

A

PROJECT REPORT ON

ESDL

**MOVIE REVIEW PORTAL**

SUBMITTED BY

**ALOK SINGH**

UNDER THE GUIDANCE OF

**MANIK HENDRE**

For the partial fulfilment of

T.E (Computer Engineering) 2015-2016

TO

**DEPARTMENT OF COMPUTER ENGINEERING**

**ARMY INSTITUTE OF TECHNOLOGY**

**DIGHI HILLS, PUNE 411015**

UNDER

**University of Pune**

**DEPARTMENT OF COMPUTER ENGINEERING**

ARMY INSTITUTE OF TECHNOLOGY

DIGHI HILLS, PUNE 411015

2015-2016

**ACKNOWLEDGEMENT**

We take this opportunity to express our gratitude to **MANIK HENDRE**, our project guide, Department of Computer Engineering, Army Institute of Technology, Pune.

We would like to express our profound thanks to our professors of Department of Computer Engineering especially to **Prof. S R Dhore**, H.O.D (COMP), for their valuable guidance and keen interest, which motivated us to put in our best.

**ALOK SINGH**

**CERTIFICATE**

This is to certify that **A(3303)** have successfully submitted their project report on **MOVIE REVIEW PORTAL** under the academic year 2015-2016 in the partial fulfillment towards completion of **TE** in the Computer Engineering under University of Pune.

**MANIK HENDRE** **PROF. S R Dhore**

Guide Head of

Department Department

Computer Engineering Computer Engineering

**ABSTRACT**

*Our topic is****Hadoop****which is cluster computing framework. Apache Hadoop is a software framework that supports data-intensive distributed applications under a free license. Hadoop was inspired by Google's MapReduce and Google File System (GFS) papers. Hadoop, however, was designed to solve a different problem: the fast, reliable analysis of both structured data and complex data. As a result, many enterprises deploy Hadoop alongside their legacy IT systems, which allows them to combine old data and new data sets in powerful new ways. The Hadoop framework is used by major players including****Google****,****Yahoo****and****IBM****, largely for applications involving search engines and advertising. I am going to represent the History, Development and Current Situation of this Technology. This technology is now under the Apache Software Foundaion via Clodera.*

**Contents**

Chapter 1: Introduction to HADOOP............................................................... 5

Chapter 2: History of HADOOP....................................................................... 6

Chapter 3: Key Technology............................................................................. 7

3.1 MapReduce................................................................................... 7

3.2 HDFS (Hadoop Distributed File System)....................................... 8

Chapter 4: HADOOP Single Node Setup.......................................................... 9

Chapter 5: Movie Review Portal ..................................................................... 11

Bibliography.................................................................................................... 22

**Chapter 1: Introduction to HADOOP**

Today, we’re surrounded by data. People upload videos, take pictures on their cell phones, text friends, update their Facebook status, leave comments around the web, click on ads, and so forth. Machines, too, are generating and keeping more and more data.

The exponential growth of data first presented challenges to cutting-edge businesses such as Google, Yahoo, Amazon, and Microsoft. They needed to go through terabytes and petabytes of data to figure out which websites were popular, what books were in demand, and what kinds of ads appealed to people. Existing tools were becoming inadequate to process such large data sets. Google was the first to publicize *MapReduce—*a system they had used to scale their data processing needs.

This system aroused a lot of interest because many other businesses were facing similar scaling challenges, and it wasn’t feasible for everyone to reinvent their own proprietary tool. Doug Cutting saw an opportunity and led the charge to develop an open source version of this MapReduce system called Hadoop . Soon after, Yahoo and others rallied around to support this effort. Today, Hadoop is a core part of the computing infrastructure for many web companies, such as Yahoo , Facebook , LinkedIn , and Twitter. Many more traditional businesses, such as media and telecom, are beginning to adopt this system too**.**

Hadoop is an open source framework for writing and running distributed applications that process large amounts of data. Distributed computing is a wide and varied field, but the key distinctions of Hadoop are that it is

■ *Accessible*—Hadoop runs on large clusters of commodity machines or on cloud computing services such as Amazon’s Elastic Compute Cloud (EC2 ).

■ *Robust*—Because it is intended to run on commodity hardware, Hadoop is architected with the assumption of frequent hardware malfunctions. It can gracefully handle most such failures.

■ *Scalable*—Hadoop scales linearly to handle larger data by adding more nodes to the cluster.

■ *Simple*—Hadoop allows users to quickly write efficient parallel code.

Hadoop’s accessibility and simplicity give it an edge over writing and running large distributed programs. Even college students can quickly and cheaply create their own Hadoop cluster. On the other hand, its robustness and scalability make it suitable for even the most demanding jobs at Yahoo and Facebook. These features make Hadoop popular in both academia and industry.

**Chapter 2: History of HADOOP**

Hadoop was created by Doug Cutting, the creator of Apache Lucene, the widely used text search library. Hadoop has its origins in Apache Nutch, an open source web search engine, itself a part of the Lucene project.

**The Origin of the Name “Hadoop”:**

The name Hadoop is not an acronym; it’s a made-up name. The project’s creator, Doug Cutting, explains how the name came about:

*The name my kid gave a stuffed yellow elephant. Short, relatively easy to spell and pronounce, meaningless, and not used elsewhere: those are my naming criteria. Kids are good at generating such. Googol is a kid’s term.*

Subprojects and “contrib” modules in Hadoop also tend to have names that are unrelated to their function, often with an elephant or other animal theme (“Pig,” for example). Smaller components are given more descriptive (and therefore more mundane) names. This is a good principle, as it means you can generally work out what something does from its name. For example, the jobtracker keeps track of MapReduce jobs.

