COLABORAÇÃO PARA AVALIAÇÃO DE ALGORITMOS DE CONSENSO

Lucca Dornelles Cezar
Prof. Dr. Fernando Luís Dotti

- Graduação
  - Ciência da Computação (PUCRS)
  - Prof. Fernando Dotti
- Mestrado – CC (PUCRS)
  - Ciência da Computação (PUCRS)
  - Prof. Fernando Dotti
  - Algoritmos distribuídos → Blockchain

- Mestrado – CC (PUCRS)
  - Colaboração Informal
  - USI - MaRS (Prof. Fernando Pedone)

Índice

- Tendermint
- Comet/Malachite
- Projeto

Tendermint

TENDERMINT

- Algoritmo de consenso
  - Multiplas máquinas
  - Ordenar valores
    - Edição de texto online
    - Bancos
    - Blockchains

TENDERMINT

- Ambiente bizantino (Internet)
  - Problemas de rede
  - Comportamento arbitrários
    - Clientes maliciosos
    - Hackers
    - Bugs

TENDERMINT

"In essence, to make a reliable system from unreliable parts."

- *Buchman, Ethan. Tendermint: Byzantine fault tolerance in the age of blockchains. Diss. University of Guelph, 2016.*

TENDERMINT

- Blockchain
  - Bitcoin
    - " Banco" decentralizado
  - Etherium
    - " Computador" decentralizado
  - Tendermint
    - Aplicação decentralizada

Cosmos/Malachite

IMPLEMENTAÇÕES

- CometBFT
  - Informal
  - Go
  - Blockchain
  - ABCI (Application Blockchain Interface)

IMPLEMENTAÇÕES

- Malachite
  - Informal
  - Rust
  - Valor genérico
  - Framework em rust

Projeto

PROJETO

- PUCRS – Informal
  - Estudo de performance
    - CometBFT
    - Malachite

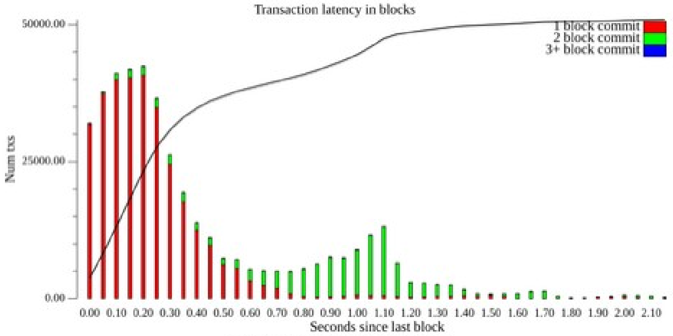
PROJETO

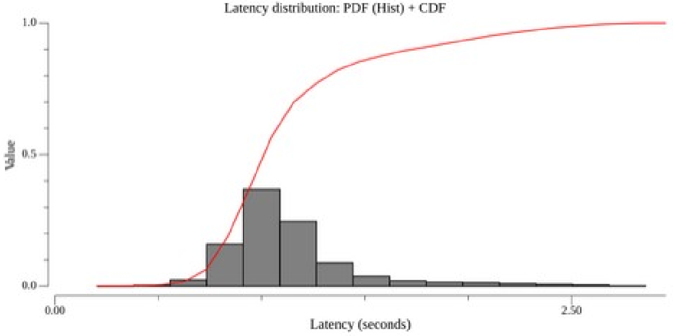
- Ferramentas
- Experimentos
- Processamento de dados

PROJETO

- Relatório, grafícos, dados

This happens because there is a small pooling window (hardware dependent) in which transactions can be received for the first next block.





- Note that while large scale experiments usually show two latency peaks, this is a local experiment with only a single node. There is a considerable amount of transactions with larger latency but not enough to form a second latency peak due to factors such as the lack of network delay and no transaction propagation.

- when the mempool size approaches 2 block sizes the buffer zone allows for transactions to be queued for the next block even when the current block is already full. Queuing ensures blocks are always full (given enough load) and considerably increases and stabilizes transaction throughput at the cost of latency, as the first next block is always full.

