TigerBeetle Integration

PI-17 OSS Community Meeting

26 January 2022

Jason Bruwer, Matseliso Thabane

Agenda

- Mission & Overview
- Timeline
- TigerBeetle for Mojaloop (central-ledger)
- Testing
- Initial Findings
- Next steps

- Appendix: extended demo

Mission & Overview

Mission: Make it easier to build the next generation of financial services systems.

TigerBeetle is a distributed financial accounting database built on 3 pillars:

- **Safety**: fault awareness & recovery that supports data quality & high-availability.
- **Performance**: optimised processing & batching that enables 1 million transactions per second on consumer-grade hardware.
- **Experience**: simple data structures & interfaces support the developer experience.

Previous presentation (**30 Jan 2021**): https://www.youtube.com/watch?v=6D7B6v06mBo

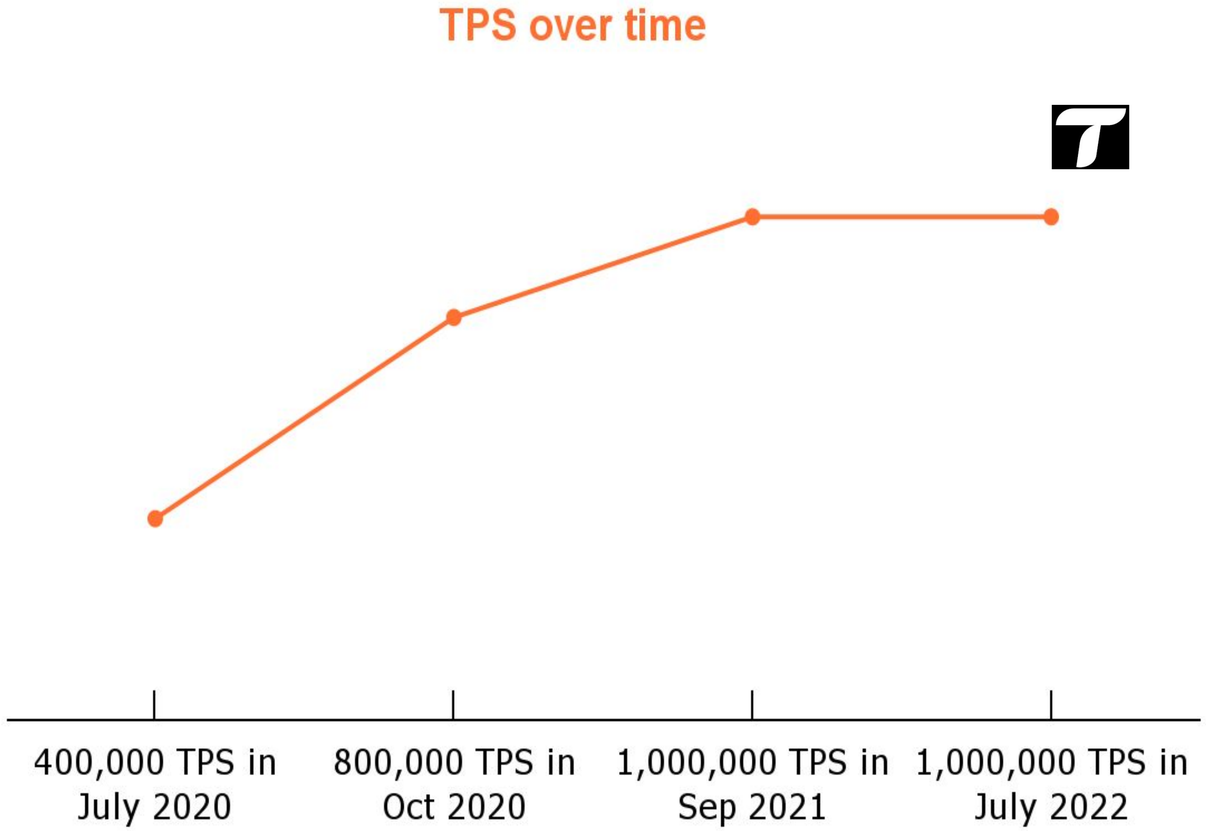
Timeline

Past

- ProtoBeetle
- AlphaBeetle
- BetaBeetle

Today

- Showcase Mojaloop Central-Ledger running on TigerBeetle



TigerBeetle for Mojaloop

Began building TigerBeetle with Mojaloop in mind, as the system of record for Accounts & Balances, to support & amplify performance, availability & safety.

