



Comparing the performance of blockchain implementations

ITU WORKSHOP ON DISTRIBUTED LEDGER TECHNOLOGY SCALABILITY AND INTEROPERABILITY

2019.8

About me

Ruifeng(Victor) HU

- ▶ Research director at Huawei
- ▶ ITU FG DLT WG3 leader
- ▶ Hyperledger Caliper project founder



Me and my decorator

Problems Faced

- ▶ More nodes usually wouldn't bring higher performance
- ▶ Large numbers of implementations
- ▶ Different implementation concept from smart contracts to consensus
- ▶ And more...
 - ▶ I wouldn't let you know those I can't solve

Preparation

- ▶ Identify yourself
 - ▶ User
 - ▶ Operator
 - ▶ Developer
- ▶ Know your limitations
 - ▶ Business
 - ▶ Cost

Concept - Users

- ▶ Concerns:
 - ▶ When my transactions will be confirmed
 - ▶ How many transactions can be handled at a period of time
 - ▶ How much would transactions cost me
- ▶ Black box testing
- ▶ Cost come from transaction fees

Concept - Ops

- ▶ Concerns:
 - ▶ How to enhance my customers experience
 - ▶ How to provide same customer experience with lower cost
- ▶ Black box testing
- ▶ Cost come from hardware investment
 - ▶ Hardware cost include: CPU, Memory, Disk and Network

Concept – Devs

- ▶ Concerns:
 - ▶ How to improve performance of my platform
 - ▶ Faster and more transactions
 - ▶ Lower overhead
- ▶ ~~Black box testing~~
- ▶ ~~White box testing~~
- ▶ Shattered back box
 - ▶ RPC call
 - ▶ Transaction propagation
 - ▶ Contract execution
 - ▶ State updating
 - ▶ Consensus and commit



Rule No. 1

DO consider the environment

Metrics

- ▶ TPS
- ▶ Latency

----- I am break -----

- ▶ TPS per (CPU/Memory/DiskIO/NetworkData)

----- I am break too -----

- ▶ RPC response time
- ▶ Transaction propagating rate
- ▶ Contract execution time
- ▶ State updating time
- ▶ Consensus cost time

