CSE403: Advanced Operating Systems

Jacob Nelson, jjn217

HW1 – Feb 29th, 2018

**Introduction**

The aim of this project was to develop a distributed hash table (DHT). Each node in a cluster is responsible for some subset of the key value pairs. Together, they contain the entire hash table, but they do not have any direct knowledge other than their own local table, and where to send requests given a key that is not local to that node. My implementation is broken into two parts. A node library, used for creating and maintaining a node in the cluster, and a client library to interact with the existing DHT. The next section details their architecture.

**Architecture**

Each DHTNode contains an instance of a HashTable which, as per the specifications, can be added to and read from. Once a key-value pair is added to the table it cannot be edited. A few custom operations I included for easy of testing include a size function and a clear function. They are not safe and will not behave correctly if run simultaneously with put and get operations. The DHTNode also includes a Network object, which handles all communication with the other nodes. It has knowledge of the network mapping and is responsible for sending and receiving messages (from both the client and other nodes). Each node’s Network listens for incoming connections from clients. When one is received the Network parses the incoming message and passes it to the node for processing. The node then decides if it should be processed locally or on another node. In the local case, the node adds the key-value pair to the local HashTable and sends a response to the client. For a key that exists on a remote node, the network connects to that node and passes a request as if it were itself a client. When the remote node is done handling the request the original recipient of the client message relays the message back. In this way, each node is responsible for delegating and responding to the messages it receives from clients.

The second piece of the implementation is the DHTClient, which is packaged as a client library. The client does not know any internal mappings of the DHT and only connects to a single IP address. In fact, in the configuration I tested, the network