White Paper

Architecture:

We will use a traditional client server model where the clients will discover the leader by connecting to one any of the servers in the static list given to it as a command line argument. That server will tell the client who the leader is. The client then sends the RMI call to that server.

• Coordinator:

We choose to use the bully algorithm for our election system. Instead of using PID we will use timestamps for the comparator of who the coordinator will be, with the smallest value winning. When a client requests the leader's address, the server will check if it is up. If it isn't, the election will be held, and the new leader's address will be returned.

• Consistency model:

The consistency model we will use is the sequential consistency model. Any rmi call to the leader will be propagated to the backup servers in the order that the events

happened. The inconsistency window will only be the amount of time to hold a re-election of the leader.

• Server synchronization:

Updates will be sent to backup servers through multicasting as soon as the leader receives RMI request from the client.

– Granularity:

Each server will contain an individual copy of the database and the leader will forward every rmi client call to those servers to update their local database.