testing nodes in a circuit, e.g. in-circuit testing (ICT). While white-box testing can be applied at the unit, integration and system levels of the software testing process, it is usually done at the unit level. It can test paths within a unit, paths between units during integration, and between subsystems during. Though this method of test design can uncover many errors or problems, it might not detect unimplemented parts of the specification or missing requirements.

Black-box testing treats the software as a "black box", examining functionality without any knowledge of internal implementation.

The testers are only aware of what the software is supposed to do, not how it does it. Black-box testing methods include equivalence partitioning, boundary value analysis, all-pairs testing, state transition tables, decision table testing, fuzz testing, model-based testing, use case testing, exploratory testing and specification-based testing.

4.2 Design of test cases

I have discussed the various test case designs that can be implemented in this project.

4.3 File Upload

The test case design of File Upload is given below.

Table 4.1: Test case of file upload

Description Of test case	This test case is to check whether file is uploading or not
Pre-condition	File selected or not
iDescription	File uploading ! OK
Actual Result	File uploading ! OK
Pass/Fail Criteria	Pass when lok` else fail

4.4 File Download

The test case design of File Download is given below.

Table 4.2: Test case of file download

Description of test case	This test case is to check whether file is downloading or not
Pre-condition	File selected or not
Expected Result	File downloading ! OK
Actual Result	File downloading ! OK
Pass/Fail Criteria	Pass when 'ok' else fail

4.5 File List

The test case design of File List Store in HDFS is given below

Table 4.3: Test case for File list

Description of test case	This test case is to check whether file is generated or not
Pre-condition	File generated or not
Expected Result	File generated ! OK
Actual Result	File generated ! OK
Pass/Fail Criteria	Pass when `ok` else fail

4.6 File Deletion

The test case design of File Deletion from HDFS is given below

Table 4.4: Test case for File Delete

Description of test case	This test case is to check whether tile is deleted or not
Pre-condition	File deleted or not
Expected Result	File deleted ! OK
Actual Result	File deleted ! OK
Pass/Fail Criteria	Pass when `ok` else fail

4.7 Directory Deletion

The test case design of Directory Deletion from HDFS is given below

Table 4.5: Test case for Directory Delete

Description of test case	This test case is to check whether directory is deleted or not
Pre-condition	Directory deleted or not
Expected Result	Directory deleted! OK
Actual Result	Directory deleted! OK
Pass/Fail Criteria	Pass when 'ole else fail

4.8 Validation

Validation is the process of checking that a software system Meets specifications and that it fulfills its intended purpose. It may also be referred to as software quality control. It is normally the responsibility of software testers as part of the software development Lifecycle.

Validation checks that the product design satisfies or fits the intended use (high-level checking), i.e., the software meets the user requirements. This is done through dynamic testing and other forms of review. Results of validation testing of test case scenarios are:-

Table 4.6: Validation

TestCase ID	Expected Result	Actual Result	Pass/Fail
6.2.1	File Uploading! Ok	OK	Pass
6.2.2	File Downloading! Ok	OK	Pass
6.2.3	File List Generated! OK	OK	Pass
6.2.4	File Deleted! OK	OK	Pass
6.2.5	Directory Deleted! OK	OK	Pass

4.9 Conclusion on Testing

In this chapter various test cases were designed and validated. The project passed the all five test cases designed. The file is being uploading, downloading, deletion, file listing, directory deletion. This project meets all the specification.

5. Conclusion and Future Work

5.1 Conclusion

Hadoop Distributed File System provides efficient storage of large dataset. Developed by Apache and co-developed by Yahoo it has come a long way from experimental implementation to real world scenario. Providing efficient management for data storage and retrieval it is now one of the most widely used distributed file system for unstructured data. It's read and write algorithm are open to changes.

5.2 Future Work

Hadoop is a fast growing open source technology which constantly needs refinements. One such field is its read-write algorithm. Its many other components such as Hive, map reduce needs development. It posses a big challenge to all BIG DATA managers and developers to define more efficient way to manage and store data.

Furthur extension of project, we can develop an application for novice users or business clients for storing, deleting, uploading, modify huge data of any organization in more ease way.

6. Installation prerequisites"

This project is a simulation of hadoop. Here I am giving the pre-requisites for installing hadoop.

