



CeADAR - The National Centre for Applied Data Analytics and Machine Intelligence

Blockchain Scalability Testing (WP-3) Report

Document Type:	Scalability report
Project Title:	Blockchain Scalability Testing (WP-3) - Report
Work Package:	3
Authors:	Prabhakaran AK, Saad Shahid, Oisín Boydell
Document Version:	0.1
Date:	July 2020
Number of pages:	9

ABSTRACT

Blockchain Scalability Testing - report (WP-3).

Copyright © the authors. Confidential – not to be circulated without permission.

CeADAR is a research partnership comprising University College Dublin and the Technical University of Dublin.

<http://www.ceadar.ie>

Contents

1	Introduction	3
2	Tool	3
2.1	Requirements	3
2.2	Assumptions	3
2.3	About Hyperledger Caliper	4
2.4	Why Hyperledger Caliper	4
2.5	Performance & Scalability Working Group - metrics	4
2.5.1	Success Rate	4
2.5.2	Transaction & Read Latency	4
2.5.3	Transaction & Read Throughput	4
2.5.4	Dependency Latency vs Throughput	5
2.6	Resource Consumption	5
3	Network setup	5
3.1	Version	5
3.2	Consensus	5
3.3	Organizations	5
4	Experiment	5
4.1	Transaction volume	5
4.2	Throughput	5
4.3	Node count	6
4.4	Plotting the results	6
4.4.1	Results for 5% of the network scale	6
4.4.2	Graph for 20% of the network scale	7
4.4.3	Graph for 50% of the network scale	7
4.4.4	Graph for 75% of the network scale	8
4.4.5	Graph for full network scale	8
A	Caliper reference	9
B	Fabric reference	9
	Bibliography	9

1 Introduction

Blockchain scalability testing predicts the performance of the blockchain network while the consortium grows over the period of time. As the number of nodes in the consortium grows, volume of the transactions going through the network is expected to grow as well.

This document presents the result of the scalability testing. Several tests are run with varying volume and frequency of the transaction and the results are compared and presented for the better understanding of users.

2 Tool

For the scalability testing purpose we are going to use Hyperledger caliper¹. It supports various version of hyperledger fabric. For this document purpose, Hyperledger caliper V0.3 will be used.

2.1 Requirements

The requirement is to analyze the scalability performance of a smart contract in a blockchain network. The scalability check will determine the latency variation against the variation in transaction volume and throughput.

2.2 Assumptions

The following are the assumptions for this test run

- This document use Hyperledger caliper version 0.3.
- Full size of network is assumed to be 136 Organizations
 - Distributor - 1
 - Publishers - 111
 - Providers - 24
- It covers the product key and product key batch request smart contract
- Since both the smart contract have similar overload, for testing purpose we will use
 - Product key smart contract
 - Create, Allocate and Query method
- We will run the experiments against the following metrics

Transaction volume - Fixed at 10,000

# of nodes / TPS	10	25	50	100	500
07(5%)					
27 (20%)					
68 (50%)					
102 (75%)					
136 (100%)					

- The results of the experiments will be consolidated into excel sheet and presented in a chart format

¹<https://www.hyperledger.org/projects/caliper>

2.3 About Hyperledger Caliper

We are going to use Hyperledger Caliper [1], a blockchain bench-mark tool to measure the performance of the blockchain implementation with a set of predefined use cases.

Hyperledger caliper is a blockchain performance bench-marking tool from the Linux foundation.

2.4 Why Hyperledger Caliper

Caliper has a defined Performance & Scalability Working Group (PSWG) [2] which contains definitions and metrics for the blockchain network bench-marking.

It has support for multiple versions of hyperledger fabric which is our blockchain platform for the present use case

Caliper helps us to determine the various-metrics for given volume of transactions at defined throughput.

2.5 Performance & Scalability Working Group - metrics

The metrics that are defined by the PSWG of Hyperledger caliper are as follows:

2.5.1 Success Rate

Measures all successful and failed transactions for a test cycle.

- This metrics is based on the volume of the transactions.
- Caliper allows users to configure the transaction volume for the testing.
- Caliper final report includes the success and failure rate for the given volume of transactions.

2.5.2 Transaction & Read Latency

Measures the time for an issued transaction to be completed and a response being available to the application that issued the transaction.

