Survey of Serverless Edge Computing Frameworks

Iftekhar Rahman (MEMS) , Mahdi Safahieh (IIW)

Communication Networks

ComNets
Institute of Communication Networks

Introduction

Goals of this poster :

- Presenting a framework to support multi-provider serverless on edge (MPSC)
- Evaluating the performance of proposed framework
- Surveying the role of serverless edge computing in two other frameworks

Limitation of Serverless Edge :

- Lock-in effect
- Limited resources
- Distributed scheduling problem

Solution :

- Presenting a framework, where Users can define their own scheduling algorithm
- Recommends best serverless provider based on user schedule
- Provides Real time monitoring of different serverless providers including edge

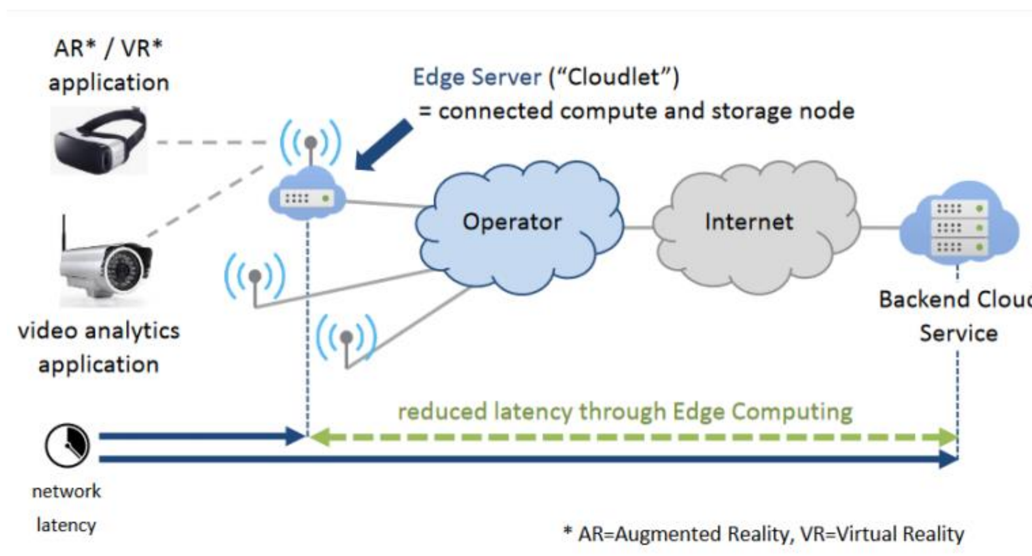

Figure 1 : Serverless edge computing schema , Source: ETSI MEC, September 2016

MPSC FRAMEWORK

- **Monitoring Controller :** Responsible for coordinating objects.
- **REST API :** Accommodates communication between user and MPSC.
- **Function Executors :** Invokes and measures performance of functions.
- **Configuration File :** Stores user specific configuration.
- **Metrics Database :** Stores benchmark results for each provider.
- **User Defined Schedules:** Scheduling algorithm defined by the use

