PROJECT

INSTALLATION OF HADOOP

10.1 Configuration

1. Installing VM Ware 12
2. Installing CentOS 6.7
3. Assigning proper ip Address for interconnection
4. Creating a master node
5. Copy master as slave node
6. Controlling all slaves from master using ssh Command
7. Copy Hadoop file to master node using Winscp
8. Install Hadoop on Master node
9. Install Hadoop to every node on cluster

10.2 Configuration file required for Hadoop installation

1. Hadoop Environmental Setup

1.2.1.1 .bash_profile

Open the .bash_profile file and copy the given lines to set Hadoop Environment Variable

```
export JAVA_HOME=/usr/java/jdk1.7.0_65
export HADOOP_HOME=/home/hadoop/hadoop
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
```

2. Hadoop Configuration Files

Fin d all the Hadoop configuration files on location

```
$HADOOP_HOME/etc/hadoop
```

core-site.xml

The core-site.xml file contains information such as the port number used for Hadoop instance, memory allocated for the file system, memory limit for storing the data, and size of Read/Write buffers.

Open the core-site.xml and add the following properties in between <configuration>, </configuration> tags.

```
<configuration>
<property>
  <name>fs.default.name</name>
    <value>hdfs://master1:9000</value>
    </property>
</configuration>
```

hdfs-site.xml

The hdfs-site.xml file contains information such as the value of replication data, namenode path, and datanode paths of your local file systems.

```
<configuration>
<property>
 <name>dfs.replication</name>
 <value>2</value>
</property>
<property>
  <name>dfs.name.dir</name>
    <value>file:///home/hadoop/hadoopdata/hdfs/namenode</value>
</property>
<property>
  <name>dfs.data.dir</name>
    <value>file:///home/hadoop/hadoopdata/hdfs/datanode</value>
</property>
</configuration>
```

mapred-site.xml

Copy the file from mapred-site,xml.template to mapred-site.xml file using the following command.

```
<configuration>
 <property>
  <name>mapreduce.framework.name</name>
   <value>yarn</value>
 </property>
</configuration>
```

yarn-site.xml

This file is used to configure yarn into Hadoop.

```
<configuration>
 <property>
  <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
 </property>
</configuration>
```

hadoop-env.sh

To develop Hadoop programs in java, you have to reset the java environment variables in hadoop-env.sh file by replacing JAVA_HOME value with the location of java in your system.

```
export JAVA_HOME=/usr/java/jdk1.7.0_65
```

3. To install Hadoop in slaves copy hadoop directory to slaves

```
scp -r hadoop slave1:/home/hadoop
scp .bash_profile slave1:/home/hadoop
scp -r hadoop slave2:/home/hadoop
scp .bash_profile slave2:/home/hadoop
```

10.3 Verify Hadoop Installation

1. Format the namenode

hdfs namenode -format

2. Start Hadoop

start-all.sh

3. Verification of working node

slaves.sh /usr/java/jdk1.7.0_65/bin/jps|sort

```
[hadoop@master1 ~]$ slaves.sh /usr/java/jdk1.7.0_65/bin/jps|sort
master1: 2900 NameNode
master1: 3004 DataNode
master1: 3153 SecondaryNameNode
master1: 3330 ResourceManager
master1: 3441 NodeManager
master1: 4659 Jps
slave1: 2718 DataNode
slave1: 2831 NodeManager
slave1: 3398 Jps
slave2: 2575 DataNode
slave2: 2651 NodeManager
slave2: 2801 Jps
```

Fig. 9.1 Verification of nodes

10.4 Report file system info and statistics

hadoop dfsadmin -report|more

```
[hadoop@master1 ~]$ hadoop dfsadmin -report|more
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.

Configured Capacity: 65021792256 (60.56 GB)
Present Capacity: 46213558272 (43.04 GB)
DFS Remaining: 41492791296 (38.64 GB)
DFS Used: 4720766976 (4.40 GB)
DFS Used%: 10.22%
Under replicated blocks: 1057
Blocks with corrupt replicas: 0
Missing blocks: 1

-------------------------------------------------
Datanodes available: 3 (3 total, 0 dead)

Live datanodes:
Name: 4.4.4.102:50010 (slave2)
Hostname: slave2
Decommission Status : Normal
Configured Capacity: 21673930752 (20.19 GB)
DFS Used: 1489973248 (1.39 GB)
Non DFS Used: 5413949440 (5.04 GB)
DFS Remaining: 14770008064 (13.76 GB)
DFS Used%: 6.87%
DFS Remaining%: 68.15%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Last contact: Sun Oct 09 03:06:19 IST 2016

Name: 4.4.4.100:50010 (master1)
Hostname: master1
Decommission Status : Normal
Configured Capacity: 21673930752 (20.19 GB)
DFS Used: 1588498432 (1.48 GB)
Non DFS Used: 7976669184 (7.43 GB)
DFS Remaining: 12108763136 (11.28 GB)
DFS Used%: 7.33%
DFS Remaining%: 55.87%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Last contact: Sun Oct 09 03:06:19 IST 2016


Name: 4.4.4.101:50010 (slave1)
Hostname: slave1
Decommission Status : Normal
Configured Capacity: 21673930752 (20.19 GB)
DFS Used: 1642295296 (1.53 GB)
Non DFS Used: 5417615360 (5.05 GB)
DFS Remaining: 14614020096 (13.61 GB)
DFS Used%: 7.58%
DFS Remaining%: 67.43%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
```