Building a web search engine from scratch was an ambitious goal, for not only is the software required to crawl and index websites complex to write, but it is also a challenge to run without a dedicated operations team, since there are so many moving parts. It’s expensive too: Mike Cafarella and Doug Cutting estimated a system supporting a 1- billion-page index would cost around half a million dollars in hardware, with a monthly running cost of $30,000. Nevertheless, they believed it was a worthy goal, as it would open up and ultimately democratize search engine algorithms. Nutch was started in 2002, and a working crawler and search system quickly emerged.

However, they realized that their architecture wouldn’t scale to the billions of pages on the Web. Help was at hand with the publication of a paper in 2003 that described the architecture of Google’s distributed filesystem, called GFS, which was being used in production at Google.# GFS, or something like it, would solve their storage needs for the very large files generated as a part of the web crawl and indexing process. In particular, GFS would free up time being spent on administrative tasks such as managing storage nodes. In 2004, they set about writing an open source implementation, the Nutch Distributed Filesystem (NDFS).

In 2004, Google published the paper that introduced MapReduce to the world. Early in 2005, the Nutch developers had a working MapReduce implementation in Nutch, and by the middle of that year all the major Nutch algorithms had been ported to run using MapReduce and NDFS. NDFS and the MapReduce implementation in Nutch were applicable beyond the realm of search, and in February 2006 they moved out of Nutch to form an independent subproject of Lucene called Hadoop. At around the same time, Doug Cutting joined Yahoo!, which provided a dedicated team and the resources to turn Hadoop into a system that ran at web scale (see sidebar). This was demonstrated in February 2008 when Yahoo! announced that its production search index was being generated by a 10,000-core Hadoop cluster.†

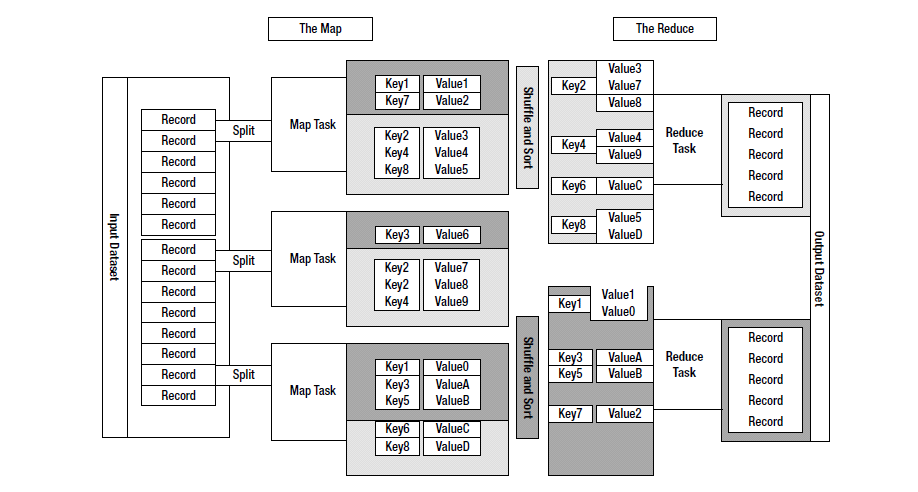
In January 2008, Hadoop was made its own top-level project at Apache, confirming its success and its diverse, active community. By this timem Hadoop was being used by many other companies besides Yahoo!, such as Last.fm, Facebook, and the *New York Times.*

**Chapter 3: Key Technology**

The key technology for Hadoop is the MapReduce programming model and Hadoop Distributed File System. The operation on large data is not possible in serial programming paradigm. MapReduce do task parallel to accomplish work in less time which is the main aim of this technology. MapReduce require special file system. In the real scenario , the data which are in terms on perabyte. To store and maintain this much data on distributed commodity hardware, Hadoop Distributed File System is invented. It is basically inspired by Google File System.

**3.1 MapReduce**

MapReduce is a framework for processing highly distributable problems across huge datasets using a large number of computers (nodes), collectively referred to as a cluster (if all nodes use the same hardware) or a grid (if the nodes use different hardware). Computational processing can occur on data stored either in a filesystem (unstructured) or in a database (structured).



**Figure 3.1 MapReduce Programming Model**

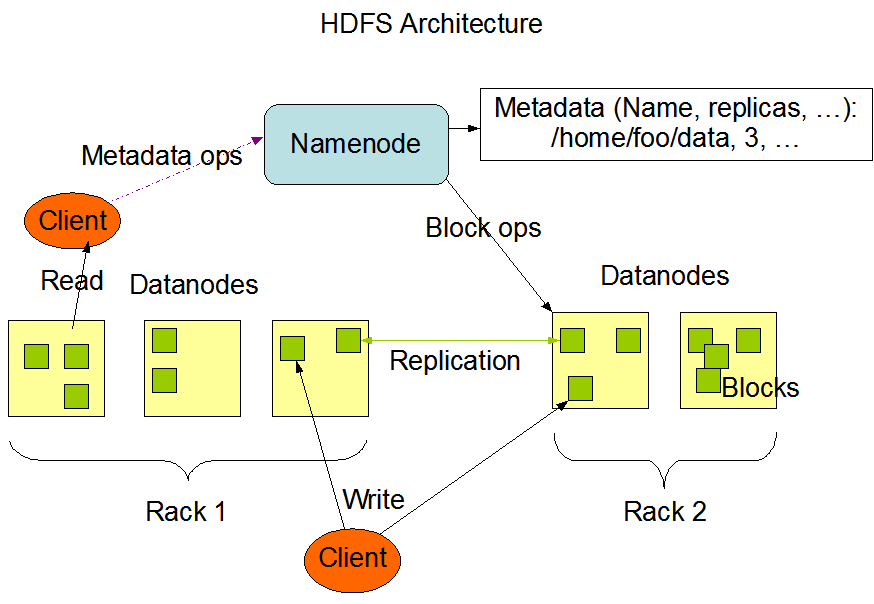
**"Map" step:** The master node takes the input, partitions it up into smaller sub-problems, and distributes them to worker nodes. A worker node may do this again in turn, leading to a multi-level tree structure. The worker node processes the smaller problem, and passes the answer back to its master node.