- This metrics is based on the time taken for one single transaction in seconds.
- Latency is the output by hyperledger caliper for given volume of transactions at defined throughput.
- Maximum, minimum and average latency in seconds for the test cycle is provided.

2.5.3 Transaction & Read Throughput

Measures the flow rate of all transactions through the system, in transactions per second, during the a cycle.

- This metrics defines the flow rate of transaction into the blockchain system.
- Throughput can be configured at caliper.
- Unit of measurement for throughput is Transactions per second (TPS).
- Various types of throughput feeding is possible in caliper.
 - fixed-rate : Fixed rate of throughput is maintained from start to end of the transaction volume.
 - linear-rate : Variable rate between beginning and towards end of the transaction volume.
 - composite-rate: Composite rate allows both fixed rate and linear rate throughput to be used on given transaction volume.

2.5.4 Dependency Latency vs Throughput

Hyperledger caliper provides sample smart contracts to run the scalability check, and based on the results of running it against different transaction volume and throughput we arrived at the following assumption:

- Latency increases linearly with number of nodes in the network.
- Throughput is configurable for each test run.
- Latency increases when the volume of the transaction and throughput is increased.

The same will be verified against each of our smart contract across WP-6 in phases.

2.6 Resource Consumption

Measures the following resource parameters of the blockchain network

- Max. and Min, memory - Memory used in MegaBytes
- CPU resource consumption - CPU usage in percentage
- I/O traffic - KiloBytes/ MegaBytes

3 Network setup

The blockchain network structure on which the experiments will be run are as follows

3.1 Version

Hyperledger Fabric version 1.4.4 will be used

3.2 Consensus

RAFT consensus mechanism with two orderer node will be used

3.3 Organizations

A total of 136 organizations will be used with a single peer structure. All the organizations will be hosted in Azure cloud

4 Experiment

The scalability experiment will be executed as follows

4.1 Transaction volume

A total of 10000 transactions will be used for each round of experiment. The transaction volume is fixed for all the experiments

4.2 Throughput

Totally five sets of throughput will be used in a fixed rate manner.

- 10 TPS
- 25 TPS
- 50 TPS
- 100 TPS
- 500 TPS

4.3 Node count

Totally 136 nodes will be used under that assumption, the network can contain a maximum of one distributor, 111 publishers and 24 providers. The node count will be spread across for experiment as follows