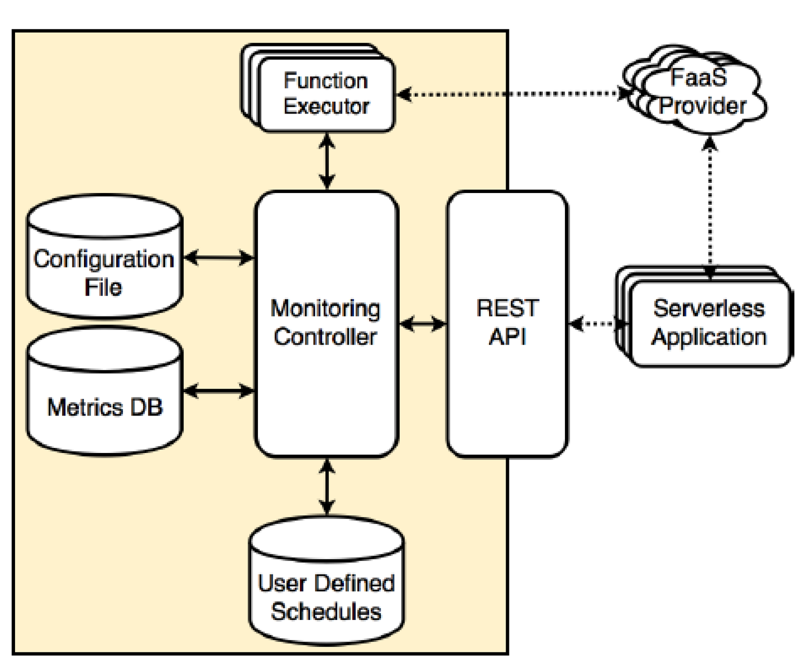

Figure 2 : MPSC System Architecture [1]

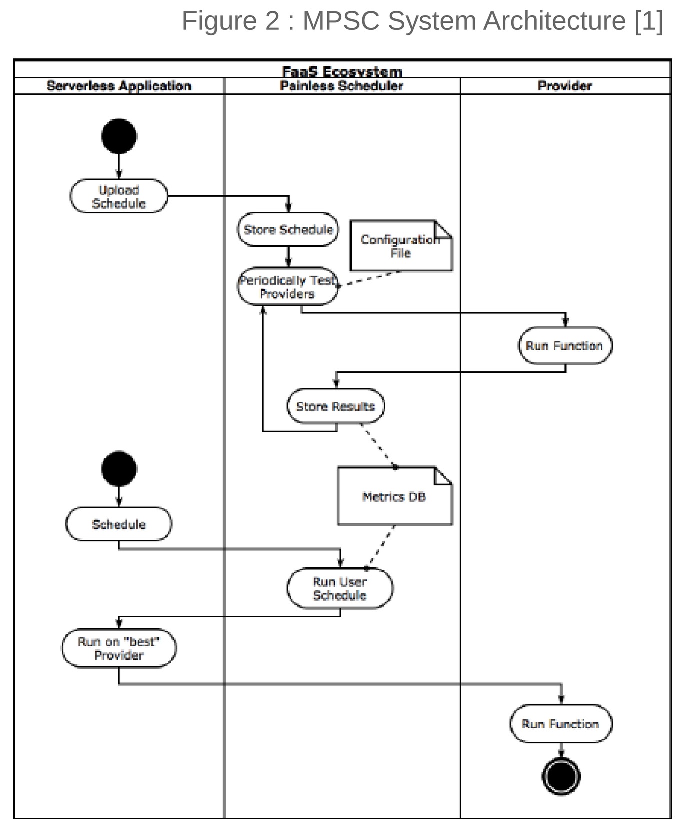

Figure 3 :MPSC Workflow [1]

Evaluation of MPSC

Low Latency scheduling

- Each provider runs Miller-Rabin primality test (on one millionth prime).
- MPSC framework runs for 400s , testing every provider every 5s.
- OpenWhisk offers lower RT(s), less Var and Std.

