

An Evaluation of Serverless Data Processing Frameworks

Presented at: Sixth International Workshop on Serverless Computing (WoSC6)
2020



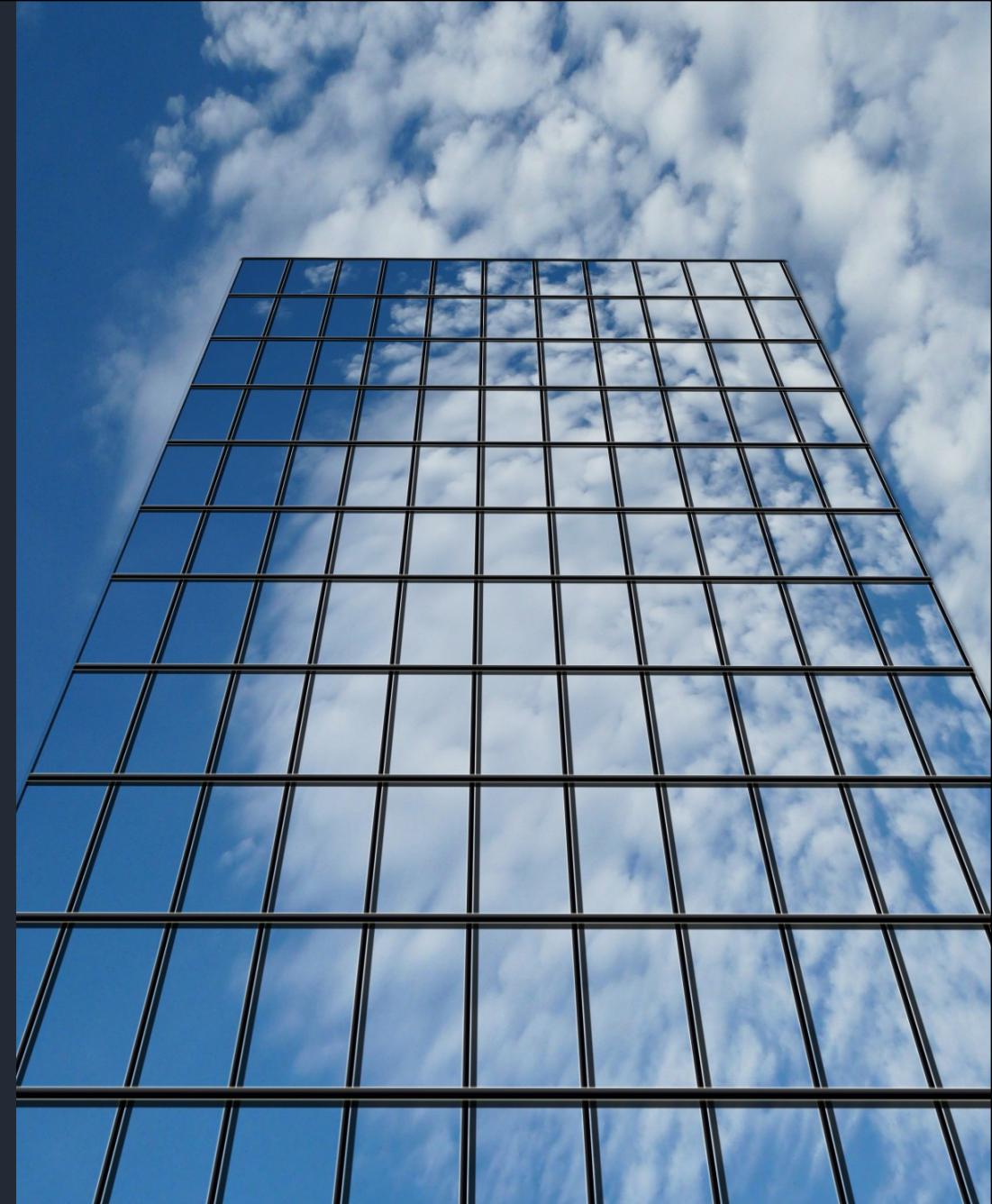
Sebastian Werner, Richard Girke, Jörn Kuhlenkamp

Thursday, November 26, 2020

*Information Systems Engineering
TU Berlin - Germany*

Motivation

- Serverless is a new cloud execution model offering auto-scaling and a pay-as-you-go cost model[1]
- High elasticity and the flexible cost model is a perfect fit for ad-hoc data processing needs[2]
- A combination of Function-as-a-Service (FaaS) and fully managed storage services as a good first architecture for exploratory data analysis[3]
- Several serverless data processing frameworks exist, but data analysts, and system researchers are unaware how these frameworks compare



Research Question:

How can data analysts assess the quality of serverless based frameworks for ad-hoc data processing?