Fig. 9.2 Filesystem Info Statistic and

10.5 Access Hadoop on browser

Use the the given below URL to access hadoop on browser

http://4.4.4.100:50070

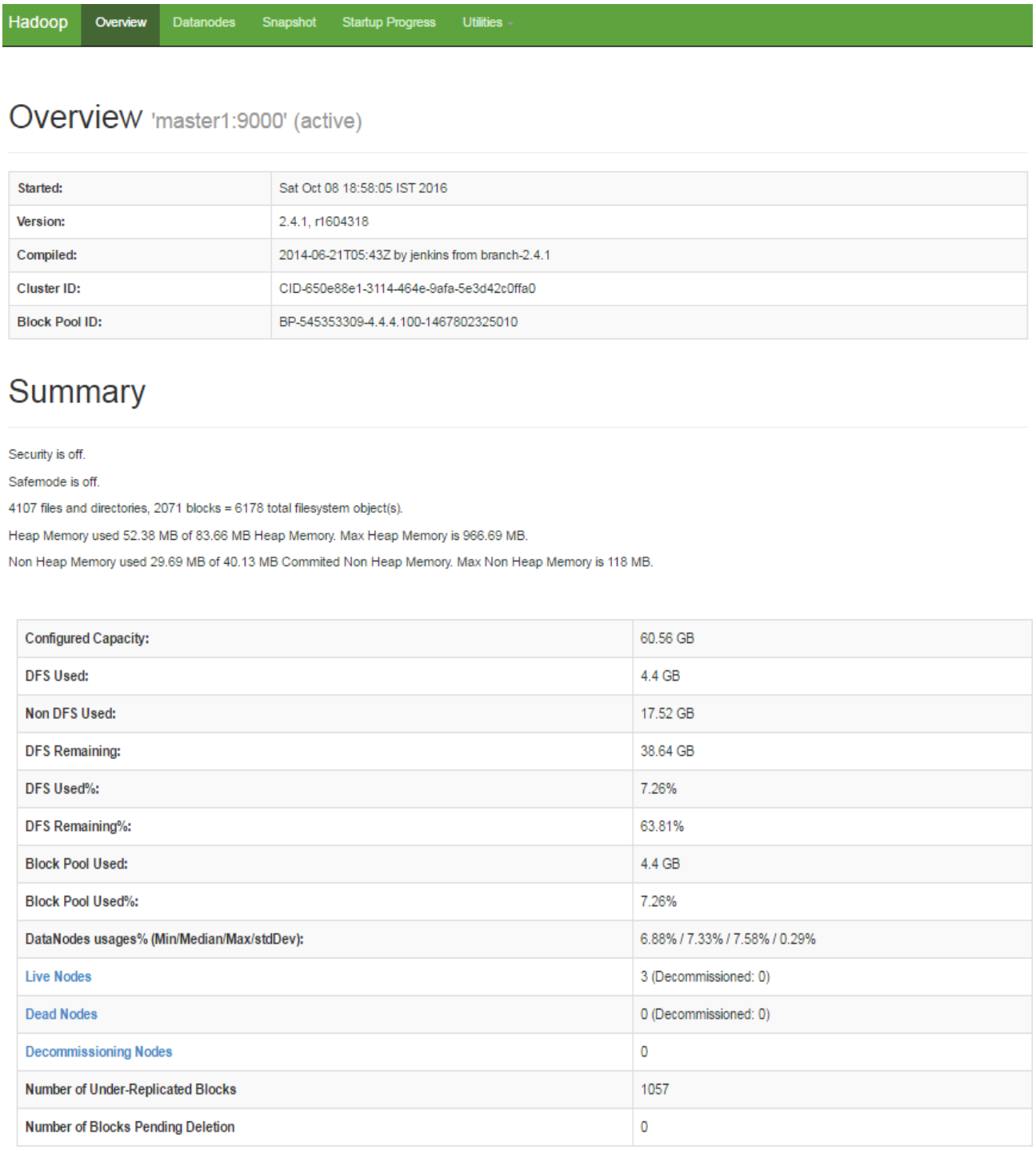


Fig.9.3 Web UI of Hadoop

PROJECT DETAIL

Our project is based on the analysis of data using Hadoop file system and its various ecosystems. Here we have used only two ecosystems of Hadoop i.e. Hive and Pig. We've collected the data from http://www.the-numbers.com and using these ecosystems we have analyzed them.

