BGWS: A Dependable Decentralized Ledger for a Blockchained Global Wallet Service

Diogo Cebola, 52718 and Gonçalo Areia, 52714

Abstract—

Index Terms—Dependable Distributed Systems, Blockchain, Byzantine Fault Tolerance

1 Introduction

This work was proposed by professor Henrique Domingos as the course project of “Dependable Distributed Systems“ 2020-21.

From cloud and world-wide e-commerce platforms to banking replacements, distributed systems are widely used nowadays for various tasks. Such tasks can range from simple consultations (e.g. querying products available in an online store) to critical operations (e.g. money transfers). Frequently these operations need to be ordered and the processes involved are required to reach an agreement. To solve this problem distributed systems often rely on state machine replication and a consensus algorithm.

With the increasing number of distributed systems being used for critical tasks, it is imperative that they are dependable and ensure high degree of reliability, availability, safety, confidentiality, fault tolerance, integrity and maintainability. When designing dependable systems it is necessary to consider the execution environment and the correct adversary model.

Inspired by the rise of blockchain technologies and the way such technologies deal with consensus, in the realm of financial operations, we will present the implementation of a dependable decentralized ledger for a blockchained global wallet service. From this point onwards we will refer to the system using BGWS. BGWS will operate in an asynchronous environment, with no timing assumptions, and consider that all faults are byzantine.

We will study how the number of replicas impact the overall performance of the system, how the size of blocks affect the latency of successful mining operations and how the system tolerates faults.

The remaining of the document is organized as follows. In Section 2, we introduce important concepts related to our work, more specifically Byzantine Fault Tolerance and the Blockchain technology. In Section 3, we present BGWS system model and architecture. Internal mechanisms, design assumptions and the service planes are covered in section 4. In Section 5, we explain the technology stack used and development issues. Section 6 will present the experimental setting and discussion of the obtained results. In Section 7 we evaluate the current state of the BGWS system, its correctness, limitations and trade-offs. Finally, in Section 7 we conclude the document and mention future work.

2 Background

2.1 State Machine Replication

We can define a state machine as a set of states, a set of transitions, and a current state. When a user issues an operation to the machine, that operation triggers a transition from its current state to a new one, producing an output.

In State Machine Replication, multiple replicas of a system are created, each one being a deterministic state machine. If they are given an input of the same operations in the same order, all the replicas will transition to the same state and produce the same output. The ordering of the operations will have to be decided through an agreement protocol.

2.2 Consensus

There are various forms of consensus, from binary consensus, multi-valued consensus, to probabilistic consensus that relies on randomization algorithms. Let us consider the following specification of consensus:

C1 (Termination): Every correct process eventually decides a value.
C2 (Validity): If a process decides v, then v was proposed by some process.
C3 (Integrity): No process decides twice.
C4 (Agreement): No two correct processes decide differently.

In a byzantine fault environment

2.3 Blockchain Technology

3 System Model and Architecture

4 Mechanisms and Service Planes

5 Implementation

5.1 Technology Stack

5.2 Issues

6 Experimental Evaluation

6.1 Experimental Setup & Methodology

6.1.1 Validation

6.1.2 Evaluation

6.2 Experimental Results

In this section we present the results as an average of the three runs executed, and describe the patterns observed.

6.3 Discussion

In this section we will discuss the variation in the results obtained for the different experiments.

7 Coverage of Requirements

8 Conclusion & Future Work

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Manubot, Tech. Rep., 2019.

[2] V. Buterin and V. Griffith, "Casper the friendly finality gadget," *arXiv preprint arXiv:1710.09437*, 2017.

[3] O. Moindrot and C. Bournhonesque, "Proof of stake made simple with casper," *ICME, Stanford University*, 2017.

[4] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the thirteenth EuroSys conference*, 2018, pp. 1–15.

[5] J. Sousa, A. Bessani, and M. Vukolic, "A byzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform," in *2018 48th annual IEEE/IFIP international conference on dependable systems and networks (DSN)*. IEEE, 2018, pp. 51–58.

[6] M. Castro, B. Liskov *et al.*, "Practical byzantine fault tolerance," in *OSDI*, vol. 99, no. 1999, 1999, pp. 173–186.

[7] A. Bessani, J. Sousa, and E. E. Alchieri, "State machine replication for the masses with bft-smart," in *2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*. IEEE, 2014, pp. 355–362. [Online]. Available: https://github.com/bft-smart/library

[8] M. J. Fischer, N. A. Lynch, and M. S. Paterson, "Impossibility of distributed consensus with one faulty process," *Journal of the ACM (JACM)*, vol. 32, no. 2, pp. 374–382, 1985.