**"Reduce" step:** The master node then collects the answers to all the sub-problems and combines them in some way to form the output – the answer to the problem it was originally trying to solve.

MapReduce allows for distributed processing of the map and reduction operations. Provided each mapping operation is independent of the others, all maps can be performed in parallel – though in practice it is limited by the number of independent data sources and/or the number of CPUs near each source. Similarly, a set of 'reducers' can perform the reduction phase - provided all outputs of the map operation that share the same key are presented to the same reducer at the same time. While this process can often appear inefficient compared to algorithms that are more sequential, MapReduce can be applied to significantly larger datasets than "commodity" servers can handle – a large server farm can use MapReduce to sort a petabyte of data in only a few hours. The parallelism also offers some possibility of recovering from partial failure of servers or storage during the operation: if one mapper or reducer fails, the work can be rescheduled – assuming the input data is still available.

**3.2 HDFS (Hadoop Distributed File System)**

The Hadoop Distributed File System (HDFS) is a distributed file system designed to run on commodity hardware. It has many similarities with existing distributed file systems. However, the differences from other distributed file systems are significant. HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware. HDFS provides high throughput access to application data and is suitable for applications that have large data sets. HDFS relaxes a few POSIX requirements to enable streaming access to file system data. HDFS was originally built as infrastructure for the Apache Nutch web search engine project. HDFS is now an Apache Hadoop subproject.



**Figure 3.2 HDFS Architecture**

HDFS has a master/slave architecture. An HDFS cluster consists of a single NameNode, a master server that manages the file system namespace and regulates access to files by clients. In addition, there are a number of DataNodes, usually one per node in the cluster, which manage storage attached to the nodes that they run on. HDFS exposes a file system namespace and allows user data to be stored in files. Internally, a file is split into one or more blocks and these blocks are stored in a set of DataNodes. The NameNode executes file system namespace operations like opening, closing, and renaming files and directories. It also determines the mapping of blocks to DataNodes. The DataNodes are responsible for serving read and write requests from the file system’s clients. The DataNodes also perform block creation, deletion, and replication upon instruction from the NameNode.

The NameNode and DataNode are pieces of software designed to run on commodity machines. These machines typically run a GNU/Linux operating system (OS). HDFS is built using the Java language; any machine that supports Java can run the NameNode or the DataNode software. Usage of the highly portable Java language means that HDFS can be deployed on a wide range of machines. A typical deployment has a dedicated machine that runs only the NameNode software.

**Chapter 4: HADOOP Single Node Setup**

The steps involved in setting up a single node Hadoop cluster are as follow:

* Download the Hadoop Software, the hadoop.tar.gz file using the [**ftp://hadoop.apche.org**](ftp://hadoop.apche.org/) URL, and ensure that the software is installed on every node of the cluster. Installing the Hadoop Software on all the nodes require unpacking of the software, the hadoop.apache.org URL, on the nodes.
* Create the keys on local machine such that ssh, required by Hadoop, does not need password. Use following command to create key on local machine:

**$ ssh-keygen -t rsa -P “ “**

**$ cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys**

* Modify the environment parameters in the **hadoop-env.sh** file. Use the following command to change the environment parameter:

**Export JAVA\_HOME=/path/to/jdk\_home\_dir**

* Modify the configuration parameters in files given below as shown below.

