

Spring 2015


Big Data Analytics by Using Hadoop

Chaitanya Arava
Governors State University

Sudharshan Bandaru
Governors State University

Saradhi Bhargava Reddy Tiyyagura
Governors State University

Follow this and additional works at: <http://opus.govst.edu/capstones>

 Part of the [Databases and Information Systems Commons](#), and the [Systems Architecture Commons](#)

Recommended Citation

Arava, Chaitanya; Bandaru, Sudharshan; and Tiyyagura, Saradhi Bhargava Reddy, "Big Data Analytics by Using Hadoop" (2015). *All Capstone Projects*. 97.
<http://opus.govst.edu/capstones/97>

For more information about the academic degree, extended learning, and certificate programs of Governors State University, go to http://www.govst.edu/Academics/Degree_Programs_and_Certifications/

Visit the [Governors State Computer Science Department](#)

This Project Summary is brought to you for free and open access by the Student Capstone Projects at OPUS Open Portal to University Scholarship. It has been accepted for inclusion in All Capstone Projects by an authorized administrator of OPUS Open Portal to University Scholarship. For more information, please contact opus@govst.edu.

Abstract

Data is large and vast, with more data coming into the system every day. Summarization analytics are all about grouping similar data together and then performing an operation such as calculating a statistic, building an index, or just simply counting.

Filtering is more about understanding a smaller piece of your data, such as all records generated from a particular user, or the top ten most used verbs in a corpus of text. In short, filtering allows you to apply a microscope to your data. It can also be considered a form of search.

Hadoop allows us to modify the way data is loaded on disk in two major ways: configuring how contiguous chunks of input are generated from blocks in, and configuring how records appear in the map phase. The two classes you'll be playing with to do this are Record Reader and Input Format. These work with the Hadoop MapReduce framework in a very similar way to how mappers and Reducers are plugged in.

This is about the analytics side of Hadoop or MapReduce. Computation in Hadoop MapReduce is performed in parallel, automatically, with a simple abstraction for developers that obviate complex synchronization and network programming. Unlike many other distributed data processing systems, Hadoop runs the user-provided processing logic on the machine where the data lives rather than dragging the data across the network; a huge win for performance.

At Q&A sites such as Experts exchange service developed and the number of users grew from thousands to millions, storing, processing, and managing all the incoming data became increasingly challenging

There were several reasons for adopting Hadoop:

- The distributed file system provided redundant backups for the data stored on it at no extra cost
- Scalability was simplified through the ability to add cheap, commodity hardware when required.
- Hadoop provided a flexible framework for running distributed computing algorithms with a relatively easy learning curve

Hadoop can be used to form core backend batch and near real-time computing infrastructures. It can also be used to store and archive massive datasets.

Table of Contents

<i>1.What is Hadoop.....</i>	<i>1</i>
<i>2.Hadoop Distributed File Systems.....</i>	<i>3</i>
<i>3.Map Reduce.....</i>	<i>3</i>
<i>4.How to Install Hadoop.....</i>	<i>4</i>
<i>5.How to Run Project.....</i>	<i>6</i>
<i>6.System Requirements.....</i>	<i>7</i>
<i>7.Results.....</i>	<i>7</i>
<i>8.References.....</i>	<i>11</i>

1.What is Hadoop:

Hadoop is an open-source software framework for storing and processing big data in a distributed fashion on large clusters of commodity hardware. Essentially, it accomplishes two tasks: massive data storage and faster processing.

Hadoop has become one of the most important technologies, the key factors of hadoop are as follows:

Low Cost :

Hadoop is an free open Source frame work, and uses commodity hardware to store substantial amount of data. Hadoop additionally offers a practical stockpiling answer for organizations' blasting information sets. The issue with customary social database administration frameworks is that it is amazingly taken a toll restrictive to scale to such an extent so as to process such huge volumes of information. With an end goal to lessen costs, numerous organizations in the past would have needed to down-example information and characterize it in view of specific presumptions as to which information was the most significant. The crude information would be erased, as it would be excessively taken a toll restrictive, making it impossible to keep. While this methodology may have worked in the short term, this implied that when business needs changed, the complete crude information set was not accessible, as it was so costly it was not possible store. Hadoop, then again, is outlined as a scale-out structural planning that can reasonably store the majority of an organization's information for later utilize. The expense reserve funds are stunning: as opposed to costing thousands to a huge number of pounds every terabyte, Hadoop offers figuring and stockpiling capacities for many pounds every terabyte.