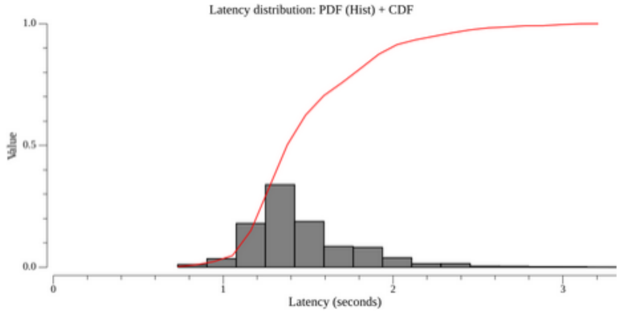
- Saturation Values

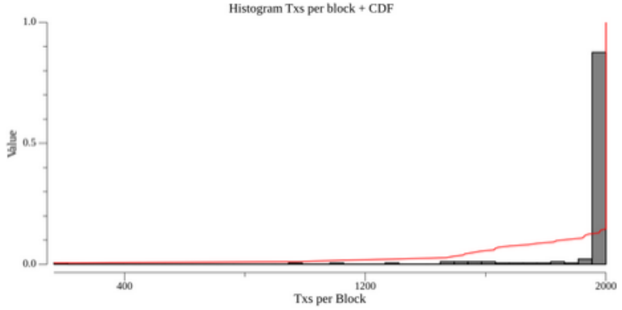
If we zoom into the saturation points we can have a more detailed look at the results.

In this ideal case where there is only a single node and it receives close to the maximum number of transactions it can possibly handle (but not more), the throughput is extremely stable. There is little variation in latency and block size. Even the fluctuating transaction rate is relatively stable when compared to other experiments. In some cases the block histogram even shows up as a solid value as every single block during the entire experiment had the exact same size with no fluctuation whatsoever.

Note, however, that the saturation points are vastly different, and in a real scenario this saturation point would not only vary with the topology and characteristics of each node, but also change depending on the current leader.

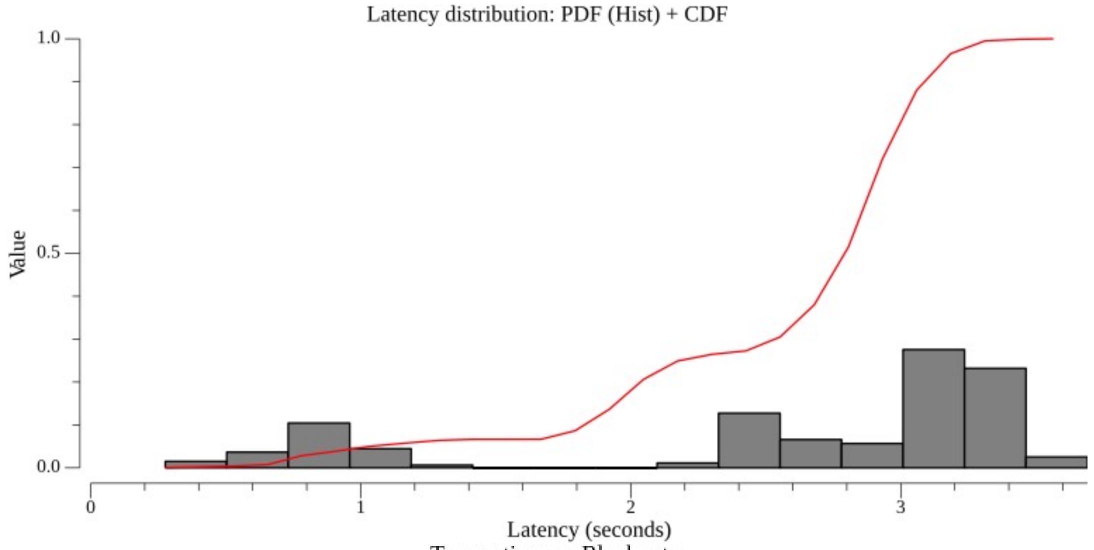
- m510 - window size = 2000txs:

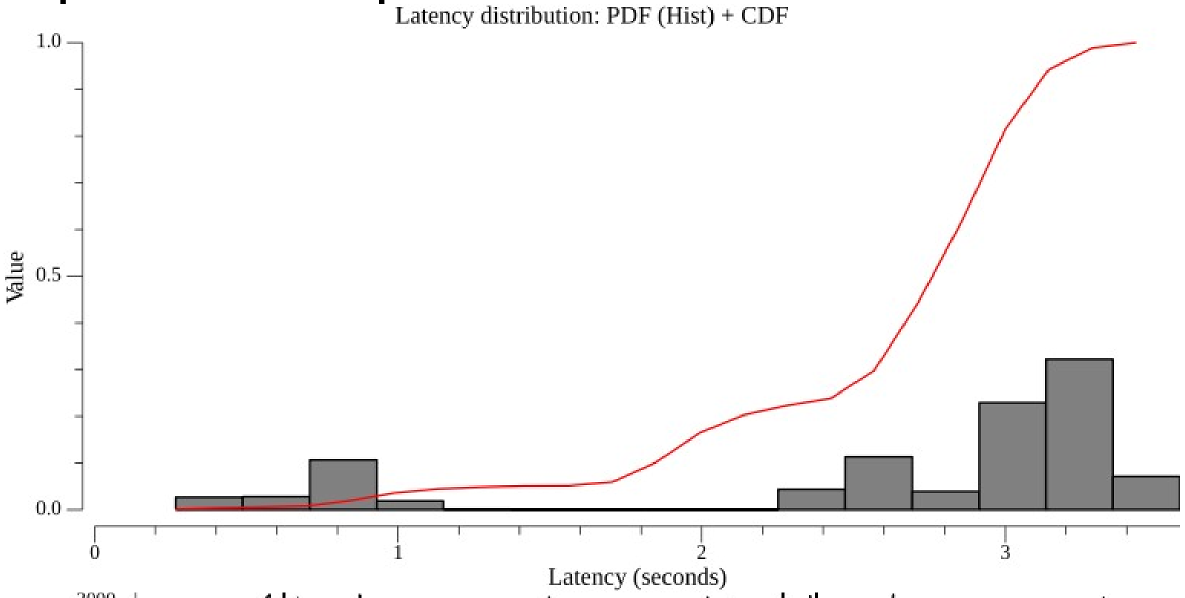


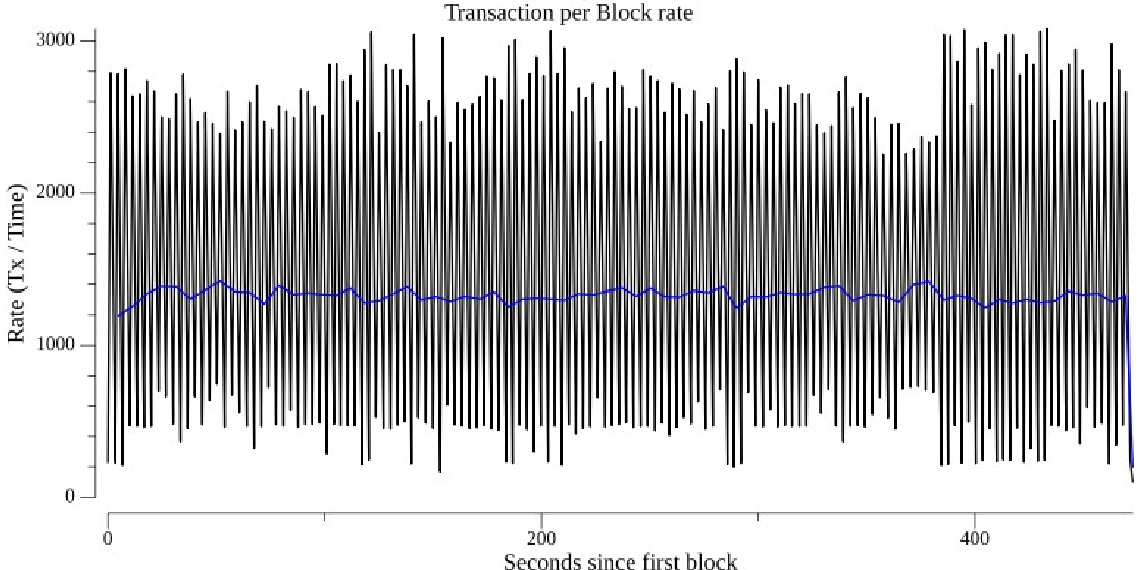


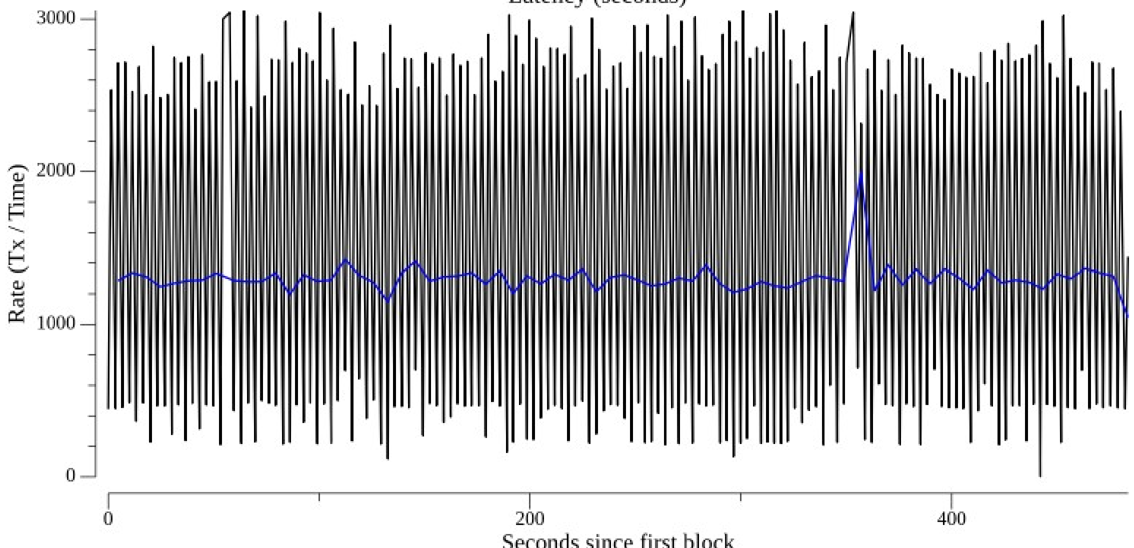
