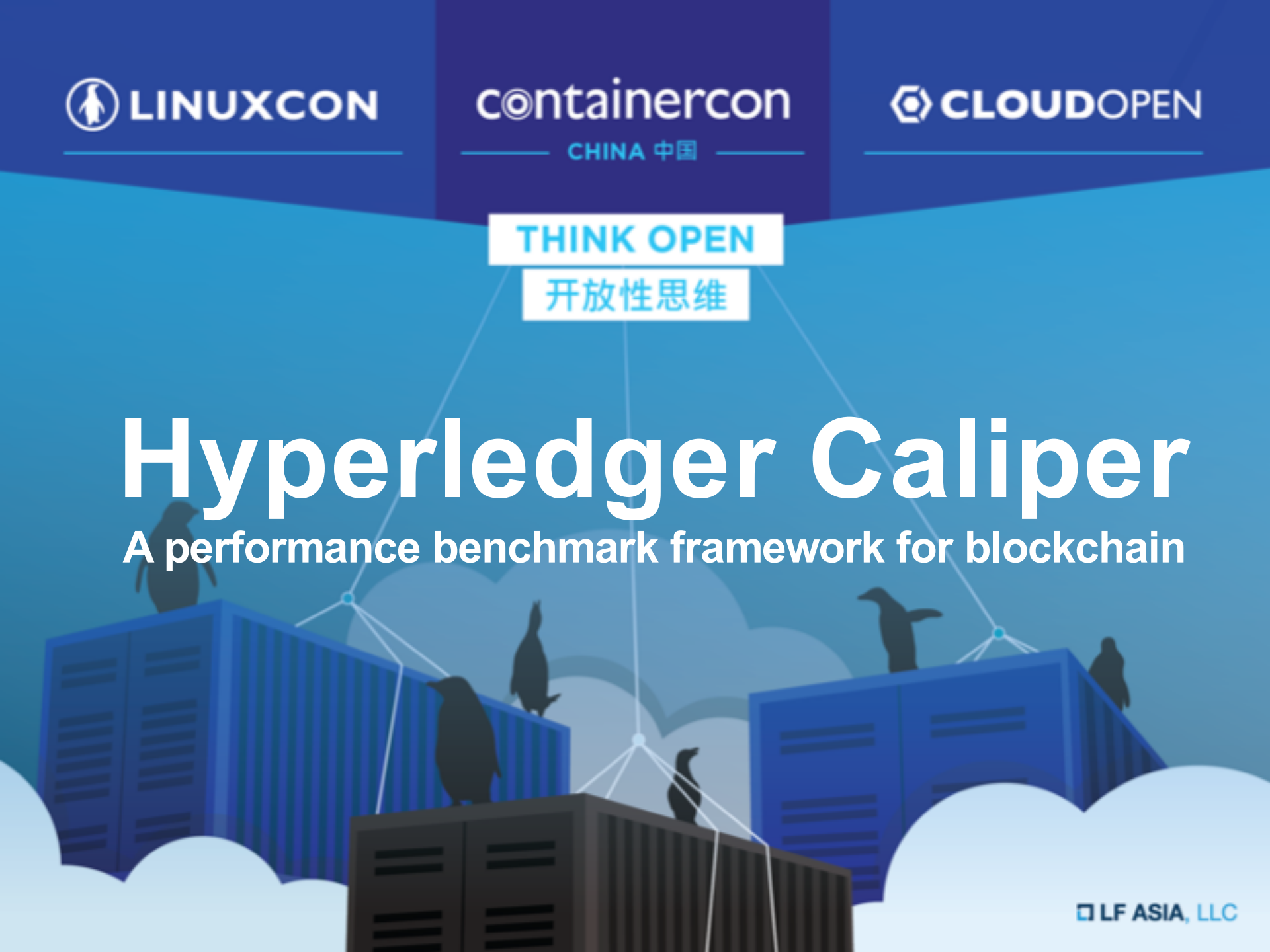
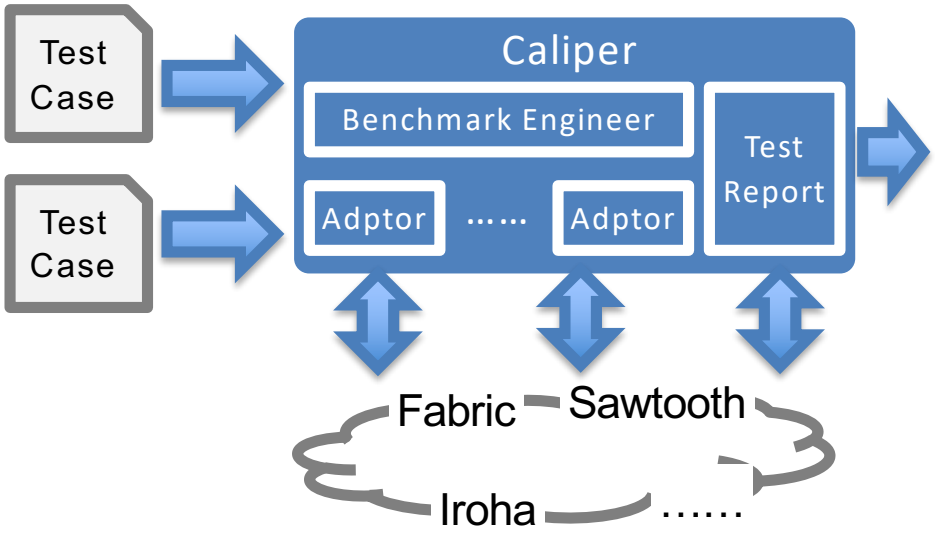