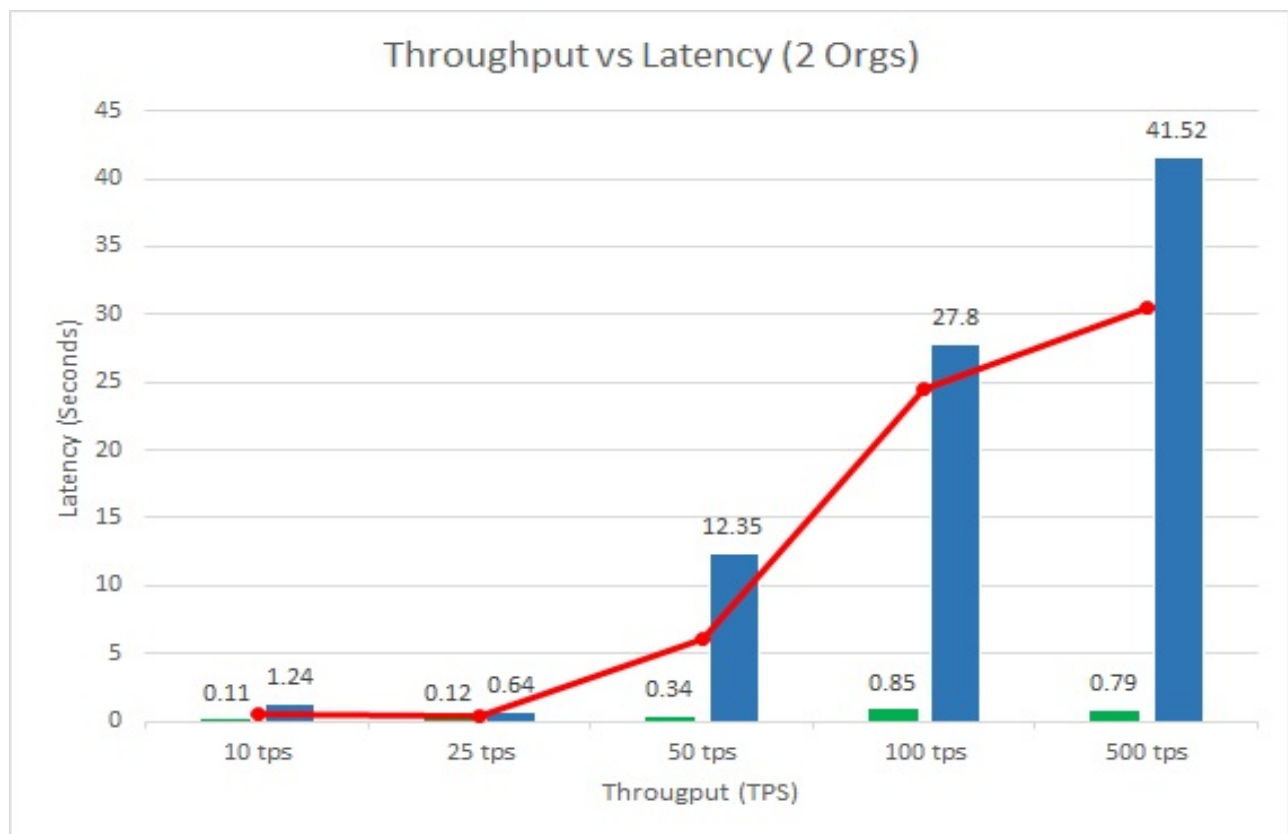
- 1 distributor - 3 publishers - 3 providers (5% of the total size)
- 1 distributor - 20 publishers - 7 providers (20% of the total size)
- 1 distributor - 77 publishers - 24 providers (50% of the total size)
- 1 distributor - 20 publishers - 24 providers (75% of the total size)
- 1 distributor - 111 publishers - 24 providers (Full network)

4.4 Plotting the results

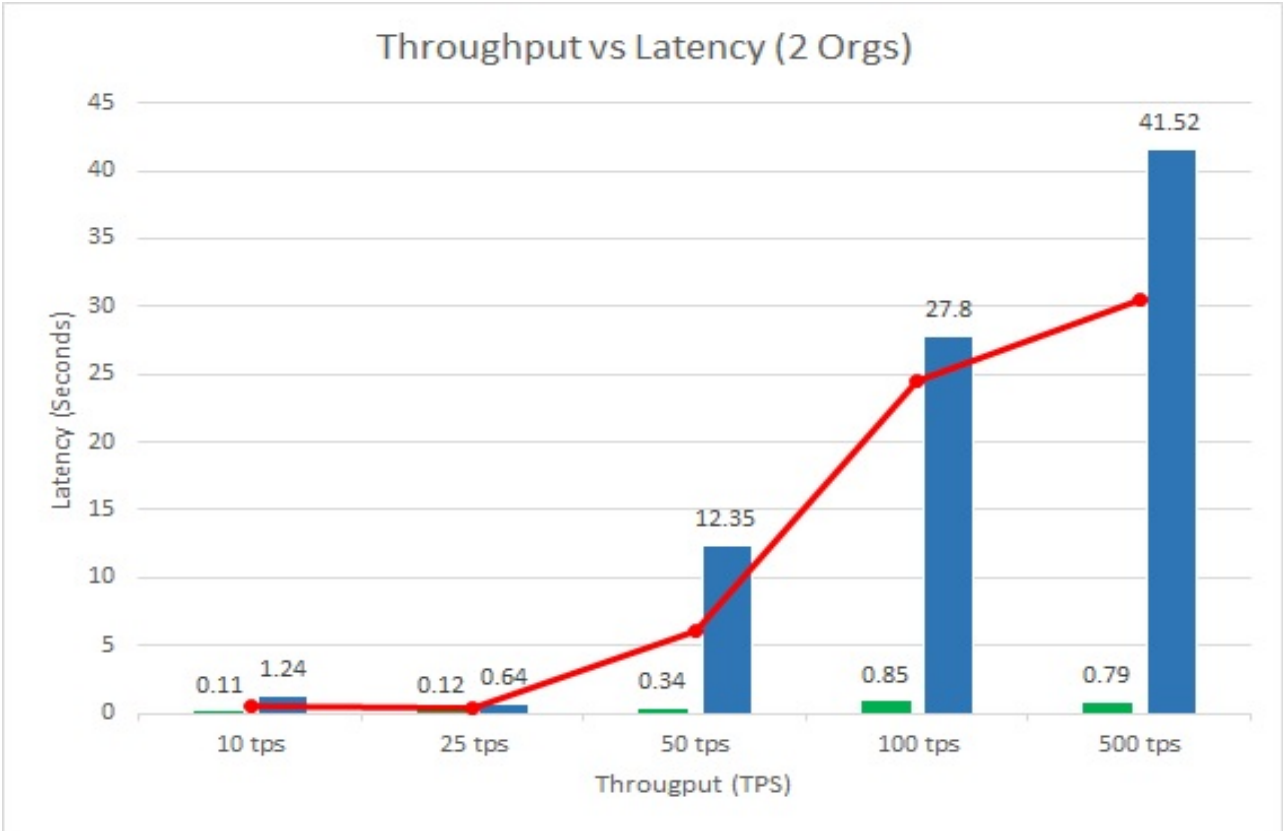
The results from the above experiments are plotted in five different charts. A chart for each scale set of nodes mentioned in section 4.3 will be plotted. The chart will contain the throughput in TPS plotted vs latency in seconds.

