Large-Scale Distributed Systems: Exercise Session 1

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

October 4, 2017

1 Introduction

The core of any distributed system is a set of distributed algorithms that ensures the behaviour and the corectness of the global state. In contrast to single monolitic systems, distributed systems have usually the advantage that other processes are conducting their computations *independently* (asynchronously) from other *entities*, i.e., a different machine or node. However, in order to ensure the correctness of the global goal, information exchange is required. The difficulty in this setting arrises from the fact that a physical communication medium is intrinsically lossy due to the fact that other (physical) processes influence the communication signal. Nevertheless, if we construct our infrastructure in such a way that we can minimize the loss, we can assume that with a certain probability some form of *useful* communication might occur between two or more entities, i.e., sometimes, the medium allows for information to pass without being destroyed. In the lecture we saw that using this principle, we can construct relatively complex abstractions, and more importantly, build on top of previous abstractions (layering) to ensure that information always arrives once it is send.

At this point, we established a method to ensure *reliable* communication. As a result, we are able to coordinate the entities to ensure the correct behaviour of the distributed system. Sadly, reliable *point-to-point* communication is probably not sufficient in a distributed system since more high level features are probably required. For example, let us assume processes p, and q, where p needs to send message m to q. Of course, p could send the message directly to q using our reliable point-to-point abstraction. However, this requires a *direct* connection from p to q. It is trivial to see that this the point-to-point protocol assumes some abstraction, i.e., a direct connection between p and q is available. As we will see in later courses, distributed systems, some of which you may or may not use (e.g., BitTorrent), usually place additional layers of abstraction on top of existing infrastructure (albeit with the same functionality) to achieve a specific goal, while at the same time, remove the abstraction when something efficient needs to be done.

In this exercise session, we will study and apply some of these abstractions in order to captivate their importance, and to what end they can be utilized.

2 Exercises

Add fancy exercises here.