**Machine Programming 3 – Simple Distributed File System**

**Yunsheng Wei (wei29) Neha Chaube (nchaub2)**

In MP2, we developed a simple distributed file system service. Some design decisions are as follows:

1. A master server called the Name Node which relies on a group membership service(in our case Gossip-style) is a daemon service which is responsible for maintaining the metadata information about the SDFS which includes getting file locations, putting request to data node for adding file, deleting file and fetching file. It also takes the responsibility of replicating file on three data nodes in case of failures.
2. A Block Report is sent from data node to name node which contains information about all the SDFS files present on that data node.
3. A Data Node is a daemon service which relies on Leader Election Service. It performs addition, deletion and replication of SDFS files after receiving request from the master node.
4. The Client node encapsulates all client operations of adding, deleting and fetching files on SDFS. The Client node basically send request to Master node which in turn send request to data node to perform the file operations.
5. In order to handle failure of Master node which is the leader, Leader Election service is built on top of Gossip Group Membership i.e. oldest alive membership is fetched from the membership list and elected as the new leader if the master node fails.

In MP3, we use MP1’s distributed log querier to query and get the results for our experiments.

We also use MP2’s group membership service to implement our leader election service by making use of the membership list.

**Experiments**

1. **Re-replication time and bandwidth upon a failure**
2. **Time between master failure and new master being reinstated**
3. **Times to read and write one file of size 20 MB, 500 MB (4 total data points), under no failure**
4. **Time to store the entire Wikipedia corpus into SDFS with 4 machines**

**Average and Standard Deviation Plots**