A Benchmark Suite for Serverless Computing

AWS Lambda - Microsoft Azure Functions -
Google Cloud Functions - IBM Cloud Functions

Master Thesis - Final Presentation

Pascal Maissen
pascal.maissen@unifr.ch

March 2, 2020

Swiss Joint Master of Science in Computer Science, University of Neuchâtel

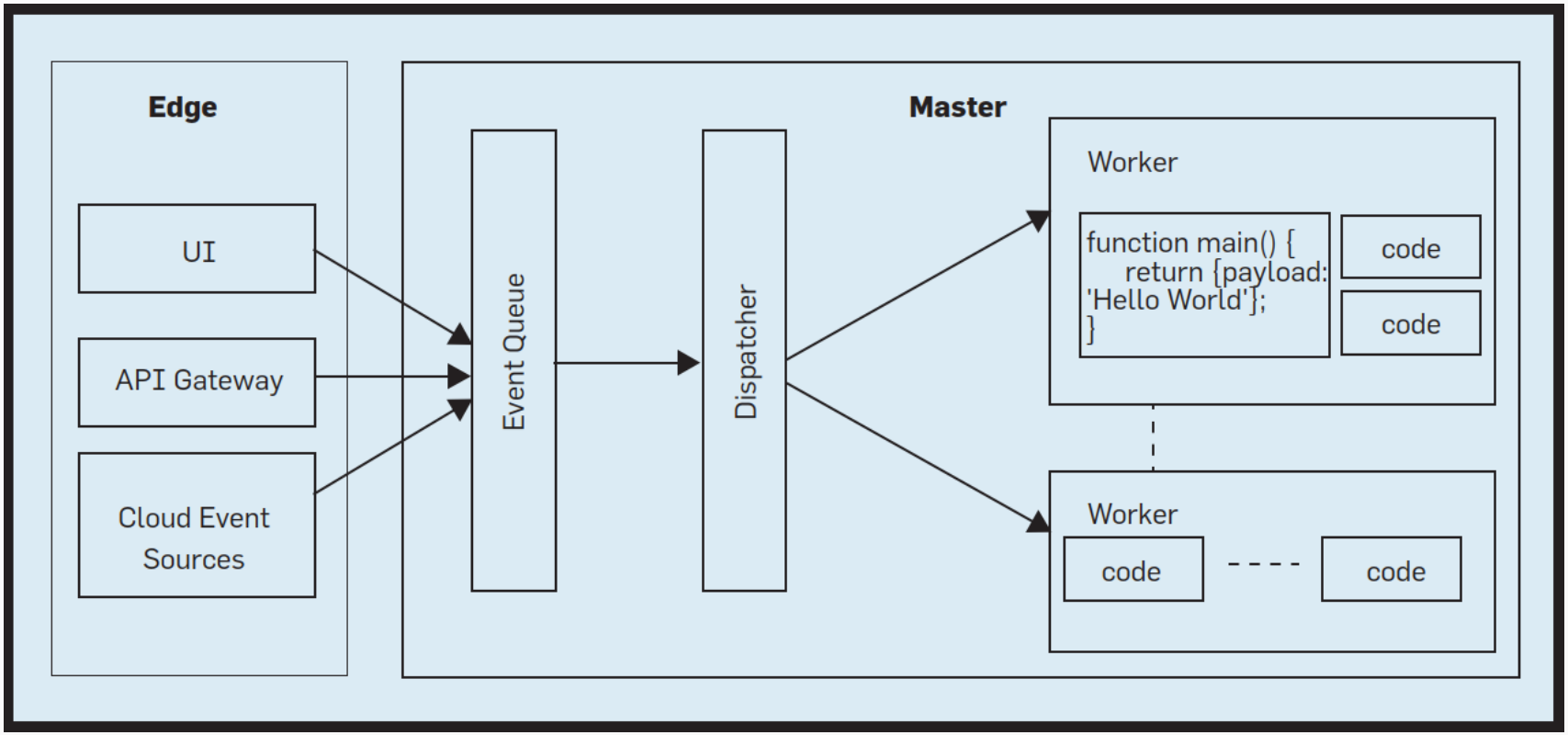
Table of contents

Problem Description and Motivation

- Trend towards cloud computing
- Many (smaller) businesses are not interested in managing and maintaining hardware, VMs or operating systems
- Solution: Serverless computing
- Focus on Function as a Service (Faas)
- Not much effort so far to test, benchmark and compare offered platforms