PROJETO

- CometBFT – 2 processos por máquina



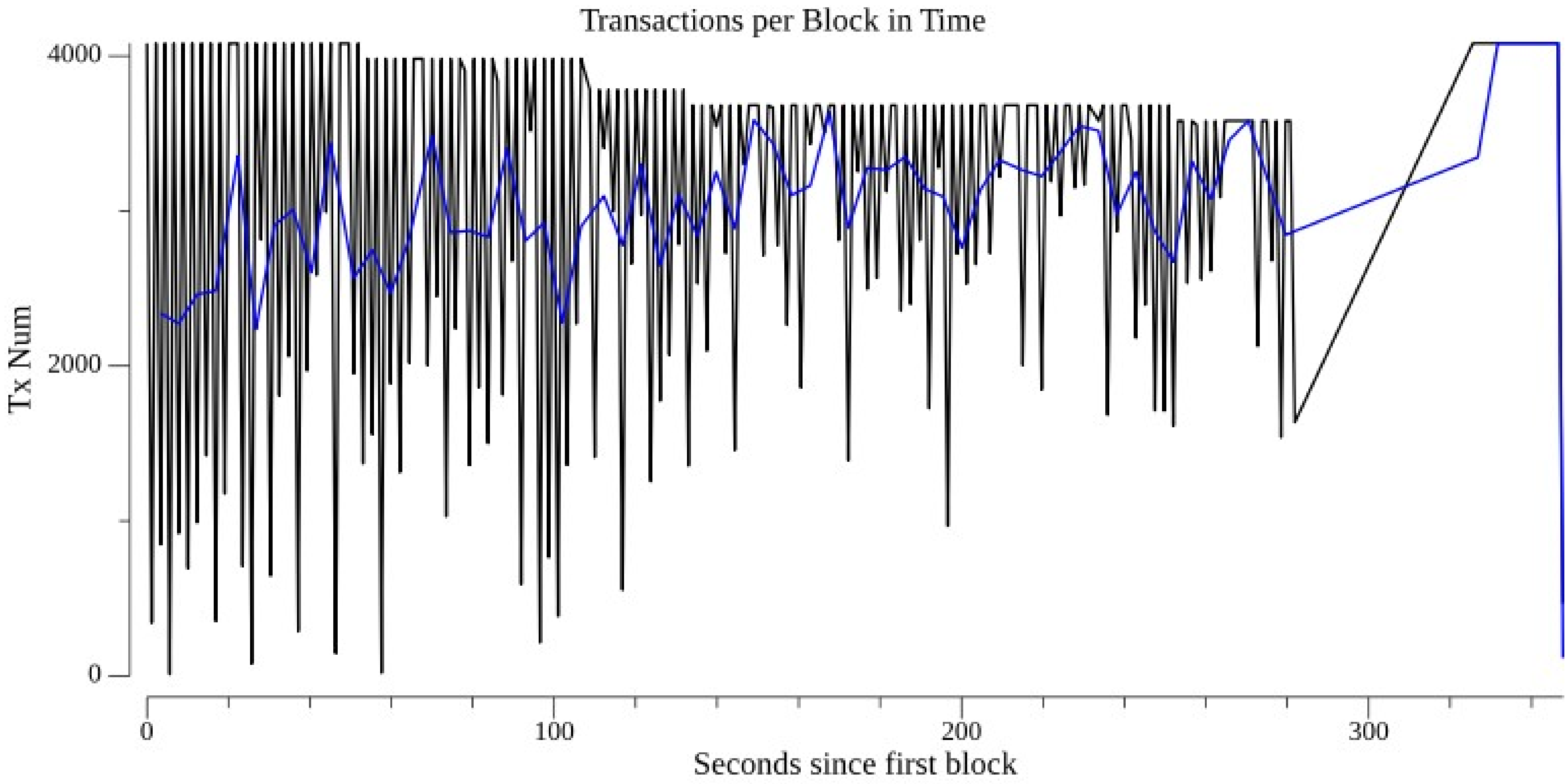






