

INF3200: Mandatory assignment 1

September 5, 2017

1 Introduction

In this assignment we want you to build a peer-to-peer network with dynamic membership based on the Chord[1] protocol.

2 Requirements

We will provide you with code that interacts with your system. To pass this assignment you need to design and implement a system with the following requirements:

- Support a network with approximately 50 nodes in a p2p network structure modelled after the Chord protocol. No centralized architectures allowed; i.e. all processes should behave similarly. You are only required to implement a subset of the chord algorithm, specifically *create*, *join* and *find_successor*.
- Support graceful shutdown of nodes. When a node receives a POST request for the URL `/shutdown`, the node should leave the network. To ensure that the network is in a stable state after the node leaves, you need to update the successor of the predecessor node.
- A POST request to any node for the URL `/addNode` should bootstrap a new node and return the ip:port¹ pair of the new node. The response body must be formatted as a single line with the ip:port (e.g. `"127.0.0.1:1234"`).
- A GET request to any node for the URL `/neighbours` should return a list of ip and port pairs of all nodes connected to the recipient node. For the chord implementation, this should be the successor(s) of the node. The response body must be formatted as a list of ip:port (e.g. `"127.0.0.1:1234"`) entries with newline separating each ip:port pair.
- Provide experimental evaluations of your systems. We would like you to measure the churn your system can handle (how many nodes can join and leave the network in a given timeframe).

¹Using the hostname of the node instead of the ip is fine

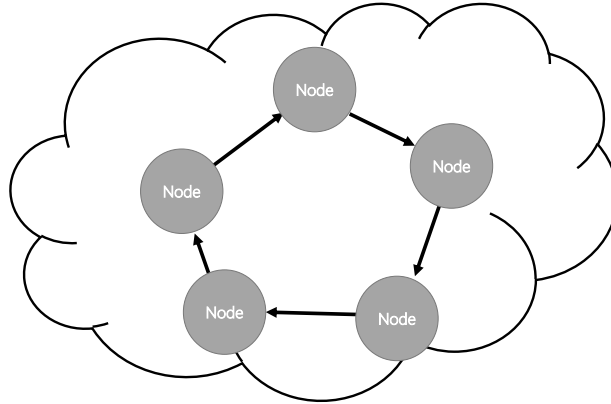


Figure 1: Architecture

3 Hand-in

In your report, briefly present the details of your approach to the assignment. Then discuss the details that are interesting to you. We want to see that you can assess the strengths and weaknesses of your implementation. Think outside of the context of the rocks cluster. The delivery should include:

1. Source code with instructions on how to run. You can use any programming language you would like as long as you support the API calls.
2. Report.

4 Other things

- There will be a demo presentation during the colloquium following the deadline. Present your work and demonstrate it. We will ask you to start a single process, and use a demo-client to grow your network.
- You share the cluster with multiple students, so please try to keep resource consumption to a minimum. You should add a time-to-live for each process so that it terminates after a given amount of time.
- Start early, fail early. (Make it better early.)
- Deadline Monday, October 2.

References

- [1] Ion Stoica et al. “Chord: a scalable peer-to-peer lookup protocol for internet applications”. In: *IEEE/ACM Transactions on Networking (TON)* 11.1 (2003), pp. 17–32.