FaaS - Overview

High-level serverless FaaS platform architecture



Source: Castro, Paul ; Ishakian, Vatche ; Muthusamy, Vinod ; Slominski, Aleksander: The Rise of Serverless Computing. In: Commun. ACM 62 (2019), November, Nr. 12, 44–54. http://dx.doi.org/10.1145/3368454. – DOI 10.1145/3368454. – ISSN 0001–0782

FaaS - Characteristics

- Stateless
- Automatic scaling
- No hardware and server management
- Paid per usage
- Usage: Application backends and data processing

FaaS - Challenges

- Performance
- Scalability
- Usability
- Pricing
- Limitations

Proposed Solution

Benchmark suite for serverless computing

- Packaged with Docker
- Easy to use

4 Clouds:

- AWS Lambda
- Azure Functions
- Google Functions
- IBM Functions

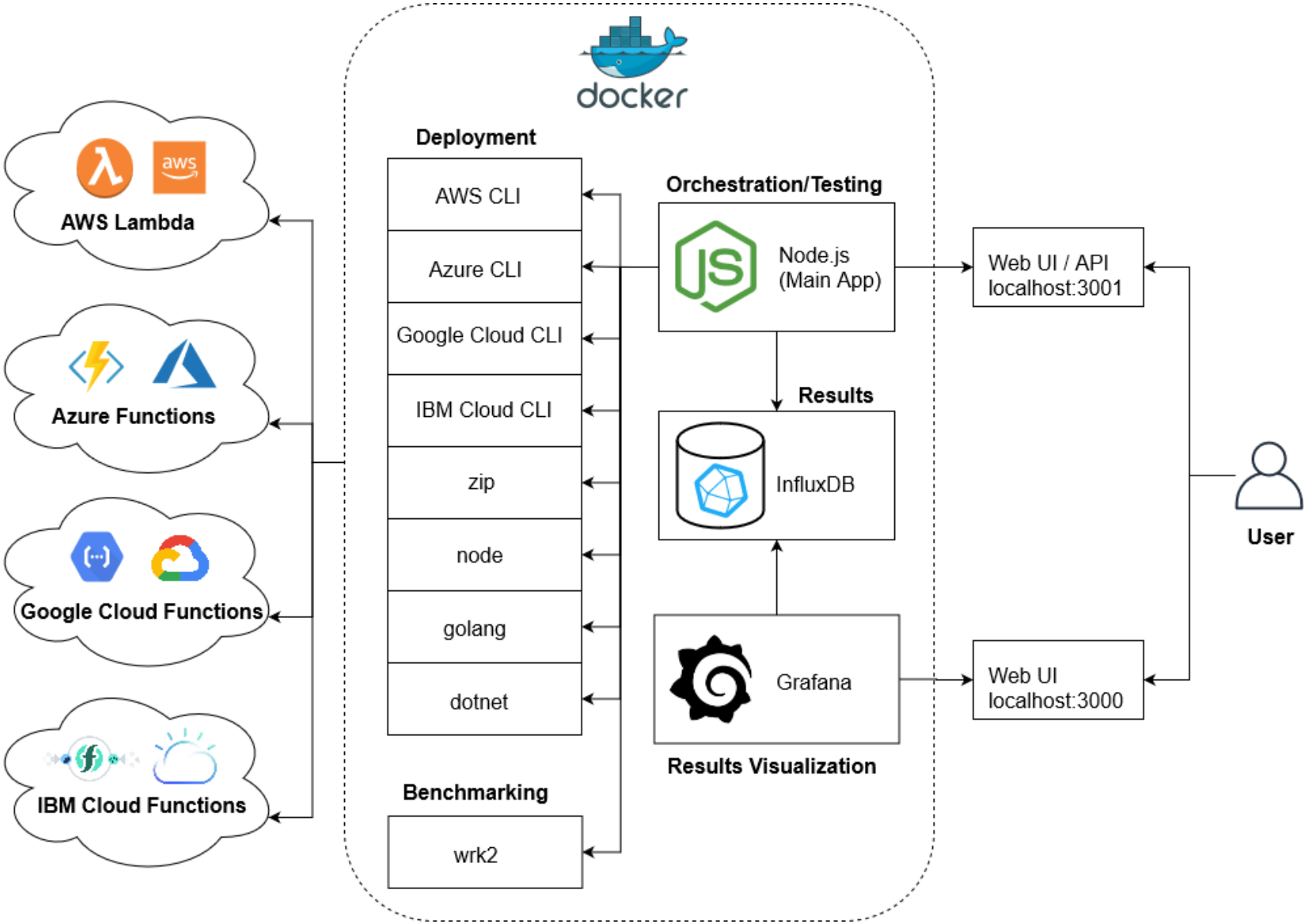
4 Runtimes:

