Weighted voting for optimising Streamlined Blockchain Consensus Algorithms

Diana Micloiu – supervised by Dr. Jérémie Decouchant and Rowdy Chotkan

1. Introduction

Consensus denotes the collective agreement of network participants, a mechanism needed to ensure proper functionality of distributed systems.

Byzantine Fault Tolerant (BFT) represents a family of protocols which enable systems to tolerate arbitrary node failures; in particular, protocols require 3f + 1 nodes to withstand f failures.

Streamlined algorithms – use leader rotation in each round to shift the communication burden from the leader.

Weighted voting – in the consensus mechanism, the voting power of a node depends on a weight metric.

3. Scientific Gap

The impact of **weighted voting** has been applied so far only on first generation consensus algorithms, in AWARE [1].

The research aims to address the benefits of **weighted voting on streamlined algorithms** such as Hotstuff [4].

The research also looks into the possibility of using a **generalised weighting scheme in AWARE** (rather than the binary one) for **optimising the recovery performance of the system.**

4. Methodology

Weighted voting on streamlined algorithms:

- Design an algorithm that would emulate Hotstuff behaviour, combined with the binary weighted voting mechanism presented in WHEAT.
- Develop a **latency prediction method** for a given distributed scenario.
- Use **Exhaustive Search or Simulated Anneal**ing for finding out the best weight distribution that would minimise latency given a network setting.

Generalised Weighting Scheme for AWARE:

- Keep $n - f$ replicas having $Vmin = 1$ voting power.
- For the $2f$ replicas holding $Vmax = 1 + \frac{\Delta}{f}$ voting power in AWARE, design a ranking mechanism to give higher power to the replicas achieving better end-to-end latency.
- Assess the system's recovery performance by introducing faulty nodes.
- Formalise the approach mathematically and ensure quorum system properties.

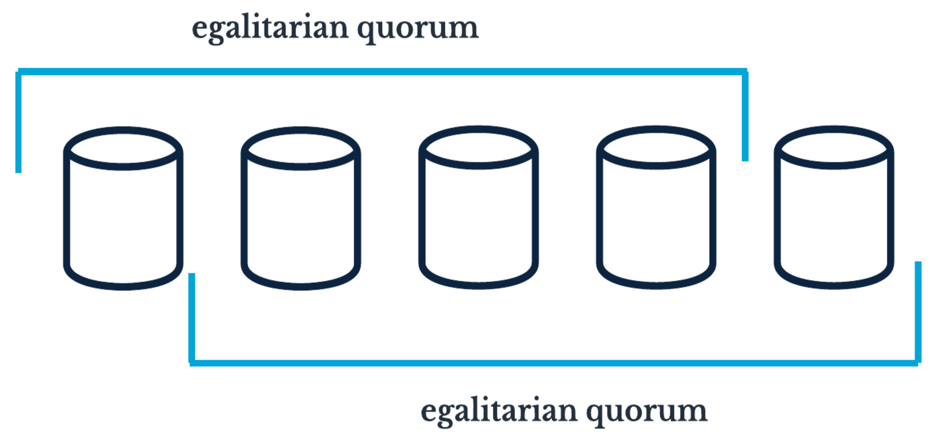
References

[1] C. Berger, H. P. Reiser, J. Sousa, and A. Bessani, "Aware: Adaptive wide-area replication for fast and resilient byzantine consensus," IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 3, pp. 1605–1620, 2020.
[2] 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.
[3] J. Sousa and A. Bessani, "Separating the wheat from the chaff: An empirical design for geo-replicated state machines," in 2015 IEEE 34th Symposium on Reliable Distributed Systems (SRDS). IEEE, 2015, pp. 146–155
[4] M. Yin, D. Malkhi, M. K. Reiter, G. G. Gueta, and I. Abraham, "Hotstuff: Bft consensus with linearity and responsiveness," in Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, 2019, pp. 347–356.

