Large-Scale Distributed Systems: Exercise Session 2

Joeri Hermans
joeri.hermans@doct.ulg.ac.be

October 23, 2017

1 Introduction

Broadcast communication is used to disseminate information among a set of processes and differ according to the reliability of the dissemination. In a traditional *server-client* architecture, interactions are often established between two processes. In this architecture, a server exposes some kind of interface to the client, which in turn exploits this interface to exchange some state with all other participating clients (e.g., as in a multiplayer game). These interactions are supported by a *direct link* between the server and all its clients. Meaning, there is a dedicated *point-to-point* link for every client with the centralized server.

Of course, this approach has several advantages and disadvantages. The main, albeit practical advantage of this paradigm is that only a single machine (or a set of machines that appear as a single service) is responsible for maintaining some state. This enables system engineers to fine-tune their service without relying on other clients which may have non-optimal connections, or have even malicious intents. This is actually what we observe nowadays. We are moving from a once idealized notion of *peer-to-peer* content delivery systems to *cloud* based architectures (NetFlix, YouTube, . . .). Of course, a very high cost is associated with building such a system. However, it provides consistent performance because it does not have to rely on the *quality* of participating nodes in the network. Nevertheless, it brings the traditional issues of introducing a single point of failure for the system. So a paradigm shift of *expecting failure* in P2P computing towards *preventing failure* can be reconsidered. Of course, these monolitic services still consist of several thousands of individual, but cooperating machine. So in a sense you still have to *expect failure* within this architecture.

On the other hand, in *multiparticipant systems* broadcasting is slightly more complex since all participating nodes have to agree on the broadcasted message they received. Then there is the issue how to distribute a message to all nodes in the network efficiently.

2 Exercises

Exercise 1

Exercise one goes here.