Flexible:

Hadoop empowers organizations to effectively get to new information sources and tap into distinctive sorts of information (both organized and unstructured) to create esteem from that information. This implies organizations can utilize Hadoop to get profitable business bits of knowledge from information sources, for example, social networking, email discussions information. Moreover, Hadoop can be utilized for a wide mixed bag of purposes, for example, log handling, suggestion frameworks, information warehousing, business sector battle investigation and misrepresentation discovery.

Computing power:

Hadoop is a distributed processing model, so it can prepare vast measure of information. The additionally registering hubs you utilize .

Fast:

Hadoop's unique storage method is based on a distributed file system that basically 'maps' data wherever it is located on a cluster. The tools for data processing are often on the same servers where the data is located, resulting in much faster data processing. If you're dealing with large volumes of unstructured data, Hadoop is able to efficiently process terabytes of data in just minutes, and petabytes in hours.

Storage Flexibility:

Unlike traditional relational databases, you don't have to preprocess data before storing it. And that includes unstructured data like text, images and videos. You can store as much data as you want and decide how to use it later.

Inherent data protection and self-healing capabilities:

Data and application processing are protected against hardware failure. If a node goes down, jobs are automatically redirected to other nodes to make sure the distributed computing does not fail. And it automatically stores multiple copies of all data.

Advantages of hadoop:

The Advantages of hadoop are as follows:

1. Distribute data and computation. The computation local to data prevents the network overload.
2. Tasks are independent The task are independent so,
 - We can easy to handle partial failure. Here the entire nodes can fail and restart.
 - it avoids crawling horrors of failure and tolerant synchronous distributed systems.
 - Speculative execution to work around stragglers.
3. Linear scaling in the ideal case. It used to design for cheap, commodity hardware.
4. Simple programming model. The end-user programmer only writes map-reduce tasks.
5. HDFS store large amount of information.
6. HDFS is simple and robust coherency model.
7. That is it should store data reliably.
8. HDFS is scalable and fast access to this information and it also possible to serve s large number of clients by simply adding more machines to the cluster.
9. Data will be written to the HDFS once and then read several times.
10. HDFS is a block structured file system: – Each file is broken into blocks of a fixed size and these blocks are stored across a cluster of one or more machines with data storage capacity.

Disadvantages Of Hadoop:

The following were the disadvantages of hadoop:

1. As big data is not suitable for small business
2. As Java is used and without an expert this is prone for hacking
3. There is a missing encryption methodology for storage and network levels.
4. There are lot of stability issues in Hadoop.

2.Hadoop Distributed File Systems:

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. Developed specifically for large-scale data processing workloads where scalability, flexibility and throughput are critical, HDFS accepts data in any format regardless of schema, optimizes for high bandwidth streaming, and scales to proven deployments of 100PB and beyond.

Key HDFS Features:

Scale-Out Architecture: We can add Servers to increase capacity.

High Availability : Serve mission critical workflows and applications.

Fault Tolerance : Automatically recover from failures.

Flexible Access : Multiple and open frameworks for serialization and file system mounts.

Load Balancing : place data intelligently for maximum efficiency and utilization.

Tunable Replication : It provides multiple copies of each file provide data protection and computational performance.

Security : POSIX-based file permissions for users and groups with optional LDAP integration.

3.Map Reduce:

Map Reduce is a programming model and an associated implementation for processing and generating large data sets with a parallel, distributed algorithm on a cluster. Conceptually similar approaches have been very well known since 1995 with the Message Passing Interface standard having reduce and scatter operations.

