combine()
Region1
Region2
DFS
Input file
split1
split2
split3
split4
split5
R1
DFS
Task Tracker
Output file 1
M2
Task Tracker
Region1
Region2
R2
Task Tracker
sort
read
reduce()
OutputFormat
DFS
Output file 2
M3
Task Tracker
Region1
Region2
Map Phase
Reduce Phase

Workflow

- Master Program handles the Map Reduce Job
- The master program initiates the Map job at first.
- The input reader retrieves the data from the input directory.
- The input reader then splits the data into small chunks and submits them to randomly chosen mapper programs
- After receiving the data, the mapper program executes a user supplied map function, and generates a collection of [key, value] pairs
- Each produced item is sorted and submitted to the reducer.
- The reducer program collects all the items with the same key values and invokes a user supplied reduce function to produce a single entity as a result
- The output of the reduce program is collected by the output writer and this process basically terminates the parallel processing phase.

Map Reduce

- Hadoop MapReduce is a software framework for easily writing applications which process vast amounts of data (multi-terabyte data-sets) in-parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner.

- A MapReduce *job* usually splits the input data-set into independent chunks which are processed by the *map tasks* in a completely parallel manner. The framework sorts the outputs of the maps, which are then input to the *reduce tasks*. Typically both the input and the output of the job are stored in a file-system. The framework takes care of scheduling tasks, monitoring them and re-executes the failed tasks.

Map Reduce Continued

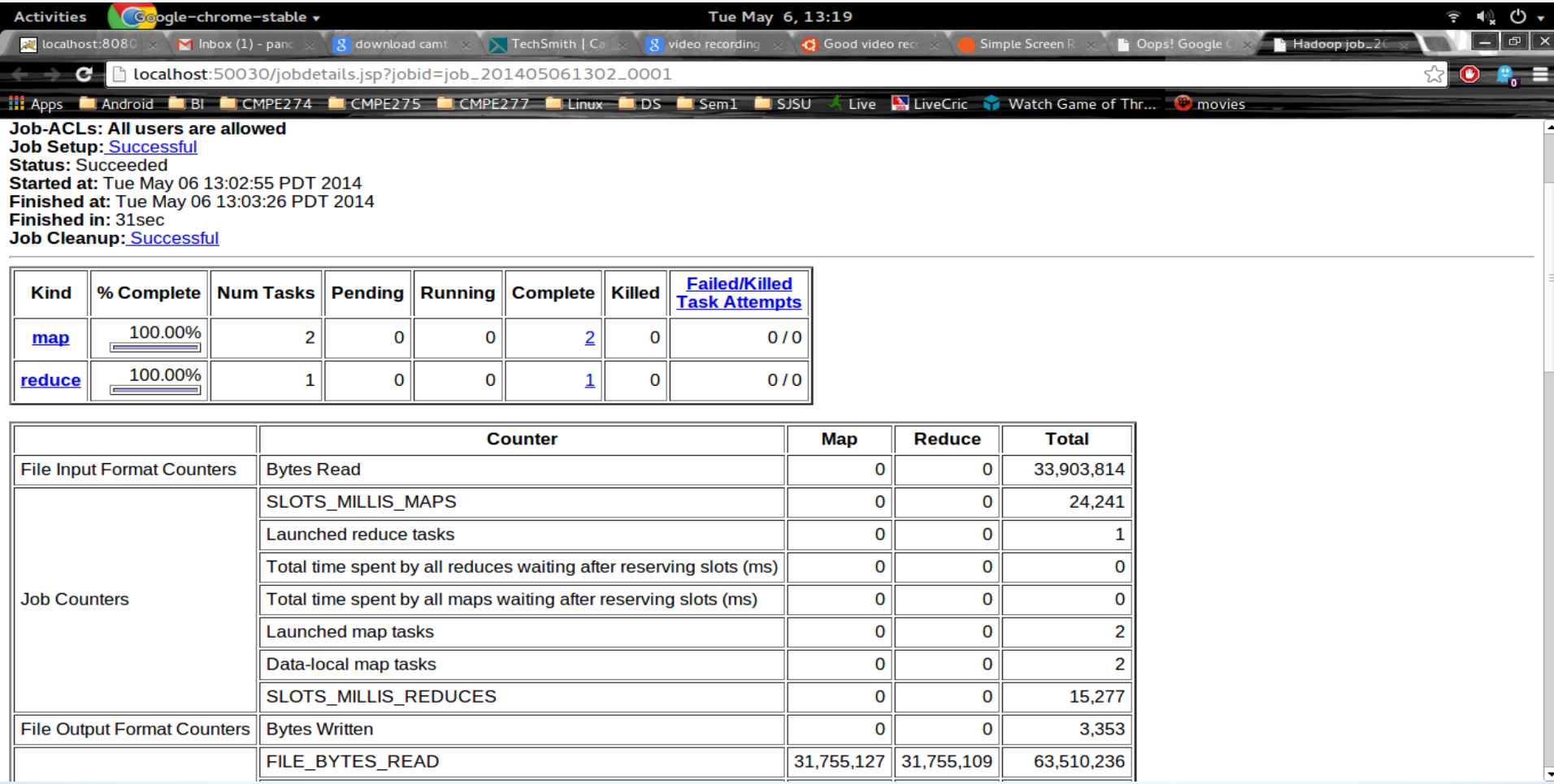
- The MapReduce framework consists of a single master JobTracker and one slave TaskTracker per cluster-node.
- The master is responsible for scheduling the jobs component tasks on the slaves, monitoring them and re-executing the failed tasks. The slaves execute the tasks as directed by the master.
- The MapReduce framework operates exclusively on <key, value> pairs, that is, the framework views the input to the job as a set of <key, value> pairs and produces a set of <key, value> pairs as the output of the job, conceivably of different types.