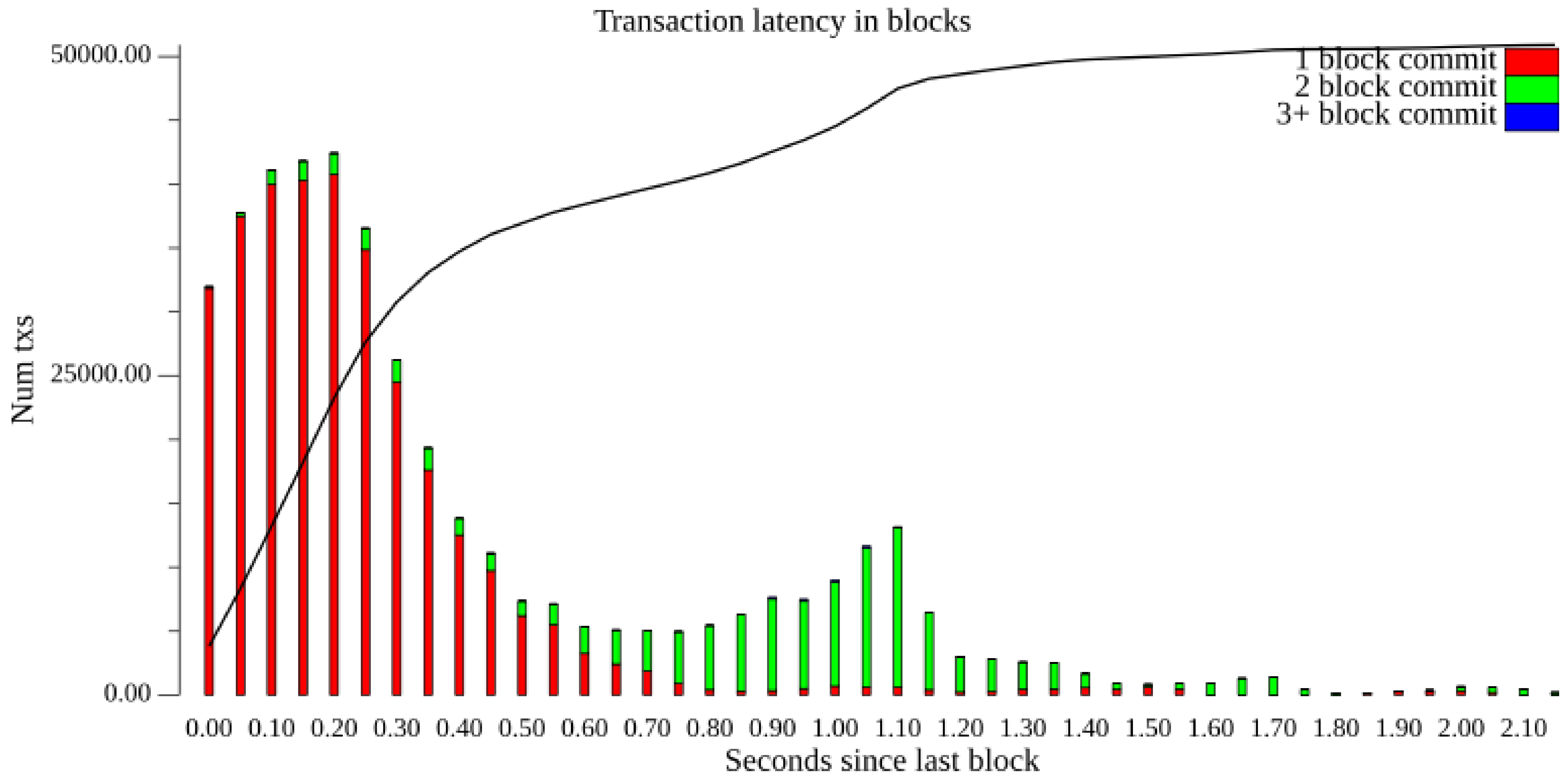
PROJETO

- CometBFT – Zigue-zague



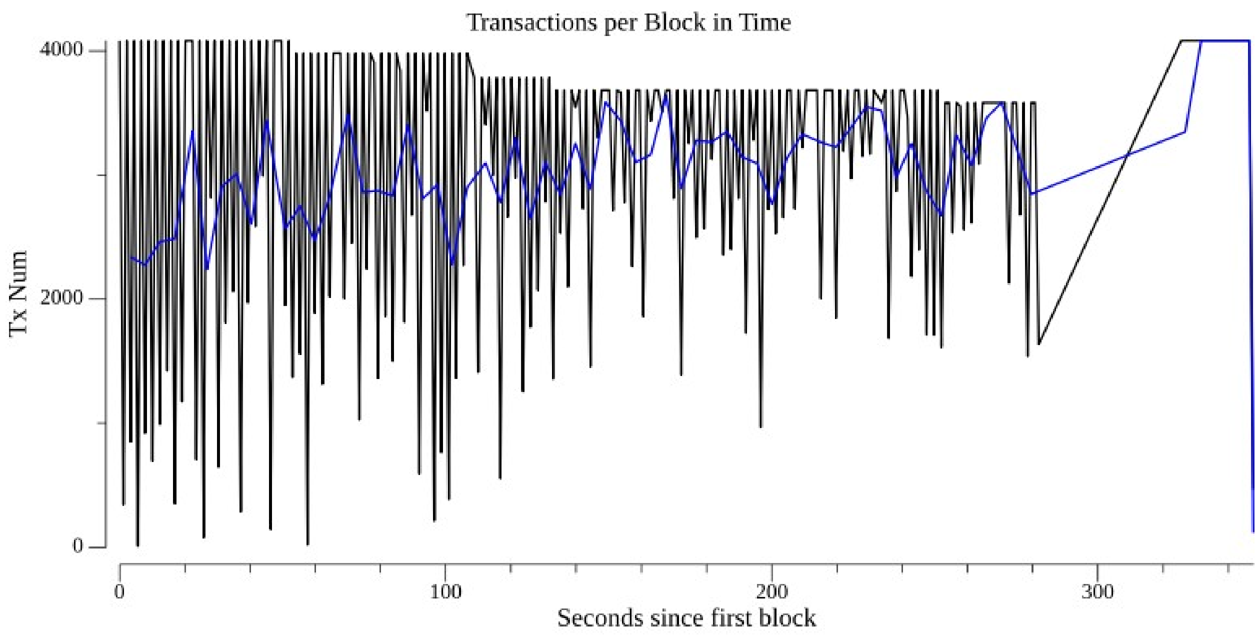
