COP5615: Fall 2015

PROJECT - 3

Chord algorithm implementation

**Project Members**

Praneeth Singh Rajput UFID: 6159 1691 prajput@ufl.edu

Sirivella Ananda Kishore UFID: 9951 5080 asirivella@ufl.edu

**READ ME**

The directory hierarchy is as follows:

**Project2**

**|+ chord**

**|+ src**

**|+ main**

**|+ scala**

**|+ ChordImpl.scala**

**|+resources**

**|+ localapplication.conf**

**|+ build.sbt**

**How to execute:**

Input: The input provided (as command line to your project2.scala) will be of the form:

**sbt “run numNodes numRequests”**

numNodes: *Number of actor to spawn.*

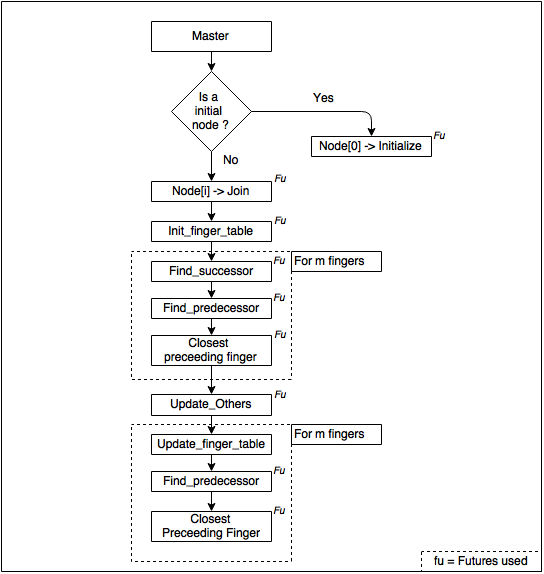
numRequest: *Number of request for the hop*

The project deals with implementation of chord protocol for a peer to peer network. We have defined two classes for the implementation.

Master: Class for the Master with only task being initialization of the Nodes

PeerNode: Class for the Node with the keys, helper functions and values for the Node.

We have implemented the synchronization of the mutual-message exchange between node using futures. We have utilized the onsuccess call back feature of the futures. With the need to specify a time out, we have fixated the timeout to be 100ms for each of the futures. Below is the flowchart of the model.



Sample Futures usage:

**val** Future = self ? **Func**(actorRef)

**val** result = Await.result(Future, timeout.duration).asInstanceOf[*String*]

Future onSuccess {

**case** "FuncCompleted" => {

*// Callbacks*

}

}