**Data Structure Term Project Report**

Muhammad Umer Ahsan 19i-2184

Usman Shafique 19i-0495

Haider ail 19i-0401

**GIVEN PROBLEM**

In this project, we have implemented a special type of DHT that has a circular identifier space, named Ring DHTs. Usually we traverse over the linked list to search any data but this simple algorithm has a complexity of O (n) so in order to reduce this complexity we have implemented a special type of search algorithm which reduces the complexity to O (log n).

**Hashing:**

We uses Sha-1 to hash keys. For machine ID, user can enter any key of any type T and it will be hashed into numerical machine ID in the range of the identifier space and same goes for the data. Data is in the form of key-value pair and the key is hashed to produce numerical data id in the range of the identifier space.

**Class associations:**

The main class I created is Ring dht in which I have an object of list class. List is a circular linked list of machines. Each node of list is a machine node. A machine node has an object of class Table and Avl. The table class is routing tables implemented as a doubly linked list. Avl tree is used to store data.

**Implementation:**

Ring dht is implemented as a circular linked list .Each node of a list act as an active machine p with some machine id in the range of identifier space .Each machine node has an object of avl tree and a routing table. Routing table is implemented as doubly linked list. Each node has a pointer which contain address of other actual machines according to the formula FTp[i] =succ (p + 2i-1).Each actual machine is responsible for its own space/associated key as p >=e. The data is stored in these avl trees. Each avl node has a line number, filename/file path and a data id .The user will enter the data in the form of key value pair. The key will be hashed to produce data id and a value will be stored on any file and the name of that file and line on which the value is stored is saved in the avl node along with the data id.

**Avl Data Redistribution:**

In order to do data redistribution at the time of machine removal or insertion, we uses stack as a container to store data. When a new machine joins or leave, all the data first stored in the stack and then after insertion or removal of machine, all the data stored back to their associated machines.

**Data insertion and removal:**

The request to store or remove data can arrive on any machine. On receiving the key, value pair, a machine will First find the data Id, i.e., H (key) = e. We uses the same search algorithm to insert or remove data from the respective machine which has the complexity of O (log n).

**Machine joining or leaving:**

A machine can join or leave the ring dht without disrupting the functionality of the ring dht. All the routing tables and avl trees are updated when the machine joins or leave. The id range of neighboring machines also get updated.