- Node.js (JavaScript)
- Python
- Go
- .NET Core (C#)

5 Tests:

- Latency
- Factors (CPU)
- Matrix (CPU)
- File system
- Custom

Implementation - Overview



Implementation - Functionality

Main application:

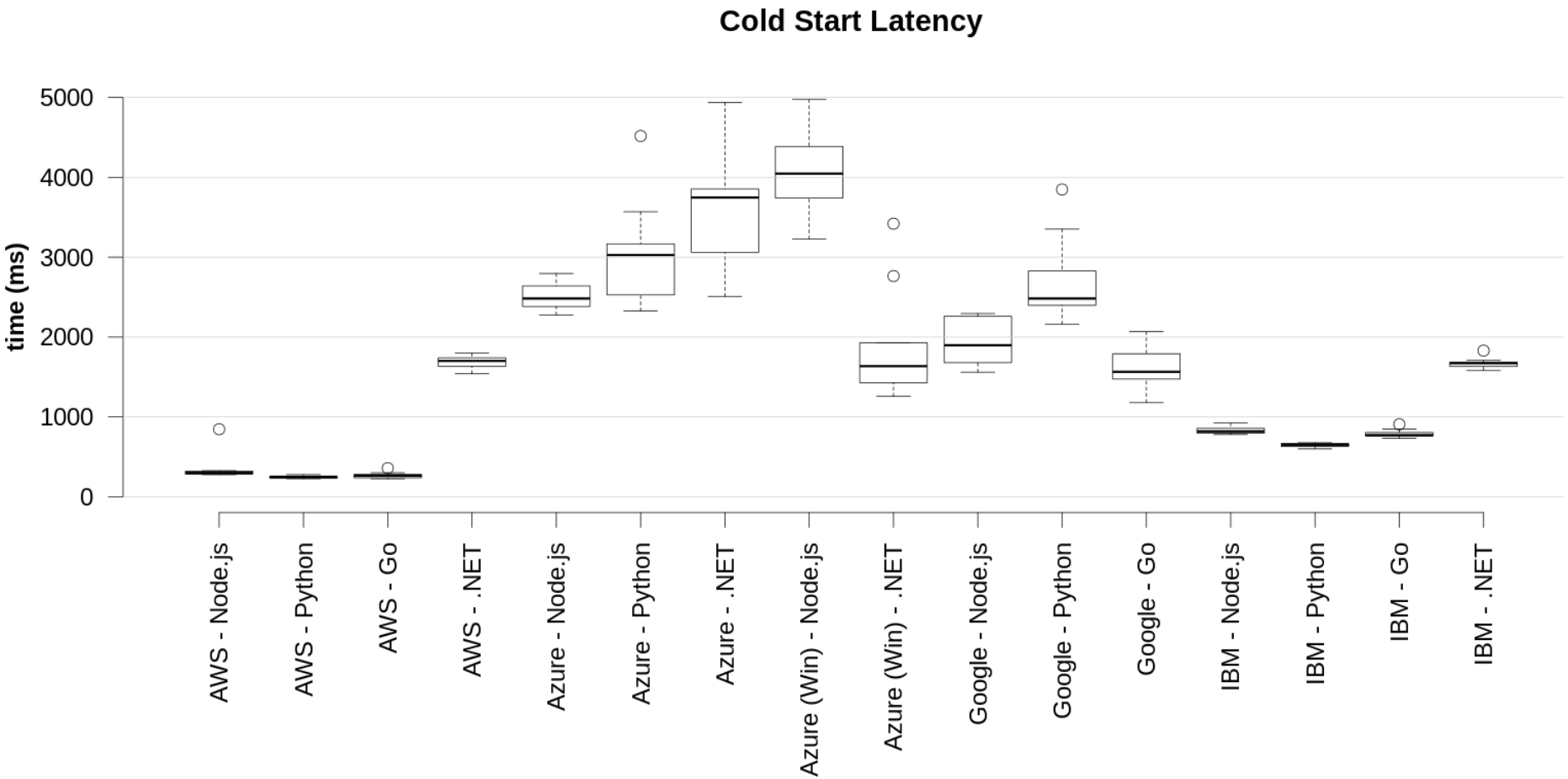
- Deployment and cleanup of tests
- Testing: Observer general performance and handling
- Benchmarking: Load test a function
- Pricing: Theoretical and practical

