Ali Azhar 1000616620
Waref Haque 1000265615
Jeffrey (Yufei) Hua 1000067205

M2 Project Report
M2 Design and Design Decisions

Client Application and Client Library:

Implementation of three additional errors:

The client application keeps all the functionalities from Milestone 1 which allows a user to store key value pair on remote servers. Provided that the storage server advanced from basic storage server to scalable storage server, a user might experience SERVER_WRITE_LOCK and SERVER_STOPPED errors and will be asked to make another attempt.

SERVER_NOT_RESPONSIBLE is the third additional error in this milestone. However, this error will not reach the client shell, instead it is handled within client library. In the case that client receives such an error, it updates its metadata, looks up the responsible server and retries. It will repeats the same process until it no more receives SERVER_NOT_RESPONSIBLE error.

Implementation of metadata:

Metadata is received along SERVER_NOT_RESPONSIBLE error and is formatted as following:
<hashedStart>,<hashEnd>,<IP>,<port> <hashedStart>,<hashEnd>,<IP>,<port>
<hashedStart>,<hashEnd>,<IP>,<port> …

The client library utilizes **Consistent Hashing** class provided by ECS to manage metadata, in which case both client and ECS share the same ring hash structure of the servers.

Upon reception of metadata, the ring structure will be removed from consistent hashing class and a completely new structure will be created based new metadata.

Server:

The ECS server instantiates servers with their port initially. The server is set up such that a int state variable decides whether a client who has connected will have their requests processed (initially set to SERVER_STOPPED). Once a server has been started, the state variable is set to SERVER_READY to indicate that clients will have their requests processed. In order to stop a server, the state variable is set to SERVER_STOPPED, but once all clients processing their requests have finished processing, then will the server reply back to the ECS server to indicate a successful stopping. The same goes for the newly added shutDown method.

Ali Azhar 1000616620
Waref Haque 1000265615
Jeffrey (Yufei) Hua 1000067205

LockWrite is implemented in a similar way, such that every client currently processing their write requests can finish processing before we notify the ECS server (the banWrite variable is set to BAN_WRITE so that no one else can attempt to process their write requests). UnLockWrite is basically setting the banWrite variable to ALLOW_WRITE.

External Configuration Service:

The external configuration services comprises of the Ecs class, the Consistent Hashing class, the Ecs Client class and the configuration file which contains the server ip addresses and ports.

Ecs Client: Responsible for accepting commands from the admin. Commands include:

- **connect <number of servers>** : Takes the IP address and port from random servers in the configuration file and then runs the ms3-server.jar <port> on separate ports which causes servers to listen on these ports.
- **start**: Tells the servers that were initiated using the connect command to take commands from client now.

Consistent Hashing: Consists of a Sorted Map called "circle" which represents the unit circle. The keys of the Sorted Map are hash values that are provided by an MD5 hash function and the values are Hashed Server objects. The Hashed Server consists of:

- **Array:** which holds the start and end hash that each server is responsible for
- **String:** which holds the ip address and the port.

Key functions:

- **add:** adds a Hashed Server object to the sorted map, then re computes the ranges that all the servers in the Sorted Map are responsible for.
- **remove:** removes a Hashed Server object from the sorted map, then re computes the ranges that all the servers in the Sorted Map are responsible for.
- **get:** given the ip address and port as a string, it computes a hash value and then finds the next largest hash value in the Sorted Map. It then returns the corresponding Hashed Server.

Ecs Client: Binds commands from the client terminal and the Consistent Hashing class to instantiate servers using SSH and then creates sockets to communicate with the corresponding servers.

Ali Azhar 1000616620
Waref Haque 1000265615
Jeffrey (Yufei) Hua 1000067205