Following data files that are collected:-

1. **movietotal.dat**

   This file contains name of movies, type and gross collection. On the basis of this data we have analysed higher and lower return of the movies released since 2000.

2. **moviedaily.dat**

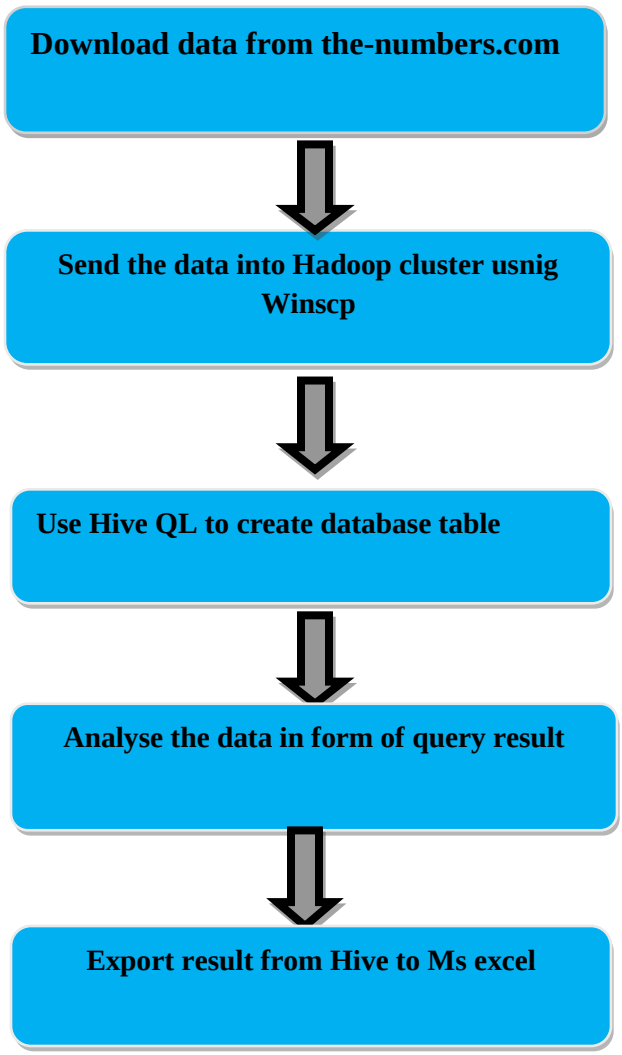
   This file contains name of movies, day number, daily collection and date.These results are namely average per day collection, which movies have maximum and minimum collection, top ten movies, and Sunday collection. Finally a chart is prepared in excel on these basis.

3. **movieweekend.dat**

   This file contains name of movies, number of weeks, collection per weekend and weekend date. They are further used to find out for how many days the movies were running on weekend and what the total collection on weekend is. Using the data file movietotal.dat, various descriptive statistics, including the mean, median, variance, standard deviation, minimum and maximum, for the movies in each category are calculated.

10.6 Flow Chart of Project

First of all the data are downloaded from the-numbers.com. These data are sent into the Hadoop system using WINSCP. Then we used Hive QL to create external database table. From there the data were analyzed using various query results. These results shows us the various trends which were not clear earlier. Finally all these results were exported from Hive to MS Excel using Hive ODBC connector. In Excel, again various charts have been prepared to show different results.



10.7 Data Dictionary

TABLE 10.1

Detail of movietotal.dat

Variable	Data Type	Description
Movie	String	It contain movie name
Type	String	It contain category of movie
Total	Int	It contain gross collection of each movie

TABLE 10.2
Detail of moviedaily.dat

VARIABLE	DATA TYPE	DESCRIPTION
MOVIE	String	It contain movie name
DAY_NUM	Int	It contain no. of day movie running
DAILY_PER_THEATER	Int	It contain daily collection
DATE	Date	It contain the date

TABLE 10.3
Detail of movieweekend.dat

VARIABLE	DATA TYPE	DESCRIPTION
MOVIE	String	It contain movie name
WEEK_NUM	Int	It contains no of weeks movies running
WEEKEND_PER_THEATER	Int	It contains weekend collection
WEEKEND_DATE	Date	It contains weekend date of movie running

10.3 Analysis of data movietotal.dat

The chart here compares the performance of all the movies released since 2000 that have earned less than US $100 million at the domestic box office. The data (movietotal.dat) is collected from the website (www.the-numbers.com).

A table movietotal is created and a chart is prepared in excel that tells which movie has higher return and lower return.