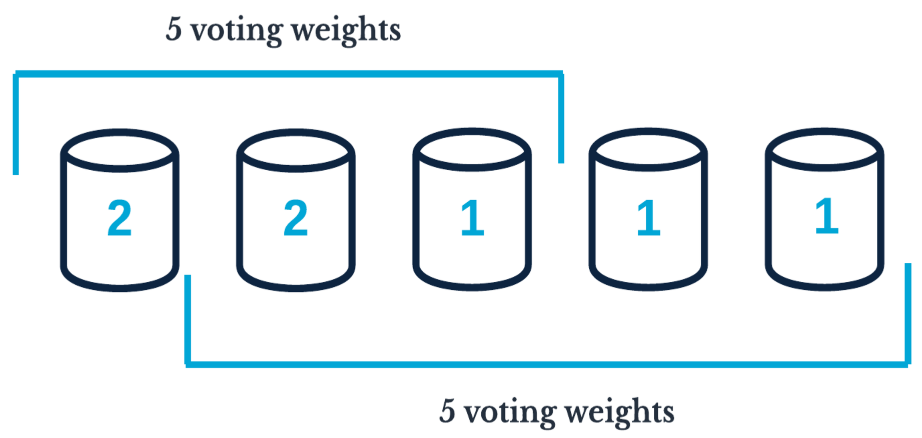
2. Background

AWARE (Adaptive Wide-Area REplication) [1]:

- Deterministic, self-monitoring and self-optimising algorithm for optimising the latency of the blockchain.
- Combines **BFT-SMaRt** [2] as replication protocol and **WHEAT** [3] for the underlying weighting distribution scheme ($Vmax = 1 + \frac{\Delta}{f}$ or $Vmin = 1$ voting power of each replica).
- Self-monitoring is achieved through deterministic latency prediction.
- Self-optimisation is employed by voting weights tuning and leader relocation



(a) Egalitarian: all quorums contain $\lceil \frac{n+f+1}{2} \rceil$ replicas.



(b) Weighted: quorum contains $2f + 1$ replicas in the best-case scenario and $n - f$ in the worst.

Figure 1: Possible quorums for $n = 5$, $f = 1$, $\Delta = 1$ additional weights

PBFT (Practical Byzantine Fault Tolerance) [4]:

- Designed in the late 90s by Liskov and Castro to work efficiently in asynchronous systems.
- One leader which gets re-elected in later rounds *if idle.*
- $O(n^2)$ communication complexity

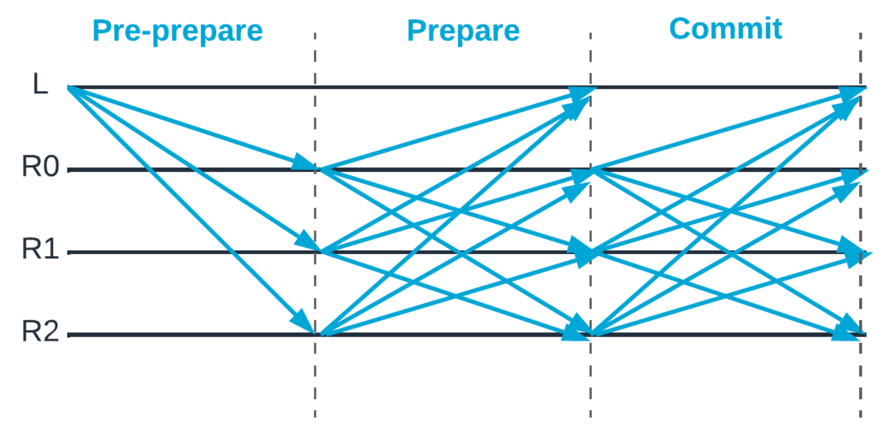


Figure 2: PBFT communication phases.

Hotstuff [5]:

- **Streamlined algorithm** comprising 5 communication phases.
- New leader is randomly assigned in each round.
- $O(n)$ communication complexity
- Basic Hotstuff - nodes vote on a single block per round.
- Chained Hotstuff – enable a pipelined voting mechanism to simultaneously progress on several blocks per round.