Each round of experiment will give two sets of latency values, min latency and max latency. For each throughput mentioned in section 4.2, min latency, max latency and average latency will be marked.

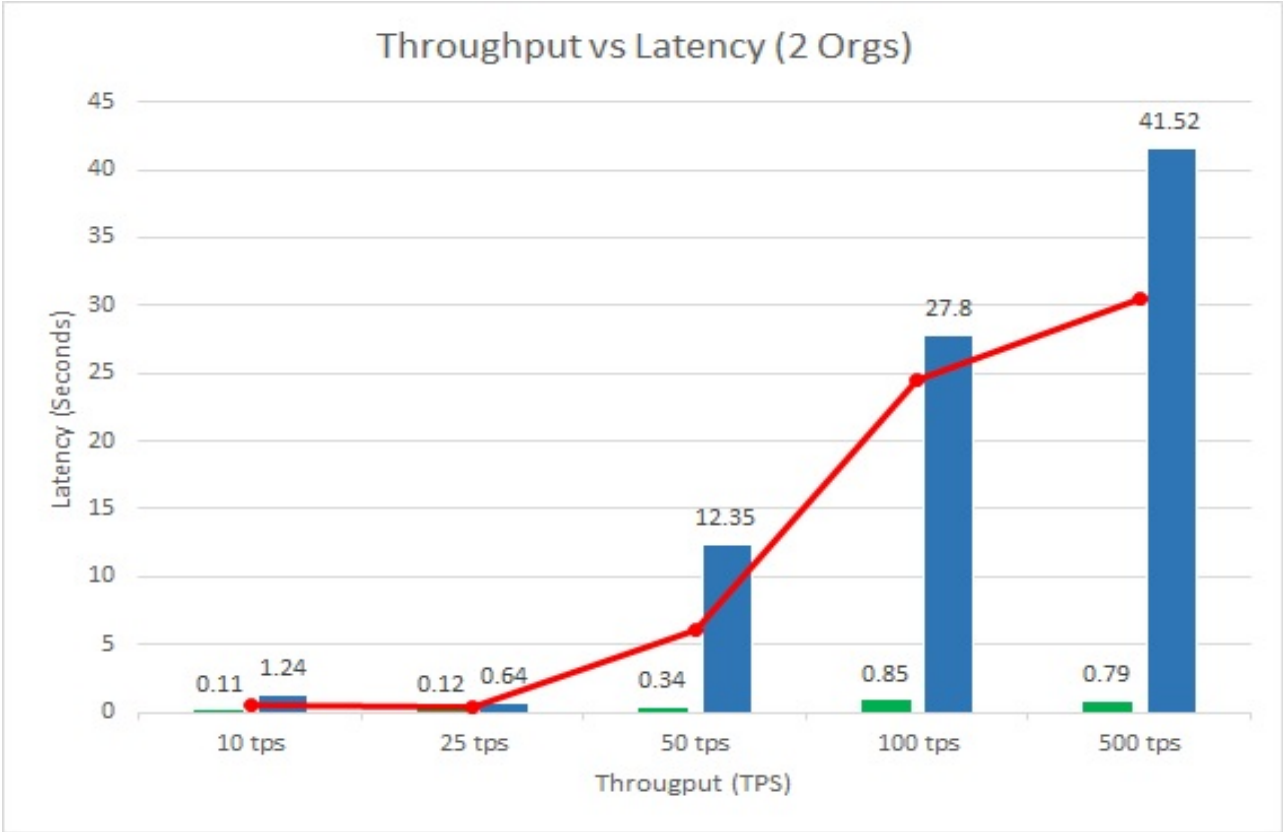
4.4.1 Results for 5% of the network scale