10.4 Procedure

1. Copy the movietotal.dat file from window to linux using winscp.
2. Put movietotal.dat file from linux to Hadoop distributed file system.
3. Using Pig, the data is filtered out which has earning less than US $100 million

4. Then filtered data is sent to Hive and ordered as per the earning.
5. After connecting Excel to Hadoop with the Hive ODBC driver, the data is transferred to Excel.
6. A Bar Graph is created on the basis of data and thereby movies with higher and lower return are found.

10.5 Source Code

1. LOAD movietotal.dat.txt' INTO PIG

```
n = LOAD '/pig/movietotal.dat.txt' as (number,movie,type,total);
```

2. FILTER the data by TOTAL>100.

```
f = filter n by total>100;
```

3. STORE the FILTERED data into /pig/gt100.

```
store f into '/pig/gt100';
```

4. To exit from pig

```
quit;
```

10.6 Source Code for HIVE

1. **Create a Table gt100**

```
create table gt100(serial int,movie string,type string,total int) row format delimited fields terminated by '\t' ;
```

2. **LOAD data in Hive**

```
LOAD DATA local INPATH '/home/hadoop/gt100' OVERWRITE INTO TABLE gt100;
```

3. **Create a Table movie_order to find the movie which return max or min.**

```
create table movie_order as select movie,type,total from gt100 order by total desc;
```