Do the following changes to the configuration files under **hadoop/conf**

* **core-site.xml**

**<configuration>**

**<property>**

**<name>hadoop.tmp.dir</name>**

**<value>TEMPORARY-DIR-FOR-HADOOPDATASTORE</value>**

**</property>**

**<property>**

**<name>fs.default.name</name>**

**<value>hdfs://localhost:54310</value>**

**</property>**

**</configuration>**

* **mapred-site.xml**

**<configuration>**

**<property>**

**<name>mapred.job.tracker</name>**

**<value>localhost:54311</value>**

**</property>**

**</configuration>**

* **hdfs-site.xml**

**<configuration>**

**<property>**

**<name>dfs.replication</name>**

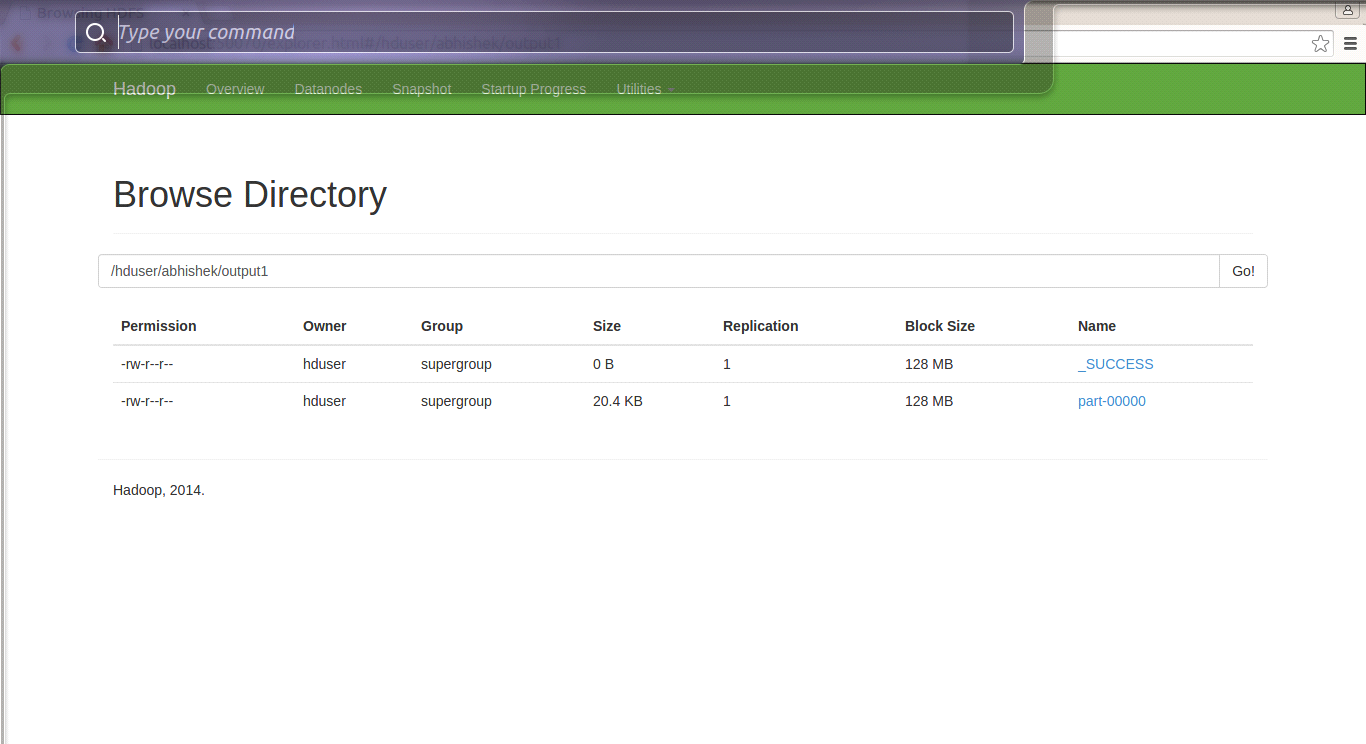
