FED4FIRE
Federation for FIRE Plus

BICEPS Benchmarking

Experiment description

Background and motivation

- Storing Efficiently Our Software Heritage
- APIs to efficiently store and retrieve billions small objects
- Bulk ingestion/mirroring
- Cost effective
- Scale out

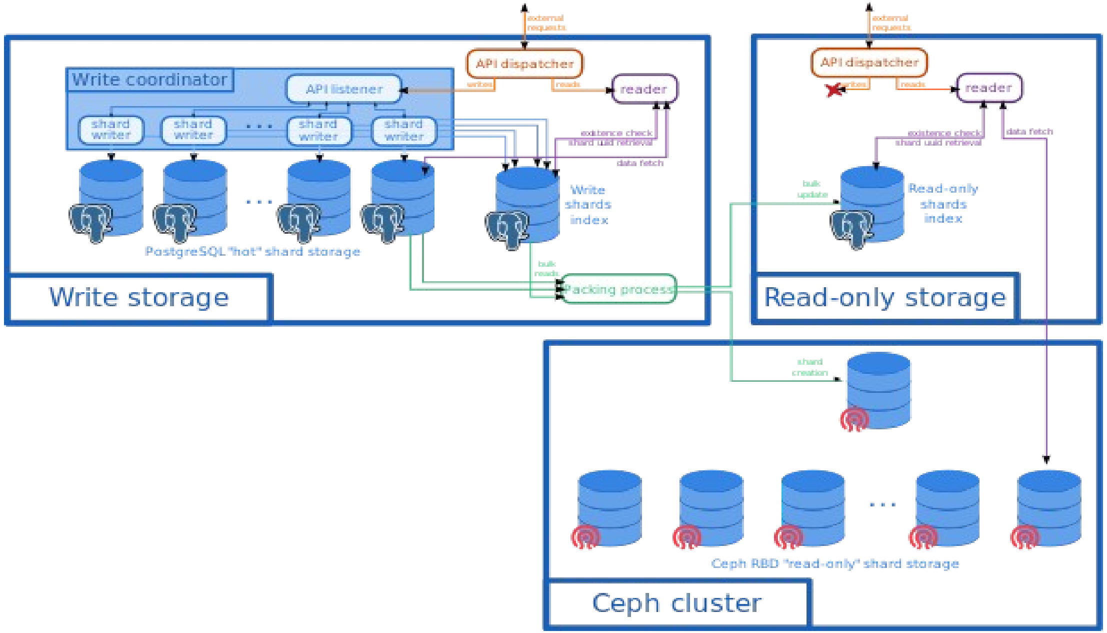
Experiment description

Concept and objectives

- **Interprocess or interthread lock contention**
- **Excessive space storage amplification**
- **Failure to scale out**
- **Ineffective IO throttling and read/write performance degradation**

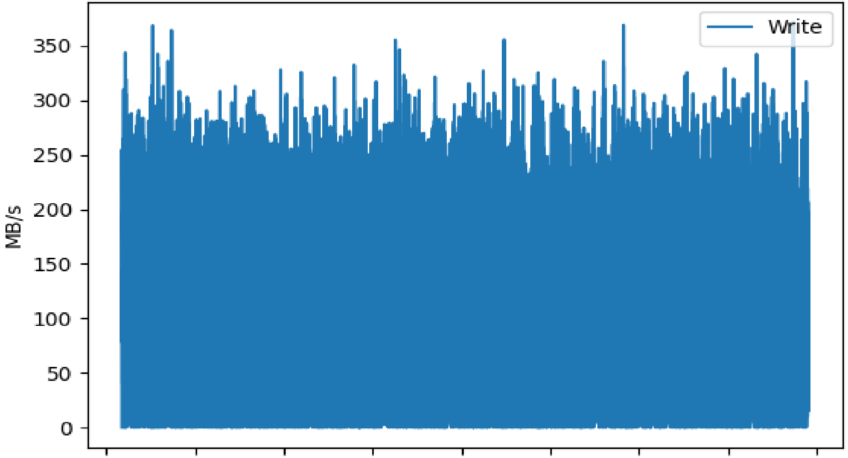
Experiment description

Experiment setup



Project results

Measurements



Project results

Lesson learned

It is **time consuming to use the native grid5000** tooling to prepare the experiment.

These **benchmarks need to be repeated every time the software implementation changes** to ensure performances are not degraded.

Project results

Lesson learned

- download the zip file wit the credentials **Fed4Fire+ jFed** client,
- configure hardware single **Ansible** command line
- run the experiment and collect the results using **tox**

Business impact

Impact on your business

- Testing a theoritcal solution to **verify the performances**
- demonstrated the **object storage behavior at scale**
- **allowing anyone to reproduce the results**

Business impact

Value perceived

- Access to test clusters
- Shorter development cycle
- **More reliable software**

Feedback

Added value of Fed4Fire

- Automation
- Portability
- Future proof

FED4FIRE

BICEPS BENCHMARKING

WWW.FED4FIRE.EU

Co-funded by the European Union

Co-funded by the Swiss Confederation

This project has received funding from the European Union’s Horizon 2020 research and innovation programme, which is co-funded by the European Commission and the Swiss State Secretariat for Education, Research and Innovation, under grant agreement No 732638.