Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Autumn Term 2018 Prof. T. Roscoe, Prof. R. Wattenhofer

Computer Systems

Assignment 12

Assigned on: **December 14, 2018**

1 piChain and PBFT

Quiz

1.1 PBFT: basics

a) At which point of the agreement protocol can a node be sure that all correct nodes can only agree on the same request for a given sequence number within the current view?

b) During a view change, how can the backups be sure the new primary did not just make up requests that he wants them to execute?

c) During a view change, will only requests that were already executed by some correct node be included in the set $\mathcal{O}$?

d) It is possible that a node collected a prepared-certificate that will not be included in a new-view-certificate. Why is this not a problem?

Basic

1.2 PBFT: we need the phases of the agreement protocol

In the PBFT agreement protocol, some phases seem superfluous. The purpose of this exercise is to get an insight to why those phases are necessary.

a) How could a byzantine client slow down the system if nodes did not forward requests to the primary?

b) Assume correct nodes do not wait for $2f+1$ `commit`-messages in phase 3 of the agreement protocol (Algorithm 25.17) and instead execute a request as soon as they have a prepared-certificate for it. How could it happen that two different correct nodes execute two different requests with the same sequence number with this change in the algorithm?

Advanced

1.3 PBFT: multiple prepared-certificates for the same sequence number!? (How does it happen)

One final corner case we did not consider in the script is implied by the answers to Quiz questions c) and d). In this exercise, we consider the possibility of multiple prepared-certificates for the same sequence number but different requests ending up in the $\mathcal{V}$-component of a `new-view`-message.

How can it happen that the $\mathcal{V}$-component of a `new-view`-message contains two prepared-certificates, one for (v, s, r) and one for (v', s, r') with $r \neq r'$?

(The Mastery Exercise 1.5, for which we will provide a solution as well, asks you to show how this can be dealt with.)

1.4 Authenticated Agreement

Algorithm 25.2 in the lecture uses authentication to reach agreement in an environment with byzantine processes.

a) Modify this algorithm in such a way that it handles arbitrary input. Write your algorithm as pseudo-code. The processes may also agree on a special "sender faulty"-value. Hint: consider a set of values that correct nodes reached agreement for, then work with the size of the set.

b) Prove the correctness of your algorithm.

Mastery

1.5 PBFT: multiple prepared-certificates for the same sequence number!? (How can we fix it)

As we saw in Exercise 1.3, it is possible that multiple contradictory prepared-certificates for the same sequence number end up in the $\mathcal{V}$-component of a `new-view`-message.

How does the primary have to choose between those prepared-certificates when generating $\mathcal{O}$ such that the system remains correct? Prove that your proposal guarantees correctness!

2 Advanced Blockchain

Quiz

2.1 Randomness from previous Block

In a simple Proof of Stake model, let's say there are a previously known set of validators who have put up some currency as stake. For every block, the protocol needs to select one of these validators randomly. The odds of a particular validator being selected should be proportional to the amount of stake they have committed to the protocol. Let's say the protocol hashes the encoded contents (transactions, metadata, timestamp, signatures, etc.) of the previous block to get a value h. Let's say that the hash function has a range $[0, H]$. We can divide this range into proportional sub-ranges based on the stake of each validator. Based on where h lands in this range, pick the validator whose sub-range overlaps with h, and you have the next validator.

How does this scheme devolve into proof-of-work?