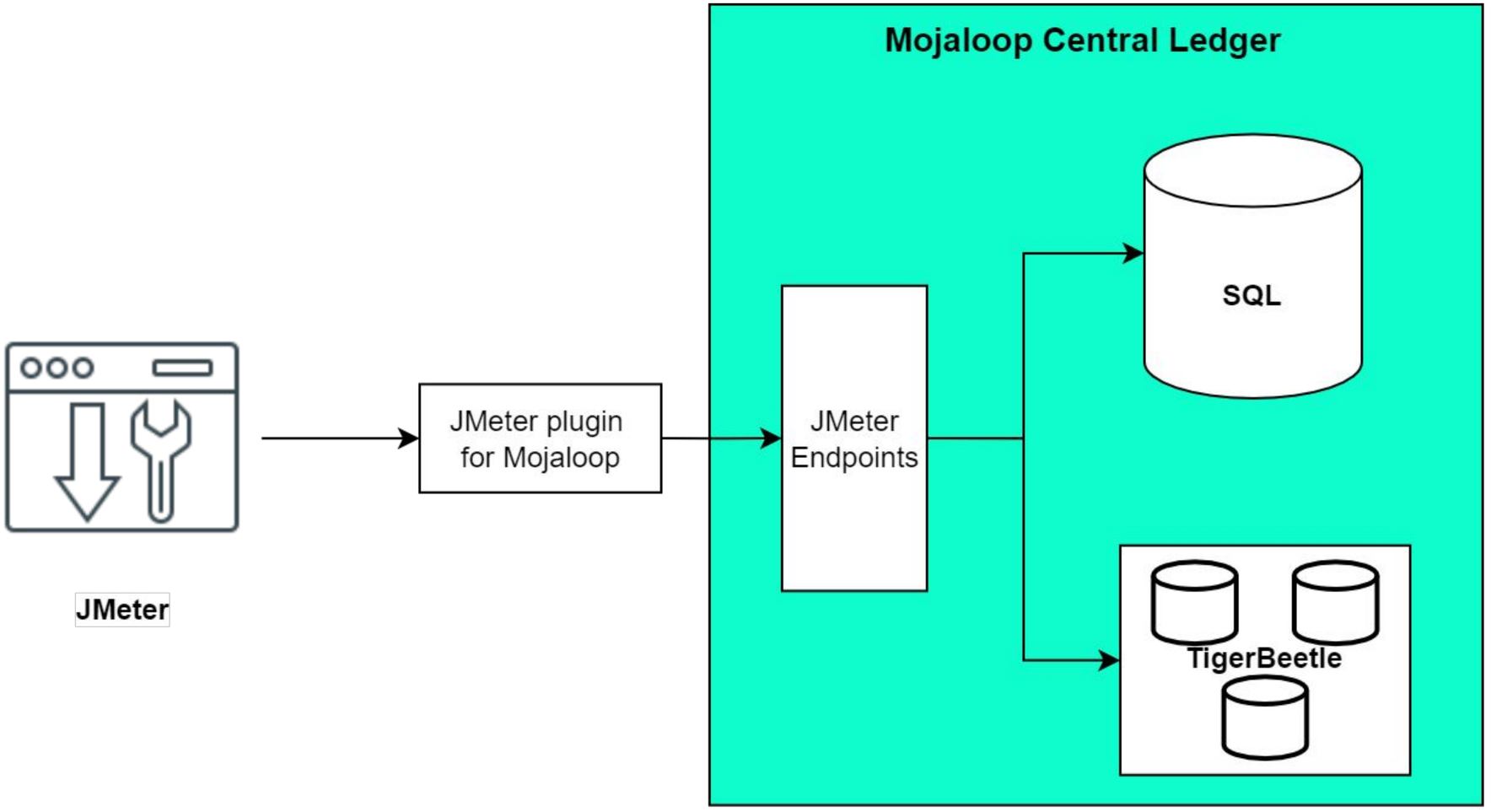
- **Current phase**: Integrate TigerBeetle into Central-Ledger to natively support:
  - native 2-phase commits
  - batch processing
  - accounting processing
  - fault-tolerant processing to alleviate single-points of failure
  - fault awareness & recovery that supports data quality
- **Next phase**: showcase integrating TigerBeetle further into Mojaloop

Testing Approach

Using Apache JMeter for performance, load & stress testing.