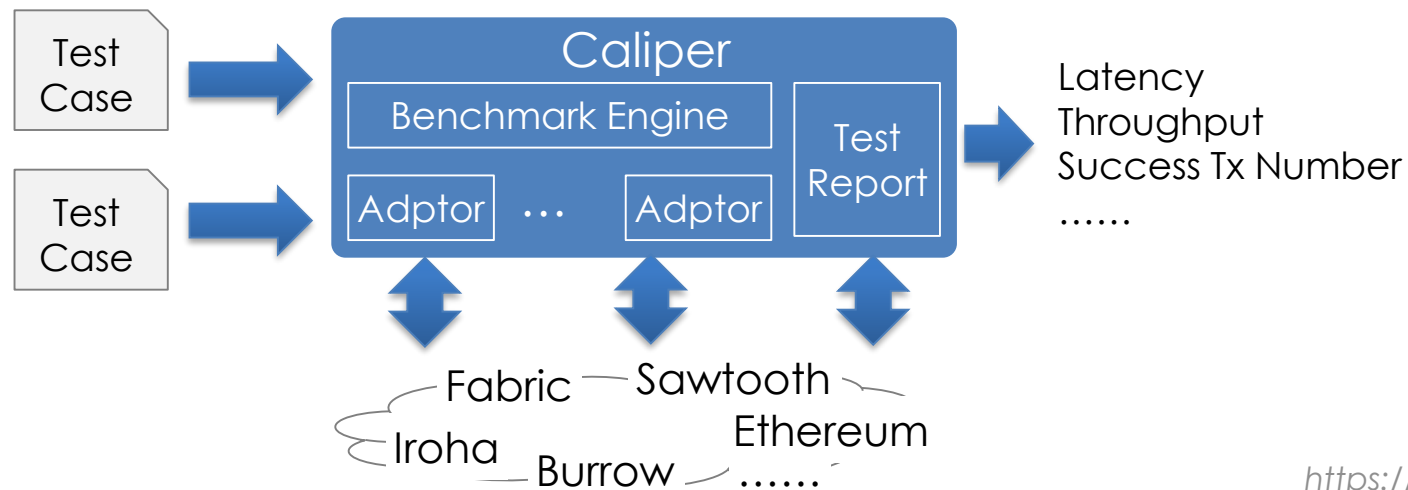
Benchmarking Approaches

- ▶ Event tracking (pretending to be an user)
 - ▶ Black box
 - ▶ Friendly to public network
- ▶ Log tracking (pretending to be an operator)
 - ▶ Shattered black box
 - ▶ Need to know whole network topology
- ▶ Code snippet (Yes I'm a developer)
 - ▶ White box
 - ▶ Everything about my platform under control

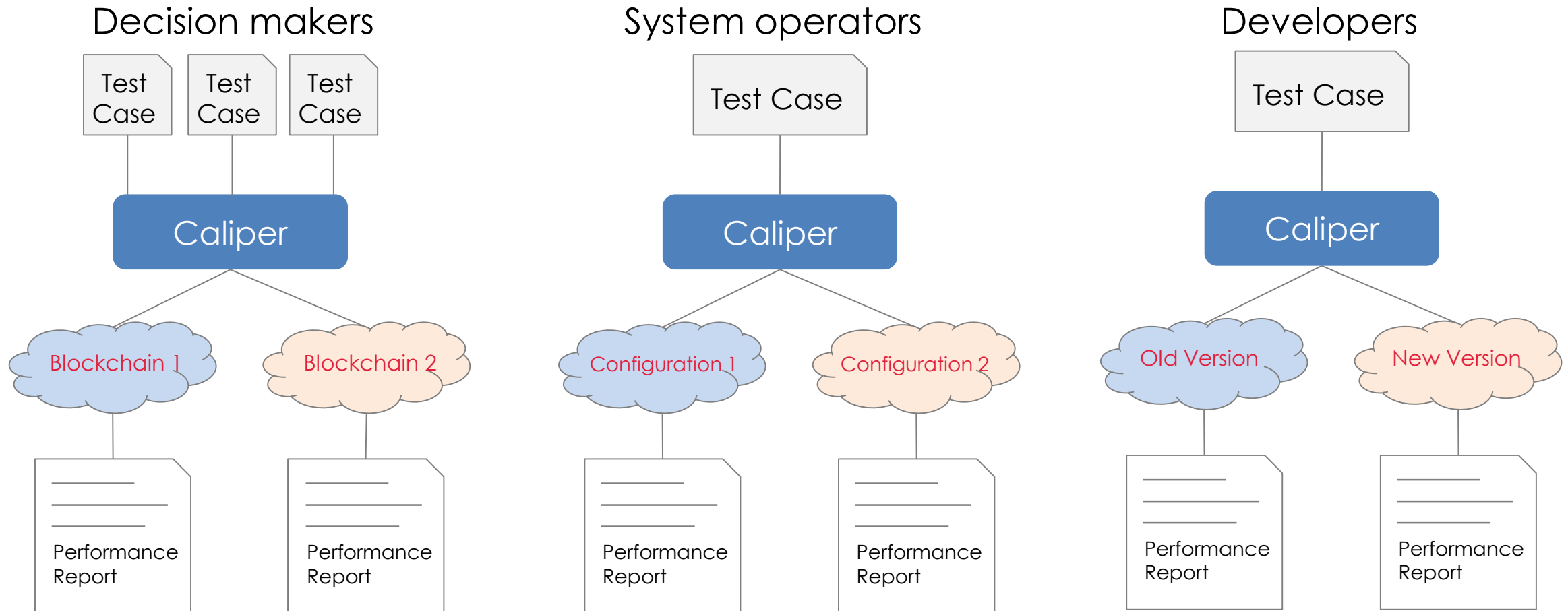
What is Caliper

Caliper is a performance benchmark framework for blockchain and one of the Hyperledger projects hosted by the Linux Foundation