LINUXCON
containercon
CHINA 中国
CLOUDOPEN
THINK OPEN
开放性思维
Hyperledger Caliper
A performance benchmark framework for blockchain
LF ASIA, LLC

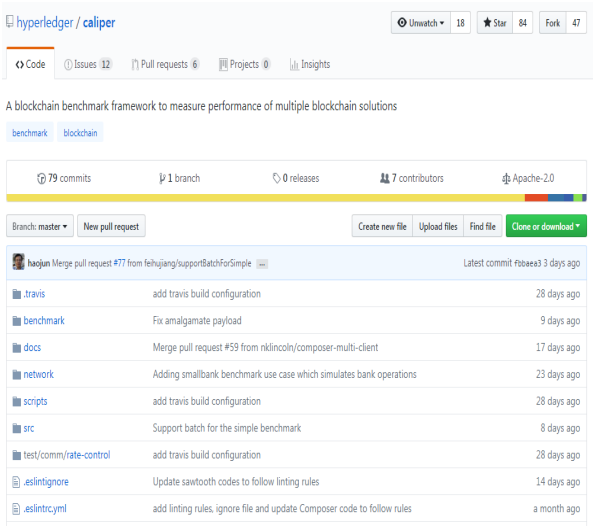
Contents

- What's Caliper
- Architecture & Design
- Roadmap

What's Caliper



Latency
Throughput
Success rate
.....



