Distributed Systems
Project Part 1
HDFS
Deadline: 7th March 5:00 PM

This part of the project requires to implement a distributed file system similar to HDFS. The distributed files system will have two major components NameNode and DataNode. Functionality of these components is as follows

- **NameNode:**
  It supports all the file operations like open, close, list. The file system supports a block mechanism similar to HDFS. So, NameNode also takes care of managing block allocations and get locations of existing blocks. It will also handle the communication with DataNode.

- **DataNode:**
  Used for reading and writing blocks.

- **Important Points:**
  - Use RMI (Java) or Sun RPC (C/C++) for communication
  - Server should be multi-threaded
  - No update operation
  - 2-way replication
  - You can have a configuration file of your own
  - Add CRC for communication (optional but recommended)
  - All the parameters and return types are byte arrays (need to marshal and unmarshal actual objects)

- **Detailed Workflows:**
  - Open
    - Assign initial block.
    - Unique handle for each opened file.
  - Write
    - If no block available then assign new block
    - Write contents to the block
  - Read
    - Get all block locations for the file
    - Read blocks in sequence
  - No update operation
  - Heartbeat should be executed periodically to know the status of DataNode

- BlockReport
  - DN sends blockReport() to inform the NN about all the blocks it has.
  - For each file, NN knows just the filename and list of blocks. NN does not store the locations. Once DNs send blockReports, NN knows the locations of each block and tracks this information in memory - it never writes this information to a disk.
  - NN has to maintain the list of files, and the blocks in each file in a directory (or a file) in the local OS.