Performance evaluation of parallel cloud functions

Maciej Pawlik, Kamil Figiela, Maciej Malawski
m.pawlik@cyfronet.pl, kfigiela@agh.edu.pl, malawski@agh.edu.pl, AGH University of Science and Technology

Introduction

Function-as-a-Service services are novel offering in cloud service provider's portfolios. FaaS enables the end user to run and manage deployed applications without the need to care for physical or virtualized infrastructure. User is only responsible for supplying the application and service provider takes care of the resource provisioning, this enables for constructing serverless applications. This poster presents research done on exploring and evaluating the potential applications of FaaS.

Objectives

- Validate **FaaS as a platform for HPC** [6] or video encoding [3]
- Test proposed means to execute scientific workflows on FaaS[4]

- **Can applications deployed on FaaS deliver the performance?**
  - Providers don't share the performance or hardware details.
  - Few function parameters: time limit, memory size, performance relative to memory.

- Extend the work done in [5] by:
  - Testing influence of parallelism on performance and resource provisioning
  - Bring the proposed benchmark closer to real life workloads
  - Provide basis for constructing a performance model

Benchmarking framework

The proposed solution is based on expanding a benchmarking framework proposed in [4]. The new benchmark combines two aspects of previous benchmarking suite:
- Workflow execution (infrastructure provisioning)
- Floating point performance

This approach allows for obtaining a more complete performance characteristics of studied infrastructures, including factors like task start delay and influence of parallelism. The testing load is generated with Linpack.

The benchmarking application was implemented as a “bag of tasks” workflow. A “bag of tasks” type workflow is depicted in Figure 1., proper application was composed of 1024 parallel tasks.

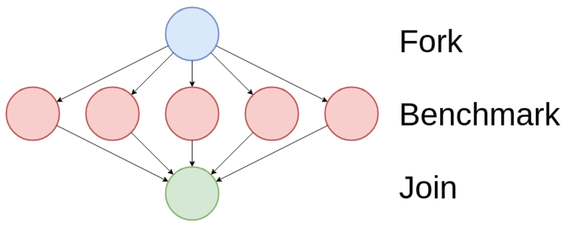

Figure 1: Benchmarking workflow graph

Workflow approach greatly simplifies managing many concurrent executions of benchmarking load. HyperFlow[2] workflow engine managed the execution.

Tested cloud function providers include:
- Amazon (Amazon Cloud Functions, abbr. **AWS**)
- Google (Google Cloud Functions, abbr. **GCF**)
- IBM (IBM Functions, abbr. **IBM**).

AGH

INFORMATYKA
Akademia Górniczo-Hutnicza w Krakowie

Results

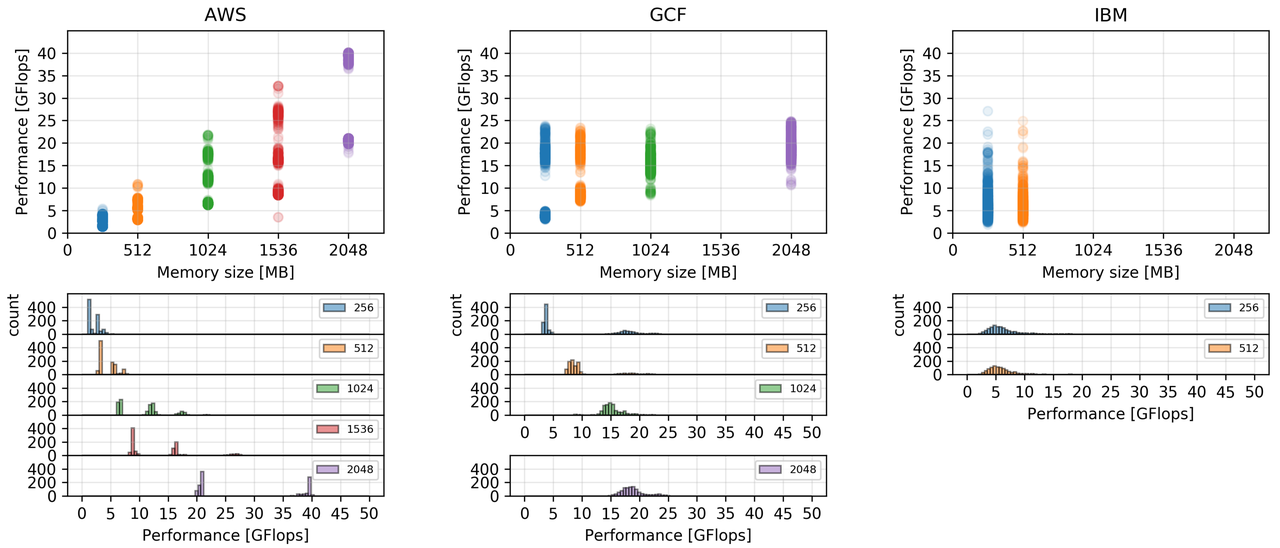

Figure 2: Measured performance in relation to function size. Each histogram contains result from 1024 samples.

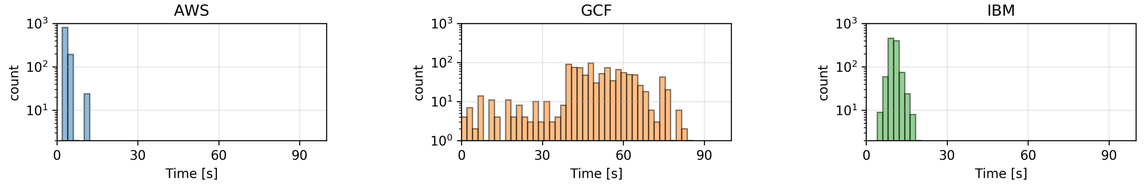

Figure 3: Histograms of execution delays for 512 MB function size and 1024 samples.