Technology stack

- Apache Hadoop
- Python scripting language
- MapReduce
- Twitter Bootstrap (for Website)
- HighChartJS (for Graph)

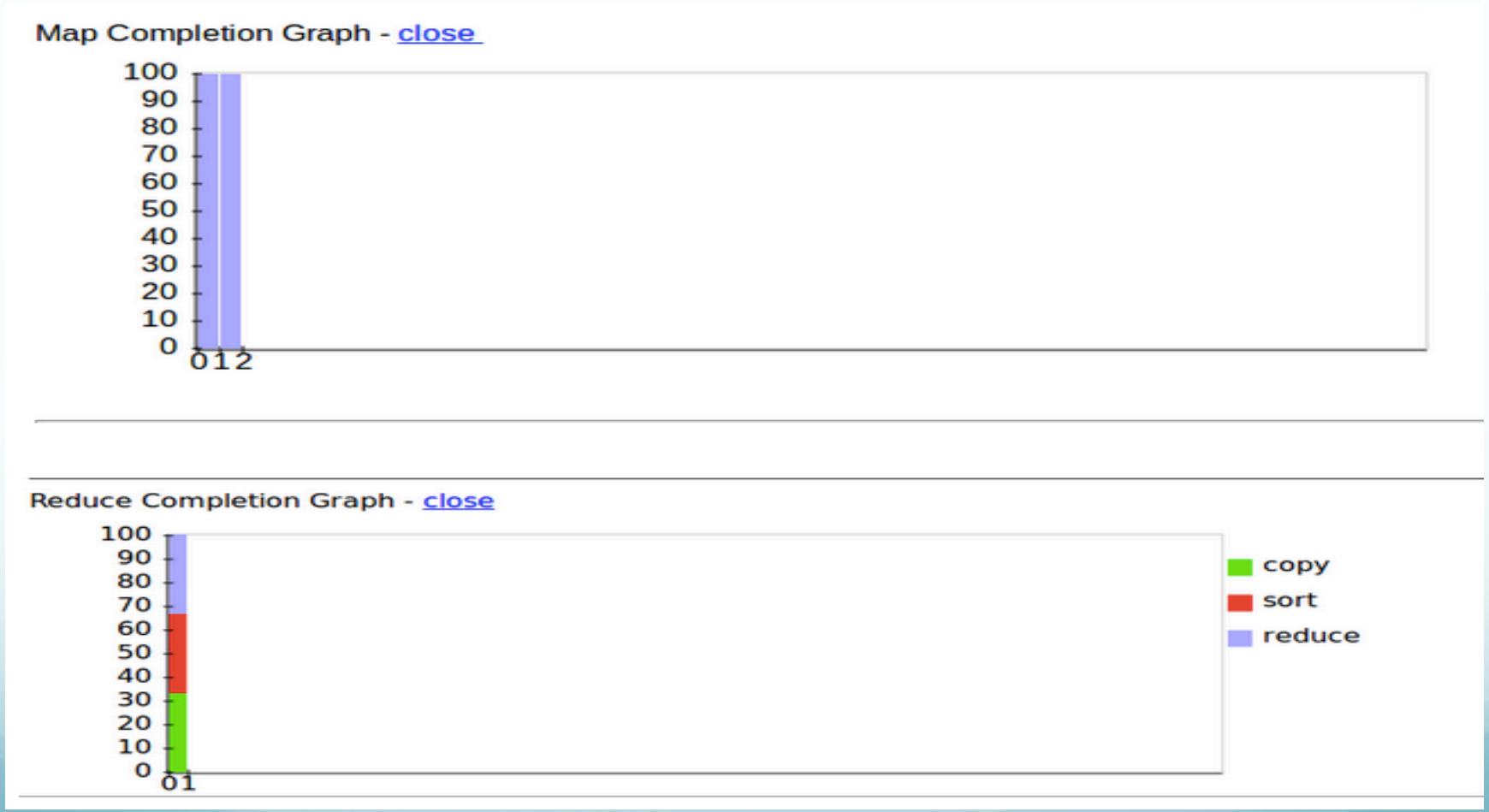
Observations

- A few observations made while working on the project.
- These are the observations while Hadoop was running on a single cluster node.