- **Artefacts**: The custom developed components for volume testing are:
  - Custom Mojaloop Central-Ledger endpoints.
  - A JMeter plug-in for the Mojaloop Central-Ledger.
- **Benefits**:
  - Reusable components for volume testing, with or without TigerBeetle.
  - Highly customizable input, to simulate real-world scenarios.
  - Enables testing the performance of the software & hardware.

Test Environment



Test Tools

Mojaloop Tooling:

jMeter Tooling:

Configure parameters	Generate test data	Define properties	Execute tests
Criteria for the test data	**Outputs the test data**	**Set test suite properties**	**Run the test suite**
Configure test parameters: - transfer amounts - number of accounts - number of transfers - rejections etc.	Generates test data as output, based on the test parameters. The output can be used in any Mojaloop Central-Ledger environment.	Configure specific properties for this test suite: - number of threads/users - warm-up/delay times - cycles etc.	Initiates the test suite & simulates test scenarios: - stores request & response data - generate HTML report - use JMeter report plugins

Testing Scope

1. Create Account

2. Create Transfer: Two-Phase commit

3. Account Lookup

4. Transfer Lookup

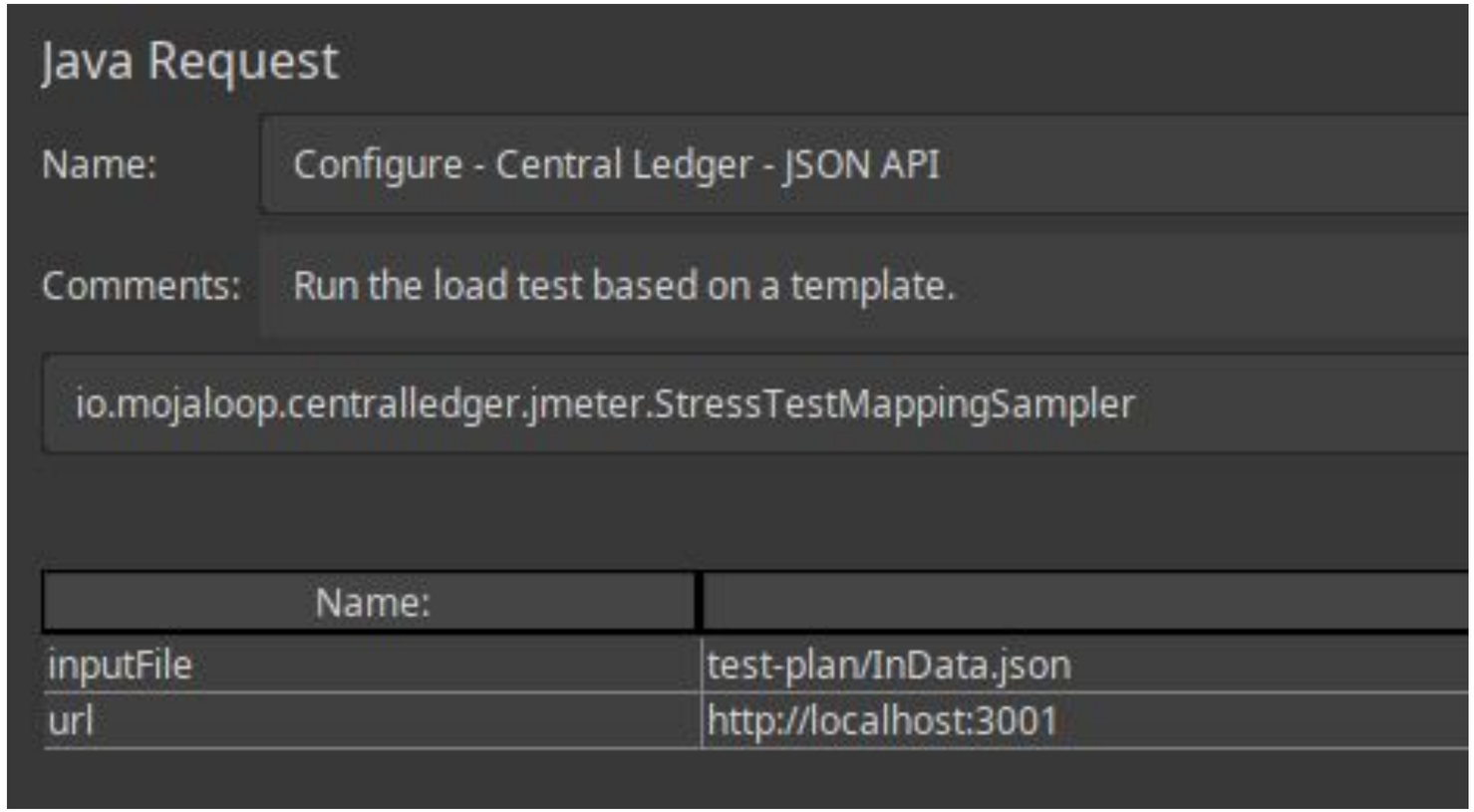
Test Configuration

Configure CL+TB - `default.json`:

```
"INTERNAL_TRANSFER_VALIDITY_SECONDS": "432000",
"ENABLE_ON_US_TRANSFERS": false,
"TIGERBEETLE" : {
  "ENABLED" : false,
  "ENABLE_BATCHING" : false,
  "CLUSTER" : 1,
  "REPLICA_ENDPOINT_01" : "5001",
  "REPLICA_ENDPOINT_02" : "5002",
  "REPLICA_ENDPOINT_03" : "5003"
},
"CACHE": {
  "CACHE_ENABLED": false,
```

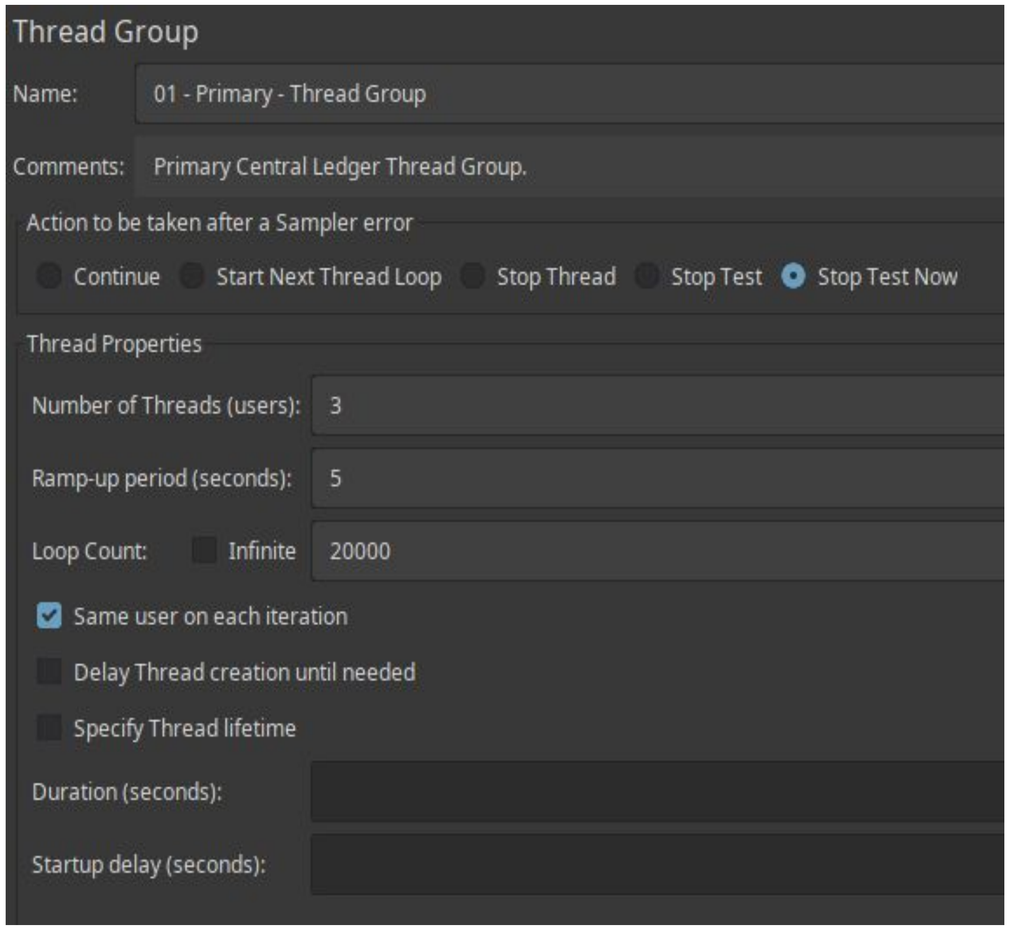
Test Configuration

Configure JMeter plugin:



Test Configuration

Set behaviour properties:



Generate Test Data

Configure Test Data *(criteria for test data)*:

Generate Test Data

Central-Ledger Endpoints:

method	path	description
GET	/enums	
GET	/health	
GET	/jmeter/participants/{name}	[jMeter] API used for retrieving a MJL participant accounts and balances by id.
GET	/jmeter/participants/{name}/transfers/{id}	[jMeter] API used for retrieving a MJL transaction by id.
POST	/jmeter/participants/create	[jMeter] API used for creating participant accounts.
GET	/jmeter/transactions/ilp/{id}	[jMeter] API used for retrieving a ILP transaction by id.
POST	/jmeter/transfers/prepare	[jMeter] API used for preparing a 2-phase transfer (optional fulfill may be provided).

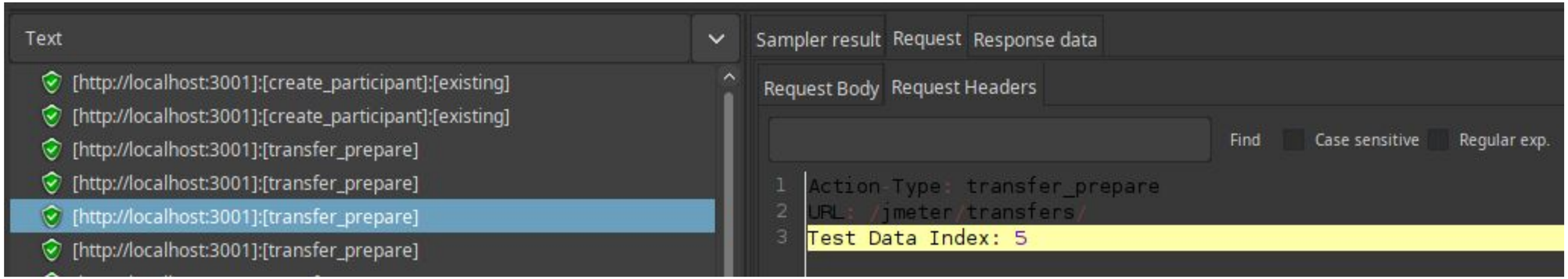
Test Execution

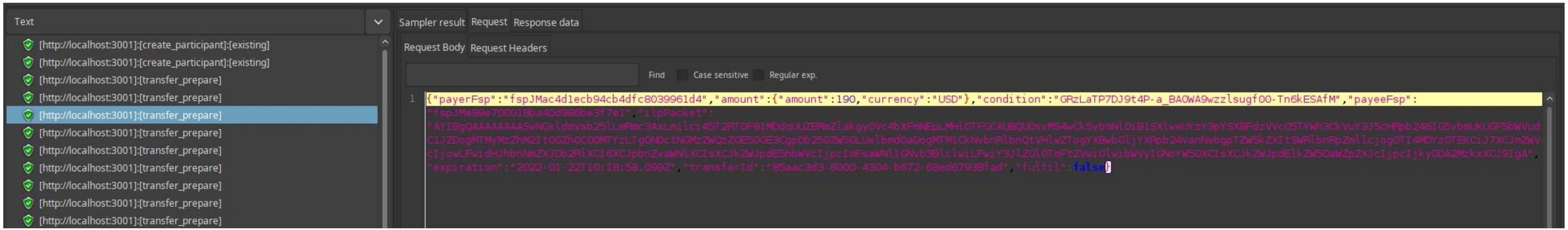
The generated test data (used as REST body):


```
{
  "actionType": "create_participant",
  "request": {
    "name": "fspJMac4d1ecb94cb4dfc8039961d4",
    "currency": "USD",
    "newlyCreated": false
  },
  "response": null
},
{
  "actionType": "transfer_prepare",
  "request": {
    "payerFsp": "fspJMe99e700018ba40d986be3f7e1",
    "amount": {
      "amount": 1,
      "currency": "USD"
    },
    "condition": "GRzLaTP7DJ9t4P-a_BA0WA9wzzlsugf00-Tn6kESAfM",
    "payeeFsp": "fspJMac4d1ecb94cb4dfc8039961d4",
    "ilpPacket": "AYIBgQAAAAAAAASwNGxldmVsb25lLmRmc3AxLm1lci45T2RT
    "expiration": null,
    "transferId": "217dbf8a-70e0-499f-84cd-7521df7ad6fa",
```

