POSTER: Scalable Communication Middleware for Permissioned Distributed Ledgers

Artem Barger[1], Yacov Manevich[1], Benjamin Mandler[1], Vita Bortnikov[1], Gennady Laventman[1], and Gregory Chockler[2]

[1]IBM Haifa Research Lab, {bartem, yacovm, mandler, vita, gennady}@il.ibm.com
[2]Royal Holloway, University of London, Egham, UK, gregory.chockler@rhul.ac.uk

1 Motivation and Background

Distributed Ledger Technology (DLT) is rapidly emerging as a new paradigm for automating complex business processes in secure and decentralised fashion. Currently, however, its wider adoption is hampered by scalability problems [3] rooted in an inherent tension between stringent consistency, security, and robustness requirements on one hand, and growing application demand coupled with high performance expectations on the other. For example, popular peer-to-peer DLTs based on proof-of-work consensus [4] can only improve the transaction throughput by degrading their security and consistency guarantees, which is unacceptable in the enterprise and mission-critical settings.

To improve scalability, recently proposed permissioned DLT systems utilise stronger trust models to replace slow proof-of-work consensus with more efficient Byzantine fault-tolerant (BFT) replication protocols [2]. Specifically, the reference architecture [1] introduced by the HyperLedger Project (HLP), delegates the task of running BFT to a small trusted core of dedicated entities, called *orderers*, and offloads the management of the application state to a separate set of entities, called *peers*. The orderers assemble client requests into transaction blocks, and put them through the BFT consensus to establish a global total order; the peers process the incoming ordered transaction blocks, and apply their constituent transactions to the application state. All entities are connected via a *reliable broadcast layer*, which is utilised by the orderers to propagate ordered transaction blocks to the peers.

2 Our contribution

In this work, we consider the problem of scaling up the HLP architecture in terms of both transaction throughput and the application hosting capacity. Since BFT consensus is bandwidth bound, its throughput is capped by the network capacity within the trusted core and cannot be improved significantly by increasing its size. On the other hand, the application capacity is mainly restricted by the available CPU and storage resources, and therefore, can be scaled by growing the number of participating peers. This however, puts the spotlight on the broadcast layer whose scalability now becomes critical for the overall system performance. We therefore ask the following question: *Is it possible to design a broadcast layer capable of matching the BFT throughput while being secure and robust under attacks?*

To answer this question, we embarked on a study to explore the design space of reliable broadcast protocols as a function of the performance, costs, and robustness. Our initial results indicate that dynamic peer-to-peer solutions are more scalable than those based on fixed topologies due to better load distribution, robustness and churn resistance. In particular, a star-like propagation topology (with the trusted core being at the centre) originally used by HLP resulted in a chunk of the BFT bandwidth being wasted on block propagation activities thus degrading its throughput.

As a proof-of-concept, we implemented BlockStorm, a simple broadcast protocol based on epidemic diffusion, and integrated it into HLP. BlockStorm leverages randomized choice and HLP's public-key authentication capabilities to limit influence of adversarial nodes and tolerate message corruption. Our preliminary simulation studies indicate that BlockStorm is capable of supporting adequate levels of performance while tolerating a fixed fraction of faulty peers.

Ongoing and Future Work: We are currently working on extending the performance analysis of BlockStorm to better understand the impact of various epidemics parameters (such as fan-out, propagation depth, and push/pull balance) on performance and fault-tolerance under a variety of adversarial models. In the future, we plan to study more efficient strategies for block verification to reduce the block processing overheads as well as Byzantine-resilient overlay network topologies as a gossip replacement.

References

[1] E. Androulaki et al. Hyperledger fabric proposals: Next consensus architecture proposal. `http://github.com/hyperledger/fabric/blob/master/proposals/r1/Next-Consensus-Architecture-Proposal.md`, 2016.

[2] M. Castro and B. Liskov. Practical byzantine fault tolerance and proactive recovery. *ACM Transactions on Computer Systems (TOCS)*, 20(4):398–461, 2002.

[3] K. Croman et al. *On Scaling Decentralized Blockchains*, pages 106–125. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016.

[4] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2009.