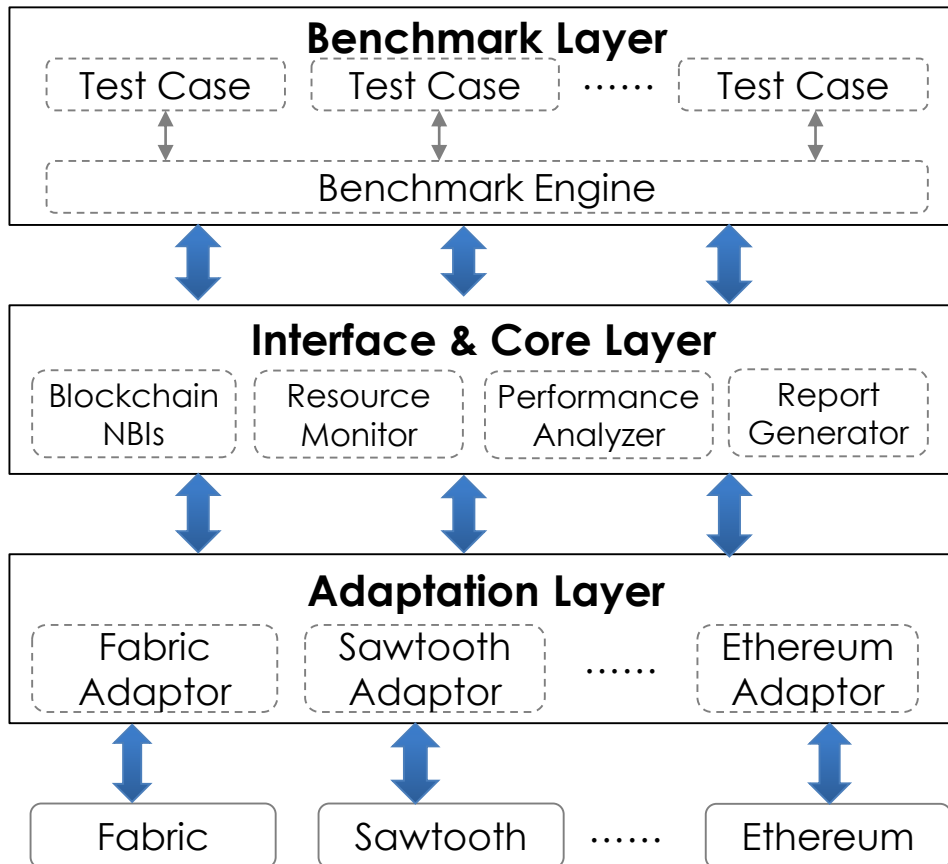
- Integrate with multiple existing DLTs (Distributed Ledger Technology)
- Measure the performance of specific blockchain systems with predefined test cases
- Reports containing standard performance indicators defined by Hyperledger Performance and Scale WG
- Provide abstract NBIs (Northbound Interface) to help extend test cases