1. ubuntu 12.04: This is the most stable linux package from ubuntua can be installed using wubi.exe software which installs ubuntu above windows. Ubuntu can also be installed using virtual box software.
2. linux-kernal 3.9.0.29:It automatically gets installed with ubuntu 12.04.
3. hadoop-1.0.4:This can be downloaded as a tar package from hadoop.apache.org This is the package with the help of which hadoop will be installed on system.
4. openjdk-6-jdk:Java version 6 or more is required to configure hadoop.
5. Now I am going to discuss how to install hadoop and all other softwares related to it.

6.1 Install openjdk-6-jdk

Open terminal and type the following commands
sudo add-apt-repository ppa:webupd8team/java
sudo apt-get update
sudoapt-get install openjdk-6-jdk
This will install Java-6 on ubuntu.

6.2 Create a dedicated hadoop user

It is recommended (not necessary) to add a dedicated user for hadoop with
sudo permissions.
sudo adduser hadoop
sudo adduser hadoop sudo
Now move to the folder where hadoop-1.0.4.tar file is kept and extract it
using following command
tar -xvf hadoop-1.0.4.tar
Now login as hadoop user.
su hadoop

6.3 Installing openssh and generating keys

sudo apt-get install openssh-server
sudo apt-get install openssh-client
ssh –keygen
cd /home/hadoop/.ssh
is (list the keys)
ssh-copy-id -i /home/hadoop/.ssh/id rsa.pub localhost

6.4 Create a folder named temp and change permissions

Before creating tmp folder permission for hadoop-1.0.4 folder needs to be changed.
sudo chown -R hadoop hadoop-1.0.4
sudo chmod 777 hadoop-1.0.4
sudo mkdir /home/hadoop tmp
sudo chown -R hadoop /home/hadoop/tmp
sudo chmod 777 /home/hadoop/tmp

6.5 Start configuring hadoop

sudo gedit /home/hadoop/.bashrc
In this file add the following two lines.
export JAVA HOME---/usr/lib/jvm/java-6-openjdk-amd64
export HADOOP HOME= path to hadoop-1.0.4 directory
Change to bin folder of hadoop-1.0.4/bin
sudo gedit hadoop-env.sh
Add the following line in it
export JAVA HOME=/usr/libijvm/java-6-openjdk-amd64
sudo gedit core-site.xml
Add the following lines in it

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xs1" href="configuration.xs1"?>

<!-- Put site-specific property overrides in this file. - - >

<configuration>
<property>
<name>hadoop.tmp.dir</name>
<value>/home/hadoop/tmp</value>
</property>
<property>
<name>fs.default.name</name>
<value>hdfs://localhost:54310</value>
</property>
</configuration>
```

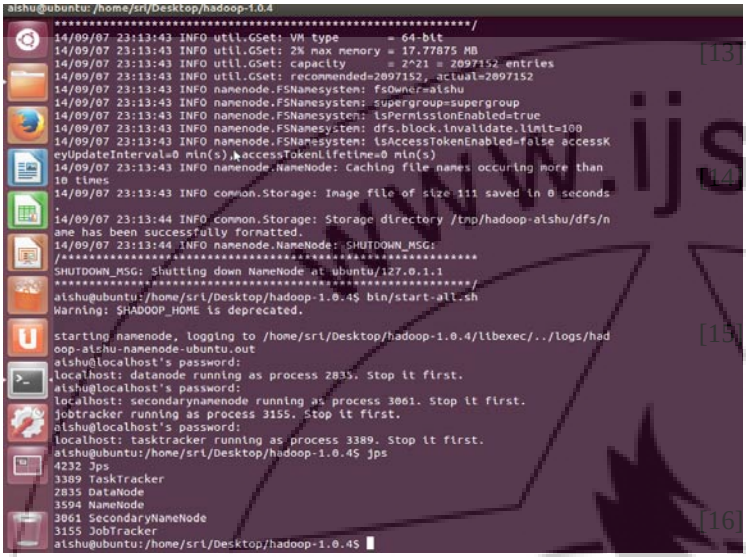
sudo gedit mapred-site.xml
Add the following lines in it

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xs1"
href="configuration.xs1"?>
<!-- Put site-specific property overrides in this file. -->
<configuration>
<property>
<name>mapred.job.tracker</name>
<value>localhost:54311</value>
</property>
</configuration>
```

sudo gedit hdfs-site.xml
Add the following lines in it

```
<?xml version="1.0"?>
<?xml-stylesheet type="textixs1"
href="configuration.xs1"?>.
<!-- Put site-specific property overrides in this file:- -->
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
```

</property>
<property>
</configuration>
Format namenode as
bin/hadoop namenode -format
It will show you a message regarding namenode has been succesfully
formated.



