Performance evaluation of parallel cloud functions

Extended Abstract

Maciej Pawlik
AGH University of Science and Technology
Krakow
m.pawlik@cyfronet.pl

Kamil Figiela
AGH University of Science and Technology
Krakow
kfigiela@agh.edu.pl

Maciej Malawski
AGH University of Science and Technology
Krakow
malawski@agh.edu.pl

ABSTRACT

This paper depicts results of benchmarking a novel, cloud based *Function as a Service* infrastructures, with a compute intensive load based on Linpack. In order to obtain the results, a benchmarking framework is proposed and applied to AWS Lambda, Google Cloud Functions and IBM Cloud Functions. Results of running 1024 Linpack processes in parallel show differences between the cloud providers and non trivial characteristics of performance and delays. The measurements can provide a baseline for estimating serverless application run times that can be useful for resource management.

CCS CONCEPTS

• **Computing methodologies** → **Massively parallel and high-performance simulations**; • **Computer systems organization** → **Cloud computing**; • **Theory of computation** → Massively parallel algorithms; • **Software and its engineering** → *Cloud computing*;

KEYWORDS

Performance Analysis and Optimization, Performance Tools, Clouds and Distributed Computing

ACM Reference Format:
Maciej Pawlik, Kamil Figiela, and Maciej Malawski. 2018. Performance evaluation of parallel cloud functions: Extended Abstract. In *Proceedings of 47th International Conference on Parallel Processing (ICPP 2018).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Function-as-a-Service services are a 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. The user is only responsible for supplying the application, resource provisioning is handled by service provider. This approach enables constructing so called serverless applications. This paper presents research done on exploring and evaluating the potential applications of FaaS.

ICPP 2018, August 13-16, 2018, Eugene, Oregon, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 OBJECTIVES

There is a significant amount of research done on new use cases for FaaS [1]. One of the ideas is to exploit the computing power of FaaS by using it as an environment for running HPC applications [6] or video encoding [3]. While not all workloads can be adapted to run as cloud functions, means for executing workflow-type HPC applications were proposed in [4].

In order to assess the feasibility of running a given application on FaaS we need to determine if the application performance will be acceptable. This can be achieved by constructing a reliable performance model, which in turn requires the knowledge about performance of the infrastructure. Cloud service providers rarely supply such details as hardware configuration, usually limiting the available information to function time limit, maximum memory (or *function size*) and note that memory size affects available CPU quota. Tested infrastructure providers follow this practice. In contrast to previous studies done in [5], work presented in this paper is closer to simulating a real life application. The tested scenario includes execution of up to 1024 function tasks in parallel. We try to provide a baseline performance, its relation to the function size and delays for highly parallel function spawning.

3 BENCHMARKING FRAMEWORK AND RESULTS

The work is based on expanding a benchmarking framework proposed in [5]. The new benchmark combines two aspects of previous benchmarking suite: testing workflow execution (infrastructure provisioning) and floating point performance into one. This 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, an industry standard benchmark with large set of results available for comparison. The benchmarking application was implemented as a workflow utilizing a bag of tasks model, which simplifies managing multiple instances of tasks. HyperFlow[2] a proven management engine for scientific workflows, was chosen to manage the execution. Tested cloud function providers include: Amazon (Amazon Cloud Functions, abbr. AWS), Google (Google Cloud Functions, abbr. GCF) and IBM (IBM Functions, abbr. IBM).

Presented results focus on two factors, the delay of starting computation and achieved performance. The factors were chosen based on their significance for building a performance model, where the information about infrastructure availability (delays), and execution time (performance) is crucial.

Figure 1 presents histograms of delays, encountered while running the benchmark for 512MB function size. In case of AWS the

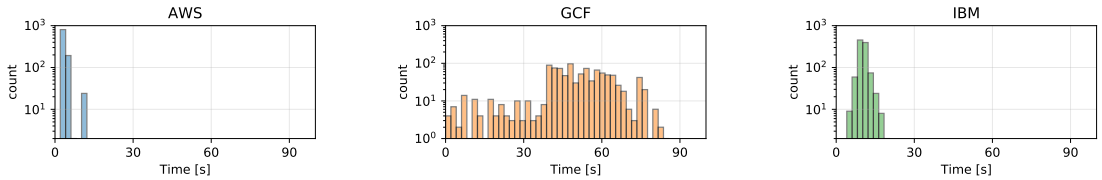

Figure 1: Histograms of execution delays for 512 MB function size and 1024 samples.

