

Shoal: Improving DAG-BFT Latency and Robustness

FC'24

Alexander Spiegelman, Rati Gelashvili, Balaji Arun, Zekun Li. Aptos

Contents

1 Context: DAG-based BFT Consensus	3
2 Problem	11
3 Solution: Pipelining	13
4 Evaluation	16

1 Context: DAG-based BFT Consensus

BFT Consensus



- $N \geq 3f+1$ validators in total
- At most f validators are faulty

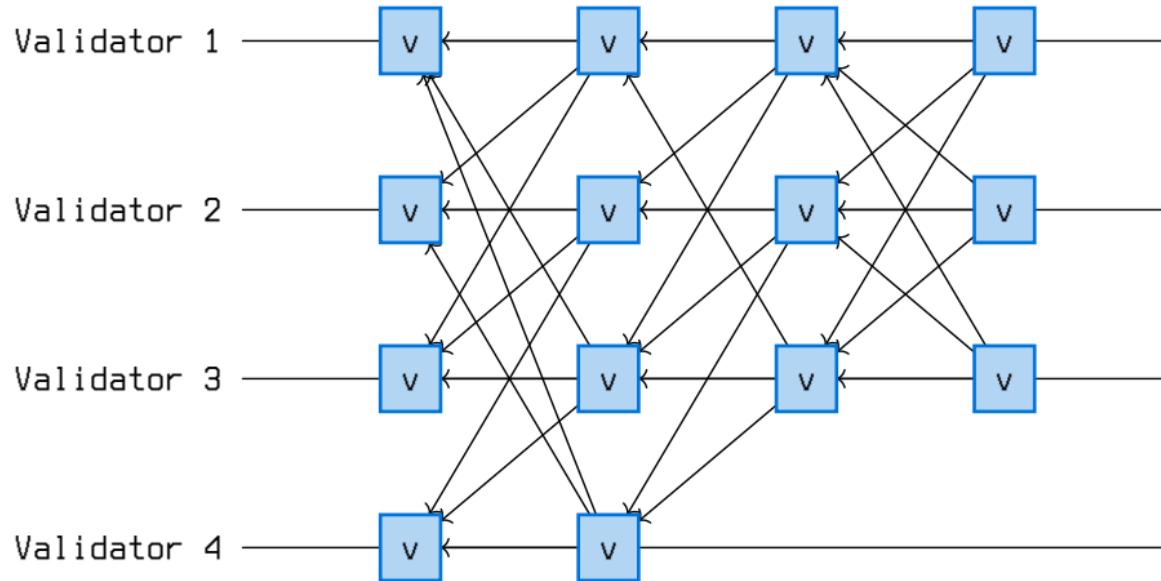
Global agreement on an infinitely growing sequence of some values.

Idea

Separate the network communication layer from the consensus logic.

- Each message contains a set of transactions, and a set of references to previous messages.
- Together, all the messages form a DAG that keeps growing - a message is a vertex and its references are edges.

DAG Example



Unifies abstraction

Reliable BFT broadcast (Not all protocols)

Result:

- All honest validators eventually deliver the same vertices and all vertices by honest validators are eventually delivered.
- Causal history of any vertex in both local views is exactly the same.

Consensus Mechanism



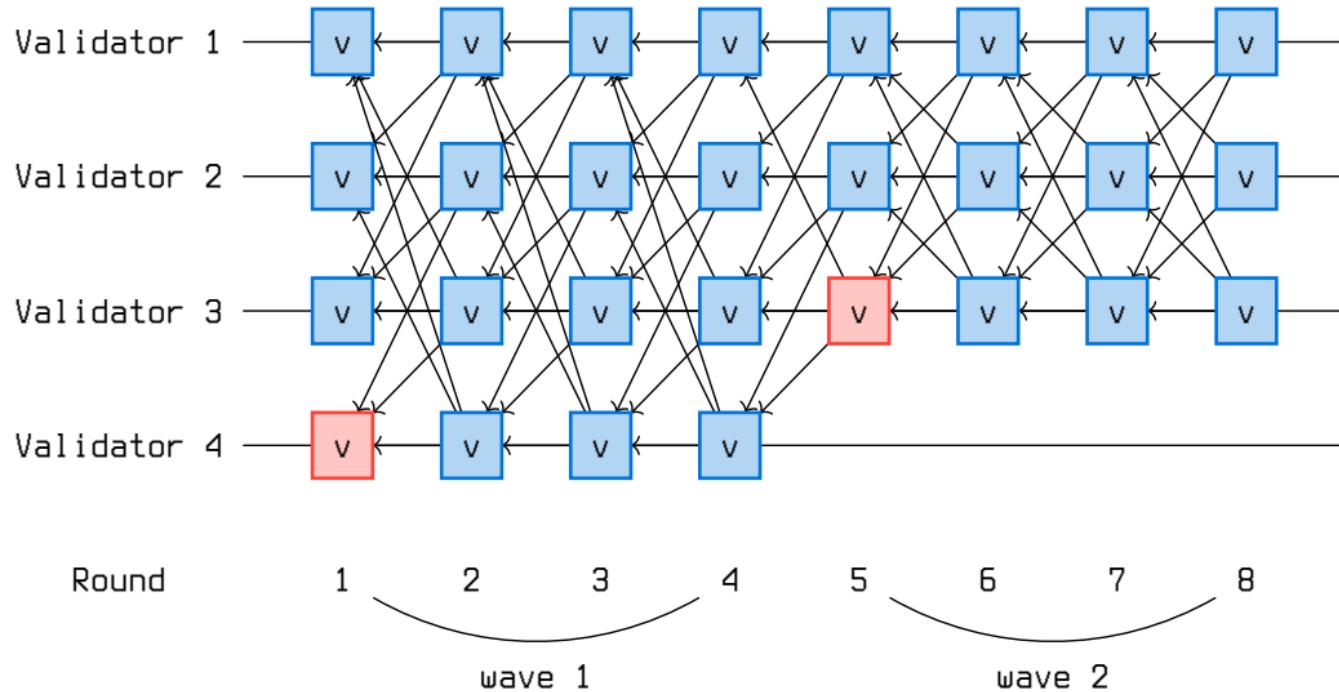
- Solving locally!!
- No need of any extra communication.

