Fault recovery in HDFS

By Group No. 4

Members:

2017MCS2074 - Harish Chandra Thuwal
2017MCS2076 - Shadab Zafar
2017CSZ8058 - Kolluru Sai Keshav
2017MCS2102 - Chrystle Myrna Lobo

Steps used to install HDFS

1. **Update all Virtual Machines**

```
sudo apt update
sudo apt upgrade
sudo shutdown -r now # restart
```

2. **Install Java Libraries on all the VMs**

```
sudo apt install -y openjdk-7-jre openjdk-7-jdk
```

3. **Add this information to the Hosts file (/etc/hosts) at each machine.**

```
10.17.50.109    vm1
10.17.50.169    vm2
10.17.51.41     vm3
10.17.6.91      vm4
```

4. **Setup Passwordless SSH**

   a. Add this info to the SSH Config file at each machine (~/.ssh/config)

```
Host vm1
       User baadalservervm

Host vm2
       User baadalvm

Host vm3
       User baadalservervm

Host vm4
       User baadalvm
```

b. Generate Key-pairs & Copy the public keys to all machines. Run the following set of commands on all machines.

```
ssh-keygen
ssh-copy-id -i ~/.ssh/id_rsa.pub vm1
ssh-copy-id -i ~/.ssh/id_rsa.pub vm2
ssh-copy-id -i ~/.ssh/id_rsa.pub vm3
ssh-copy-id -i ~/.ssh/id_rsa.pub vm4
```

5. Upload & extract the Hadoop tarball on all machines.

```
rsync -aq ~/Downloads/hadoop-2.6.5.tar.gz vm1:”~/”     # Uploading to vm1
```

```
ssh vm1
tar -xvzf ~/hadoop-2.6.5.tar.gz
mv ~/hadoop-2.6.5 ~/hadoop
```

6. Add necessary environment variables to the ~/.bash_profile file (on all machines)

```
# Hadoop
export HADOOP_HOME=~/hadoop
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64

# for pseudo distributed mode
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export HADOOP_INSTALL=$HADOOP_HOME

export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
```

7. **Edit the Hadoop configuration files, located in the ~/hadoop/etc/hadoop folder**

   a. **masters:** contains a hostname of the NameNode, and secondary NameNode servers.

```
vm1
```

   b. **slaves:** contains a list of hostnames of DataNodes, one per line.

```
vm2
vm3
vm4
```

   c. **core-site.xml:** This file informs the Hadoop daemon where NameNode runs in the cluster and some other I/O settings.

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
      <property>
      <name>fs.default.name</name>
      <value>hdfs://vm1:9000</value>
      </property>
</configuration>
```

   d. **hdfs-site.xml:** contains configuration setting for HDFS daemons: the NameNode, and the DataNodes. Configuration include the default replication factor, path of namenode and datanode directories, permission checking etc.

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <property>
      <name>dfs.replication</name>
      <value>2</value>
  </property>
  <property>
      <name>dfs.permissions</name>
      <value>false</value>
  </property>
  <property>
      <name>dfs.namenode.name.dir</name>
      <value>/home/vm*/hdfs/namenode</value>
  </property>
  <property>
      <name>dfs.datanode.data.dir</name>
      <value>/home/vm*/hdfs/datanode</value>
  </property>
</configuration>
```

e. **mapred-site.xml:** contains the configuration settings for MapReduce daemons: the job tracker and the task-trackers.

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
      <property>
      <name>mapreduce.job.tracker</name>
      <value>vm1:5431</value>
      </property>
      <property>
      <name>mapreduce.framework.name</name>
      <value>yarn</value>
      </property>
</configuration>
```

f. **yarn-site.xml:** contains configuration setting and address of the resource manager.