Target Users and Typical Scenarios



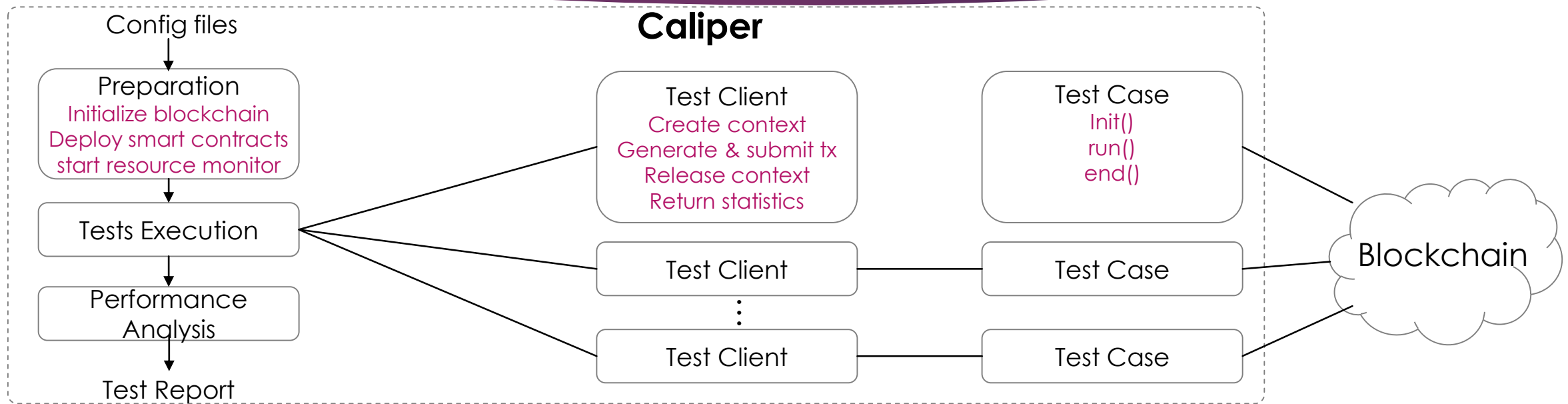
Architecture



Node.js based, 3 layers from top to bottom

- Benchmark Layer
 - Predefined benchmark test cases
 - Pluggable & configurable benchmark engine
- Interface & Core Layer
 - Blockchain NBIs – install, invoke, query.....
 - Resource Monitor – memory, cpu, network io
 - Performance Analyzer – latency, throughput
 - Report Generator – HTML format test report
- Adaptation Layer
 - Translate NBIs into DLT protocols

How it works



Master

Execute the test flow according to the configuration

- Preparation: prepare the test context, e.g. installing smart contracts
- Test Execution: assign tasks to clients to run the test
- Performance Analysis: gather test results & generate report

Clients

Run test case according to the specific workload

- Transaction count based test or duration based test
- Pluggable rate controller: Fixed submitting rate, Dynamic submitting rate based on specific schema, ...

Test Case

Scripts which define interactions with the system under test

- Use Caliper's NBIs to define common script for multiple blockchain systems

Roadmap

- ▶ Support of Fabric v1.1-1.4 (two adapters were provided : fabric and fabric-ccp) & Sawtooth v1.0 & Burrow & Iroha & Ethereum & BCOS
- ▶ Added Fabric kafka example and raft example
- ▶ Provided file mode to support long time test;
- ▶ Refactor code for npm package publishing
- ▶ Created CLI and sample package
- ▶ Improved documentation
- ▶ Enhanced abilities (configurable logging mechanism, fabric network creating detection, automatically generate crypto config, a license check process, Caliper running environment check)
- ▶ Npm package
- ▶ API interface
- ▶ Docker image
- ▶ Upload/edit test configuration
- ▶ Better visualization for configuration, execution and results display
- ▶ Corda adaptor
- ▶ Sufficient test coverage
- ▶ Quorum adaptor and more
- ▶ Sufficient use cases
- ▶ Metrics extension
 - ▶ Optional Metrics Framework
 - ▶ Costs for a certain TPS

Jun 2019

Sep 2019

Dec 2019

Welcome to join Caliper team

- ▶ Questions and suggestions
 - ▶ Rocket chat (<https://chat.hyperledger.org/channel/caliper>)
 - ▶ Issues (<https://github.com/hyperledger/caliper/issues>)
- ▶ Regular meeting: UTC 9am Wednesday (<https://wiki.hyperledger.org/display/caliper>)

Thank You!

