Uppsala University

Computer Networks and Distributed Systems, 10c

Project, 1c

Authors:
Ashraf, Pouya
Billman, Linnar
Boström, Carl
Emrén, Krister

December 12, 2015

Contents

1 Summary

This project had the aim to explore how one may approach the task of creating a system that runs on a distributed system, and how to solve the problems that occur with that task.

2 Introduction

For this project we chose to implement a distributed buffer of sorts, which in turn implements a distributed server polling system, to ensure that the system will not fail when the designated server crashes/goes down (when the current server crashes, or loses connection with the connected nodes, a different node will take the server responsibilities.).

3 Original Design Idea

Initially, the project was envisioned as a distributed command line buffer, which also included a distributed server mechanism. The purpose of the distributed server mechanism was to ensure that the system would be able to recover by itself, in case the node currently acting as the server was to crash/get disconnected from the network. This was to be done by the server, which would with a fixed delay send the most recent information regarding the structure and hierarchy of the network to all connected nodes.

4 Revised Design

5 The System

The system consists of <X> main parts, outlined below.

5.1 Node.java

5.2 LockList.java

LockList is a class designed as a linked list, with the sole purpose of keeping track of which nodes have requested write access to which parts of the text.

For instance, if node no.3 requests write access to a file att offset 450 bytes, and length of segment 100 bytes, other nodes will be unable to write information in that segment of the file (between offset 450 and 550 bytes) until node no.3 forfeits its privileges.

5.3 EditPane.java & EditWindow.java

EditPane and EditWindow are classes designed to display information to the user. This information consists of the file currently being edited, status/error messages and a means to cancel/submit the changes that have been made to the server. The graphical interface is entirely cross platform, since it builds on the Java Swing framework.

6 Results and Discussion

6.1 What we did right

6.2 What could have been implemented better

6.3 Possible Further Development

7 Conclusions