```
<?xml version="1.0"?>
<configuration>
  <property>
      <name>yarn.resourcemanager.hostname</name>
      <value>vm1</value>
  </property>
  <property>
  <name>yarn.resourcemanager.resource-tracker.address</name>
```

```
      <value>vm1:8025</value>
  </property>
  <property>
      <name>yarn.resourcemanager.scheduler.address</name>
      <value>vm1:8035</value>
  </property>
  <property>
      <name>yarn.resourcemanager.address</name>
      <value>vm1:8050</value>
  </property>
</configuration>
```

8. **Create the HDFS directories at each machine**

```
mkdir -p ~/hdfs/namenode && mkdir -p ~/hdfs/datanode
```

9. **Format the NameNode (VM1)**

```
hadoop namenode -format
```

10. **Once the setup phase is complete, we can now start HDFS**

    Run the following command at VM1 the namenode:

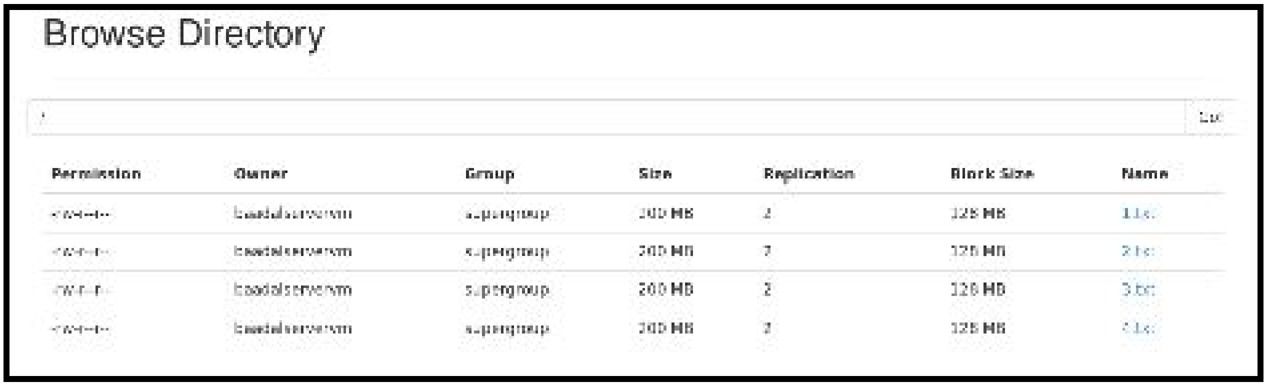
```
start-all.sh
```

Distribution of File Blocks

In order to explore the file-block distribution and replication in HDFS we added four **200MB** text files to it using the command on the namenode:

```
hdfs dfs -put file_num.txt
```

Following is the screenshot of the hdfs root directory after uploading the files.



Browse Directory

/

Permission	Owner	Group	Size	Replication	Block Size	Name
-rw-r--r--	baadalservervm	supergroup	200 MB	2	128 MB	1.txt
-rw-r--r--	baadalservervm	supergroup	200 MB	2	128 MB	2.txt
-rw-r--r--	baadalservervm	supergroup	200 MB	2	128 MB	3.txt
-rw-r--r--	baadalservervm	supergroup	200 MB	2	128 MB	4.txt

Initial Block Distribution

To see the block distribution of the files, SSH into the **NameNode** and run the following command:

```
hdfs fsck / -files -blocks -locations
```

Following is the output of the above command:

```
FSCK started by baadalservervm (auth:SIMPLE) from /10.17.50.109 for path / at Tue
Aug 28 12:59:43 IST 2018
/ <dir>
/1.txt 209715200 bytes, 2 block(s):  OK
0. BP-1958787498-10.17.50.109-1535117509695:blk_1073741827_1003 len=134217728
repl=2
      [10.17.50.169:50010, 10.17.51.41:50010]
1. BP-1958787498-10.17.50.109-1535117509695:blk_1073741828_1004 len=75497472 repl=2
      [10.17.51.41:50010, 10.17.50.169:50010]
```

```
/2.txt 209715200 bytes, 2 block(s):  OK
0. BP-1958787498-10.17.50.109-1535117509695:blk_1073741829_1005 len=134217728
repl=2
      [10.17.50.169:50010, 10.17.51.41:50010]
1. BP-1958787498-10.17.50.109-1535117509695:blk_1073741830_1006 len=75497472 repl=2
      [10.17.51.41:50010, 10.17.6.91:50010]