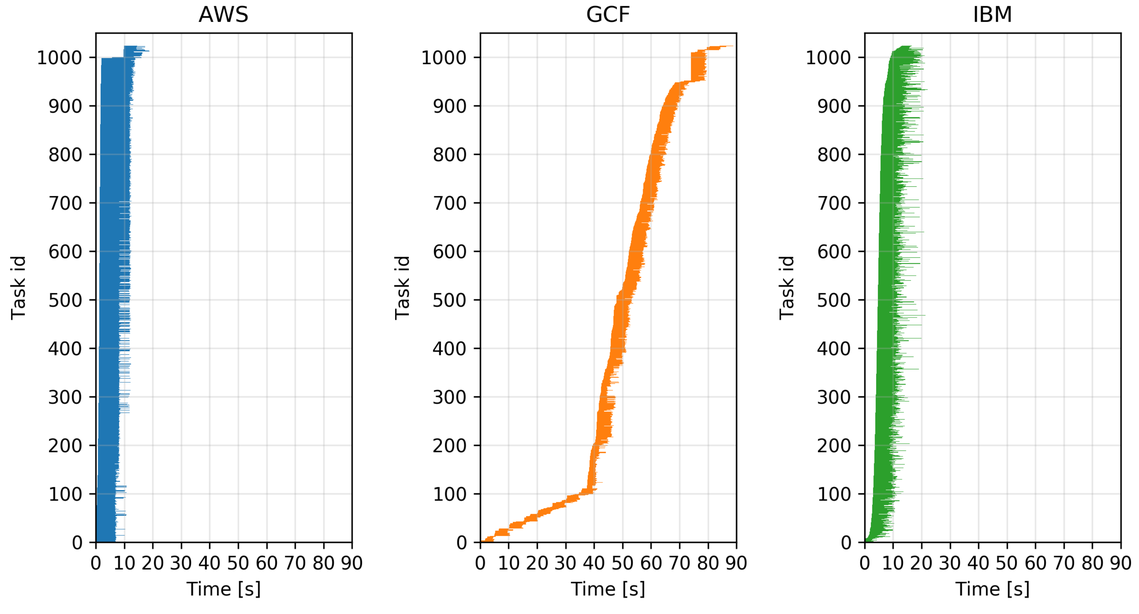

Figure 4: Charts representing execution time of 1024 individual tasks.

Discussion of results

Presented results focus on two factors:
- achieved performance (impacts execution time)
- delay of starting computation (infrastructure availability)

Performance

Figure 2. depicts achieved performance. Chart grid is organized in vendor specific columns. The upper part of the column is a scatter chart, where one can observe the measured performance in relation to function size. The lower part presents histograms of performance values for individual function sizes. Blank spaces are a result of some vendors offering only specific function sizes.

- AWS and GCF’s results show correlation of performance and function size
- IBM’s performance seems to be constant
- **All AWS sizes and GCF 256 do not have a single point of clustered results**
  - AWS 2048 performance was clustered around 20 and 40 GFlops
  - Approximately half of tasks were assigned resources with twice the computing power

Delays

Figure 3. presents histograms of task start delays:
- AWS: 1 to 3 seconds
- IBM: cluster of values near the 15. second
- **GCF: delay gradually rises with task number**

GCF’s behaviour might be a result of infrastructure provisioning policy, which includes throttling of requests. This conclusion can be drawn from Figure 4. which depicts execution period of individual tasks. In case of GCF we can see, that tasks are starting gradually with a certain rate, and right after the 35. second rate increases.

Conclusions and future work

- The proposed benchmark allowed to measure the approximate performance of FaaS providers
- Results revealed non obvious aspects of available performance and influence of parallelism on the function start delay
- Performance results, with minor differences in average values and cluster locations, are similar to ones obtained in [5]
- Presented results will be used for constructing FaaS performance models
- Periodic monitoring of performance is planned

Acknowledgements

This work was supported by the National Science Centre, Poland, grant 2016/21/B/ST6/01497.

References

[1] I. Baldini, P. Castro, K. Chang, P. Cheng, S. Fink, V. Ishakian, N. Mitchell, V. Muthusamy, R. Rabbah, A. Slominski, et al. 2017. Serverless computing: Current trends and open problems. In Research Advances in Cloud Computing. Springer, 1–20.
[2] B. Balis. 2016. HyperFlow: A model of computation, programming approach and enactment engine for complex distributed workflows. Future Generation Computer Systems 55 (2016), 147–162.
[3] S. Fouladi et al. 2017. Encoding, Fast and Slow: Low-Latency Video Processing Using Thousands of Tiny Threads. In 14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17). {USENIX} Association, Boston, MA, 363–376.
[4] M. Malawski. 2016. Towards Serverless Execution of Scientific Workflows-HyperFlow Case Study.. In WORKS@ SC. 25–33.
[5] M. Malawski, K. Figiela, A. Gajek, and A. Zima. 2017. Benchmarking Heterogeneous Cloud Functions. In European Conference on Parallel Processing. Springer, 415–426.
[6] J. Spillner, C. Mateos, and D. A Monge. 2017. FaaSter, Better, Cheaper: The Prospect of Serverless Scientific Computing and HPC. In Latin American High Performance Computing Conference. Springer, 154–168.