Accessibility

HDFS can be accessed from application in many different ways. Natively, HDFS provides a FileSystem Java API for applications to use. A C language wrapper for this JavaAPI and REST API is also available. In addition an HTTP browser can also be used to browse the files of an HDFS instance. By using NFS gateway, HDFS can be mounted as part of the client's local file system.

FS shell

HDFS allows user data to be organised in the form of files and directories. It provides a command line interface called ***fs shell*** that lets a user interact with the data in HDFS. The syntax of this command set is similar to other shells (bash) that users are already familiar with. Here are some sample action/command pairs:

Action	Command
Create a directory named /foodir	bin/hadoop dfs -mkdir /foodir
Remove a directory named /foodir	bin/hadoop fs -rm -R /foodir
View the contents of a file named /foodir/myfile.txt	bin.hadoop dfs -cat /foodir/myfile.txt

DFSAdmin

The dfsadmin command set is used for administering an HDFS cluster. These are commands that are used only by an HDFS administrator. Here are some sample action/command pairs:

Action	Command
Put the cluster in safemode	bin/hdfs dfsadmin -safemode enter
Generate a list of DataNodes	bin/hdfs dfsadmin -report
Recommission or decommission DataNode(s)	bin/hdfs dfsadmin -refreshNodes

Browser Interface

A typical HDFS install configures a web server to expose the HDFS namespace through a configurable TCP port. This allows a user to navigate the HDFS namespace and view the contents of its files using a web browser.

Space Reclamation

File deletes and undeletes

If trash configuration is enabled, files removed by FS Shell are not immediately removed from HDFS. Instead, HDFS moves it to a trash directory. Each user has his own trash directory under

```
/user/<username>/.Trash
```

The file can be restored quickly as long as it remains in trash. Most recent deleted files are moved to the current trash directory

```
/user/<username>/.Trash/Current
```

and in a configurable interval, HDFS creates checkpoints under the following directory for files in current trash directory and deletes old checkpoints when they are expired. (**NOTE:** See expunge command of FS shell about check pointing of trash.)

```
/user/<username>/.Trash/<date>
```

After the expiry of its life in trash, the NameNode deletes the file from the HDFS namespace. The deletion of a file causes the blocks associated with the file to be freed. Note that there could be an appreciable time delay between the time a file is deleted by a user and the time of the corresponding increase in free space in HDFS.

Decrease Replication Factor

When the replication factor of a file is reduced, the NameNode selects excess replicas that can be deleted. The next Heartbeat transfers this information to the DataNode. The DataNode then removes the corresponding blocks and the corresponding free space appears in the cluster. Once again, there might be a time delay between the completion of the setReplication API call and the appearance of free space in the cluster.

Sources:

1. Main