/3.txt 209715200 bytes, 2 block(s):  OK
0. BP-1958787498-10.17.50.109-1535117509695:blk_1073741831_1007 len=134217728
repl=2
      [10.17.51.41:50010, 10.17.6.91:50010]
1. BP-1958787498-10.17.50.109-1535117509695:blk_1073741832_1008 len=75497472 repl=2
      [10.17.51.41:50010, 10.17.6.91:50010]


/4.txt 209715200 bytes, 2 block(s):  OK
0. BP-1958787498-10.17.50.109-1535117509695:blk_1073741833_1009 len=134217728
repl=2
      [10.17.6.91:50010, 10.17.51.41:50010]
1. BP-1958787498-10.17.50.109-1535117509695:blk_1073741834_1010 len=75497472 repl=2
      [10.17.51.41:50010, 10.17.50.169:50010]
```

- It can be observed that the blocks corresponding to each file are replicated twice **(replication factor = 2)**.
- Also all the replicas of each block are stored on different node. That is for one block no two replicas are present on the same data node

Files block distribution	**Data Nodes**		
	10.17.50.169 (vm2)	10.17.51.41 (vm3)	10.17.6.91 (vm4)
1.txt	B0,B1	B0, B1	
2.txt	B0	B0, B1	B1
3.txt		B0, B1	B0, B1
4.txt	B1	B0, B1	B0

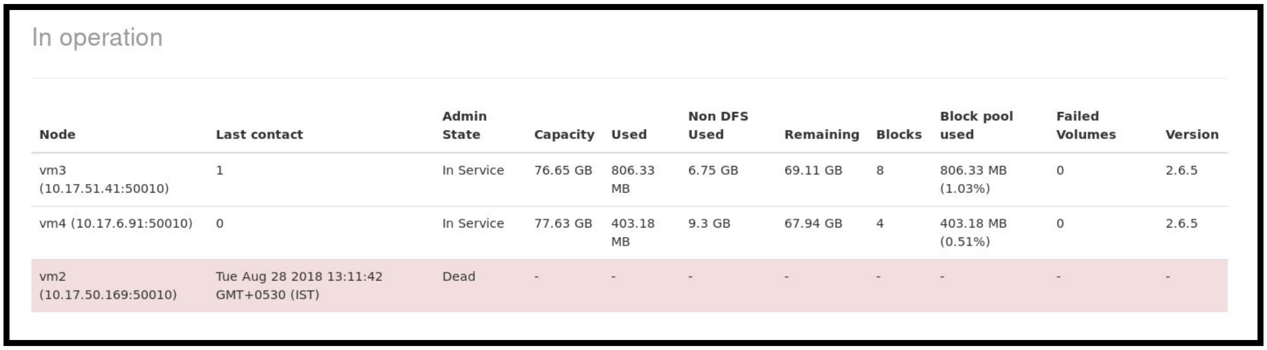
- The load on vm3 is twice as compared to vm2 and vm4. That is a load distribution of **1 : 2 : 1.**

Shutdown a DataNode (vm 2)

To observe the behaviour of the HDFS system in the scenario when one of the nodes fails or is unreachable, ssh into vm2 and shut it down by executing the following command:

```
sudo shutdown -P now
```

Following is the screenshot of the list of DataNodes in service to the NameNode. VM2 now appears dead to the namenode an it has been shut down.



In operation

Node	Last contact	Admin State	Capacity	Used	Non DFS Used	Remaining	Blocks	Block pool used	Failed Volumes	Version
vm3 (10.17.51.41:50010)	1	In Service	76.65 GB	806.33 MB	6.75 GB	69.11 GB	8	806.33 MB (1.03%)	0	2.6.5
vm4 (10.17.6.91:50010)	0	In Service	77.63 GB	403.18 MB	9.3 GB	67.94 GB	4	403.18 MB (0.51%)	0	2.6.5
vm2 (10.17.50.169:50010)	Tue Aug 28 2018 13:11:42 GMT+0530 (IST)	Dead	-	-	-	-	-	-	-	-

We observed that it took ~500 seconds for the NameNode to declare that the DataNode has been disconnected. Meanwhile it kept updating the number of seconds since last contact.

Block Distribution after Shutdown of vm2

Once the NameNode realized that the DataNode is dead, the NameNode began re-replication of Blocks because the replication factor is now no longer 2.

Output of fsck command after re-replication:

```
FSCK started by baadalservervm (auth:SIMPLE) from /10.17.50.109 for path / at Tue
Aug 28 13:24:32 IST 2018
/ <dir>
/1.txt 209715200 bytes, 2 block(s):  OK
0. BP-1958787498-10.17.50.109-1535117509695:blk_1073741827_1003 len=134217728
repl=2
       [10.17.51.41:50010, 10.17.6.91:50010]
1. BP-1958787498-10.17.50.109-1535117509695:blk_1073741828_1004 len=75497472 repl=2
       [10.17.51.41:50010, 10.17.6.91:50010]
```

```
/2.txt 209715200 bytes, 2 block(s):  OK
0. BP-1958787498-10.17.50.109-1535117509695:blk_1073741829_1005 len=134217728
repl=2
      [10.17.51.41:50010, 10.17.6.91:50010]
1. BP-1958787498-10.17.50.109-1535117509695:blk_1073741830_1006 len=75497472 repl=2
      [10.17.51.41:50010, 10.17.6.91:50010]

/3.txt 209715200 bytes, 2 block(s):  OK
0. BP-1958787498-10.17.50.109-1535117509695:blk_1073741831_1007 len=134217728
repl=2
      [10.17.51.41:50010, 10.17.6.91:50010]
1. BP-1958787498-10.17.50.109-1535117509695:blk_1073741832_1008 len=75497472 repl=2
      [10.17.51.41:50010, 10.17.6.91:50010]

/4.txt 209715200 bytes, 2 block(s):  OK
0. BP-1958787498-10.17.50.109-1535117509695:blk_1073741833_1009 len=134217728
repl=2
      [10.17.6.91:50010, 10.17.51.41:50010]
1. BP-1958787498-10.17.50.109-1535117509695:blk_1073741834_1010 len=75497472 repl=2
      [10.17.51.41:50010, 10.17.6.91:50010]
```

- The NameNode quickly replicated all the Data blocks that were lost (by shutting down vm2) to vm4 restoring the replication factor back to 2.

Files block distribution	**Data Nodes**	
	10.17.51.41 (vm3)	10.17.6.91 (vm4)
1.txt	B0, B1	B0, B1
2.txt	B0, B1	B0, B1
3.txt	B0, B1	B0, B1
4.txt	B0, B1	B0, B1

- The load in this case is equally balanced **(1:1)** between the two data nodes.
- Still no two replicas of same block are present on same node. This however, wouldn't have been the case if:
  - there were more number of blocks or
  - replication factor was > 2.

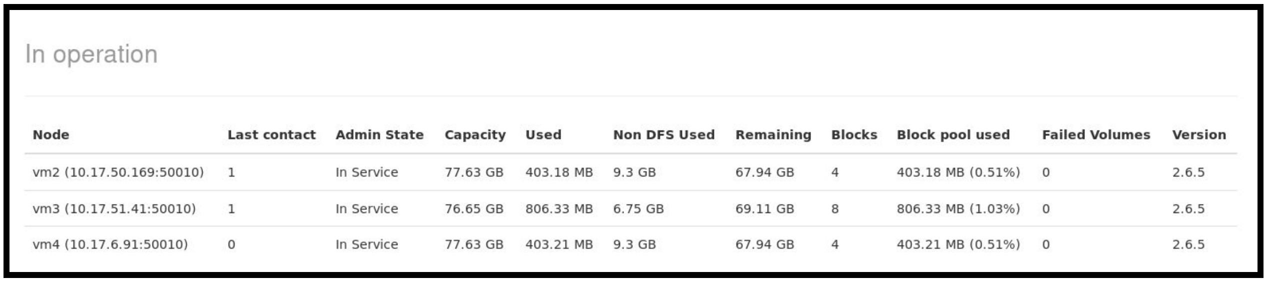
Retrieve files and compare with original files

We downloaded the files from the web file system browser. To compare the files we ran the linux **diff** command for each file.