4. **To exit from hive**

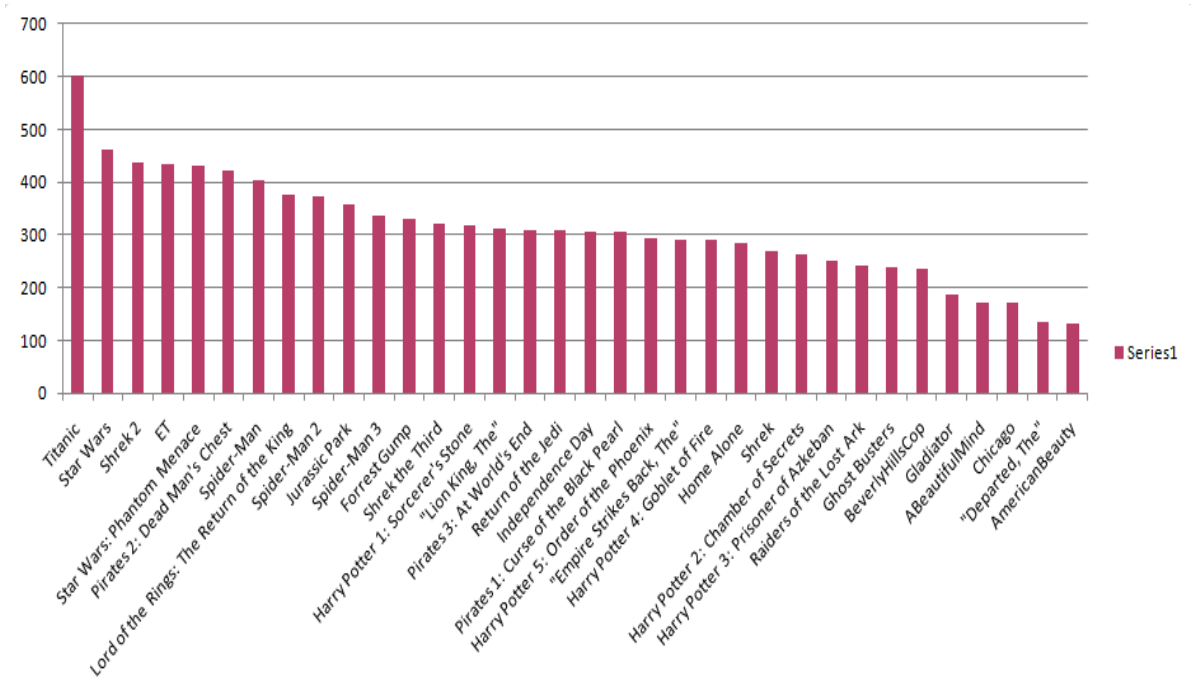
```
exit;
```

10.7 Movietotal chart:

TABLE 10.4
Showing Lowest and Highest Return

Movie	Type	Total
Titanic	Best Picture	600.79
Star Wars	Biggest Gross	461
Shrek 2	Series	436.72
ET	Biggest Gross	435.11
Star Wars: Phantom Menace	Biggest Gross	431.09
Pirates 2: Dead Man's Chest	Series	423.32
Spider-Man	Series	403.71
Lord of the Rings: The Return of the King	Best Picture	377.03
Spider-Man 2	Series	373.52
Jurassic Park	Biggest Gross	357.07
Spider-Man 3	Series	336.53
Forrest Gump	Biggest Gross	329.69
Shrek the Third	Series	321.01
Harry Potter 1: Sorcerer's Stone	Series	317.56
"Lion King, The"	Biggest Gross	312.86
Pirates 3: At World's End	Series	309.4
Return of the Jedi	Biggest Gross	309.21
Independence Day	Biggest Gross	306.17
Pirates 1: Curse of the Black Pearl	Series	305.41
Harry Potter 5: Order of the Phoenix	Series	292
"Empire Strikes Back, The"	Biggest Gross	290.27
Harry Potter 4: Goblet of Fire	Series	290.01
Home Alone	Biggest Gross	285.76
Shrek	Series	267.65
Harry Potter 2: Chamber of Secrets	Series	261.99
Harry Potter 3: Prisoner of Azkeban	Series	249.54
Raiders of the Lost Ark	Biggest Gross	242.37
Ghost Busters	Biggest Gross	238.63
BeverlyHillsCop	BiggestGross	234.76
Gladiator	Best Picture	187.68
ABeautifulMind	BestPicture	170.71
Chicago	Best Picture	170.69
"Departed, The"	Best Picture	133.31
AmericanBeauty	BestPicture	130.06

10.8 Garph:



10.9 Analysis of moviedaily.dat

The "moviedaily.dat" which contains data of various movies containing their daily performance is further analyzed using various tools like Pig and Hive.

1. Procedure

1. First data(moviedaily.dat) is collected and transferred to linux using Winscp.
2. The data is transferred to Hadoop file system from machine file system.
3. Using pig, data is filtered out which are having 'NA' in data.
4. The data is then sent to hive to analyze it.
5. In Hive, different results are found out
    a. For how many days movies were in the theatre individually.
    b. Average per day collection.
    c. Which movie has max collection, min collection, and top ten movies.
    d. Sunday collection
6. All results are sent to excel using Hive ODBC driver and Hive server and corresponding charts are prepared.

10.10 Source Code of Pig

1. LOAD movietotal.dat.txt' INTO PIG.

n=LOAD'/pig/moviedaily.dat.txt'as(number,movie,day_num,daily_per_theater,date);

2. **FILTER out the data by date==NA.**

   filter_moviedaily = FILTER n by date != 'NA'.

3. **STORE the FILTERED data into /pig/filter_m_daily**

   store f into '/pig/filter_m_daily';.

4. **To exit from pig**

   quit;

10.11 Source Code of Hive

1. **Create a Table movie_daily**

   create table movie_daily(serial int, movie string, day_num int, daily_collection int,m_date string) row format delimited fields terminated by '\t';

2. **LOAD data in Hive**

   LOAD DATA local INPATH '/home/hadoop/filter_m_daily' OVERWRITE INTO TABLE movie_daily;

3. **Create a Table avg_collection to calculate the average collection of movie.**

   create table avg_movie as select movie,count(day_num),avg(daily_per_theater) as avg_collection from movie_daily group by movie ;

4. **Create a Table Top_ten_movie to find top ten movies.**

   create table top_ten_movie as select movie,SUM(daily_per_theater) as collection from moviedaily group by movi2 order by top_collection desc limit 10;

