Bobby Jones, Conor M, Matt Hennes, Travis Michael, Juan Carlos Ferrel
Prof. Lupo
CPE 458
Chord DHT

Implemented Features

Our Chord ring supports adding nodes, adding simple (string) key-value pairs, and adding entire files to the ring, as well as retrieving simple key-values and entire files from any node. The ring also supports major faults in the chord ring, and is generally able to survive half or even more nodes going offline, depending on how much time passes between failures allowing the system to redistribute chunks.

Adding nodes is done through the console, and can take arguments of what port the node should run on as well as the address of the node you would like to join the Chord ring through. In this way, it is easy to add multiple nodes to a single machine, or to add nodes to a network from other machines. Creating the initial Chord ring (the first node) is done by creating a node with no bootstrap given.

Adding simple key-value pair consists of a String key and a String value, which is stored as a single chunk (and therefore on a single node, plus duplicates). The node that gets the key-value pair is determined through a hashing function in OpenChord.

In order to add entire files to the Chord ring, we took advantage of the already implemented functionality of adding a string key with a string value. We first add a key-value pair with the filename to be added as the key, and the number of 4kB chunks the file will consist of as the value. We then break the file into 4kB chunks, and encode them as hex ascii. This allows us to treat the chunks as strings, and easily add them to the ring. They key for each chunk is the filename with a number appended to the end representing which chunk it is.

In order to retrieve files, we first lookup the filename as the key (passed in as a parameter) to determine how many chunks we need to find. We then iterate through a loop and retrieve each chunk, convert it back into binary from hex ascii, and output it to a file.

Improvements

Areas that could be improved on for this project would be the feature to see where the chunks are distributed within the chord ring. With this feature being implemented, when something is being retrieved from the Chord ring, it would be able to balance the load across several nodes, acting in a similar fashion to a torrent. This would make sure that one node doesn't just get blasted with retrieves while the other nodes are sitting idle.

Another improvement that could be made to the Chord ring is the way that we store the chunks within it. As of right now, the chunks are stored as strings of hex values. These chunks could instead be serialized or compressed, reducing how much space it takes to store them and the amount of bandwidth that used when retrieving the chunks from the ring.

A final and crucial improvement that could be made would be handling files that are not directly in the directory that the Chord ring is being run from. As of right now, it only works with files that are in the exact path and also only with files that do not have a dash in the filename. Handling files from different directories is nearly essential for a functional Chord ring. On top of this, allowing a client to query the ring for a list of all files in it would be a nice feature.