University of St Andrews

Distributed Systems

CS4103

Ring-Based Distributed System

Author:
150008022

March 27, 2019

Goal

To demonstrate an understanding of leader election and mutual exclusion in distributed systems by developing a ring-based distributed social media application.

Contents

1 Initial Set-up 1

1.1 Configuration 1

1.2 Communication 1

2 Leader/Coordinator Election 2

3 Receive/Send Posts 2

1 Initial Set-up

Java 8 was chosen for this project due to it's friendly socket API, and Maven [1] was used as the build tool. TDD was implemented with JUnit4 as the the test suite framework.

The project was started with the intention of implementing the bully algorithm, as it provided fault tolerance with low overhead.

1.1 Configuration

Since each node would be running on an isolated machine, the planned testing environment would involve using *ssh* to start the nodes remotely. Providing the configuration as command line arguments was considered simpler than other methods, and was implemented using the Apache Commons CLI library [2].

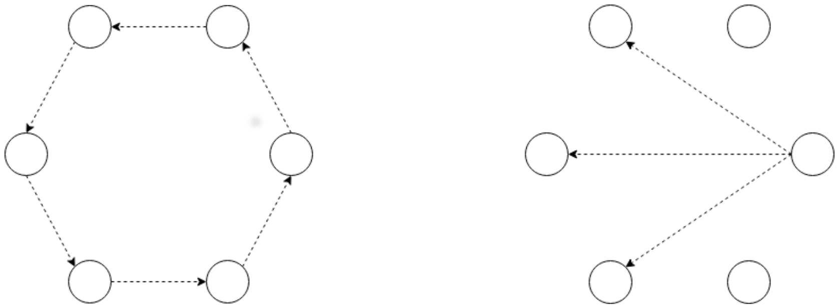
1.2 Communication

TCP was used for communication **around** the ring for the following reasons:

1. Reliable communication.
2. Keepalive functionality allows effective failure detection.
3. Connection is reused frequently between predecessors and successors, justifying handshake overhead.

UDP was used for communication **across** the ring for the following reasons:

1. Reliable communication.
2. Keepalive functionality allows effective failure detection.
3. Connection is reused frequently between predecessors and successors, justifying handshake overhead.



(a) TCP Around Ring

(b) UDP Across Ring

Figure 1: The different types of communication used

2 Leader/Coordinator Election

3 Receive/Send Posts

References

[1] Apache. Apache maven project. `https://maven.apache.org/`.

[2] Apache Commons. Apache commons cli. `https://commons.apache.org/proper/commons-cli/`.