

Large-Scale Distributed Systems: Exercise Session 3

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

October 30, 2017

1 Introduction

Consensus is the problem of making processes agree on one of the values they propose. Solving the problem of consensus is the key to solving many problems in distributed computing. In fact, any algorithm that helps multiple processes maintain a common state, or to decide on a future action in a model where processes may fail involves solving a consensus problem. This lecture shows several approaches to build robust consensus mechanisms. These mechanisms can in turn be used as abstractions to provide *robust* complex functionality in higher-level algorithms.

2 Exercises

Exercise 1

Why is the consensus problem impossible to solve in asynchronous systems?

Exercise 2

Describe Hierarchical Consensus. How does the hierarchical approach ensure the correctness of the consensus properties?

Exercise 3

How does a Leader-Driven approach achieve consensus?

Exercise 4

How does Read/Write Epoch Consensus prevent a violation of the lock-in property? Why is it important that lock-in is not violated in order to ensure consensus?

Exercise 5

We have n nodes conducting an asynchronous computation. Meaning, given an input I , every node applies some computation f to I , i.e., $f(I)$, and forwards it to the next node in the chain. However, in order to speed up the computation (due to unmodeled delays), we could pass on the result of $f(I)$ to some other node in the network if and only if the next node in the chain is not ready. However, in order to provide some sense of fairness, finished nodes should be addressed in a FIFO manner. Basically, the desired product is a shared variable, which holds a queue of nodes which are in a ready state. How would you implement this? Assume no node failures.

Exercise 6

Let us take the same setting in as Exercise 5. However, contrary to the other setting, nodes can fail. What needs to change in order to ensure consensus?