The term Map Reduce actually refers to two separate and distinct tasks that Hadoop programs perform. The first is the map job, which takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). The reduce job takes the output from a map as input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce job is always performed after the map job.

Simplicity:

Developers can write applications in their language of choice, such as Java, C++ or python, and map reduce jobs are easy to run.

Scalability:

Map reduce can process peta bytes of data, stored in HDFS on the cluster.

Speed:

Parallel processing means that map reduce can take problems that used to take these days to solve and solve them in hours or minutes.

Recovery:

Map reduce takes care of failures. If a machine with one copy of the data is unavailable, another machine has a copy of the same key/value pair, which can be used to solve the same sub-task. The job tracker keeps track of it all.

Apache Hive:

Apache Hive is data warehouse infrastructure built on top of hadoop for providing data summarization, query, and analysis. While initially developed by facebook, Apache Hive is now used and developed by other companies such as Netflix. Amazon maintains a software fork of Apache Hive that is included in Amazon Elastic Map Reduce on Amazon Web Services.

4.How to Install Hadoop:

Go to Ubuntu Virtual Machine:

1. Open terminal using `ctl+alt+t` and Create a folder called work using the command
`carava@Ubuntu$ $ mkdir project`

2. Copy hadoop.1.2.1 and jdk 1.6 to the project folder created above.

3. Use the command to install JDK if you get connected to internet

```
carava@Ubuntu$ sudo apt-get install openjdk-6-jdk
```

4. The next step is to install ssh server and client using the below command

```
carava@Ubuntu$ sudo apt-get install openssh-server openssh-client
```

5. Next extract hadoop.1.2.1.tar file under /work folder using the below command.

```
carava@Ubuntu$ tar -xvf hadoop-1.2.1.tar
```

6. Go to the location /project/hadoop.1.2.1/conf/hadoop-env.sh, add the below entries based on the mode of JDK installation.

```
export JAVA_HOME=/usr/lib/jvm/java-6-openjdk-i386/ -- incase if online mode.
```

```
export JAVA_HOME=/home/hduser/jdk1.7.0_67/ -- incase of offline mode.
```

7. Open core-site.xml located under /hadoop/conf folder and then add below entries

```
<property>
<name>fs.default.name</name>
<value>hdfs://localhost:54433</value>
</property>
```

8. Open hdfs-site.xml located under /hadoop/conf folder and then add below entries

```
<property>
<name>dfs.replication</name>
```

```
<value>1</value>
</property>
<property>
<name>dfs.name.dir</name>
<value>/home/hadoop/nn/dfs/name</value>
</property>
```

```
<property>
<name>dfs.data.dir</name>
<value>/home/hadoop/nn/dfs/data</value>
</property>
```

9. Open `mapred-site.xml` located under `/hadoop/conf` folder and then add below entries

```
<property>
<name>mapred.job.tracker</name>
<value>localhost:9001</value>
</property>
```

10. Create a folder under home directory and provide permissions for others complete read and write permissions.

11. Format name node using the below command.

```
carava@Ubuntu$ hadoop namenode -format
```

12. Go to `/project/hadoop/bin/$./start-dfs.sh` which will bring up below core demons.

```
Name node
Secondary Name node
Data node
```

13. Next steps are to run `start-mapred.sh`, which will bring below couple of core demons. By now altogether

we see 3 demons started in the earlier setup plus below couple of other demons.

```
Job tracker
Task tracker
```

14. Use the below URL to access web UI for the Namenode

```
http://localhost:50070
```

15. Default ports for the other Hadoop demons.

```
Namenode 50070
Datanodes 50075
Job tracker 50030
Task tracker 50060
Secondary Name Node 50090
```


5.How to Run Project:

1. Format the nameNode by using the following command:

a. **carava@ubuntu:~/project/hadoop\$** hadoop namenode -format

2. Start Hadoop by using following command:

a. **carava@ubuntu:~/project/hadoop/bin\$** ./sh start-all.sh

3. Check the weather the name started or not by using the following command:

a. **carava@ubuntu:~/project/hadoop/bin \$** jps

4. Create a new file System using the following command:

i. **carava@ubuntu:~/project/hadoop/bin\$** hadoop fs -mkdir /user/carava/datadump/posts

ii. **carava@ubuntu:~/project/hadoop/bin\$** hadoop fs -mkdir /user/carava/datadump/comments

iii. **carava@ubuntu:~/project/hadoop/bin\$** hadoop fs -mkdir /user/carava/datadump/user

5. load the xml files from ubuntu file system to hadoop file system.

I. **carava@ubuntu:~/project/hadoop/bin\$** hadoop dfs -put Posts.xml /user/carava/datadump/posts

II. **carava@ubuntu:~/project/hadoop/bin\$** hadoop dfs -put Comments.xml /user/carava/datadump/comments

III. **carava@ubuntu:~/project/hadoop/bin\$** hadoop dfs -put user.xml /user/carava/datadump/user

6. Connect to HIVE database by using following command:

carava@ubuntu:~/Downloads\$ hive

7. create tables for the Hive database.

8. Run hive by using the following command:

carava@ubuntu:~/Downloads\$ hive --service hiveserver

9. Start Tomcat Server in eclipse and run programs that you want to execute.

10. Start the program using local host:

<http://localhost:8080/hadoopwebproj/login.jsf>

6.System Requirements

Software requirements:

Operating System	:	Ubuntu
Technology	:	Hadoop, Hive, Java and J2EE
Web Technologies	:	Html, JavaScript, CSS
IDE	:	Eclipse
Web Server	:	Tomcat
Java Version	:	JDK1.7, Tomcat 7

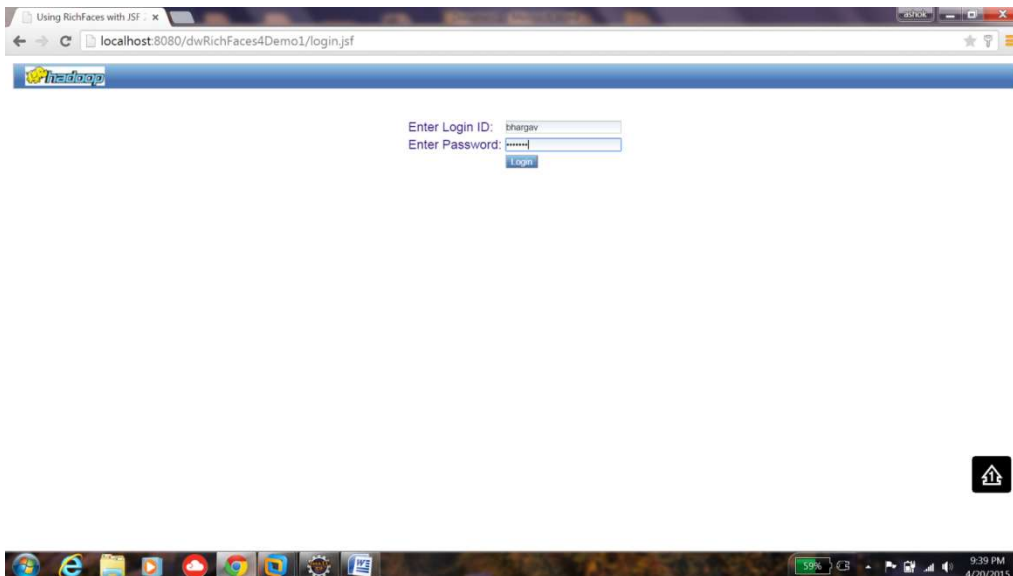
Hardware requirements:

Hardware	:	Commodity
RAM	:	4GB

7.Results:

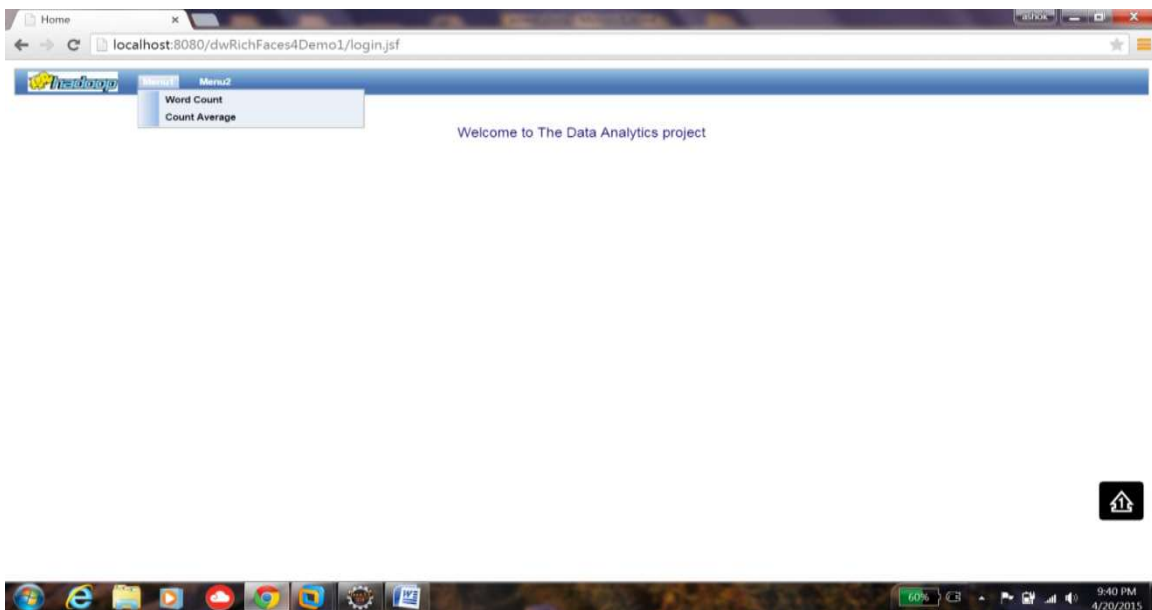
Login page :

This is the user login page to interact with Data Analytics using Hadoop .
User should login with login credentials .



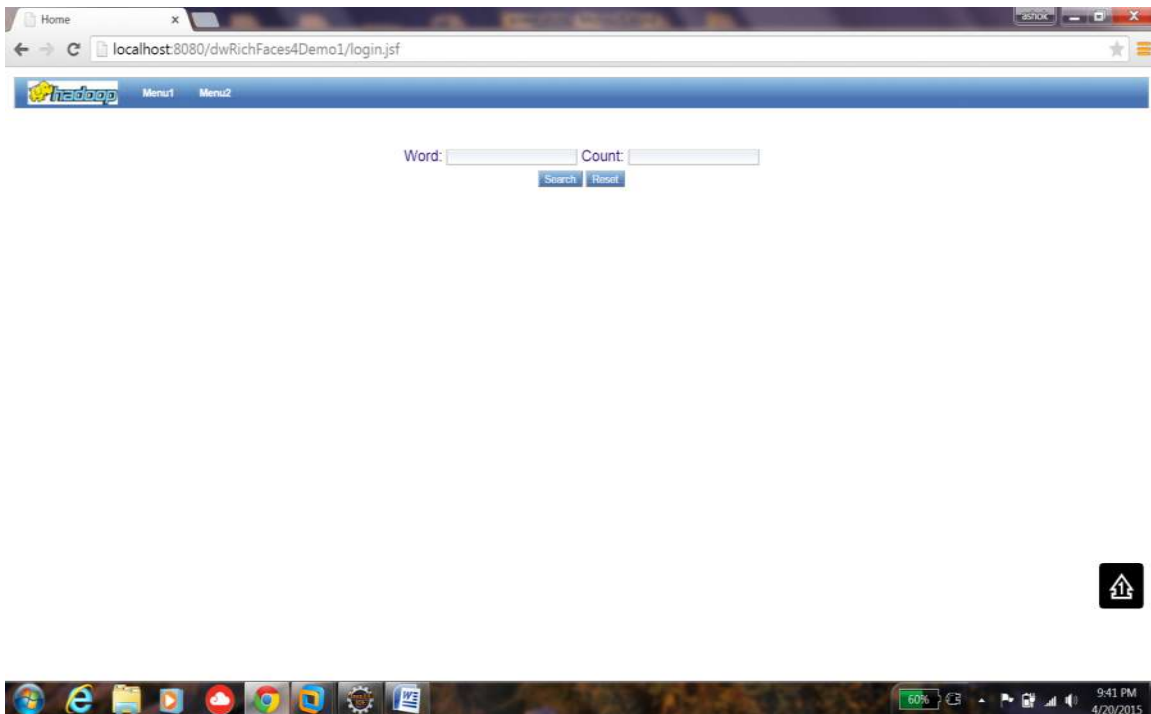
Selecting Functionality:

When the user login successfully , below is the home page in which user can select the module , based on the module functionalities like word count , count average and hourly count are appeared .



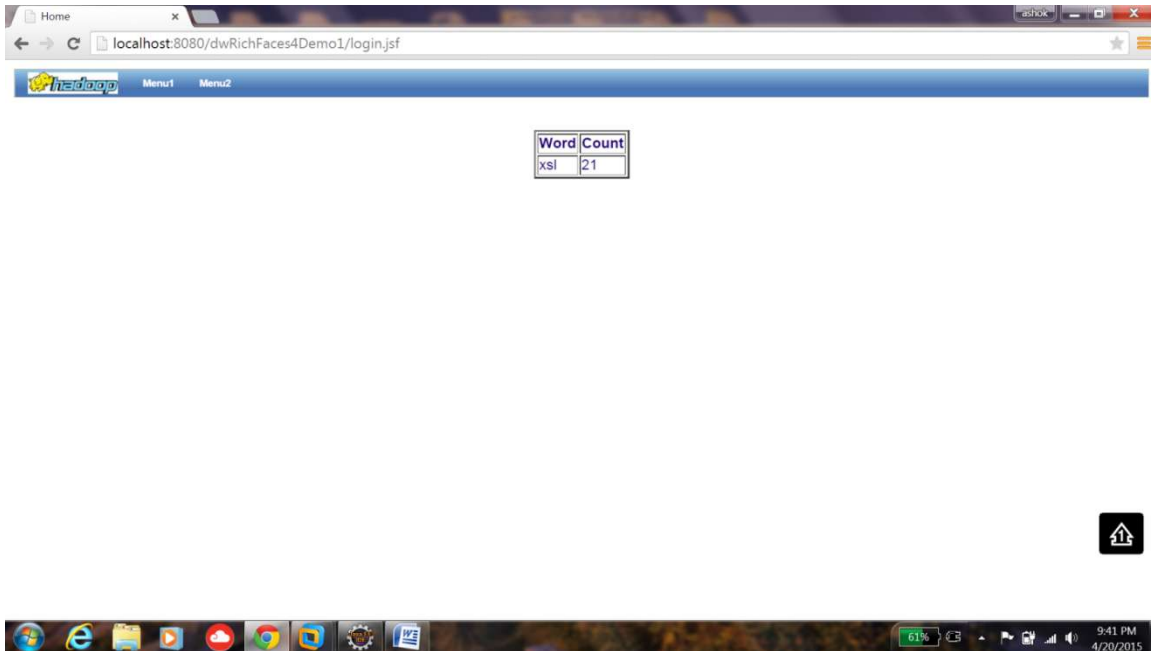
Searching word Count:

When the user want to search a word that repeated n number of times user can give the word and then press the search button .Similarly , if user wants to give the count and list of the words for the given count , user has to give the number in count and then press the search button to get the result.



Output After serching Would Count :

Here user entered word as xsl and the count get displayed as 21.