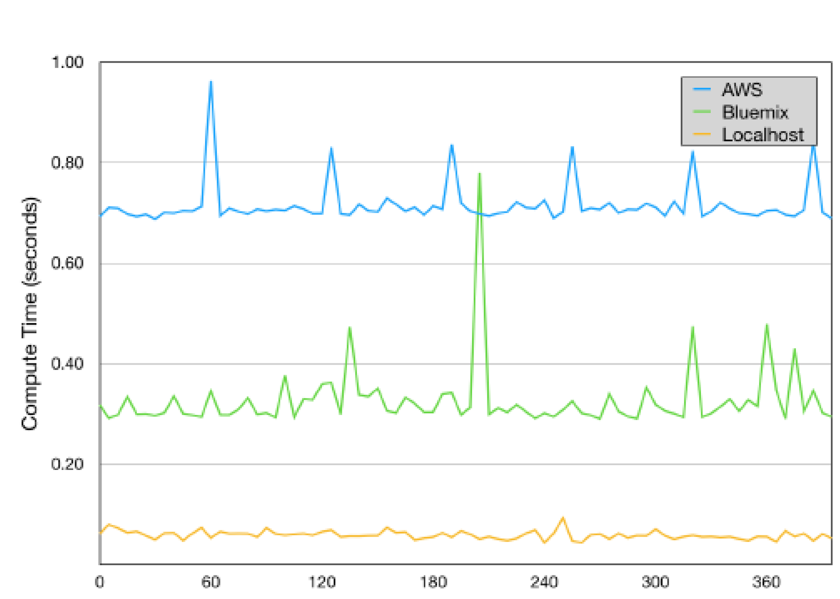

Figure 4 :The round trip times of each provider running Miller-Rabin primality test on one millionth prime (15,485,863) [1]

	AWS	Bluemix	Localhost	User Schedule
RT (s)	57.26446	26.13867	4.72654	4.72654
Var	0.00184	0.00413	0.00007	0.00007
Std	0.04292	0.06430	0.00824	0.00824

Table 1: Experimental results for low latency scheduling [1]

Low Latency scheduling with background workload:

- Same test settings as before with added background in the local sever
- Added traffic includes asynchronously 1000 function invocation to local server while MPSC is still running
- Background color refers to the chosen provider based on provided schedule

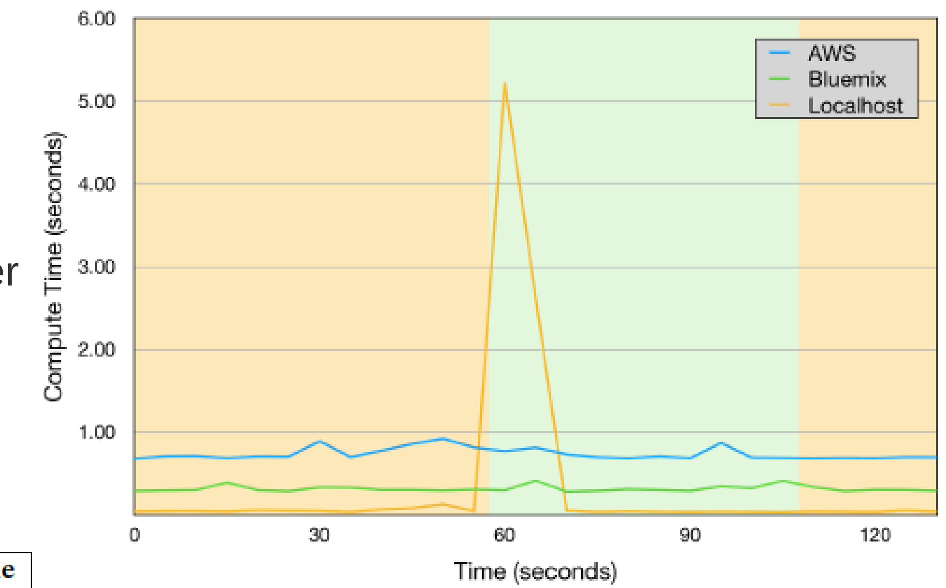

Figure 5 :The round trip times of each provider with introduction of simulated traffic on the local provider [1]

	AWS	Bluemix	Localhost	User Schedule
RT (s)	20.03395	8.61793	9.15323	4.26473
Var	0.00527	0.00131	1.19856	0.01927
Std	0.07263	0.03614	1.09479	0.13881

Table 2 : Experimental results for low latency scheduling with background workload [1]

- Each providers total execution time is presented as a base line and compared to the low latency schedule
- 468% decrease in latency, running MPSC vs AWS lambda
- 200% decrease in latency, running MPSC vs Bluemix and local openWhisk

