In distributed computing, byzantine fault tolerant (BFT) systems have been subject to extensive studies. A BFT system can tolerate at most $f < \frac{n}{3}$ arbitrarily malicious (byzantine) nodes among a set of $n$ participants. Nowadays, BFT protocols are the key to building "permissioned blockchains", an area traditionally known as "state machine replication" [9, 10].

From a research perspective, BFT protocols have regained attention with the emergence of "Practical" BFT (PBFT) systems [1–8, 11–13]. These "practical" systems were motivated by the needs of many applications: from online shopping to credit card transactions, cryptocurrencies or stock market trades; whenever a set of clients makes concurrent requests for (or with) limited resources, the service providers have an interest to both prevent fraudulent and tolerate faulty behaviour in a system. However, taking one step back allows us to observe that all these applications have one common, reoccurring theme: they are network applications.

A natural question arising could be whether the available network infrastructure is well-suited to support the demands of such an application efficiently. Hence, in this thesis, we will take a closer look at the symbiosis of modern networking capabilities with established BFT protocols known from the distributed computing domain. Specifically, we will adapt an existing BFT protocol implementation for a case-study to get a hands-on feeling what it takes to speed up BFT protocols.

Requirements

- good experience in C++ programming
- experience with P4 programming is advantageous (but not strictly required)
- general knowledge in cryptography
- thinking out of the box, independent problem solving

Contact

- Roland Schmid, roschmi@ethz.ch
- Prof. Laurent Vanbever, lvanbever@ethz.ch

References