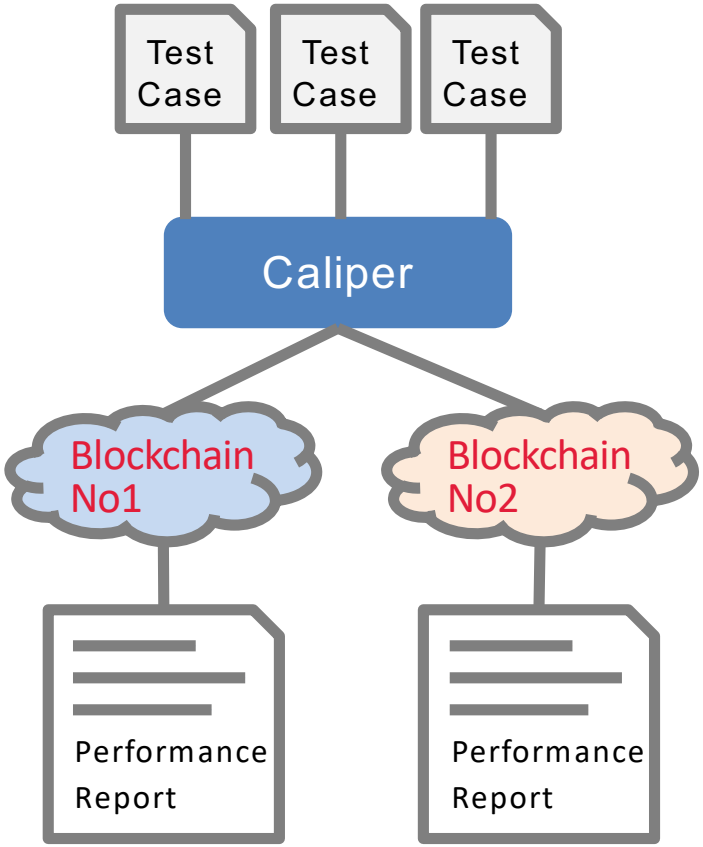
Caliper is a performance benchmark framework for blockchain and one of the Hyperledger projects hosted by The Linux Foundation

- 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

https://github.com/hyperledger/caliper

What's Caliper

Target users and typical scenarios (1)

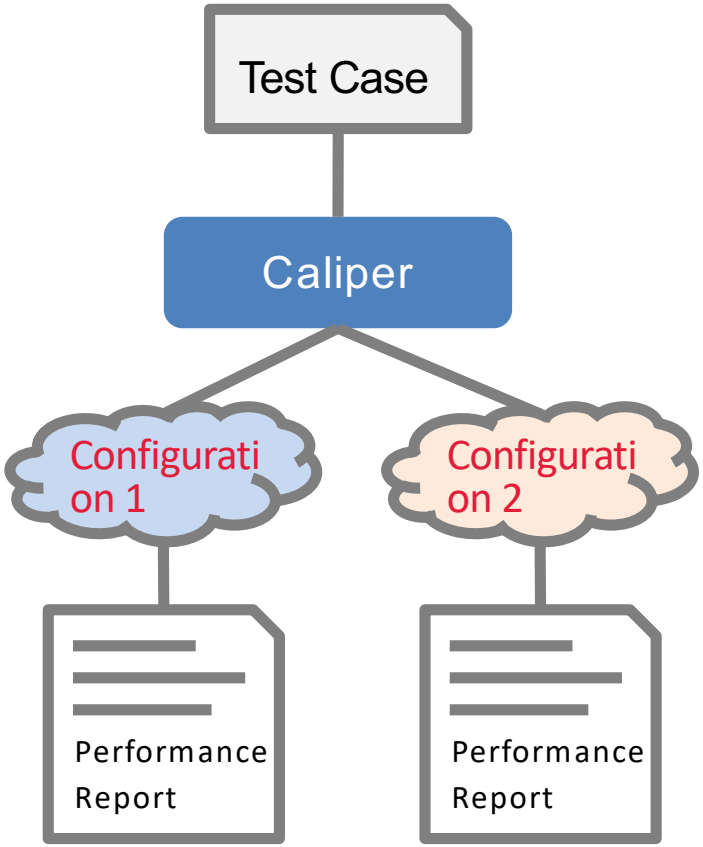


For decision makers who choose blockchain system for their business, Caliper can help:

- Test performance with specific test cases to find out which one best meet their needs
- Guarantee fairness for various systems
- Learn resource (CPU, Memory ,…) requirements and estimate costs to set up the system
- ……

What's Caliper

Target users and typical scenarios (2)