10.12 Average Collection of Movie

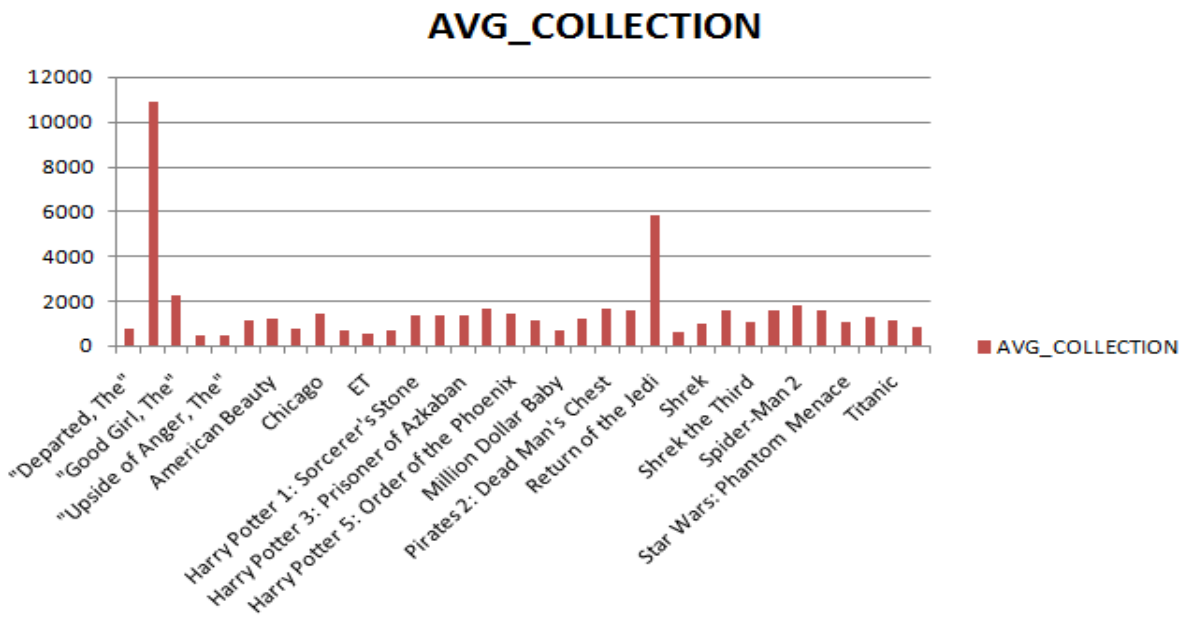
TABLE 10.5

Average Collection of Movies

Movie	Days	Avg_Collection
"Departed, The"	**62**	**729.2419355**
"Empire Strikes Back, The"	**7**	**10876**
"Good Girl, The"	**32**	**2210.1875**
"Last Mimzy, The"	**14**	**411.8571429**
"Upside of Anger, The"	**21**	**472.9047619**
A Beautiful Mind	**109**	**1105.761468**
American Beauty	**184**	**1213.431694**
Batman	**175**	**737.6114286**

Chicago	**106**	**1392.424528**
Crash	**38**	**685.4473684**
ET	**21**	**507.6666667**
Gladiator	**101**	**691.5940594**
Harry Potter 1: Sorcerer's Stone	**66**	**1327.590909**
Harry Potter 2: Chamber of Secrets	**55**	**1350.581818**
Harry Potter 3: Prisoner of Azkaban	**50**	**1370.04**
Harry Potter 4: Goblet of Fire	**47**	**1613.93617**
Harry Potter 5: Order of the Phoenix	**50**	**1448.12**
Lord of the Rings: Return	**100**	**1153.44**
Million Dollar Baby	**63**	**647.8730159**
Pirates 1: Curse of the Black Pearl	**79**	**1219.025316**
Pirates 2: Dead Man's Chest	**66**	**1654.030303**
Pirates 3: At World's End	**48**	**1581.083333**
Return of the Jedi	**12**	**5825**
Shakespeare in Love	**148**	**576.8378378**
Shrek	**78**	**1004.179487**
Shrek 2	**72**	**1589.513889**
Shrek the Third	**85**	**1046.188235**
Spider-Man	**76**	**1549.736842**
Spider-Man 2	**51**	**1803.490196**
Spider-Man 3	**53**	**1543.528302**
Star Wars: Phantom Menace	**157**	**1047.955414**
Super Size Me	**23**	**1239.478261**
Titanic	**186**	**1121.419355**
You Can Count on Me	**18**	**832.8333333**

10.13 Graph of Average Collection:



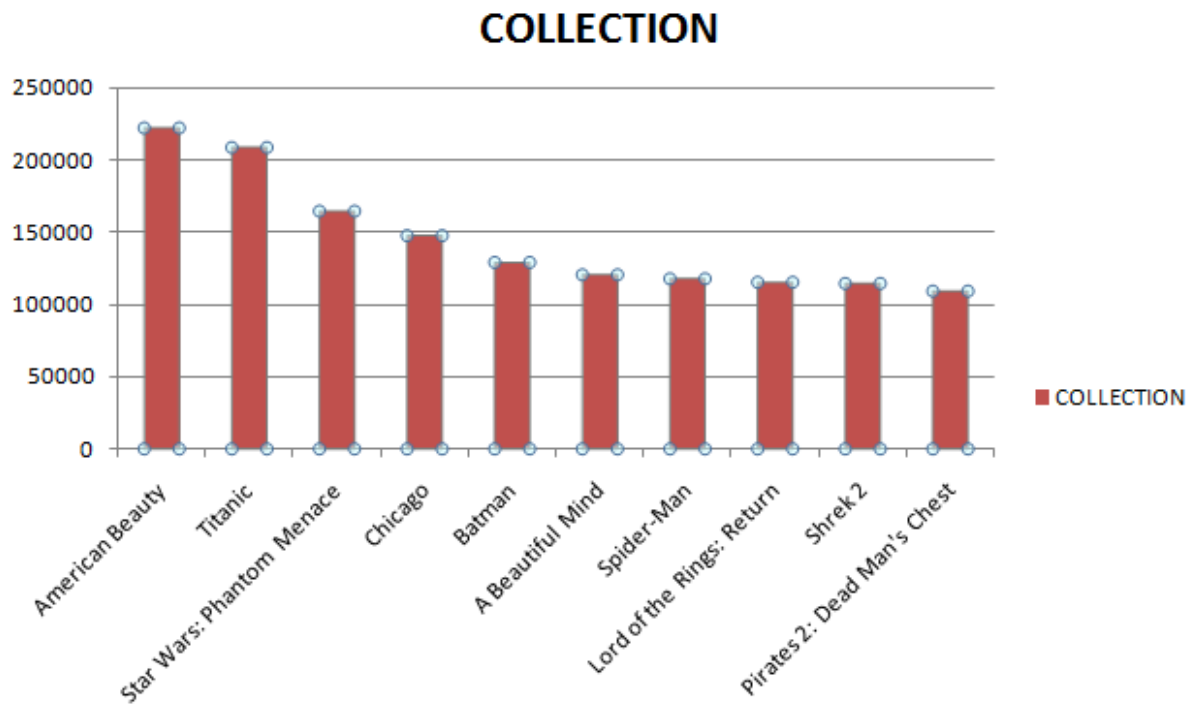
10.14 Top Ten Movie Chart:

TABLE 10.6

Top Ten Movie

MOVIE	COLLECTION
American Beauty	**222058**
Titanic	**208584**
Star Wars: Phantom Menace	**164529**
Chicago	**147597**
Batman	**129082**
A Beautiful Mind	**120528**
Spider-Man	**117780**
Lord of the Rings: Return	**115344**
Shrek 2	**114445**
Pirates 2: Dead Man's Chest	**109166**

10.15 Graph of Top Ten Collection



10.16 Analysis of Data movieweekend.dat

The moviweekend.dat compares the total collection of movies in weekends(Friday, Saturday, Sunday) and different movies are analyzed based on the number of weeks movies were in the theater and their corresponding weekend collection.

Here we use Hive and Pig to extract the data and results are sent to excel using Hive server.

10.17 Procedure

1. First data(movieweekend.dat) is collected and transferred to a Hadoop machine using WINSCP.
2. Data is then transferred to Hadoop file system using Hadoop fs commands.
3. we will load movieweekend.dat into hive and we will find and create table
    a) for how many weeks the movie is runnig on weekend.
    b) what is the total collection on weekend.

10.18 Source Code for Hive

1. **Create Table movie_weekend in Hive**

   create table movieweek(number int,movie string,week_num string,week_per_theater double,week_date string) row format delimited fields terminated by '\t' ;

2. **Load the data movieweekend.dat into Hive.**

   LOAD DATA local INPATH '/home/hadoop/movieweekend.dat' OVERWRITE INTO TABLE movie_weekend;

3. **Create a Table week_running to find the no. of weeks movie is running and collection of week.**

   create table week_running as select movie ,count(week_num) as num_of_weeks, sum(week_per_theater) as total_weekend from movie_weekend group by movie;

4. **To exit from Hive.**

   exit;