Test Execution

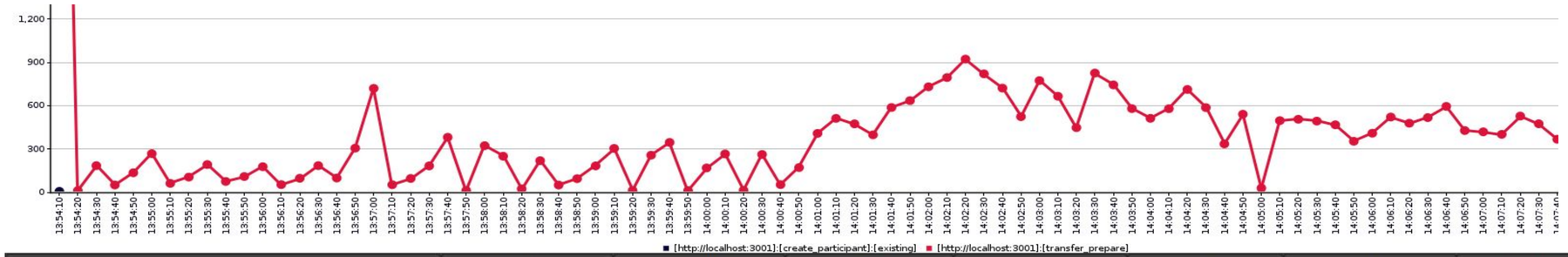
View detailed JSON request data during jMeter execution:





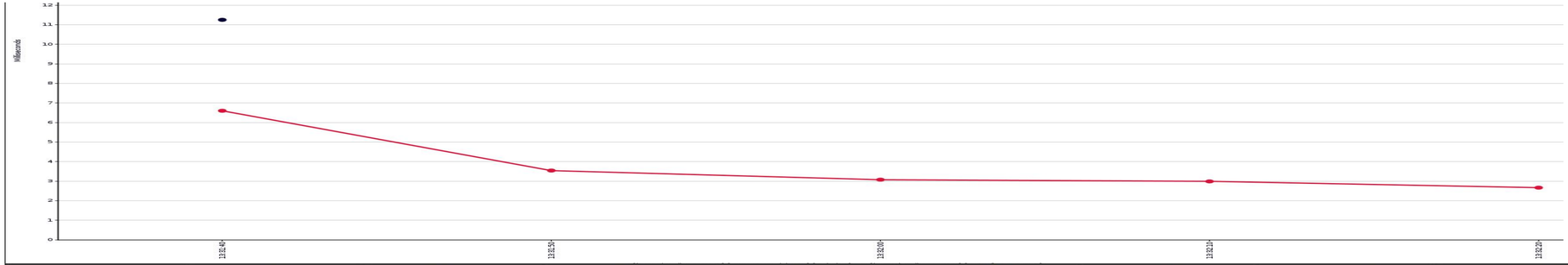
Initial Findings

Without TigerBeetle - 175ms round trip @ 17tps with 18s max:



Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
[http://localhost:3001]:[create_participant]:[existing]	6	6	4	10	2.19	0.00%	400.0/sec	30.86	58.59	79.0
[http://localhost:3001]:[transfer_prepare]	16858	175	4	18649	1525.39	0.00%	17.0/sec	13.75	14.53	828.5
TOTAL	16864	175	4	18649	1525.12	0.00%	17.0/sec	13.75	14.53	828.2

With TigerBeetle - 3ms round trip @ 1102tps with 80ms max: *(n*64)*



Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
[http://localhost:3001]:[create...	12	11	8	15	2.05	0.00%	375.0/sec	28.93	54.93	79.0
[http://localhost:3001]:[transf...	59988	3	1	80	2.54	0.00%	1102.7/sec	892.11	942.72	828.5
TOTAL	60000	3	1	80	2.54	0.00%	1102.5/sec	891.82	942.43	828.3

Next steps

- **Production release of TigerBeetle**: by July 2022
- **Release JMeter test suite for Central Ledger**: by June 2022
- **Next Demo**: showcase Prod. release integration in July 2022
- **Review**: assess possible suitability for Mojaloop v.Next

Appendix: Extended Demo Video

https://www.youtube.com/channel/UC3TIyQ3h6IC_jSWust2leGg/videos

Thank you.