Simulating HDFS, HDFS-RAID and Locally Repairable Codes (LRC)

USER MANUAL

By

Mansukh Shrimali(200901098)

Ankit Sharma(200901099)

Table of contents:

1. Introduction.. 3
2. How it works... 3
   2.1. Starting Project.. 3
   2.2. HDFS simulation... 3
   2.3. HDFS RAID simulation.. 6
   2.4. LRC simulation... 7
   2.5. More than one simulation at one time.................... 7

1. Introduction

This simulator takes a file and stores it in different nodes using 3 simulation methods: HDFS, HDFS-RAID and Locally Repairable Codes (LRC). User can also manually select any combination of these methods for encoding, decoding and regeneration techniques.

2. How it works

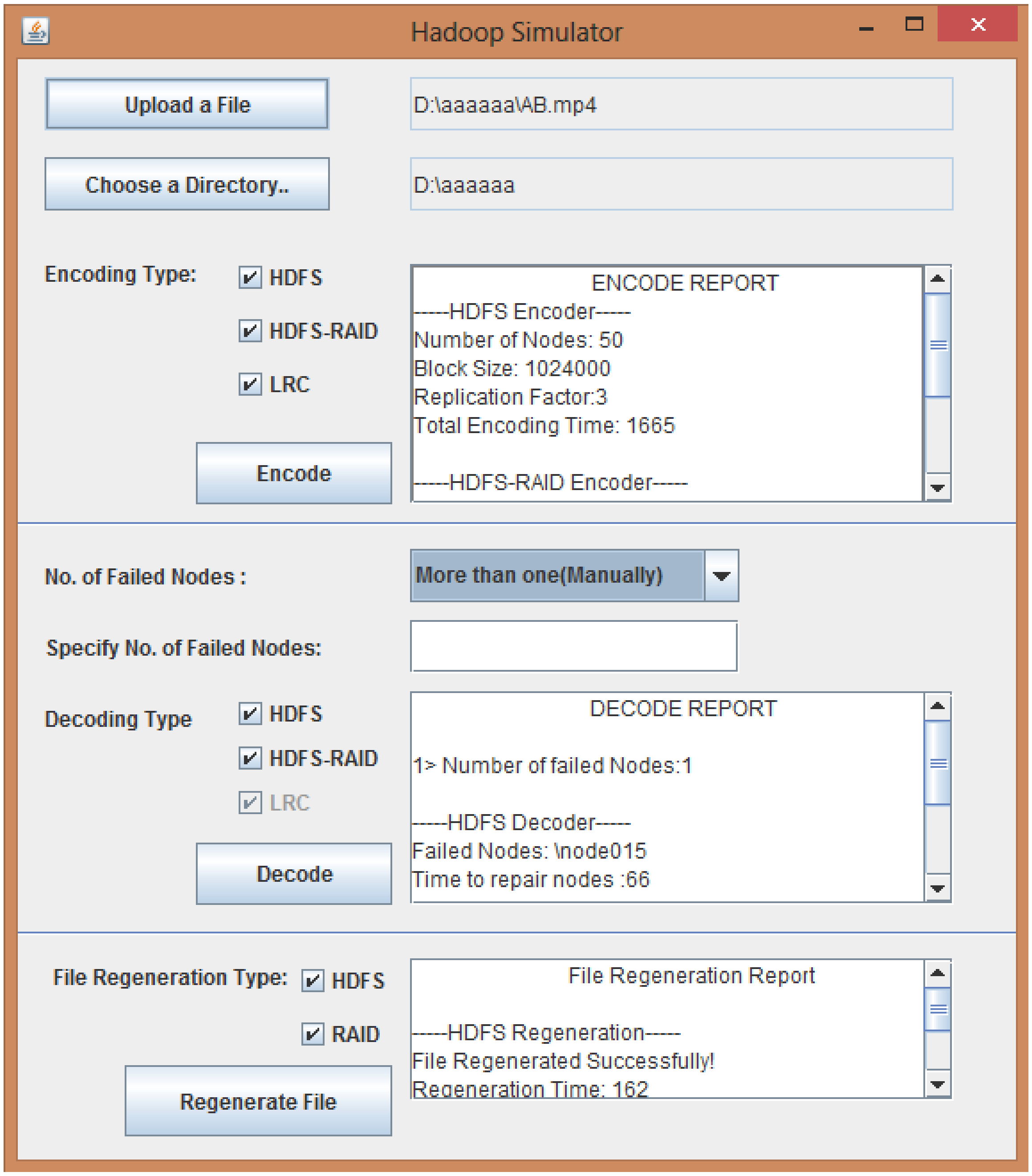
2.1 Starting and importing project

User can download a source code of the project from the following link: http://www.guptalab.org/mankg/public_html/WWW/students.html.