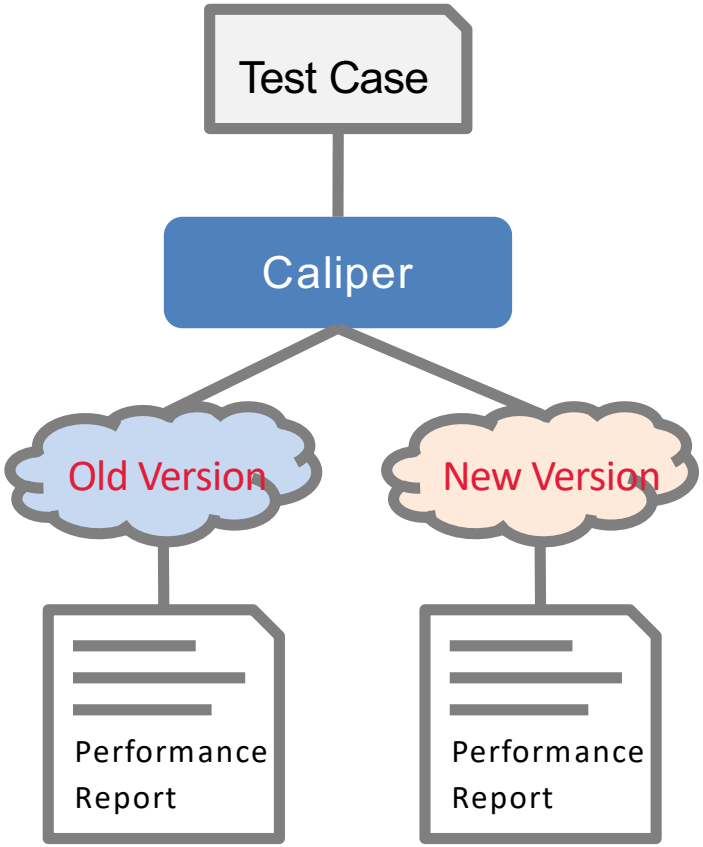


For system operators, Caliper can help:

- Evaluate performance of multiple blockchain configuration schemes and choose the best one
- Learn how network condition would affect the performance
- Find out the hardware requirements for specific SLA
- ……

What's Caliper

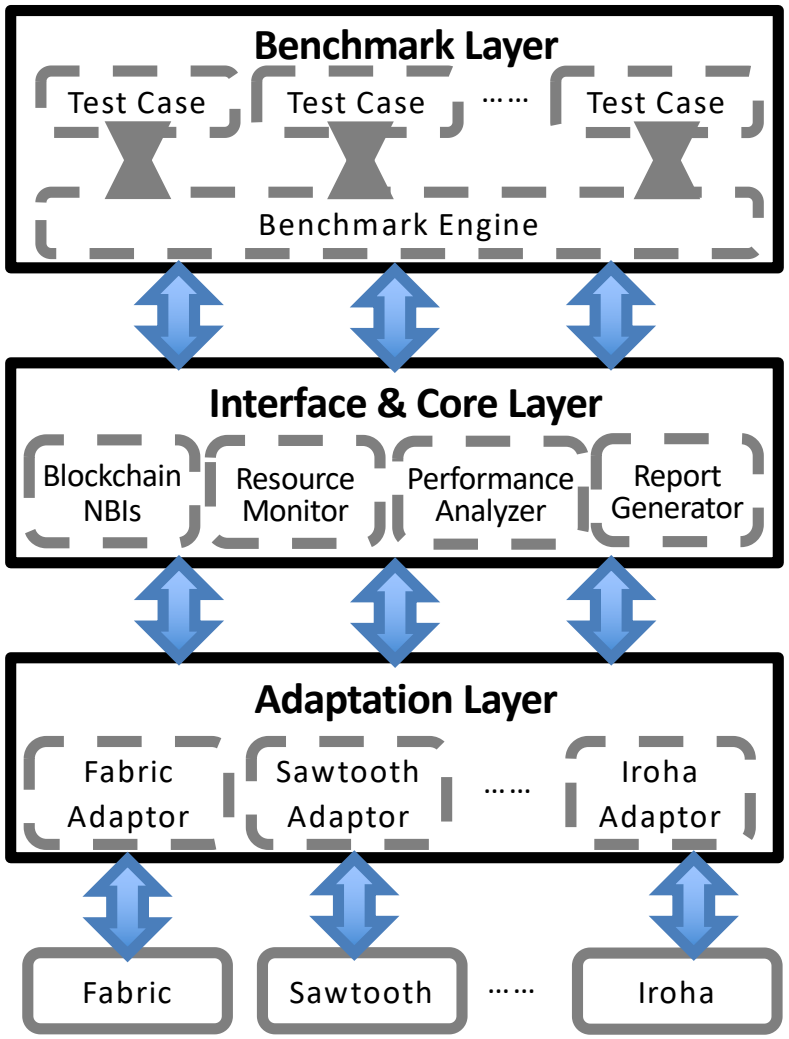
Target users and typical scenarios (3)