**<value>1</value>**

**</property>**

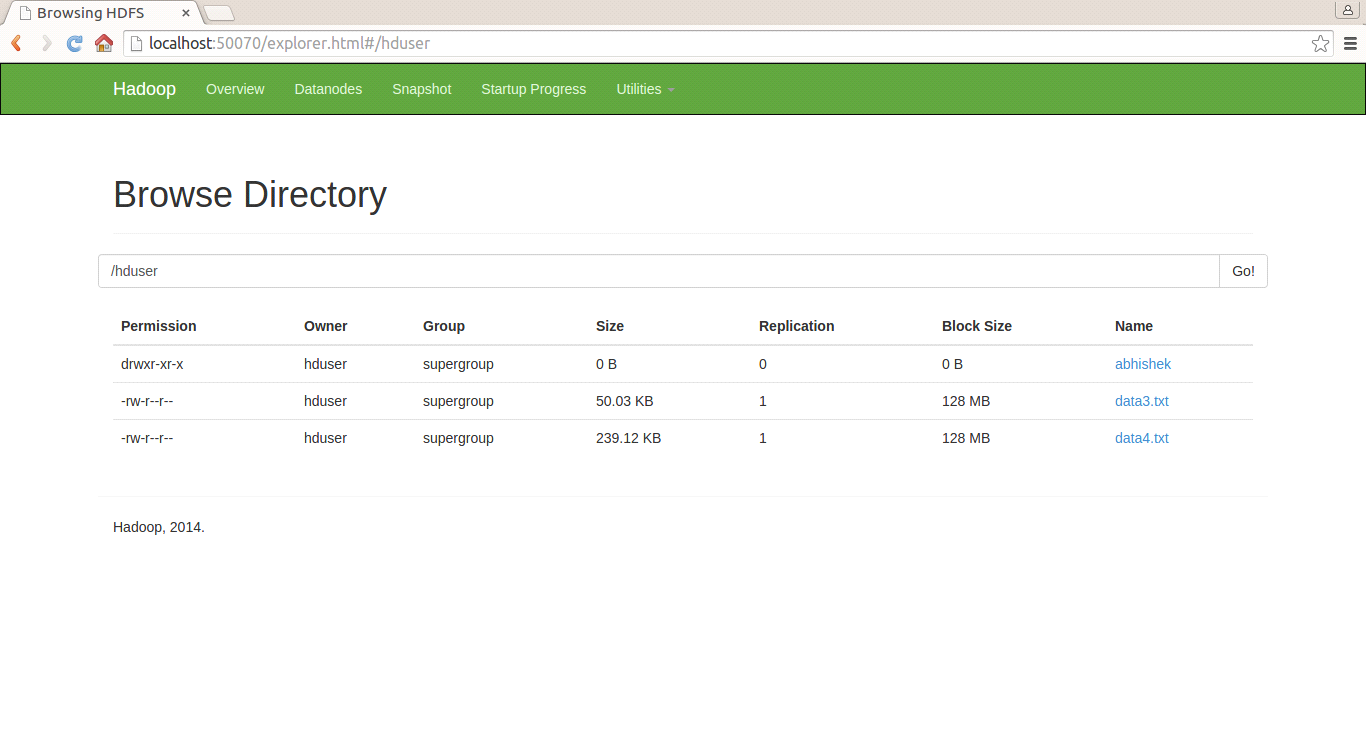
**</configuration>**

* Format the hadoop file system. From hadoop directory run the following

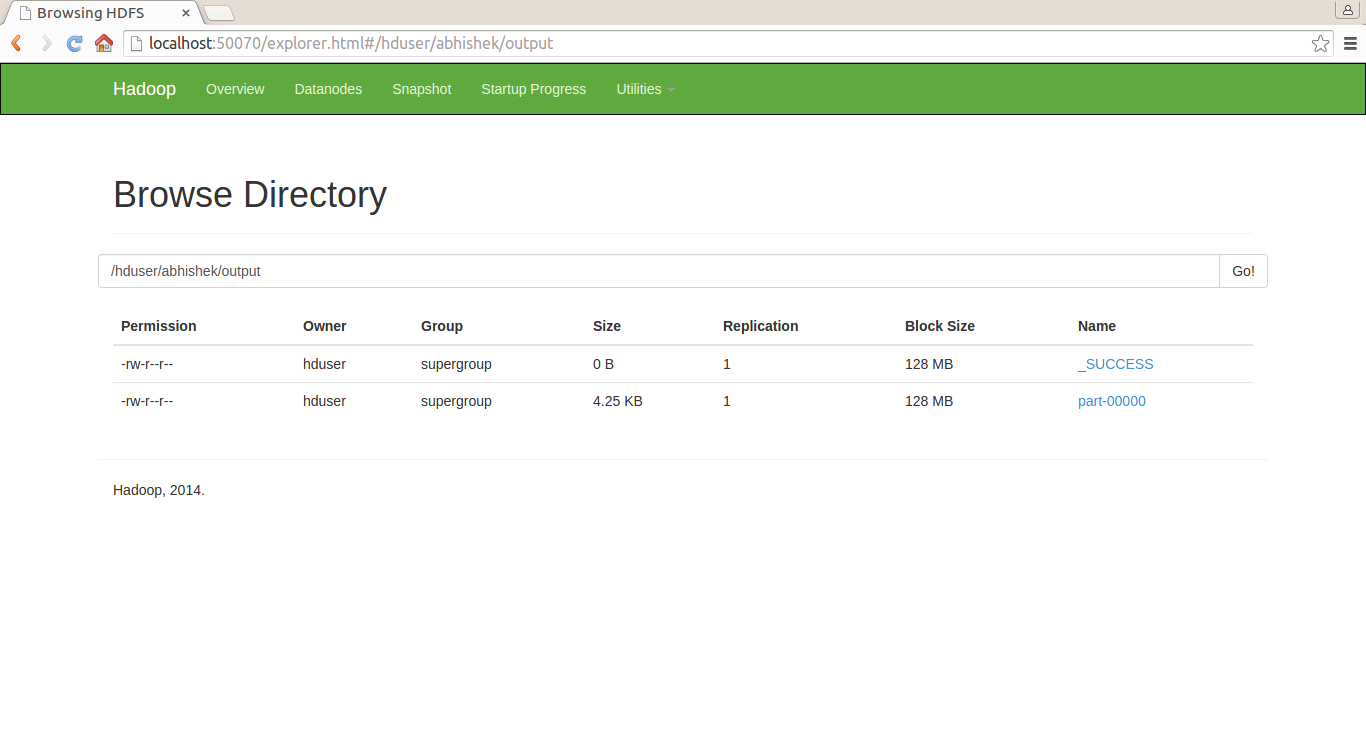
**bin/hadoop namenode –format**



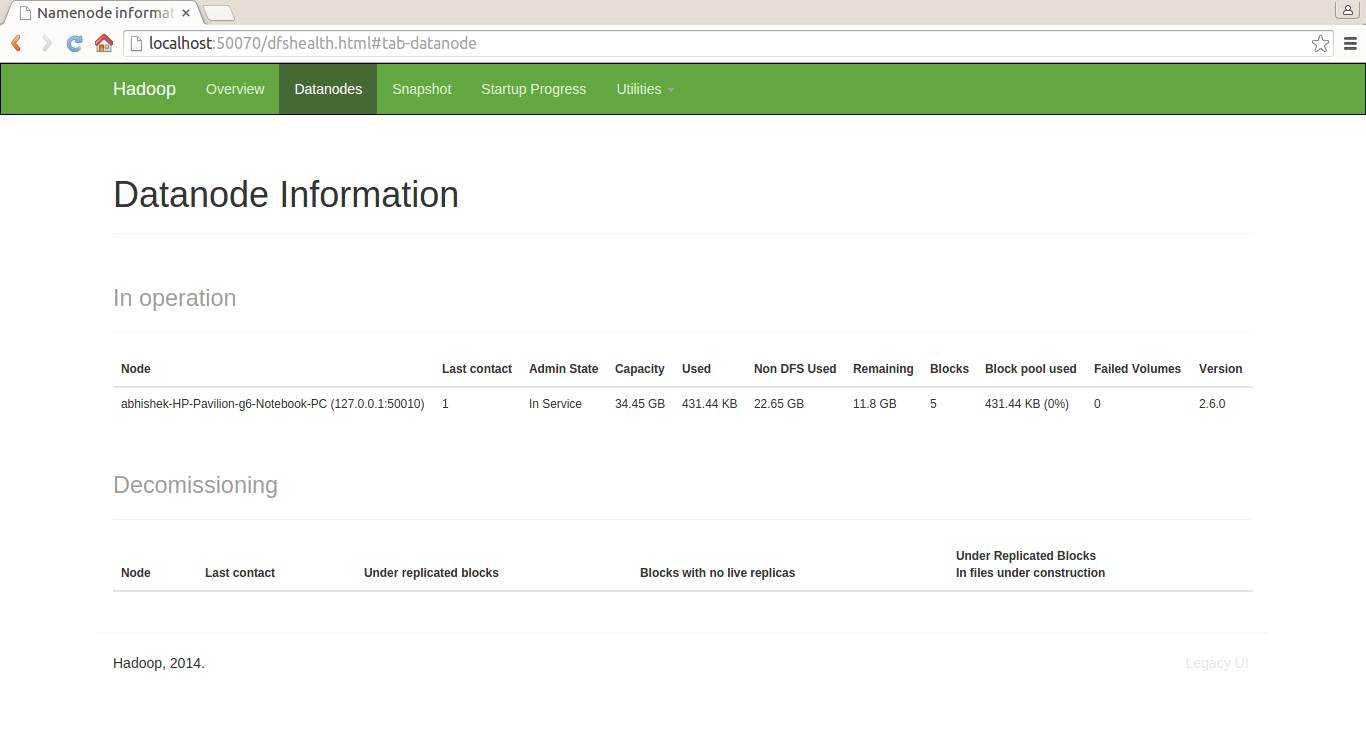
**HADOOP UTILITY FOR NAME NODE AND DATA NODE**