Once a zip file is downloaded then user can easily import the project using any java IDE with File -> import project -> zip file. After running the project, a simulation window will appear. Fig. 1 is a snapshot of that window.

2.2 HDFS Simulation

First Upload a file button allows the user to upload a file for simulation. Then *Choose a directory* button allows the user to choose a directory to store the project works. Path of the file and the directory must not contain any blank space in their location.



D:\aaaaaa\AB.mp4

D:\aaaaaa

Encoding Type: ✓ HDFS ✓ HDFS-RAID ✓ LRC

```
ENCODE REPORT
-----HDFS Encoder-----
Number of Nodes: 50
Block Size: 1024000
Replication Factor:3
Total Encoding Time: 1665

-----HDFS-RAID Encoder-----
```

No. of Failed Nodes : More than one(Manually)

Specify No. of Failed Nodes:

Decoding Type ✓ HDFS ✓ HDFS-RAID ✓ LRC

```
DECODE REPORT

1> Number of failed Nodes:1

-----HDFS Decoder-----
Failed Nodes: \node015
Time to repair nodes :66
```

File Regeneration Type: ✓ HDFS ✓ RAID

```
File Regeneration Report

-----HDFS Regeneration-----
File Regenerated Successfully!
Regeneration Time: 162
```

Fig.1 Snapshot of Simulator

Now by selecting check box HDFS in Encoding section and then pressing *Encode* button stores and replicates all file blocks in all nodes according to replication factor, Block size and Number of 5 nodes given in *config.properties* file, which is saved in folder of this project. User can manually change values in that file. Following are the default values:

1. Replication factor = 3
2. Block size = 1024000 (1000 KB)
3. Total number of nodes = 50

Encode report text area shows logs generated during encoding and storing of the file. Now User can regenerate the stored file by selecting check box HDFS and using *Regenerate File* button in last section. User can select no. of nodes to fail from given three option: *one, more than one (Automatically) and more than one (Manually).* If last option is selected then user will have to enter no. of nodes to fail in *Specify no. of Failed nodes* textbox and select check box HDFS. After that *decode* button will first fail selected no. of nodes and then tries to regenerate those failed nodes and shows log in *Decode Report*.

Decode button will fail nodes and then it will regenerate those failed nodes. By default, Decode will run only once. But to compare the results of HDFS with HDFS-RAID by running the decoder more than once, user can select manually how many time user wants to take observation by setting the node failure option to "more than one (Automatically)" and setting the value of *No. of Times* more than '1'. Graph will be generated describing the average node regeneration time between HDFS and HDFS-RAID.

2.3 HDFS RAID Simulation

When HDFS-RAID encoding mode is selected, a new subdirectory under the main project directory will be created with the name “\RAID\”. Then according to the file size, different nodes will be created. One node can contain maximum 16 blocks. And 1 block has maximum size of 1MB. Therefore, one Node can contain maximum 16MB of original data. For example, if the file has the size of 50 MB, then four Nodes will be created. In the first 3 Nodes, 16 blocks of 1MB each will be filled and in the last Node, remaining 2 blocks will filled.

After dividing the files into chunks and storing it into different Nodes, parity files will be generated. For each Node with 16 blocks, 10 parity files of 1MB each will be generated. Therefore, one node contains total parity of maximum 10MB. This encoding is done using RS encoding. And the parity files for each Node is stored under the sub-directory named “\RS_Parity\”.

Our approach can tolerate up to maximum 5 block failures. If a block is failed, the parity files will be called and the blocks will be repaired using RS decoding.

In Decode section, if HDFS RAID is selected then first simulator will fail selected no. of nodes then tries to regenerate those failed nodes and shows log in Decode Report.

2.4 LRC simulation

In Encoding section of window if LRC checkbox is selected and Encode button is pressed then local parity files will be generated and stored. Then, under each Node, a subdirectory will be created which will contain LRC parity with the name "\LRC_Parity\". Now for every 8 blocks, one local parity will be stored under the subfolder. Therefore, one Node can have maximum 2 local parities. The parity will be generated with the XOR operation between the corresponding 8 blocks. Local parity for the 10 RS parity files will also be stored under "\LRC_Parity\".

Now, if one block is failed, it can be recovered by using the local LRC parity and other remaining 7 blocks. File Regeneration in LRC is same as in HDFS RAID, because LRC is layered on HDFS RAID by adding local parity files. If a single node fails then LRC's local parity file will regenerate the failed node and restore it but if more than one node fails then LRC Decoding checkbox is disabled because LRC cannot regenerate more than one failed node but it uses HDFS RAID scheme for that situation.

2.5 More than one simulation at one time

If more than one simulation technique from HDFS, HDFS RAID and LRC is selected in Encode, Decode or Regenerate File options, then all appropriate techniques will be applied and proceed accordingly. But to Regenerate file or Decode file in one technique a file must be encoded and stored in that technique.