For developers, Caliper can be used as an internal tool to :

- Qualify the performance improvement of new version
- Assess the impact of new features on performance
- Compare with other blockchain systems
- ……

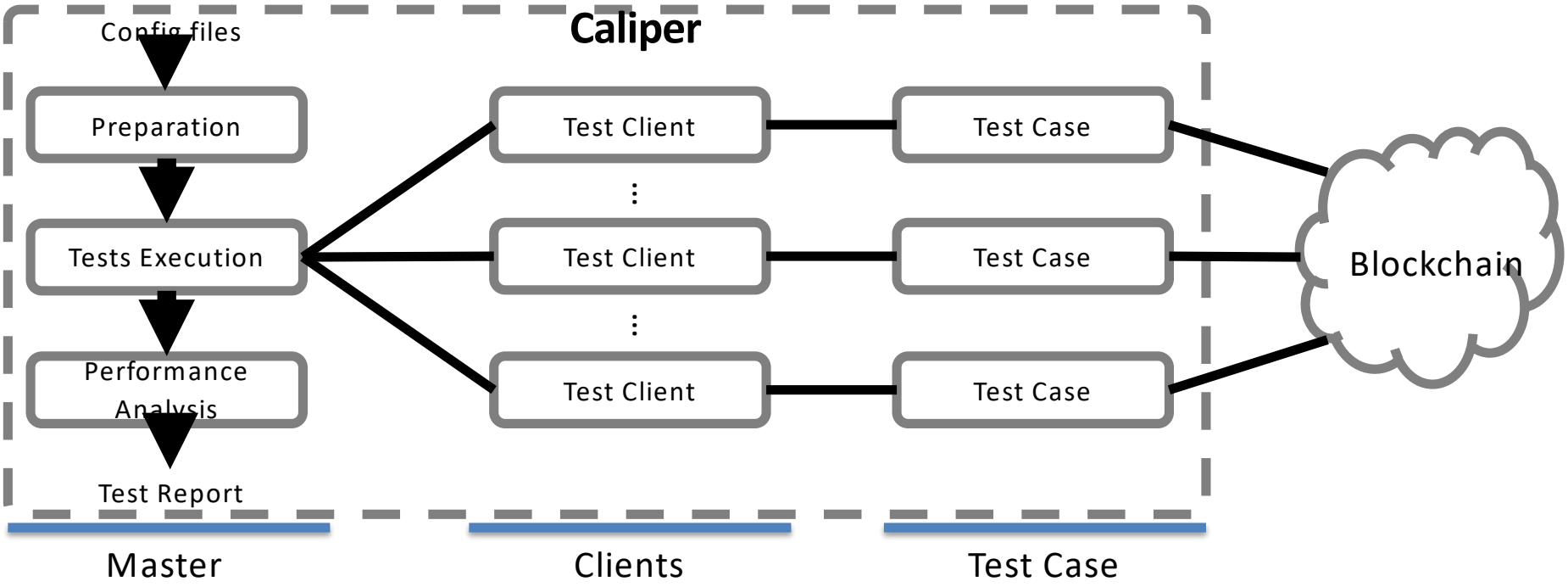
Architecture



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



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

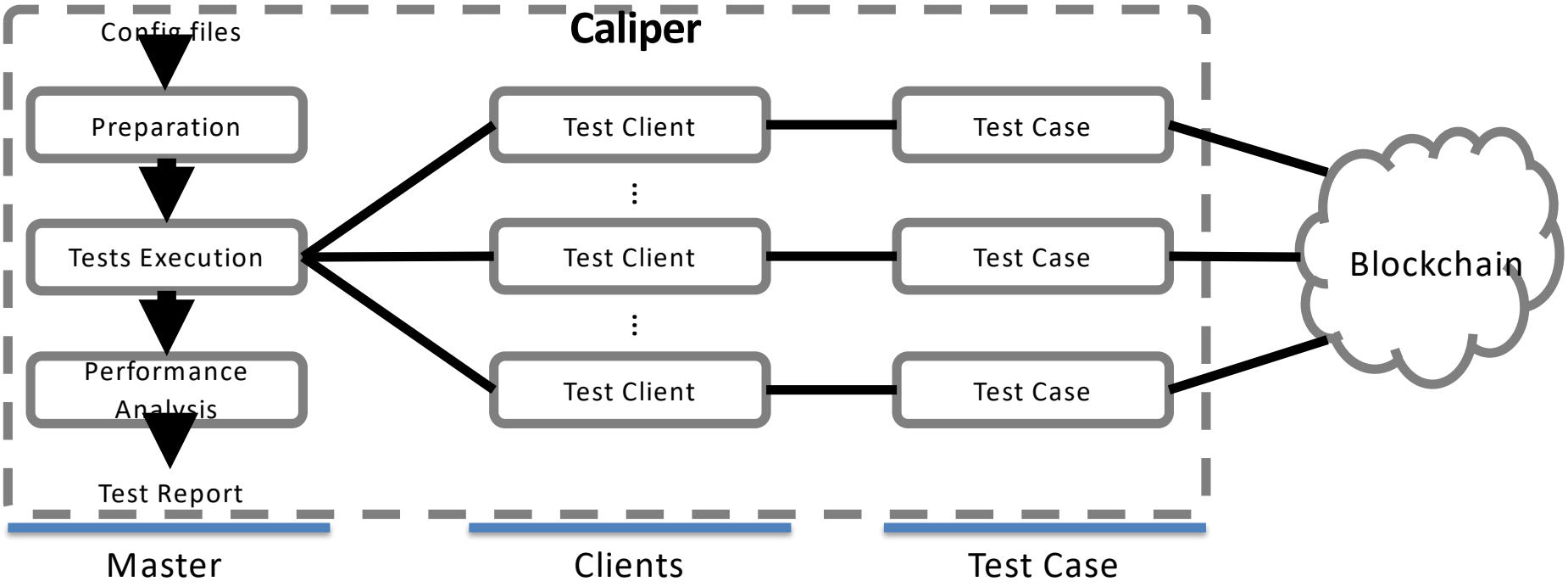
Run test case according to the specific workload

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

Scripts which define interactions with the system under test

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

How it works



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

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
  - ……

Scripts which define interactions with the system under test

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

Example: test & network config

Test configuration file is used to specify test flow and workloads, as well as other global configuration items

```
{
  "blockchain": {
    "type": "fabric",
    "config": "./fabric.json"
  },
  "command" : {
    "start": "docker-compose -f ../../network/fabric/simplenetwork/docker-compose.yaml up -d",
    "end" : "docker-compose -f ../../network/fabric/simplenetwork/docker-compose.yaml down;docker rm
$(docker ps -aq)"
  },
  "test": {
    "clients": {
      "type": "local",
      "number": 5
    },
"rounds": [{
        "label" : "open",
        "txNumber" : [5000, 10000],
        "rateControl" : [ {"type": "fixed-rate", "opts": {"tps" : 200}}, {"type": "fixed-rate", "opts": {"tps" : 300}}],
        "arguments": {  "money": 10000 },
        "callback" : "benchmark/simple/open.js"
      },
      {
        "label" : "query",
        "txNumber" : [5000],
        "rateControl" : [{"type": "fixed-rate", "opts": {"tps" : 300}}],
        "callback" : "benchmark/simple/query.js"
      }]
  },
  "monitor": {
    "type": "docker",
    "docker": { "name": ["peer0.org1.example.com", "http://192.168.1.100:2375/orderer.example.com"]
},
    "interval": 1
  }
}
```