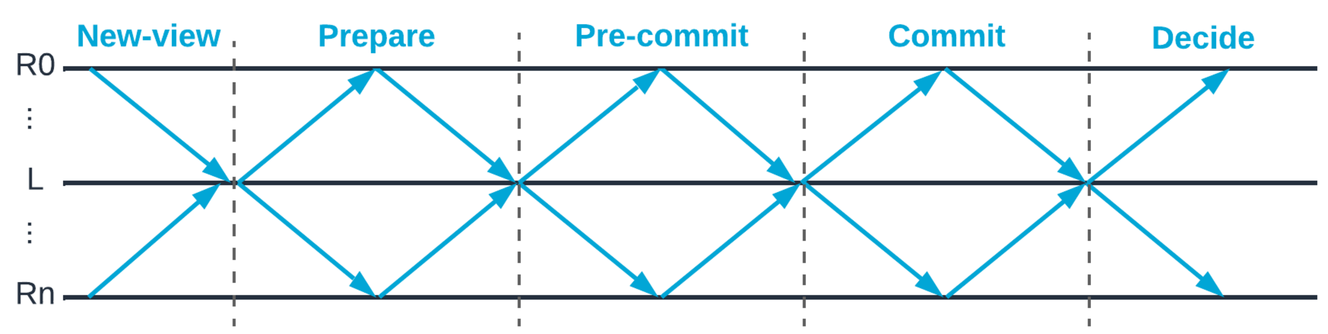


Figure 3: Hotstuff communication phases.

5. Results

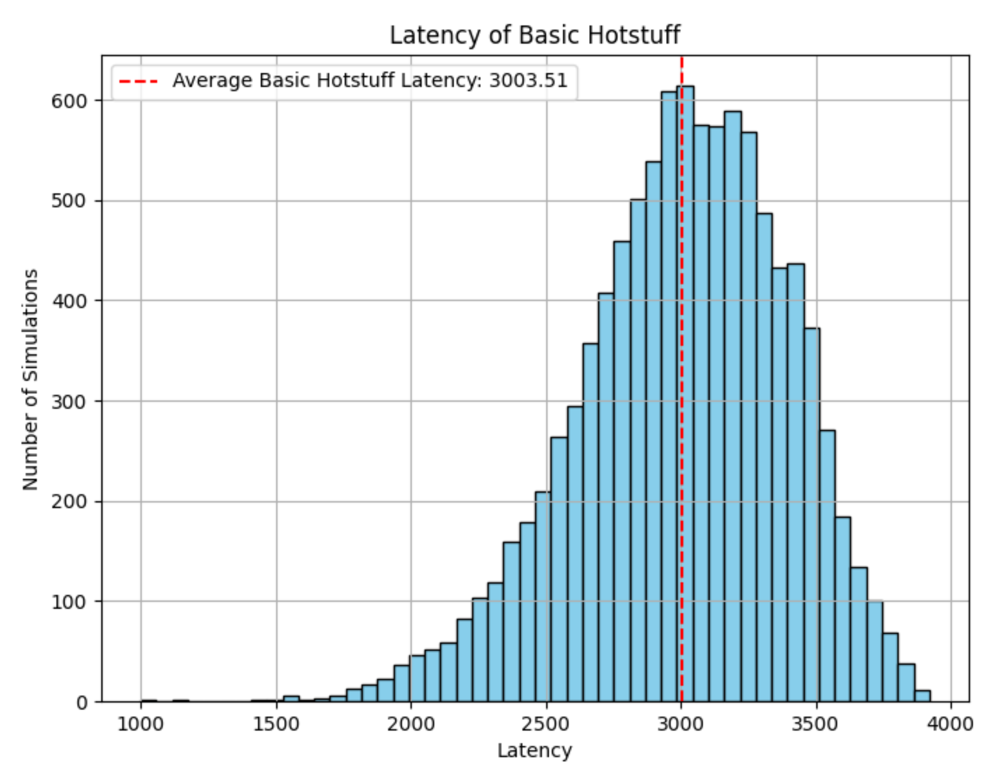


Figure 4: Basic Hotstuff latency analysis on $N = 10000$ simulations in a system with $n = 4$, $f = 1$

