# Declarative workload behavior definition for Hyperledger Caliper

| | |
|---|---|
| **Project Title** | Declarative workload behavior definition for Hyperledger Caliper |
| **Status** | IN PROGRESS |
| **Difficulty** | MEDIUM |

# Description

Performance benchmarking distributed systems (such as DLTs) is a challenging task, comprising of multiple aspects, such as scalable workload generation, representative workload definitions, and comprehensive data analysis.

Hyperledger Caliper is a general-purpose benchmarking tool with the goal of mitigating the aforementioned aspects of performance benchmarking:

1. It provides a flexible architecture to allow scalable workload generation.
2. It collects and reports results based on detailed client-observable execution traces.
3. **It allows the implementation/plug-in of custom workload behaviors to meet the diverse criteria of a wide range of business scenarios.**

Implementing a workload module for Caliper currently has some limitations and usability obstacles:

1. As of now, Caliper is built using Node.JS/JavaScript, which also means that workload modules must be implemented using JS, adhering to a predefined interface.
   Caliper worker services (which perform the workload generation) could be implemented in different languages in the future, which raises the question of **workload module migration efforts**.
2. Even though a workload module behavior can be customized to a degree using a benchmark configuration file, the high-level orchestration of workload items is still typically done in code.
   **Lowering the amount of required workload module code (or completely eliminating it for common scenarios) would greatly lower the barrier of entry for Caliper users and aid adoption.**

The goal of the project is to provide means for defining complex workload behaviors in Caliper in a declarative manner, using solely/mostly configuration files.

This should be accomplished with a general built-in workload module implementation that acts according to a configuration-based workload specification.

The exact steps of the project are detailed in the Expected Outcome section.

# Additional Information

- Caliper GitHub repository: https://github.com/hyperledger/caliper
- Caliper example benchmarks GitHub repository: https://github.com/hyperledger/caliper-benchmarks
- Caliper documentation page: https://hyperledger.github.io/caliper/
  - Architecture documentation: https://hyperledger.github.io/caliper/vNext/architecture/
  - Benchmark configuration documentation: https://hyperledger.github.io/caliper/vNext/bench-config/
  - Workload module documentation: https://hyperledger.github.io/caliper/vNext/workload-module/
- Caliper Rocket.Chat channels (LFID needed for login)
  - General: https://chat.hyperledger.org/channel/caliper
  - Contributors: https://chat.hyperledger.org/channel/caliper-contributors

# Learning Objectives

- Mastering the open-source workflow
  - Issues, pull requests, development branches, repository management
- Being part of an open community, navigating and utilizing different forums
- Working in a team
- Covering a wide spectrum of software development skills
  - Conducting unsupervised research for state-of-the-art solutions
  - Writing detailed, consistent specifications
  - Designing/implementing functionally rich, testable, maintainable software components
  - Testing methodologies for rigorous testing
  - Documentation and presentation skills
    - Developer documentation
    - User documentation
    - To-the-point, coherent presentation

# Expected Outcome

The mentee must complete the following tasks by the end of the internship:

1. Survey the workload definition capabilities of relevant workload generators/benchmark tools and typical workload requirements of relevant standards or existing workloads.
2. Specify/design a flexible and extendable YAML-based configuration schema for the declarative definition of complex workloads.
3. Implement a built-in Caliper workload module capable of generating requests based on the above configuration.
4. Thoroughly test the implemented module.
5. Provide developer and user documentation for the schema and the implemented module.
6. Port some of the Caliper microbenchmarks to the declarative schema as proof of concept.

# Relation to Hyperledger

Hyperledger Caliper: specification and implementation of an "extend-only" feature (meaning no blocking dependency on parallel implementation efforts).

# Education Level

At least ongoing B.Sc. studies in software engineering required.

# Skills

Required skills:

- Basic understanding of version control and git
- Experience with JavaScript programming (in Node.JS context)
- Intermediate verbal and writing skills in English
- High-level understanding of Hyperledger Caliper components and the benchmark workflow

Nice-to-have skills (the mentee can learn these during the internship):

- Advanced git usage (upstream repositories, branching, rebasing, etc)
- Experience in unit testing and JS-related unit testing frameworks (chai, sinon, mockery)
- Writing documentation in markdown
- Capability for unsupervised learning and research

# Future plans

The project is an important stepping stone towards the long-term Caliper goals of minimizing the effort for defining workloads and maximizing workload portability across different target platforms.

The project is the first proof-of-concept towards a unified benchmarking experience across target platforms.

# Preferred Hours and Length of Internship

Full-time (40 hours a week for 12 weeks during the summer) or Part-time (20 hours a week for 24 weeks starting in summer and ending in fall)

# Mentor(s) Names and Contact Info

Attila Klenik

Budapest University of Technology and Economics, Critical Systems Research Group
klenik@mit.bme.hu
Rocket.Chat handle: @klenik

## Mentee

Aastha Bist

GitHub: github.com/bistaastha

LinkedIn: linkedin.com/in/bistaastha/

## Project result

- Value Providers - https://github.com/hyperledger/caliper/tree/main/packages/caliper-core/lib/worker/workload/declarative/value-providers

## Final Report