Design:

We started with the framework for the recommended MP2 failure detector, put on the course website.

We decided to extend this version of the MP, because we lost points on our attempt. Our code is divided into two parts: the failure detector, and the file transfer protocol. We added additional message types to the ping-ack failure detector. We appoint the initiator as a master node when the first incoming node asks to join. If the introducer node fails, it can be re-joined by declaring any active node to be its introducer.

After this point, whenever a node is told to put a file, it first consults that master node, who coordinates where files should be put in the sdfs in order to remain failsafe for up to two consecutive failures. The master node also communicates to other nodes when a file is to be deleted, or where to retrieve a file from if it is not stored on the same node as the request was made. The master node keeps up to two backup nodes who have the same full set of file location information as the master node, and know of each other, to allow for election when a master node fails.

When a master fails, a backup who notices this will compare itself with the other backup node. If it is a better choice (existed longer), it will declare itself the new leader. Once an election is completed, the new master immediately creates a new backup node, and informs both backups of each other and the current dfs state. It then rebalances files, moving files with less than three copies on the dfs to nodes with the fewest files who do not yet have a copy of that file. After the rebalancing is done, the backups are again informed of the new state of the dfs. Backups are also informed when a file is added or deleted, even when there are no failures.

We used mutexes to guard whenever a node’s master file list or member list is being changed, as our code uses many threads and forks frequently, and we do not want to create a race condition. We have specific different messages for sending from a master node and sending to a master node, with node\_ids and filenames delimited by the ‘#’ character.

We have separate threads for user input, for listening to dfs messages, and for sending periodic pings (the failure detection app). We always fork whenever we are going to send a message that requires a response to prevent delays or false reporting of timeouts.

MP1 for debugging:

Again, we used direct printouts to terminal to debug our sdfs, rather than trying to create specific useful grep queries. More data is still better, and MP1 was minimally useful. Had we not been able to debug with print statements, MP1 would have been significantly more useful.

Measurements:

: (i) re-replication time and bandwidth upon a failure (you can measure for a 30 MB file); (ii) time between master failure and new master being reinstated; (iii) times to read and write one file of size 20 MB, 500 MB (4 total data points), under no failure; (iv) time to store the entire English Wikipedia corpus into SDFS with 4 machines and 8 machines (not counting the master): use the Wikipedia English (raw text) link at: http://www.cs.upc.edu/~nlp/wikicorpus/ . For each data point, run at least 5 trials and give the average and standard deviation bars. Discuss your plots, don’t just put them on paper, i.e., discuss trends, and whether they are what you expect or not (why or why not). (Measurement numbers don’t lie, but we need to make sense of them!)