Specify the location of blockchain network configuration file

User defined commands which are called before/after test

Specify type and number of clients used for the test

Specify test rounds

- txNumber: defines an array of sub-rounds with number based test runs
- rateControl: defines how to control the txns submitting
- arguments: user defined arguments which are passed directly to the specified test script
- callback: location of the test script

Resource monitor

- docker: local/remote containers which will be watched

Example: test & network config

Network configuration file is used to specify access points, as well as other necessary informations such as cryptographic materials required to interact with the SUT

```
{
  "fabric": {
    "cryptodir": "network/fabric/simplenetwork/crypto-config",
    "network": {
      "orderer": {
        "url": "grpcs://localhost:7050",
        "mspid": "OrdererMSP",
        "user": {
          "key": "network/fabric/……/keystore/be595….57cd_sk",
          "cert": "network/fabric/……/Admin@example.com-cert.pem"
        },
        "server-hostname": "orderer.example.com",
        "tls_cacerts": "network/fabric/……/tls/ca.crt"
      },
      "org1": {
        "name": "peerOrg1",
        "mspid": "Org1MSP",
        "user": {……},
        "peer1": {……},
        "peer2": {……}
      },
      "org2": {……}
    },
    "channel": [{
        "name": "mychannel",
        "config": "network/fabric/simplenetwork/mychannel.tx",
        "organizations": ["org1", "org2"],
        "deployed": false
    }],
    "chaincodes": [{"id": "simple", "path": "contract/fabric/simple", "language":"golang", "version": "v0",
"channel": "mychannel"}] }
    ……
}
```

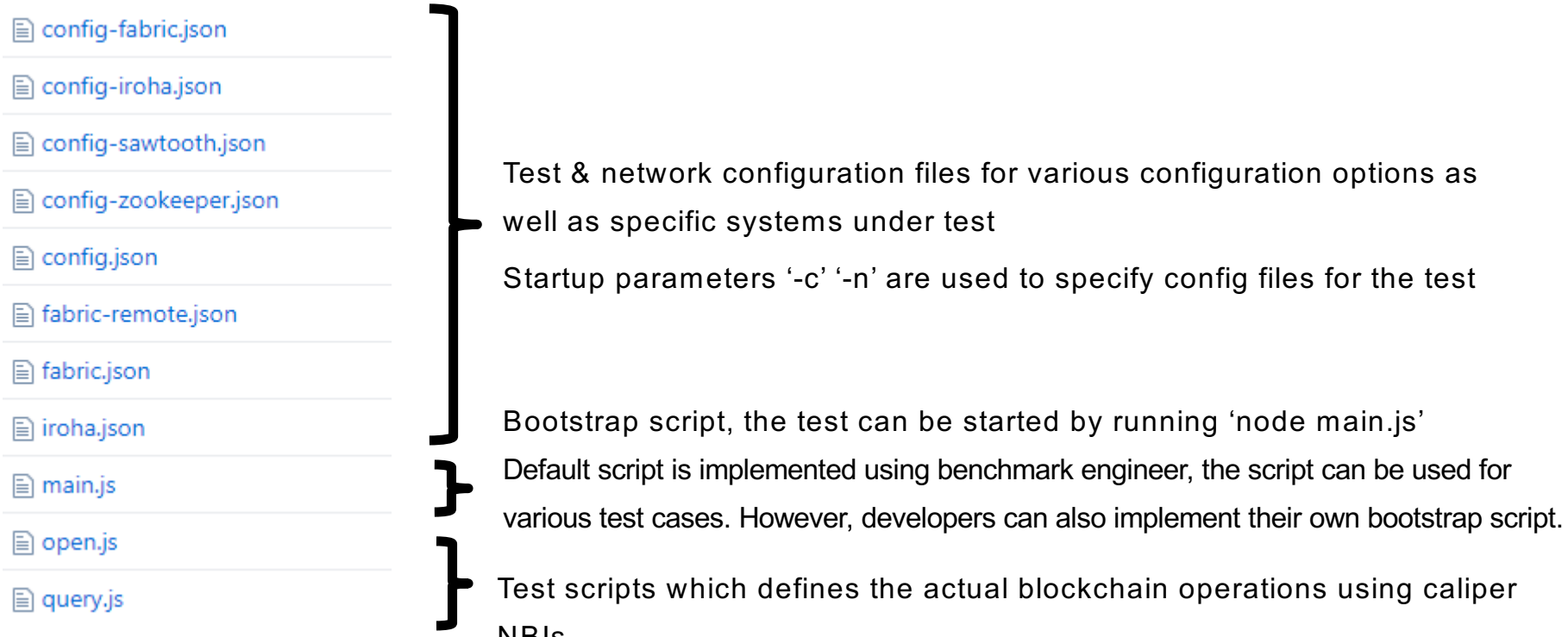
Informations of orderer and peers which can be used to submit transactions to Fabric