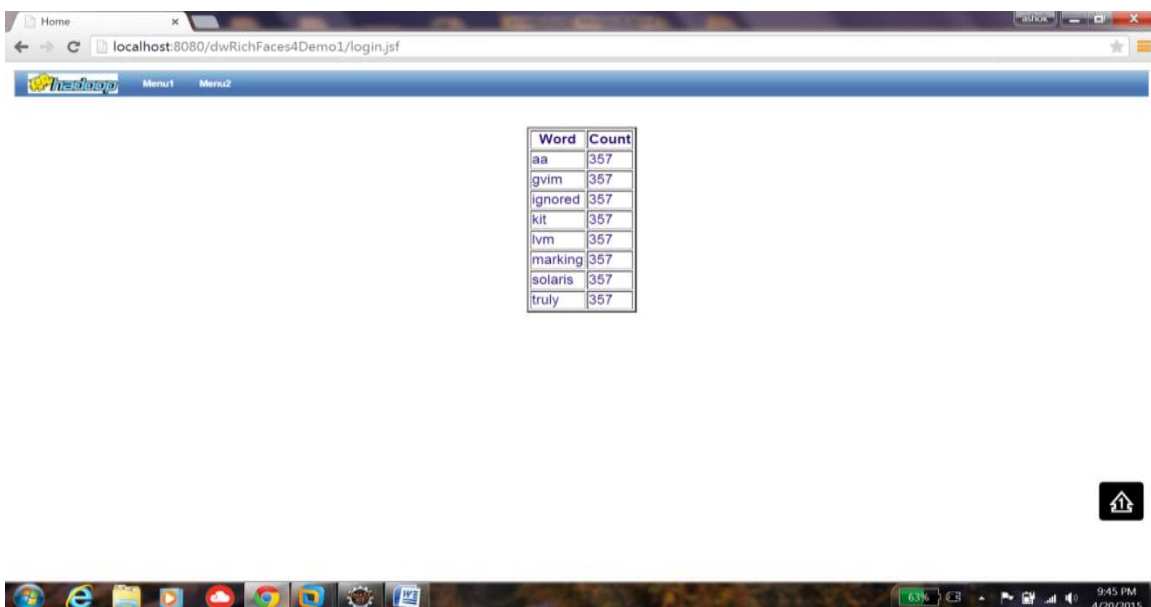
The screenshot shows a web browser window with the address bar displaying 'localhost:8080/dwRichFaces4Demo1/login.jsf'. The page content includes a table with the following data:

Word	Count
xsl	21

The browser's taskbar at the bottom shows various application icons and system tray information, including the time '9:41 PM' and date '4/20/2015'.

Multiple Would count :

Here is the result for the count equals to 357.



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/dwRichFaces4Demo1/login.jsf'. The page content includes a table with the following data:

Word	Count
aa	357
gvim	357
ignored	357
kit	357
lvm	357
marking	357
solaris	357
truly	357

The browser's taskbar at the bottom shows various application icons and system tray information, including the time '9:45 PM' and date '4/20/2015'.

Average Count:

Here is the result for Hourly Average count.

Hour	Count	Average
0	20084.0	163.15465
1	16225.0	161.04927
2	17440.0	160.31325
3	16794.0	159.82953
4	15991.0	157.18604
5	15489.0	155.47285
6	16058.0	153.61433
7	17682.0	155.21892
8	19307.0	155.35123
9	20515.0	154.07736
10	21011.0	151.32312
11	23498.0	154.56192
12	27273.0	156.88383
13	31006.0	155.40927
14	36085.0	156.74228
15	37903.0	157.6561
16	36139.0	159.64307
17	35679.0	159.44803
18	34984.0	159.60733
19	35255.0	158.9112
20	34745.0	159.59335
21	31498.0	160.2227

8.References:

- <http://hadoop.apache.org/>
- <http://www.cloudera.com/content/cloudera/en/about/hadoop-and-big-data.html>
- <http://hortonworks.com/hadoop>