```
diff original_file dowloaded_file
```

- The execution of the above command for any of the files produced no output (no differences).
- This Indicates the fact that the files were unchanged even after shutting down one of the data nodes.
- There was no loss of data because each block of each file was replicated.

Restoring VM2



In operation

Node	Last contact	Admin State	Capacity	Used	Non DFS Used	Remaining	Blocks	Block pool used	Failed Volumes	Version
vm2 (10.17.50.169:50010)	1	In Service	77.63 GB	403.18 MB	9.3 GB	67.94 GB	4	403.18 MB (0.51%)	0	2.6.5
vm3 (10.17.51.41:50010)	1	In Service	76.65 GB	806.33 MB	6.75 GB	69.11 GB	8	806.33 MB (1.03%)	0	2.6.5
vm4 (10.17.6.91:50010)	0	In Service	77.63 GB	403.21 MB	9.3 GB	67.94 GB	4	403.21 MB (0.51%)	0	2.6.5

Output of the Fsck command after restoring VM2.

```
FSCK started by baadalservervm (auth:SIMPLE) from /10.17.50.109 for path / at Tue
Aug 28 13:38:15 IST 2018
/ <dir>
/1.txt 209715200 bytes, 2 block(s):  OK
0. BP-1958787498-10.17.50.109-1535117509695:blk_1073741827_1003 len=134217728
repl=2
      [10.17.51.41:50010, 10.17.50.169:50010]
1. BP-1958787498-10.17.50.109-1535117509695:blk_1073741828_1004 len=75497472 repl=2
      [10.17.51.41:50010, 10.17.50.169:50010]
```

```
/2.txt 209715200 bytes, 2 block(s):  OK
0. BP-1958787498-10.17.50.109-1535117509695:blk_1073741829_1005 len=134217728
repl=2
      [10.17.51.41:50010, 10.17.50.169:50010]
1. BP-1958787498-10.17.50.109-1535117509695:blk_1073741830_1006 len=75497472 repl=2
      [10.17.51.41:50010, 10.17.6.91:50010]

/3.txt 209715200 bytes, 2 block(s):  OK
0. BP-1958787498-10.17.50.109-1535117509695:blk_1073741831_1007 len=134217728
repl=2
      [10.17.51.41:50010, 10.17.6.91:50010]
1. BP-1958787498-10.17.50.109-1535117509695:blk_1073741832_1008 len=75497472 repl=2
      [10.17.51.41:50010, 10.17.6.91:50010]

/4.txt 209715200 bytes, 2 block(s):  OK
0. BP-1958787498-10.17.50.109-1535117509695:blk_1073741833_1009 len=134217728
repl=2
      [10.17.6.91:50010, 10.17.51.41:50010]
1. BP-1958787498-10.17.50.109-1535117509695:blk_1073741834_1010 len=75497472 repl=2
      [10.17.51.41:50010, 10.17.50.169:50010]
```

- As the vm2 is now up again with all of its copies intact, there are three copies of each block which is 1 more than the replication factor of 2.
- The NameNode deletes the extra replicas and in such a way that the load is balanced as much as possible while simultaneously avoiding keeping two copies of same block on each node.
- The best configuration, in this case, is the one we started with. The one with **1:2:1** load distribution.

Files block distribution	**Data Nodes**		
	10.17.50.169 (vm2)	10.17.51.41 (vm3)	10.17.6.91 (vm4)
1.txt	B0,B1	B0, B1	
2.txt	B0	B0, B1	B1
3.txt		B0, B1	B0, B1
4.txt	B1	B0, B1	B0

References

https://hadoop.apache.org/docs/r1.2.1/cluster_setup.html

HDFS class notes