Results:

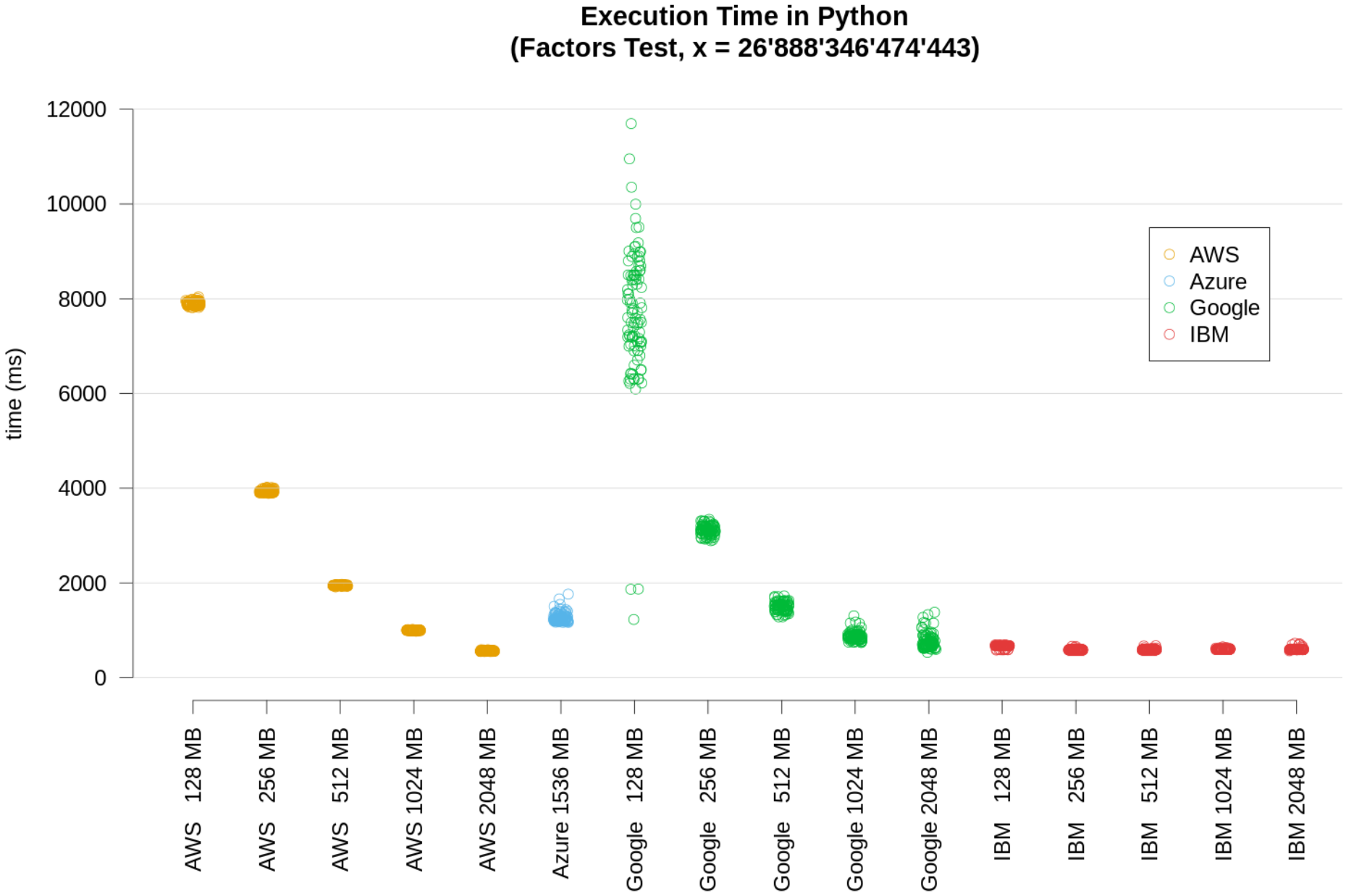
- Stored into InfluxDB
- Visualization in Grafana