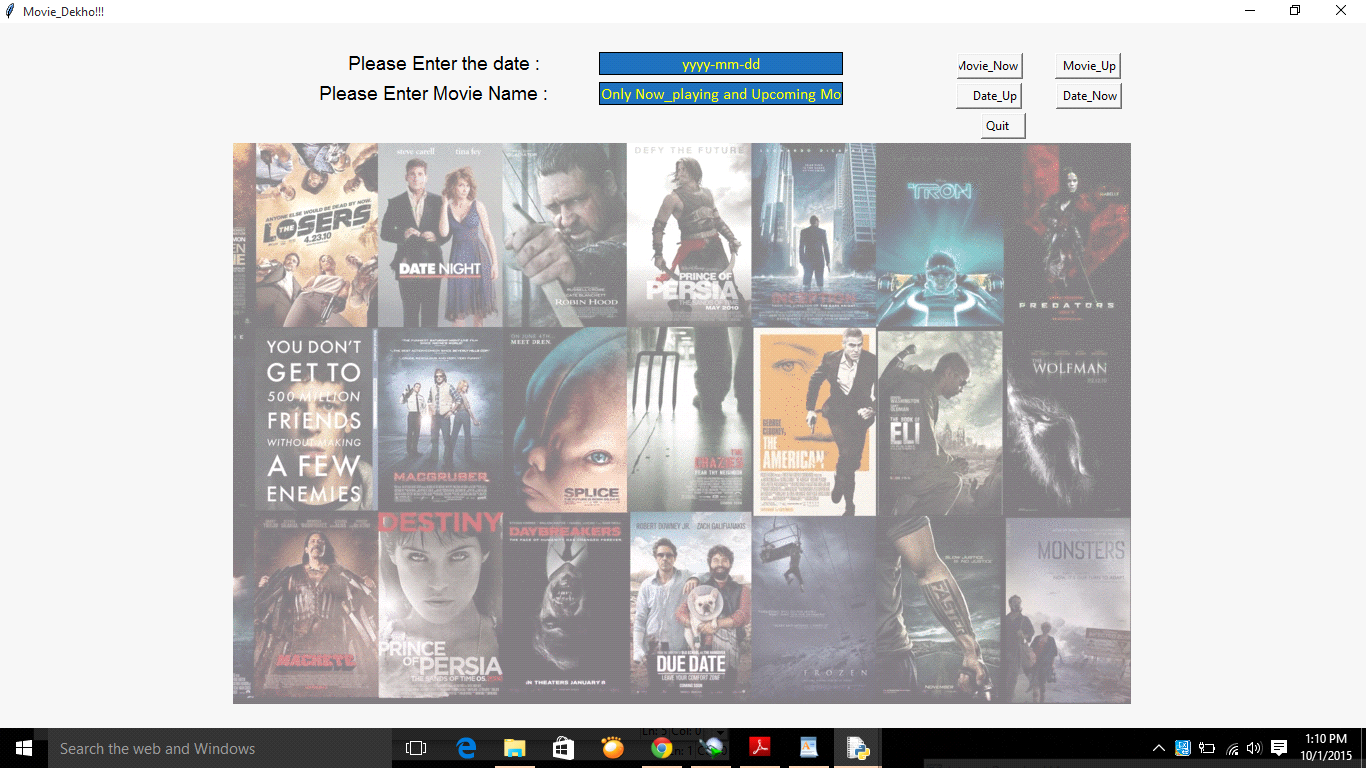
**DATA FILES UPLOADED ON HDFS**



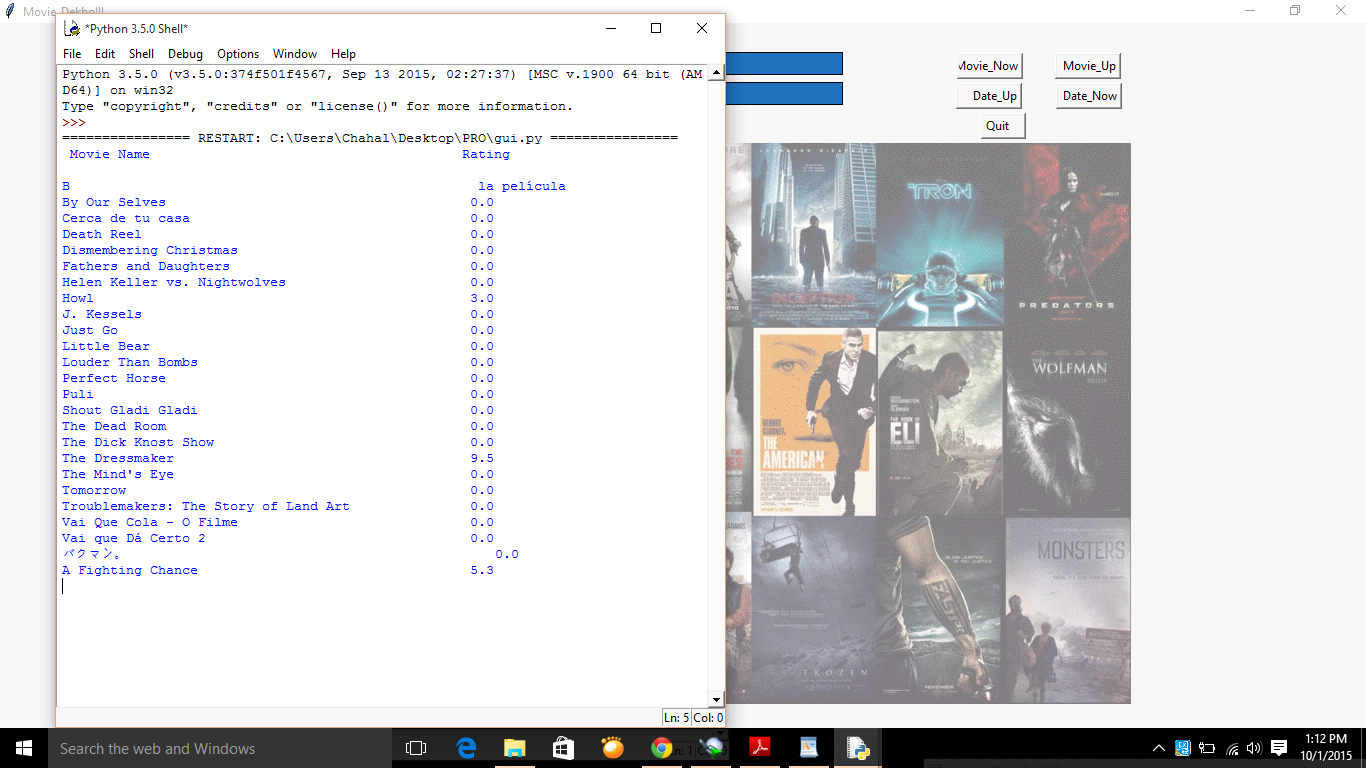
**OUTPUT FILE GENERATED ON HADOOP FRAMEWORK**