Consensus Structure



- Consensus divided into rounds
- Rounds groups waves
- Each wave contains a leader
- Commit speed not faster than a wave size

DAG Waves example



2 Problem

Consensus committing speed

Problem

Commit speed is not faster than a wave size

Protocol	Common case round latency	Async round latency
DAG-Rider	4	E(6)
Tusk	3	E(7)
Bullshark	2	E(6)

3 Solution: Pipelining

General Algorithm



Consectetur adipiscing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
consequatur. Duis aute irure dolor in reprehenderit in voluptate velit
esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat
cupidatat non proident, sunt in culpa qui officia deserunt mollit anim
ex ea commodo consequat.

Leader reputation

*Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna aliquam quaerat.*

4 Evaluation

- Machines:
 - ▶ t2d-standard-32 type virtual machine
 - ▶ 32 vCPUs, 128GB of memory, up to 10Gbps of network bandwidth.
- Cluster:
 - ▶ Google Cloud
 - ▶ Machines spread equally across regions: us-west1, europe-west4, asia-east1.
 - ▶ Latencies: us-west1 asia-east1 [118ms]; europe-west4 asia-east1 [251ms]; us-west1 europe-west4 [133ms]
 - ▶ Cluster size (N): 10 ($f \leq 3$); 20 ($f \leq 6$); 50 ($f \leq 16$)
- Data:
 - ▶ Transactions ~270B in size
 - ▶ Maximum batch size of 5000 transactions

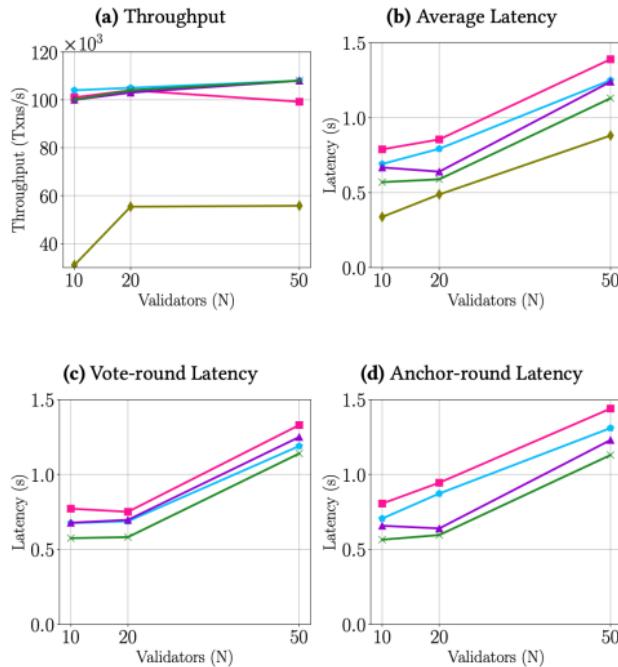
Latency

Time elapsed from when a vertex is created from a batch of client transactions to when it is ordered by a validator

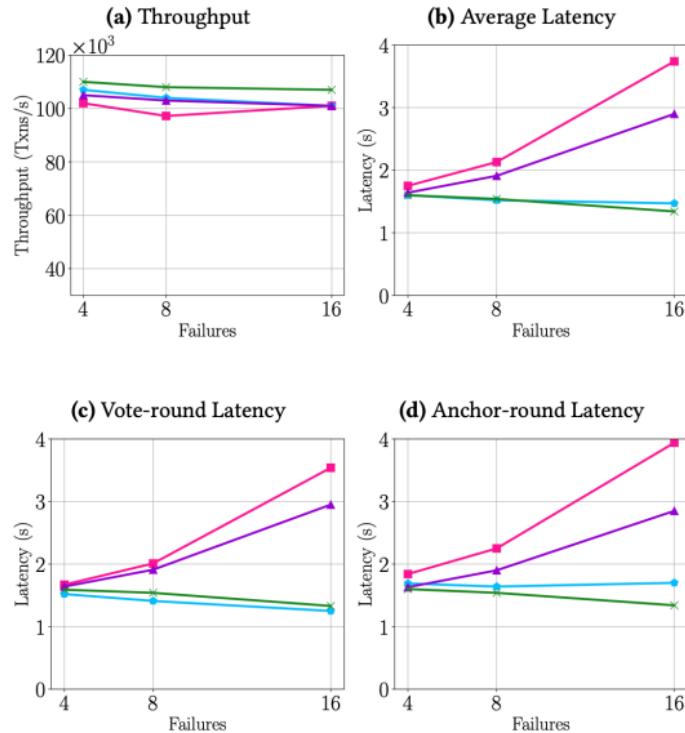
Results: No failures



■ Baseline Bullshark ■ Shoal PL ■ Jolteon
■ Shoal LR ■ Shoal



Results: With failures



Results: Skipping leaders

◀ ● ● ● ●