10.19 Week Collection Chart:

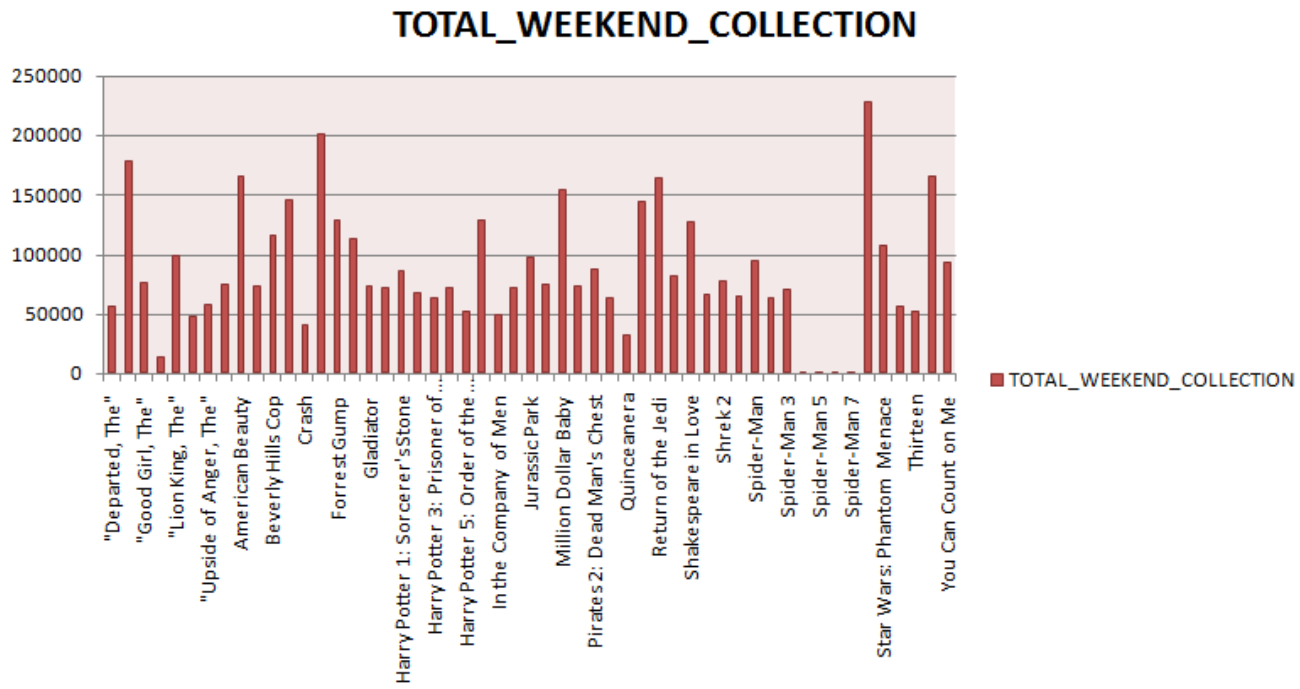
TABLE 10.7

Weekend Collection

Movie	Num_of_weeks	Total_weekend_Collection
"Departed, The"	24	55394
"Empire Strikes Back, The"	15	178013
"Good Girl, The"	18	75290
"Last Mimzy, The"	12	13853
"Lion King, The"	32	97983
"Road Home, The"	23	47026
"Upside of Anger, The"	16	57568
A Beautiful Mind	23	74851
American Beauty	38	165891
Batman	13	72861
Beverly Hills Cop	33	115915
Chicago	36	146062
Crash	23	41036
ET	52	201257
Forrest Gump	42	128534
Ghost Busters	30	113540
Gladiator	33	73643
Gods and Monsters	29	72218

Harry Potter 1: Sorcerer's Stone	**27**	**86287**
Harry Potter 2: Chamber of Secrets	**22**	**66978**
Harry Potter 3: Prisoner of Azkaban	**22**	**62522**
Harry Potter 4: Goblet of Fire	**20**	**71430**
Harry Potter 5: Order of the Phoenix	**22**	**51965**
Home Alone	**30**	**128472**
In the Company of Men	**16**	**49422**
Independence Day	**24**	**71861**
Jurassic Park	**21**	**97520**
Lord of the Rings: Return	**24**	**73870**
Million Dollar Baby	**25**	**154115**
Pirates 1: Curse of the Black Pearl	**26**	**72740**
Pirates 2: Dead Man's Chest	**22**	**87171**
Pirates 3: At World's End	**19**	**63285**
Quinceanera	**14**	**32472**
Raiders of the Lost Ark	**43**	**144778**
Return of the Jedi	**42**	**163572**
Run Lola Run	**31**	**82019**
Shakespeare in Love	**33**	**126779**
Shrek	**29**	**65716**
Shrek 2	**14**	**77896**
Shrek the Third	**12**	**64396**
Spider-Man	**16**	**93854**
Spider-Man 2	**20**	**63638**
Spider-Man 3	**12**	**70368**
Spider-Man 4	**1**	**774**
Spider-Man 5	**1**	**604**
Spider-Man 6	**1**	**514**
Spider-Man 7	**1**	**474**
Star Wars	**31**	**228181**
Star Wars: Phantom Menace	**31**	**107684**
Super Size Me	**21**	**55341**
Thirteen	**17**	**51399**
Titanic	**41**	**165701**
You Can Count on Me	**31**	**93166**

10.20 Graph of Total Weekend Collection:



10.21 Calculation of movietotal.dat

Mean, Median, Mode, Standard Deviation and Variance are calculated from data movietotal.dat of each category from Excel. Maximum and Minimum of each category is also found from the table.

10.22 Procedure

1. First data(movietotal.dat) is collected and transferred to a Hadoop machine where master node is assigned using WINSCP.
2. Data is then transferred to Hadoop file system using Hadoop fs commands.
3. Using Pig, data of each type is separated and sent to Hive.
4. Using Hive, Data is filtered so that we can do different types of calculation on each type of data and further saved as table movietotal_calculation.
5. The results are then sent to excel using Hive server and Hive ODBC Driver.
6. Using Excel Functions, mean, median, mode, standard deviation, variance, maximum and minimum of each category is calculated.

10.23 Chart Show The Calculation

TABLE 10.8

Calculation of movietotal.dat

Biggest	Best	Sundance	Series	Calculation
290.27	133.31	14.02	317.56	
312.86	170.69	21.47	261.99	
435.11	55.33	1.28	249.54	
329.69	187.68	18.76	290.01	
238.63	377.03	6.45	292	
285.76	100.42	2.88	305.41	
306.17	100.32	1.69	423.32	
357.07	600.79	7.27	309.4	
242.37		11.53	267.65	
309.21		4.6	436.72	
461		9.18	321.01	
431.09			403.71	
			373.52	
			336.53	
333.269167	215.69625	9.011818182	327.7407143	**Mean**
311.035	152	7.27	313.48	**Median**
5444.54079	33748.767	46.17349636	3591.585469	**Variance**
73.7871316	183.708375	6.795108267	59.92983788	**Deviation**
238.63	55.33	1.28	249.54	**Minimum**
461	600.79	35.49	436.72	**Maximum**