Informations of fabric channels

Informations of fabric chaincodes

Example: test case

A typical directory of test case : https://github.com/hyperledger/caliper/tree/master/benchmark/simple



Example: test report

Test results are outputted to the console in real time, and a HTML format report will be generated after the test

Caliper Report

Basic information

DLT: fabric
Benchmark: simple
Description: This is an example benchmark for caliper, to test the backend DLT's performance with simple account opening & querying transactions
Test Rounds: 5
Details

Benchmark results

Summary
round 0
round 1
round 2
round 3
round 4

System Under Test

Version: 1.0.5
Size: 4 Peers
Orderer: Solo
Distribution: Single Host
Details

Summary

Test	Name	Succ	Fail	Send Rate	Max Latency	Min Latency	Avg Latency	75%ile Latency	Throughput
1	open								
2	open								
3	open							1.1	
4	query								
5	query								

round 0 - open

performance metrics

Name	Succ	Fail	Send Rate	Max Latency	Min Latency	Avg Latency	75%ile Latency	Throughput
open								

resource consumption

TYPE	NAME	Memory(max)	Memory(avg)	CPU(max)	CPU(avg)	Traffic In	Traffic Out
Process	node local-client.js(avg)						
Docker	dev-peer1.org2.example.co...le-v0						
Docker	dev-peer0.org2.example.co...le-v0						
Docker	dev-peer1.org1.example.co...le-v0						
Docker	dev-peer0.org1.example.co...le-v0						

PSWG

https://wiki.hyperledger.org/groups/pswg/performance-and-scale-wg

Performance and Scale Working Group is a cross project forum of Hyperledger for architects and technologists to discuss, research, and identify key metrics that relate to the performance and scalability of a blockchain and blockchain related technologies.

Roadmap

- Fabric v1.1 & Sawtooth v1.0 & Iroha
- Performance metrics
  - Success Rate
  - Throughput
  - Latency
- Resource Monitor
  - Docker Container
  - Local Process
- Simple sample test cases

- Other Hyperledger blockchain (and Non-Hyperledger System?)
- Keep up with PSWG metrics
- GUI & Dashboard support
- Add more sophisticated test cases
- Stable version supports long-term & large-scale testing
- Integrate with blockchain & network operation tools

Jun 2018 Dec 2018

Contact

- Join the discussion of performance metrics and benchmark requirements

https://wiki.hyperledger.org/groups/pswg/performance-and-scale-wg

- Submit issues or PRs to contribute directly

https://github.com/Huawei-OSG/caliper

- Any questions / suggestions about Caliper

https://chat.hyperledger.org/channel/caliper

zhouhaojun@huawei.com

huruifeng@huawei.com


LINUXCON
containercon
CLOUDOPEN
CHINA 中国
THINK OPEN
开放性思维
LF ASIA, LLC