MobyChord: A chord implementation for Android Devices

ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ

ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS

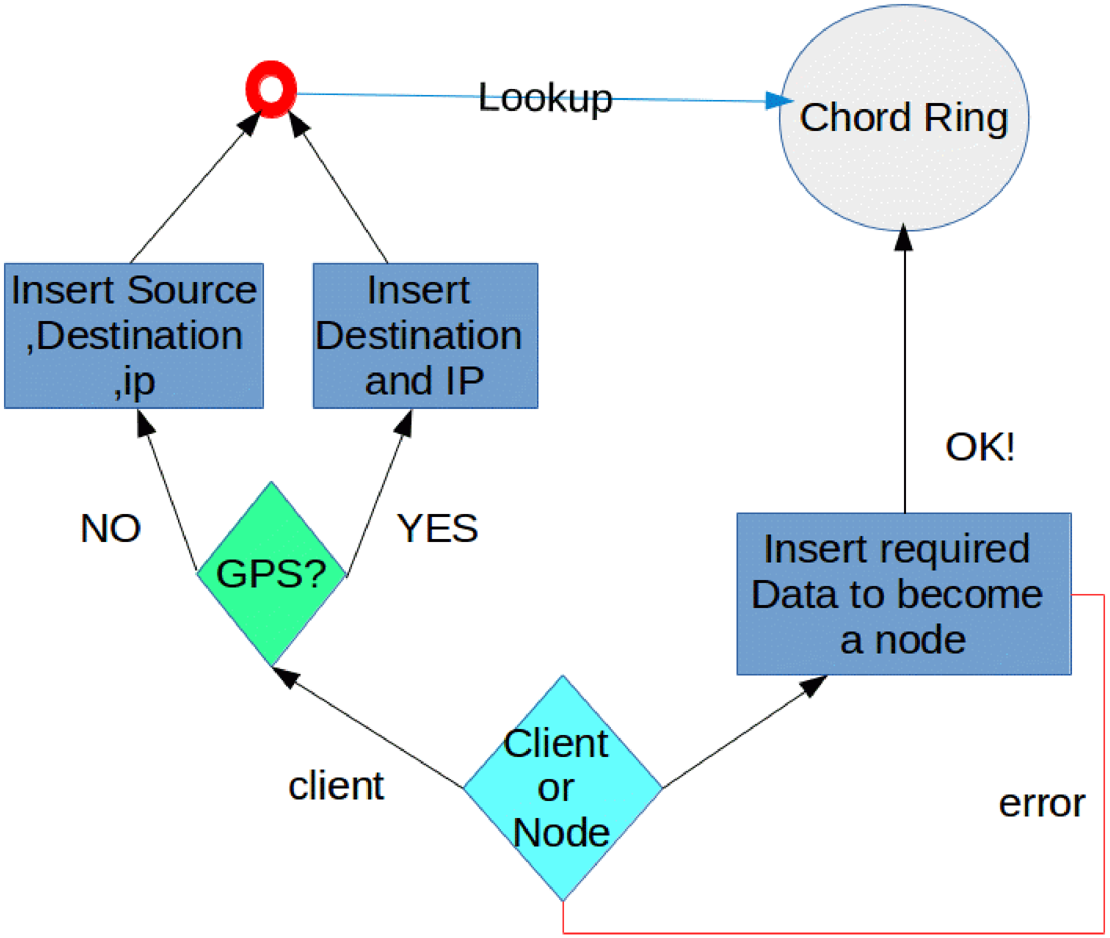
Kormaris Christos
Liosis Polyvios

Main Idea

- More adventurous and challenging than a simple Chord Implementation, where the nodes are actually some kind of Servers/Desktops

- The basic characteristic of ModyChord is that the nodes of the Chord Ring are actually Smartphones

Basic Scenario


Lookup
Chord Ring
Insert Source ,Destination ,ip
Insert Destination and IP
OK!
NO
GPS?
YES
Insert required Data to become a node
Client or Node
client
error

Basic Scenario: A Node’s Life

- An I responsible for the specific file?

  - Continue searching inside chord ring

  - Check if exists in Cache. If not the search in local storage of device and load it in cache.

  Finally if we do not have the requested route, then we download it and we send it to the starting node.

- Listen to possible changes in Chord Ring

Core Components/Functionality

- Memcached: Contains vital info for the Node (successor id & ip, fingertable info, files in cache etc.)
- OfferServiceToConnectedUser: Listens to all kind of requests from clients and other nodes in Chord Ring. The passed info is actually a string with specific id and important data: “3#ID#IP”
- The information in case we have a change in the Ring (e.g Insertion of Node etc.) is transferred clockwise. More specifically each node informs his successor.
- By default each node in his cache can have up to 3 files. We use a simple LFU technique.
- In order for a new device to become a node in a existing ring it must know 2 nodes: The successor and the predecessor!!!

Demo!!!