PROJETO

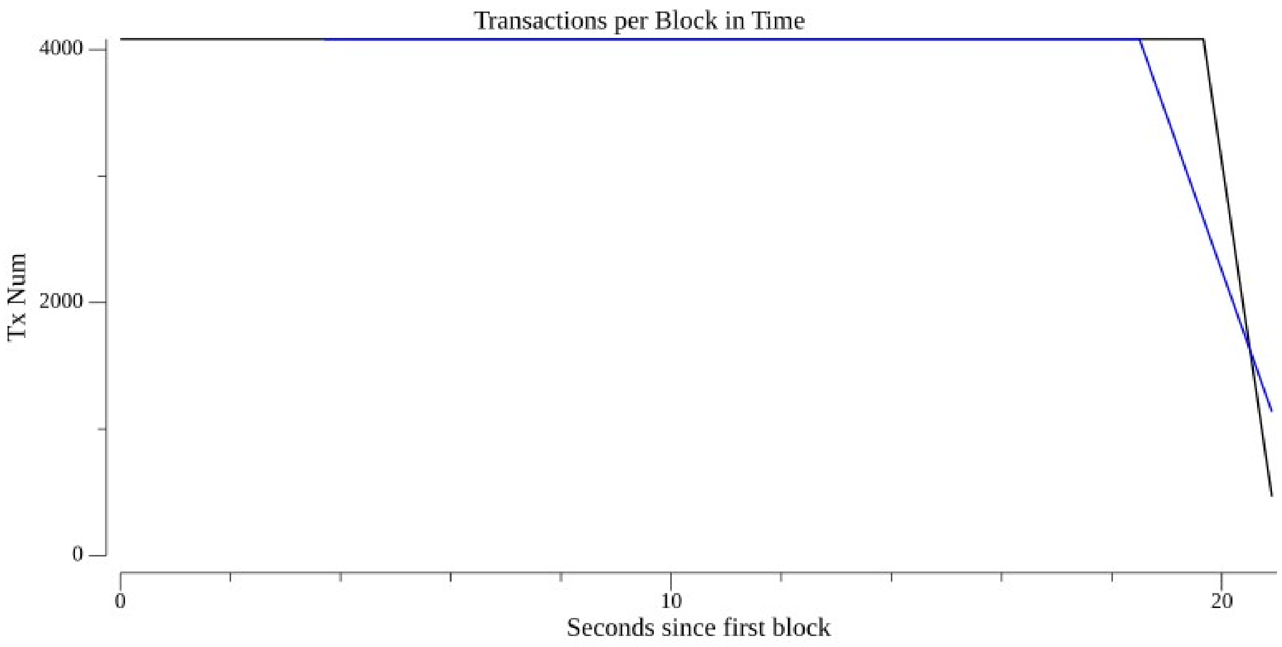
- CometBFT – Zigue-zague



PROJETO

- CometBFT – Zigue-zague





PROJETO

- Malachite
  - Ferramentas
  - Cluster (Cloudlab)
  - Experimentos de larga escala