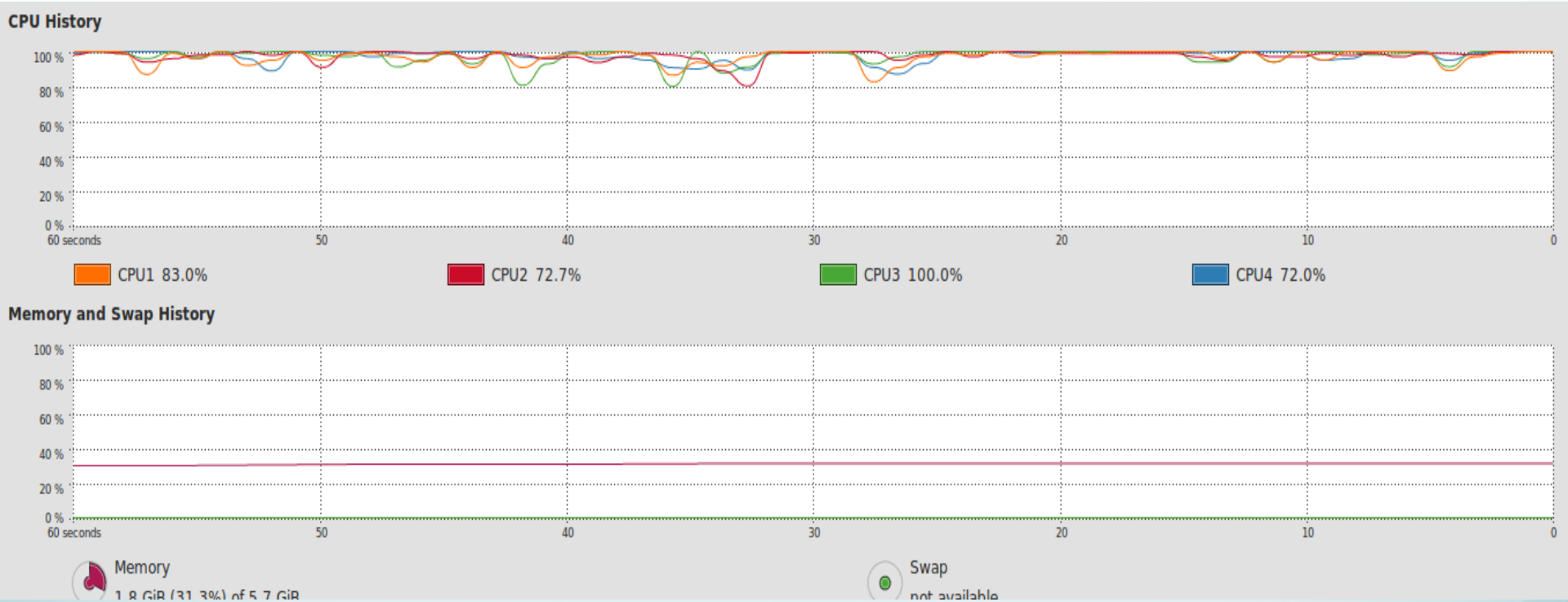


The above figure shows that 2 Map job and 1 Reduce job will be there to complete the task of analyzing the data