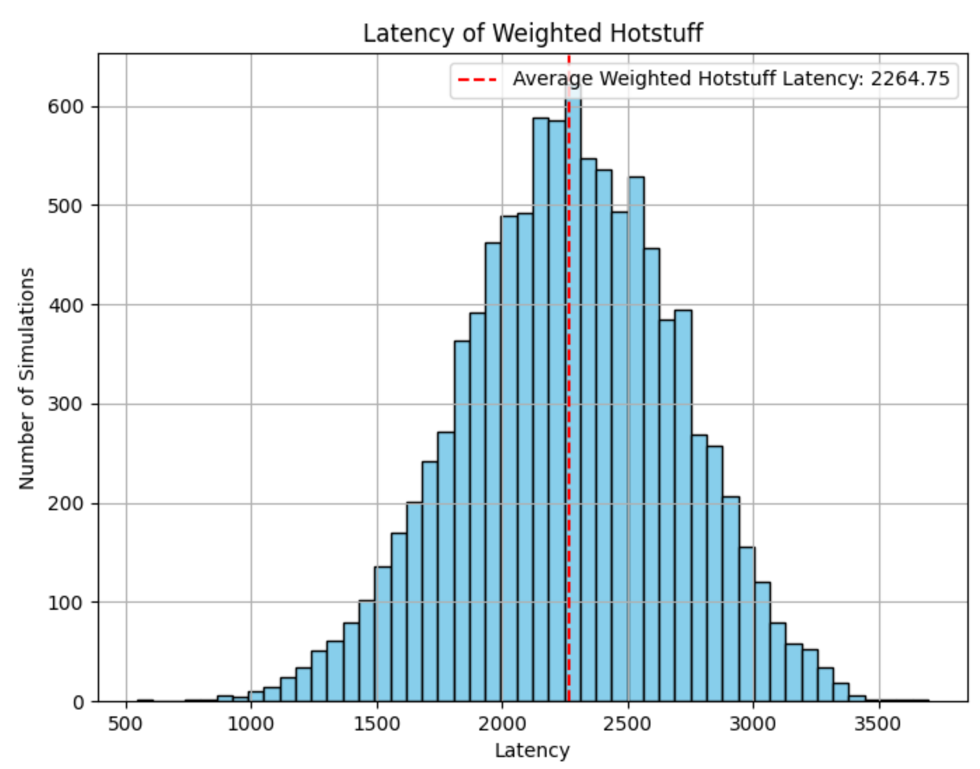


Figure 5: Weighted Hotstuff latency analysis on $N = 10000$ simulations in a system with $n = 5$, $f = 1$, $\Delta = 1$

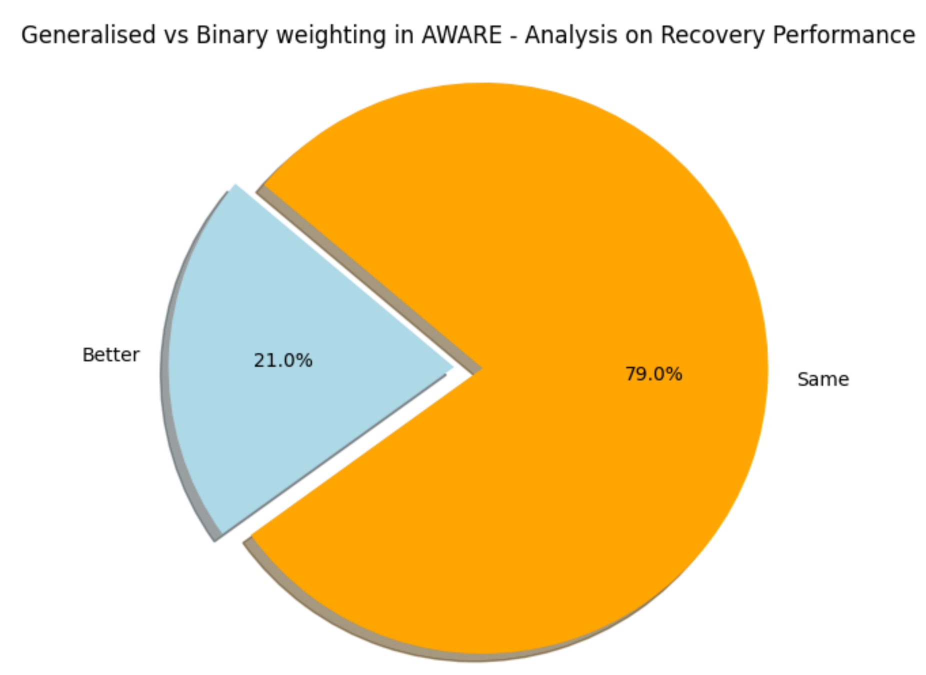


Figure 6: Recovery performance of *Generalised Weighting Scheme* on AWARE in a system with $n = 5$, $f = 1$, $\Delta = 1$