**DATA NODE INFORMATION**

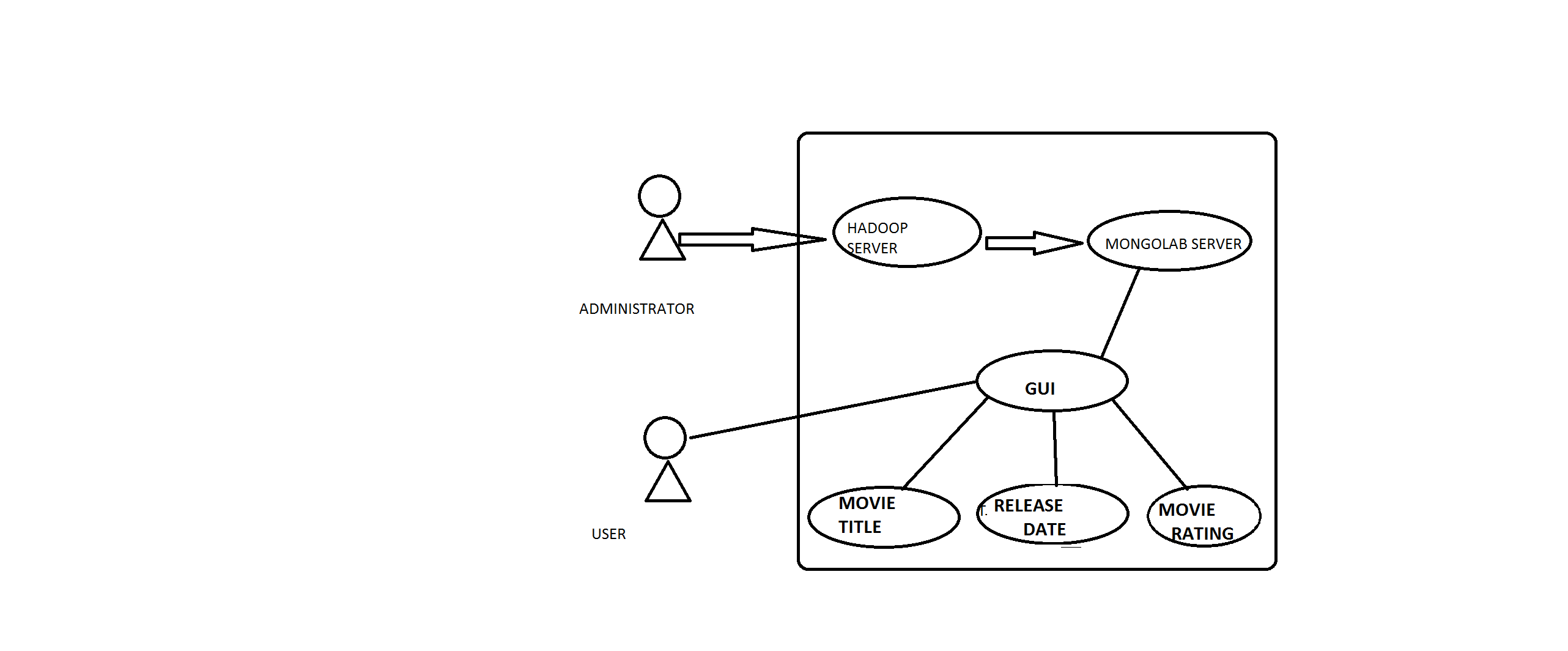


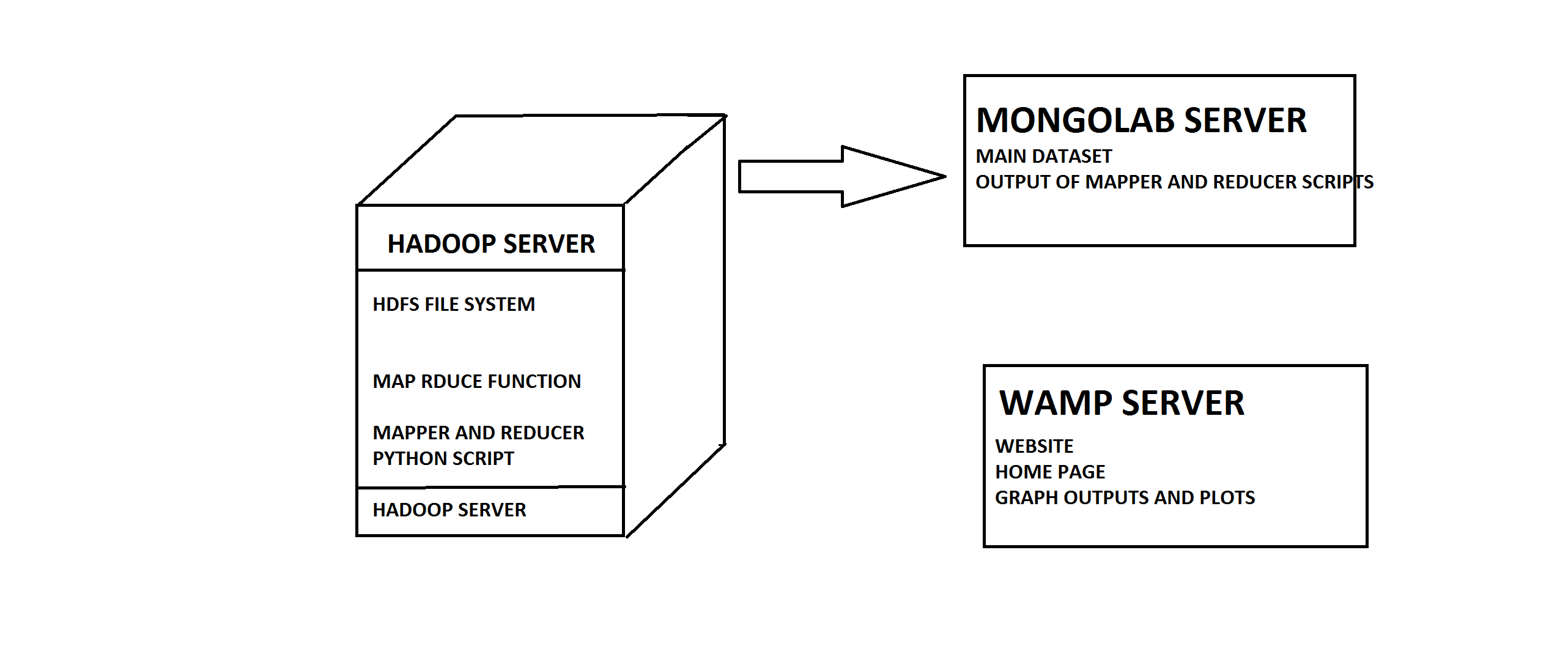
**GRAPHICAL USER INTERFACE**



**MOVIE NAME AND RATINGS INFORMATION**

**UML DIAGRAMS**





**Bibliography**

[1] Jason Venner, *Pro Hadoop*, Apress

[2] Tom White, *Hadoop: The Definitive Guide* , O’REILLY

[3] Chuck Lam, Hadoop in Action, MANNING

[4] Hadoop.apache.org

[5] sms-tsunami-warning.com

[6] Wikipedia.com