A Review of 'The Honey Badger of BFT Protocols'

Prashanth R Duggirala

Summary

Paper by Andrew Miller, Yu Xia, Kyle Croman, Elaine Shi, Dawn Song, 2016

The bandwidth limitations of current cryptocurrencies and blockchains causes them to have a low throughput and can be addressed by the asynchronicity of protocols like Honey Badger. Usually BFT protocols are believed to have higher throughput in comparison. According to the paper, timing assumptions are harmful for the practicality of a fault tolerant system. Most existing Byzantine fault tolerant (BFT) systems, even those called "robust," assume some variation of weak synchrony, where, roughly speaking, messages are guaranteed to be delivered after a certain bound and may be time-varying or unknown to the protocol designer. The authors claim that the assumption of synchrony which is utilized in other scalable proposals is unrealistic in almost all practical purposes. They argue that protocols based on timing assumptions are unsuitable for decentralized, cryptocurrency settings, where network links can be unreliable, network speeds change rapidly, and network delays may even be adversarially induced. The current assumptions underlying current practical byzantine fault-tolerance (PBFT) based protocols are revealed to fail by constructing an adversarial intermittently synchronous network call for new approaches that deal specifically with the weak synchronicity problems. Furthermore a weak asynchronous protocols is also unsuited for cryptocurrencies applications:

The HoneyBadgerBFT makes significant efficiency improvements on the best prior known asynchronous atomic broadcast protocol for a large scale transaction processing systems called SINTRA, as mentioned in the paper. HoneyBadger adapts synchronous BFT for efficiency in the batch setting, this protocol is centered around the concept of Asynchronous Common Subset (ACS) primitive. Essentially, it proceeds in epochs, with a new batch of transactions appended to the (shared) committed log at the end of each epoch. In each epoch, nodes choose a subset of the transactions in their input buffer and provide them as input to a randomized agreement protocol. At the end of the agreement protocol, the final set of transactions for the epoch is chosen. This improves throughput by O(N). Another O(N) improvement is achieved by using random selection and threshold encryption which essentially avoids sending redundant transactions. Which brings the total improvement of throughput in the order of O(N^2).

This approach is especially well suited for a cryptocurrency ecosystem, where the network bandwidth is a scarce resource, but the computation is relatively abundant. HoneyBadgerBFT's design is optimized for a cryptocurrency like deployment scenario where network bandwidth is a scarce resource, but computation is relatively ample. This allows us to take advantage of cryptographic building blocks (in particular, threshold public-key encryption) that would be considered too expensive in a classical fault-tolerant database setting where the primary goal is to minimize response time even under contention. The paper states a few deployment scenarios of this protocol in distributed computing and financial technologies perspectives. The authors transparently provide two main ecosystems where Honey Badger might encounter its ideal playground: confederated cryptocurrencies in which a conglomerate of financial institutions jointly contribute to a Byzantine agreement protocol to allow fast and robust settlement of transactions and permissionless blockchains with enrolment open to everyone.

Strong points

1. This paper refutes the prevailing knowledge that normally assumes that asynchronous atomic broadcast protocols as impractical and inefficient, with a demonstration that is sound theoretically but also supported empirically.

2. Suggested Deployment Scenarios - The paper highlights two likely deployment scenarios that are sought after by banks, financial institutions, and advocates for fully decentralized cryptocurrencies namely confederation cryptocurrencies and permissionless blockchains.

3. They demonstrate the scalability of HoneyBadgerBFT by performing an experiment in a wide area network including up to 104 nodes in five continents. and show that it can reach peak throughputs of thousands of transactions per second. And, they also demonstrate the feasibility of running asynchronous BFT over the Tor anonymous communication layer and show impressive results.

Weak points and comments

1. The paper does not talk about the symmetric leaderless consensus protocols where all servers have equal roles and clients can contact any server. These protocols are said to provide good amount of performance upgrade in an asynchronous setting. The paper also does not talk much about previous work done on randomized solutions for asynchronous BFT. There are protocols like RITAS which were proposed earlier and work on the same lines as SINTRA.

2. The performance of this protocol relies on one assumption that we need to choose a large enough batch size i.e. that the input buffers of each honest node are sufficiently full. Then efficiency is the expected communication cost for each node amortized over all committed transactions. The paper suggests a batch size of the order $\Omega(\lambda N^2 \log N)$. Which may or may not be always possible.

3. The evaluation section does not show the results for latency and throughput when the byzantine nodes really behave maliciously. They also did not simulate any crashes and show their results for these cases as well.

References

1. CCS 2016 - The Honey Badger of BFT Protocols
2. POA Network: How Honey Badger BFT Consensus Works - POA Network
3. Asynchronous BFT protocols – the case for Honey Badger – The Intelligence of Information
4. The Honey Badger of BFT protocols