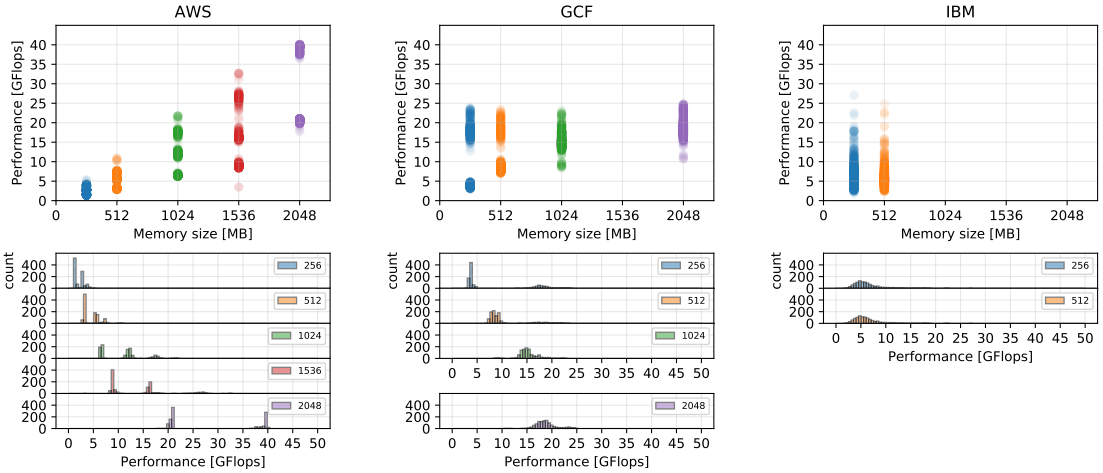

Figure 2: Measured performance in relation to function size. Each histogram contains result from 1024 samples.

delays were measured to be in range of 1 to 3 seconds. In case of IBM we have a similar chart albeit the delays concentrate around the 15 second mark. Results for GCF show, that the delay is proportional to number of task. This might be an effect of throttling function invocations, possibly due to infrastructure provisioning policy. At first small portion of tasks is executed instantly, while right after the 35 seconds mark a surge of executions occurs.

Figure 2 depicts achieved performance, column wise, for each provider. Upper charts illustrate measured performance in relation to function size. The lower charts present histograms of performance values for individual function sizes. Some vendors offer only a specific function size configurations, thus not all combinations were covered by tests.

AWS and GCF results show a direct correlation of performance and function size, whereas IBM performance seems to be constant. Besides the average performance, it is important to note that results for AWS and GCF 256 do not have a single point of clustered results. In the case of AWS 2048 achieved performance was clustered around 20 and 40 GFlops, where almost half of tasks are assigned resources with twice the computing power. This phenomena, with a varying relation between clusters, can be observed in other configurations for AWS.

4 CONCLUSIONS AND FUTURE WORK

The proposed benchmark allowed to measure and document the approximate performance available on popular FaaS platforms. Results revealed non obvious aspects of obtained performance and an influence of parallelism on the function start delay. Performance results, with minor differences in average values and cluster locations, are similar to ones included in [5]. This confirms that FaaS infrastructures can provide consistent results over time. Presented results can be used as the basis for constructing performance models of FaaS deployed applications.

This work can be extended by setting up an automated benchmarking infrastructure. Constant benchmarking over a longer period of time would allow to determine if the performance is invariant, regardless of the cloud provider's infrastructure load.

REFERENCES

[1] Ioana Baldini, Paul Castro, Kerry Chang, Perry Cheng, Stephen Fink, Vatche Ishakian, Nick Mitchell, Vinod Muthusamy, Rodric Rabbah, Aleksander Slominski, et al. 2017. Serverless computing: Current trends and open problems. In *Research Advances in Cloud Computing*. Springer, 1–20.

[2] Bartosz 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] Sadjad 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. https://www.usenix.org/conference/nsdi17/technical-sessions/presentation/fouladi

[4] Maciej Malawski. 2016. Towards Serverless Execution of Scientific Workflows-HyperFlow Case Study.. In *WORKS@ SC*. 25–33.

[5] Maciej Malawski, Kamil Figiela, Adam Gajek, and Adam Zima. 2017. Benchmarking Heterogeneous Cloud Functions. In *European Conference on Parallel Processing*. Springer, 415–426.

[6] Josef Spillner, Cristian Mateos, and David A Monge. 2017. FaaSter, Better, Cheaper: The Prospect of Serverless Scientific Computing and HPC. In *Latin American High Performance Computing Conference*. Springer, 154–168.