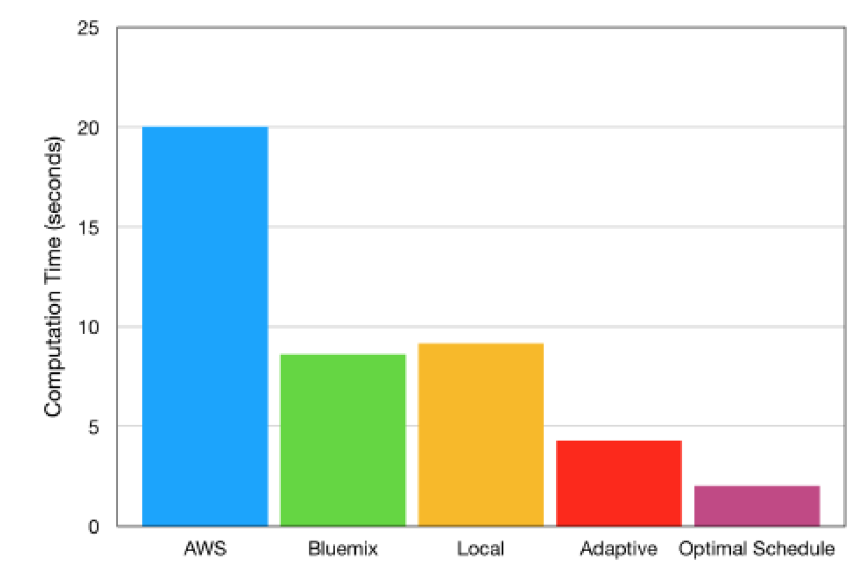

Figure 6 :Total round trip times for each provider compared with our adaptive schedule and the optimal schedule [1]

Related works

Low-Latency Applications Through a Serverless Edge Computing Architecture :

- Offering serverless architecture on the edge to gain Low-Latency
- The feasibility of this approach is shown through an augmented reality application
- Results show 80% improvement in throughput and latency compared to cloud

Latency-aware Video Analytics on Edge Computing Platform :

- This research provides a low-latency video edge analytic System
- LEVEA framework offloads computation between clients and edge nodes
- Results reveal that the client-edge configuration improves speed of system from 1.3x to 4x (1.2x to 1.7x) compare to running in client-cloud configuration

Comparison of MPSC and related works :

Features	MPSC	MEC	LAVEA
Low-latency scheduling	✓	×	✓
Adaptability	✓	×	✓
Real-time monitoring	✓	×	✓
High throughput	✓	✓	✓
Offloading Service	×	✓	✓

Table 3 : MPSC vs Related works

Conclusion

- Serverless-Edge technology makes microservices easier to develop while taking the resources closer to users
- Although, developers are left with little control over the performance of their code. The MPSC framework proposes a solution to cope with this problem
- MPSC framework increases system agility (200% faster) even working on a resource limited structure like the edge
- Serverless-Edge architecture also promises better performance for advanced applications like virtual reality, augmented reality, video analytics, face recognition etc.
- It will provide robust infrastructure for coming 5G networks which requires low latency and high throughput

References

[1] Austin Aske and Xinghui Zhao. 2018. " Supporting Multi-Provider Serverless Computing on the Edge". In ICPP '18 Comp: 47th International Conference on Parallel Processing Companion, August 13–16, 2018, Eugene, OR, USA. ACM, New York, NY, USA, 6 pages.

[2] Luciano Baresi, Danilo Filgueira Mendonça, and Martin Garriga. 2017. "Empowering Low-Latency Applications Through a Serverless Edge Computing Architecture ".In European Conference on Service-Oriented and Cloud Computing. Springer, 196–210.

[3] Shanhe Yi, Zijiang Hao, Qingyang Zhang, an Zhang, Weisong Shi, and n Li. 2017. LAVEA:" Latency-aware Video Analytics on Edge Computing Platform". In Proceedings of SEC '17, San Jose / Silicon Valley, CA, USA, October 12–14, 2017, 13 pages.

TUHH
Technische Universität Hamburg