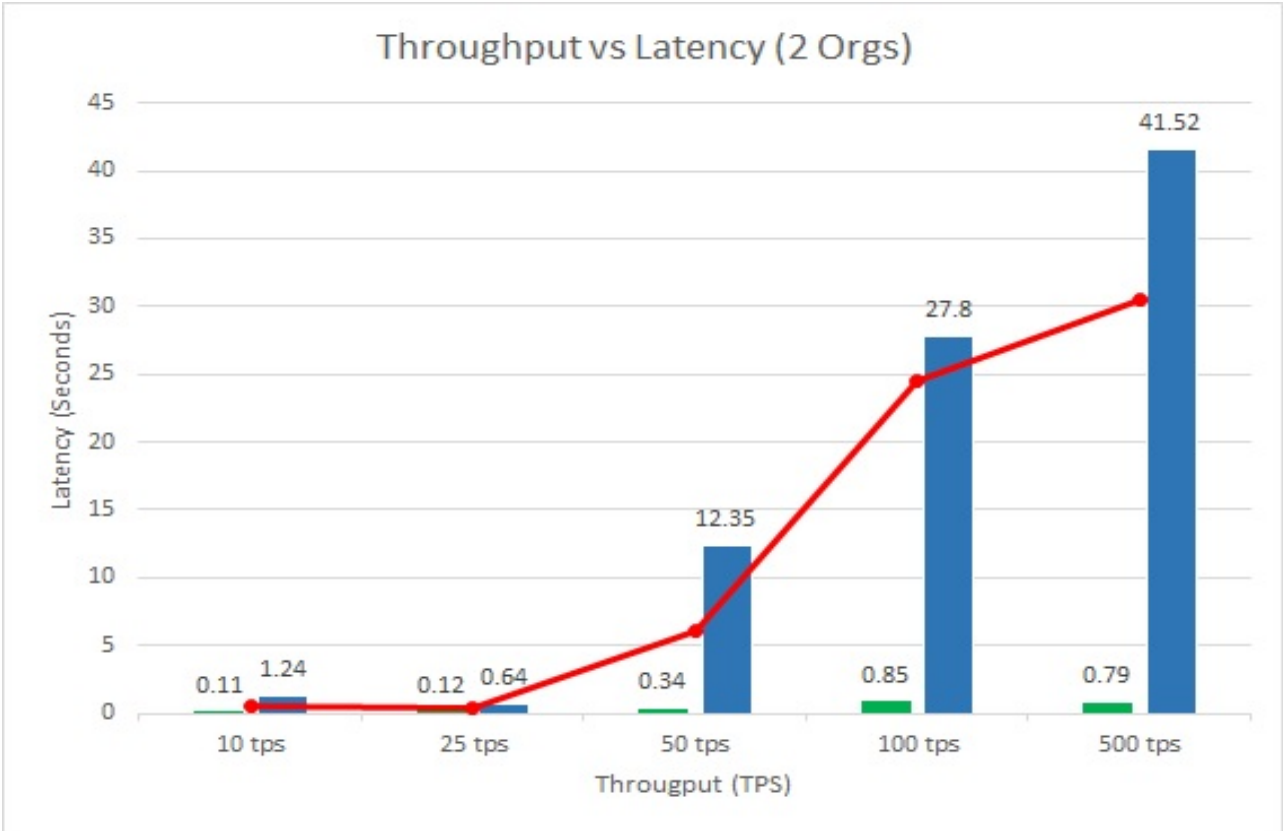
4.4.2 Graph for 20% of the network scale