Start the name node by typing **bin/start-all.sh**.
It will start all hadoop daemons.
To check whether all are working type jps.
To stop hadoop type **bin/stop-all.sh.**

References

[1] Konstantin Shvachko, Hairong Kuang, Sanjay Radia, Robert Chansler "The Hadoop Distributed File System". Yahoo!,Sunnyvale, California USA,2010,pp.1-10
[2] Apache Hadoop. http://hadoop.apache.org/
[3] S. Ghemawat, H. Gobioff, S. Leung. "The Google filesystem," In Proc. of ACM Symposium on Operating Systems Principles, Lake George,NY, Oct 2003, pp 2943.
[4] IEEE Standard for Software Test Documentation, IEEE Std 829,1998
J. Dean, S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," In Proc. of the 6th Symposium on Operating Systems Design and Implementation, San Francisco CA, Dec. 2004,pp.1-13
Common IO Download http://commons.apache.org/proper/commons-io/download_io.cgi
[5] K. V. Shvachko, "HDFS Scalability: The limits to growth," ;login:.April 2010, pp. 616.
[6] J. Venner, Pro Hadoop. Apress, June 22, 2009.
[7] T. White, Hadoop: The Definitive Guide. O'Reilly Media, Yahoo! Press,June 5, 2009.
[8] InterMezzo.http://www.inter-mezzo.org
[9] Lustre.http://www.lustre.org
[10] Barbara Liskov, Sanjay Ghemawat, Robert Gruber,Paul Johnson, Liuba Shrira, and Michael Williams. Replication in the Harp file system. In 13th Symposium on Operating System Principles, pages 226238, Pacific Grove, CA,October 1991.
[11] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung.The Google file system. In 19th Symposium on Operating Systems Principles, pages 29.43, Lake George,New York,2003.
[12] Douglas Thain, Todd Tannenbaum, and Miron Livny.Distributed computing in practice: The Condor experience. Concurrency and Computation: Practice and Experience,2004.
[13] P.H. Carns, W.B. Ligon III, R. B. Ross, and R. Thakur . "PVFS: A parallel file system for Linux Clusters," in proc. Of 4th Annual Linux Showcase and conference,200,pp. 317327
[14] Lusture File System. http://www.lustre.org
Software Testing.http://en.wikipedia.org/wiki/Software_testing
Creating Tables with Latex www1.maths.leeds.ac.uk/latex/TableHelp1.pdf
[15] A. Gates, O.Natkovich, S. Chopra, P.kamath, S.Narayanam, C.Olston, B. Reed, S. Srinivasan, U. Srivastava. "Building a Higjh-LevelDataFlow System on top of MapReduce : The Pig Experience," In Proc. Of Very Large Data Bases, vol 2 no. 2, 2009, PP. 14141425 .
[16] S. Radia, "Naming Policies in the spring system," In Proc. Of 1st IEEE Workshop on Services in Distributed and Networked Environments , June 1994, PP. 164171.

Author Profile

Dubellam Dhananjay received Bachelor of Engineering in Computer Science and Engineering from TRRCE, JNTUH. He is pursuing Master of Technology in Computer Science. His research interests are Big Data and Analytics, Operating Systems, Data mining, Networking, Web Technologies, Image Processing, Computer Graphics.

G. Venkata Rami Reddy has completed his Master of Technology in Computer Science from School Of IT, JNTU, Kukatpally Hyderabad. He is the Associate Professor and course coordinator of Software Engineering for School of IT, JNTUH. His subjects of interests are Image Processing, Computer Networks, Analysis of Algorithms, Data mining, Operating Systems and Web technologies.