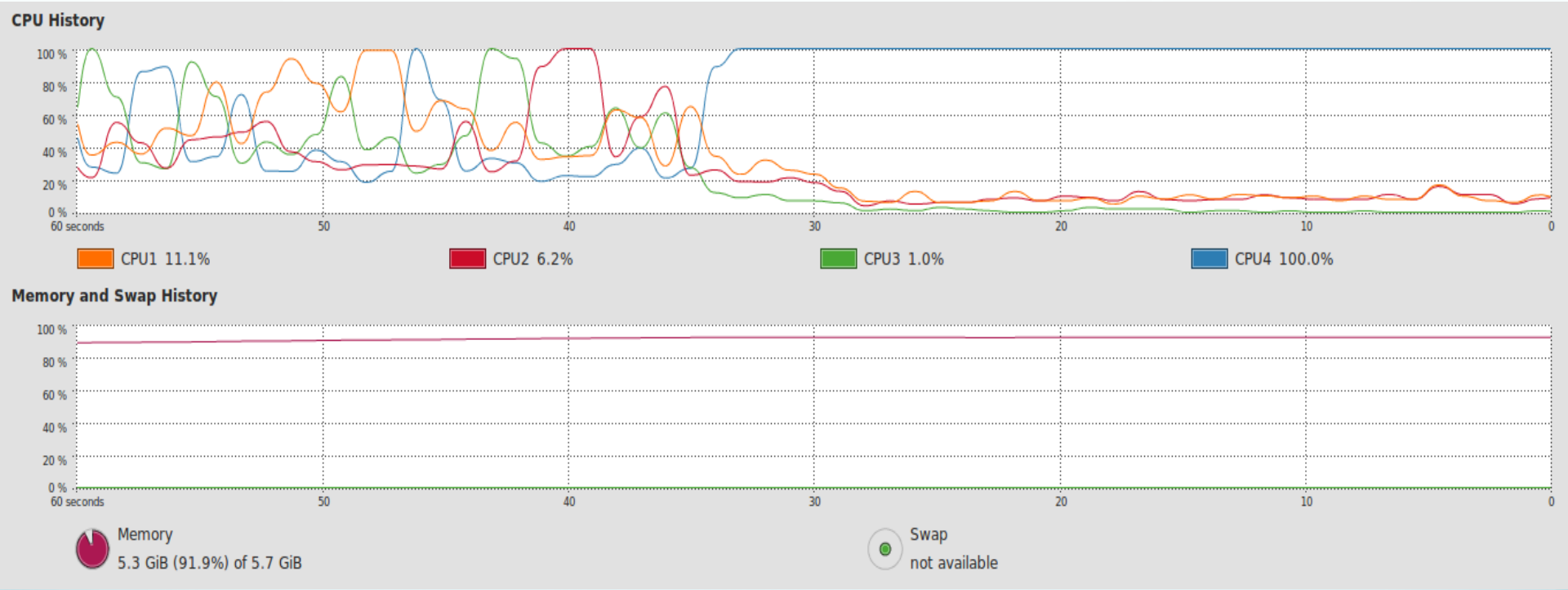
Graph when jobs were completed



CPU Utilization : During Mapper Jobs



CPU Utilization : During Reduce Jobs



Demo

Future Enhancement

Several ways to expand on what we have done :

- Our analysis could be used by recommendation services such as Pandora and iHeartRadio to generate better suggestions for listeners. However, we could also create a recommendation service of our own based on the results generated using predictive analysis.
- There are more track attributes in the Million Song Dataset that we can aggregate to get more accurate recommendation.
- There is also the possibility of finding more song data. It may have more tracks or more attributes than what we used for this analysis.

Conclusion

- Over the course of this project, we have managed to take a large, complex datasets and from that compute information that market researchers in the music industry can understand and put to good use in analyzing industry trends and making recommendations to listeners.

- Using MapReduce via Hadoop proved to be an ideal solution to manipulating massive amounts of data, as it allowed us to work through a big data to deliver results within minutes. Any other approach like using RDBMS would have been hopelessly inefficient.

- Making this project help us to understand well about a lot of new technologies like Hadoop, Map Reduce.

References

- Bertin-Mahieux, Thierry, Daniel P. W. Ellis, Brian Whitman, & Paul Lamere. “The Million Song Dataset,” ISMIR 2011: Proceedings of the 12th International Society for Music Information Retrieval Conference, October 24-28, 2011, Miami, Florida (2011).
- Liang,Dawen,HaijieGu,andBrendanO’Connor."Music Genre Classification with the Million Song Dataset." Machine Learning Department, CMU (2011).
- Density of Song with Million Song Dataset. http://musicmachinery.com/2011/09/04/how-to-process-a- million-songs-in-20-minutes/
- Courtney Love’s Letter to Recording Artists, Gerry Hemingway, [online] http:// www.gerryhemingway.com/piracy2.html (Accessed: 28 November 2013).
- Gettingthedataset,MillionSongDataset,[online] http://labrosa.ee.columbia.edu/ millionsong/pages/getting- dataset (Accessed: 12 October 2013).
- http://www.ibmbigdatahub.com/sites/default/files/infographic_file/4-Vs-of-big-data.jpg
- http://en.wikipedia.org/wiki/Business_intelligence
- http://www.applieddatalabs.com/content/new-reality-business-intelligence-and-big-data
- www.highcharts.com
- http://en.wikipedia.org/wiki/Apache_Hadoop
- http://hadoop.apache.org/