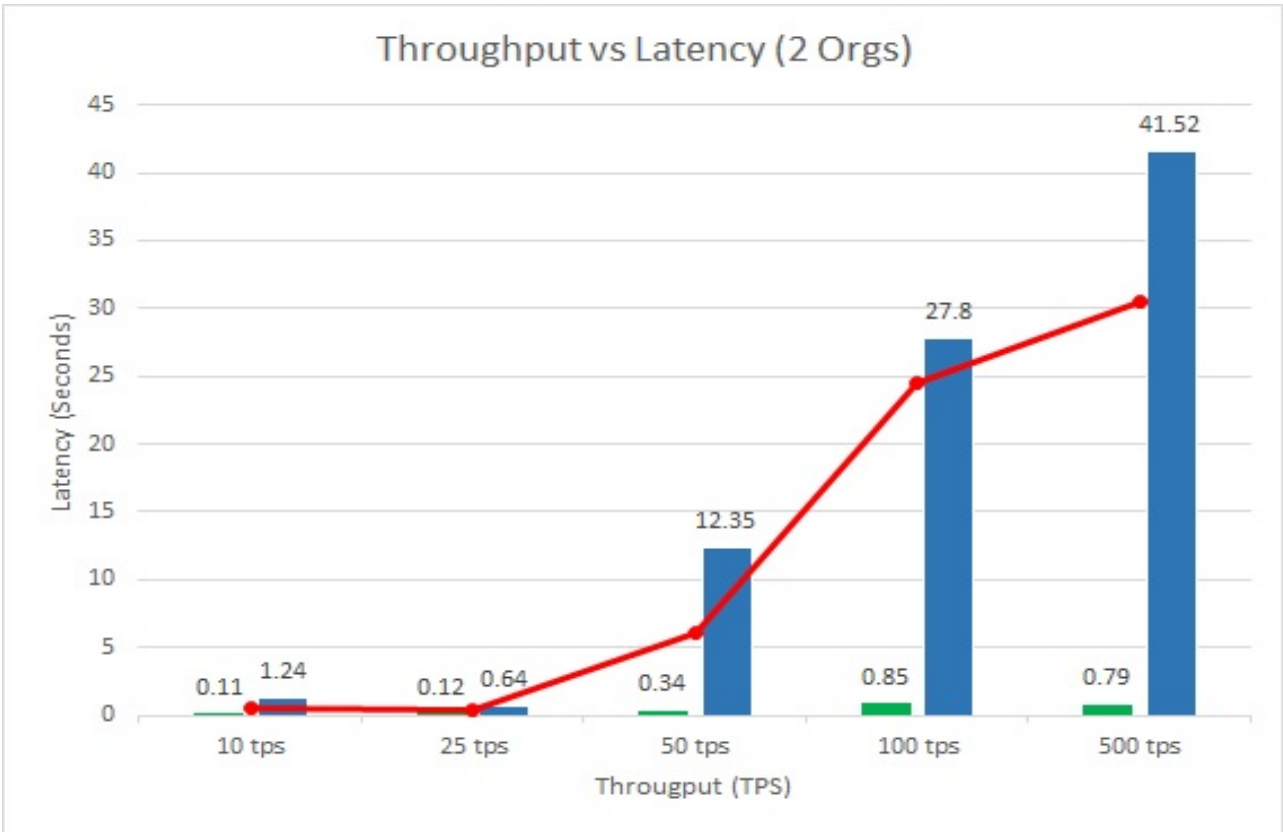
4.4.3 Graph for 50% of the network scale



4.4.4 Graph for 75% of the network scale



4.4.5 Graph for full network scale



A Caliper reference

The caliper reference files used for the experiments can be found at gitlab location

<https://www.hyperledger.org/projects/caliper>

B Fabric reference

The Hyperledger fabric network reference files used for the experiments can be found at gitlab location

<https://www.hyperledger.org/projects/caliper>

References

- [1] Hyperledger. Hyperledger Caliper. <https://hyperledger.github.io/caliper/>, 2020.
- [2] Hyperledger-Caliper. Performance & Scalability Working Group. <https://www.hyperledger.org/resources/publications/blockchain-performance-metrics/>, 2020.