Demo

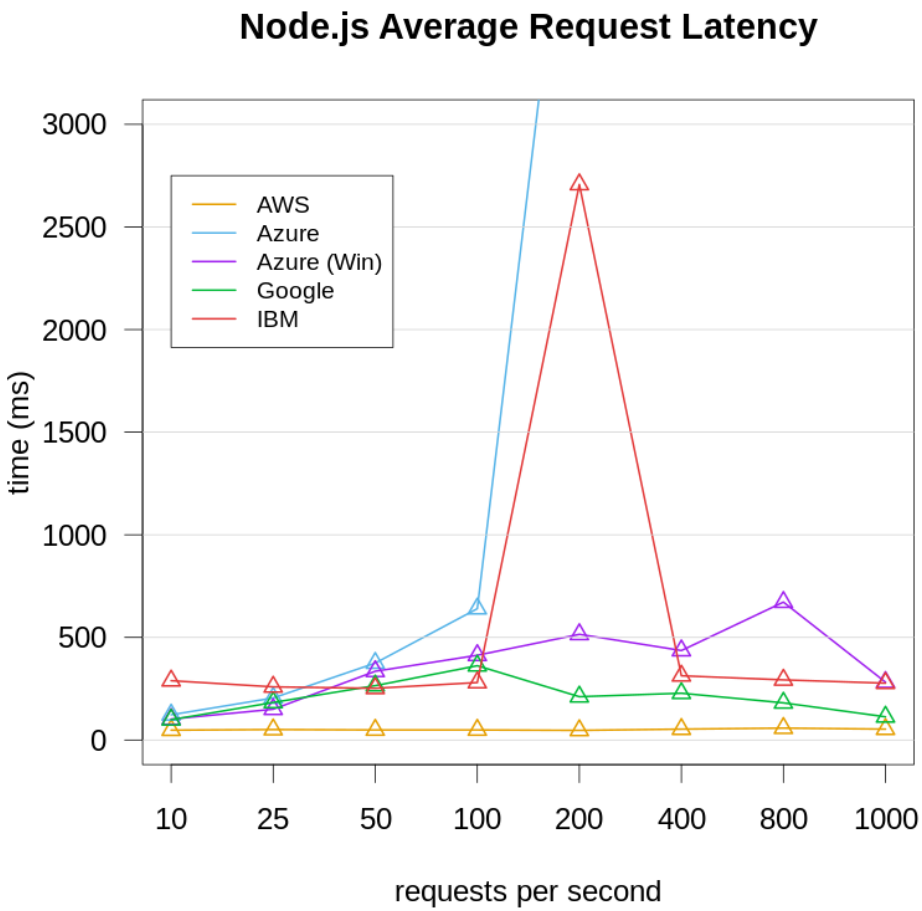
Results - Cold Start

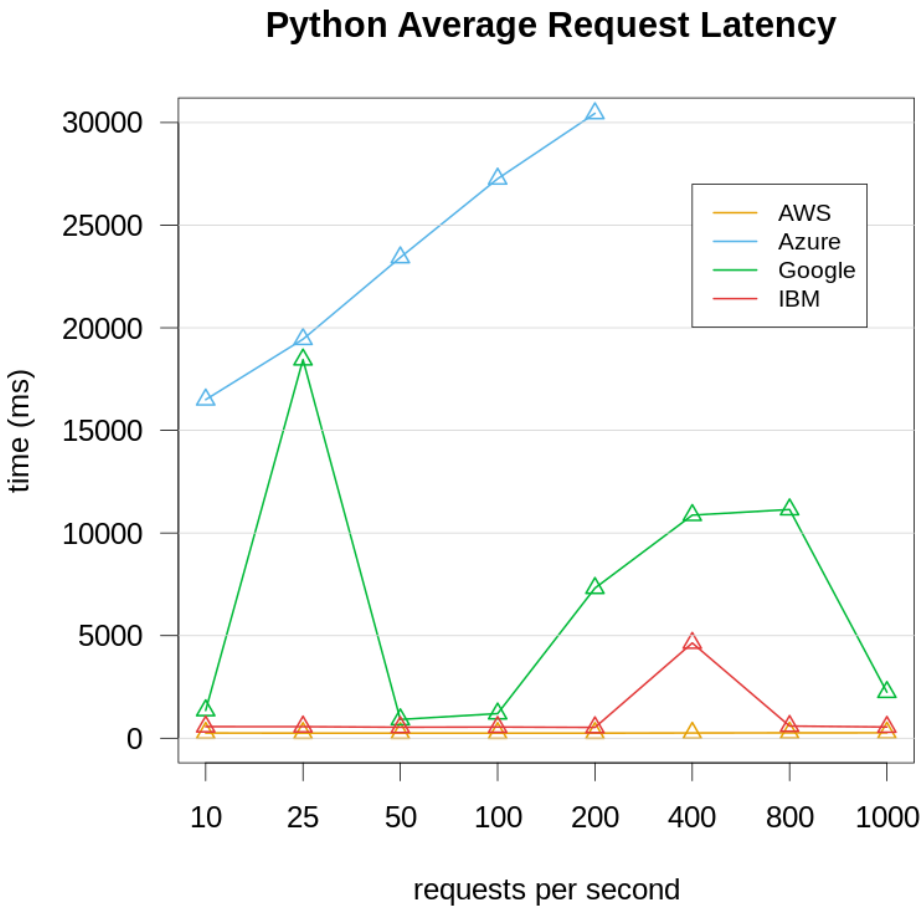


Results - General Performance

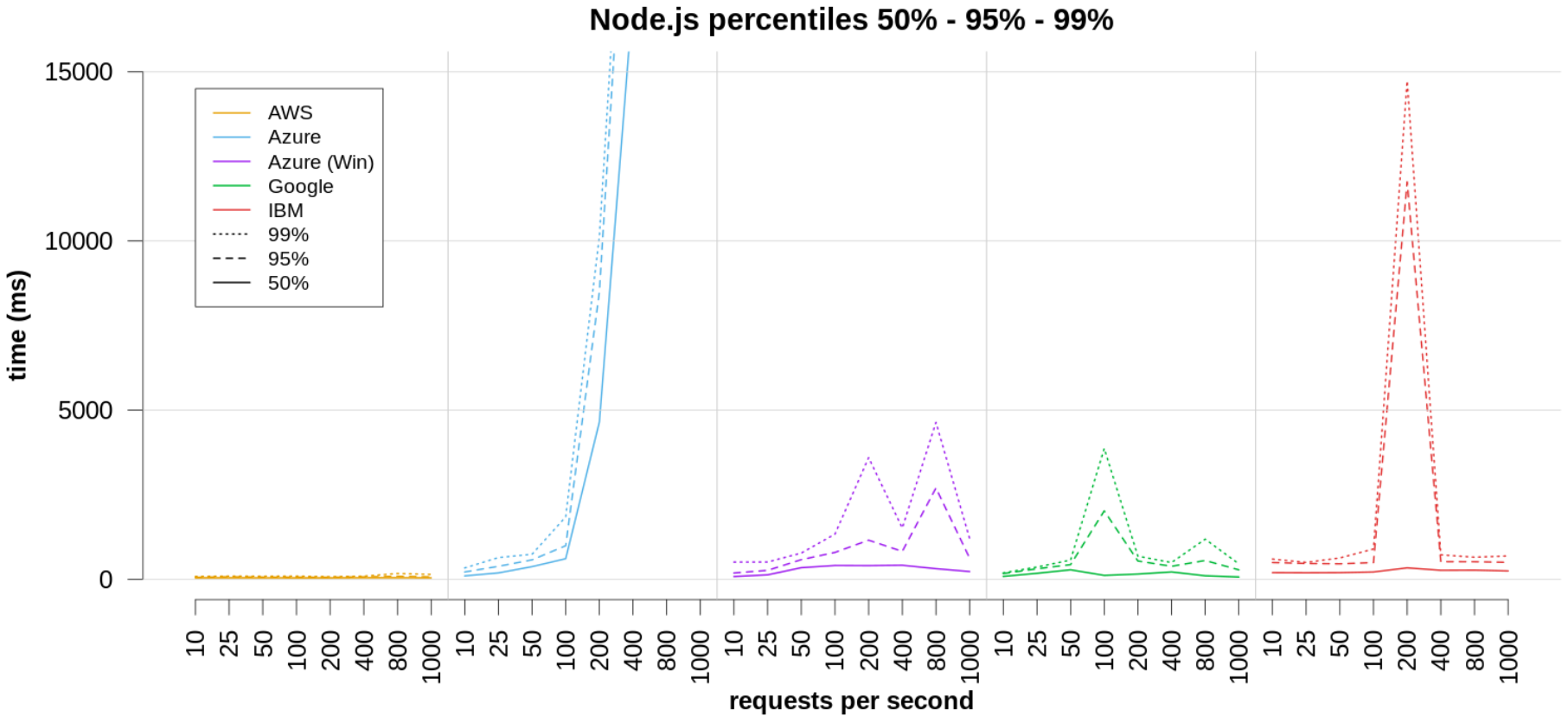


Results - Load Test (Average Latency)





Results - Load Test (Percentiles)



Future Work

- General optimizations and improvements
- Plotting integration (e.g. R)
- Continuous deployment
- More detailed load test
- Real world application benchmark
- Test docker images instead of given runtimes

Summary and Discussion

Testing: Important to test and optimize the code

- Execution time
- Memory needed
- Save costs

Scaling: Test a scenario as close as possible to the actual load. Clouds and runtimes can differ a lot in scaling.

Pricing: Estimate prices according to test results and not just using the offered pricing calculator

No standard: There is no standard framework between the clouds

Serverless can be a valid option depending on the workload and the requirements. This suite helps with the analysis and the decision.