

Mandatory assignment two, INF3200 A15

Peer-to-peer leader election

October 19, 2015

1 Introduction

In this assignment we want you to implement leader election on top of a peer-to-peer network. All the peers in the network must agree on who the leader is and only one peer can be leader at a time. To make this interesting we want you to support peers joining & leaving the network.

2 Leader Election

It is difficult to make everyone agree on something in a distributed system, yet sometimes necessary. It's for eg often necessary to have one node in your system act as leader or coordinator. Algorithms of varying complexity with varying consistency guarantees exist, a very simple algorithm for leader election is the bully algorithm, while a fairly complex one is paxos.

3 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 at least 10 nodes in a p2p network structure of your own choice. No centralized architectures allowed; i.e. all processes should behave similarly.
- Support graceful shutdown of nodes. On receiving a signal to shut down (SIGTERM), a node should leave the network.
- Support adding nodes on demand. Adding a new process allows you to grow the system as the demand increases.
- Leader election. There should at all times be a single leader. A pertinent Q: What happens if the leader leaves the network?

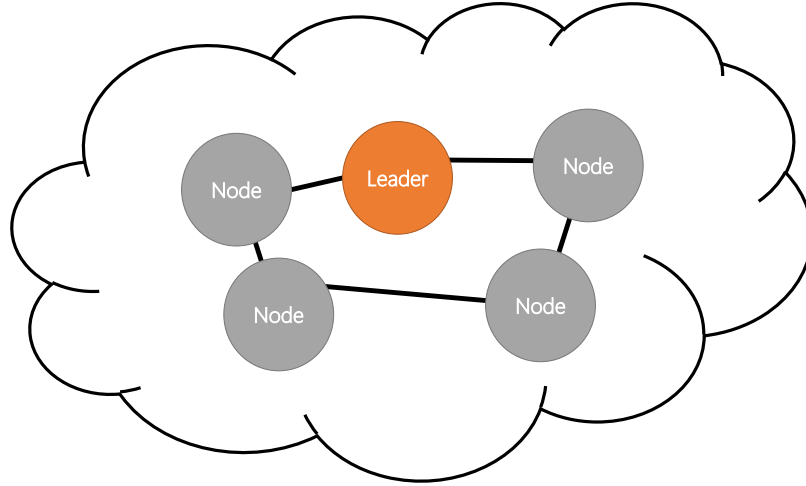


Figure 1: Architecture

- A GET request to any node for the url `“/getCurrentLeader”` should return the ip and port of the current leader. The response body must be formatted as a single ip:port (e.g. `“127.0.0.1:1234”`) entry.
- A GET request to any node for the url `“/getNodes”` should return a list of ip and port pairs of all nodes connected to the recipient 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.
- Conscientious students have good experimental evaluations of their systems. We would like you to measure the time it takes to elect a leader when the number of nodes changes.

4 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.
2. Report.

5 Other things

- There will be a demo presentation during the colloquium following the deadline. Present your work and demonstrate it.
- The hand in will still be on github.
- Start early, fail early. (Make it better early.)
- Deadline Monday, November 2nd.

6 References

1. <https://google.com>