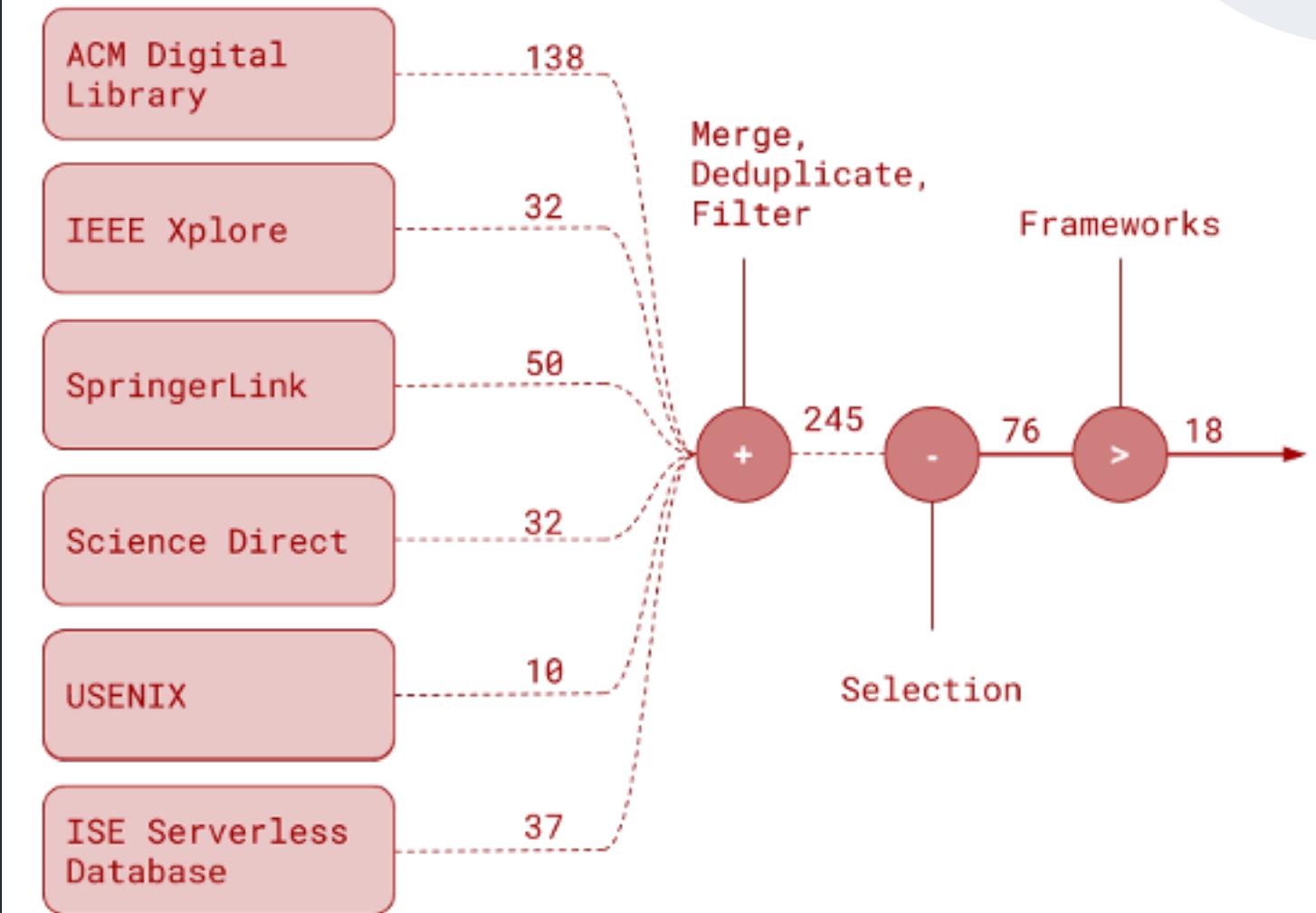
Contributions:

- A structured overview of existing serverless data processing frameworks
- A qualitative architectural comparison of existing serverless data processing frameworks
- An experimental comparison of publicly available serverless data processing frameworks and AWS EMR

Multi-vocal Literature Review

Q1: What serverless data processing tools exist in research and industry?

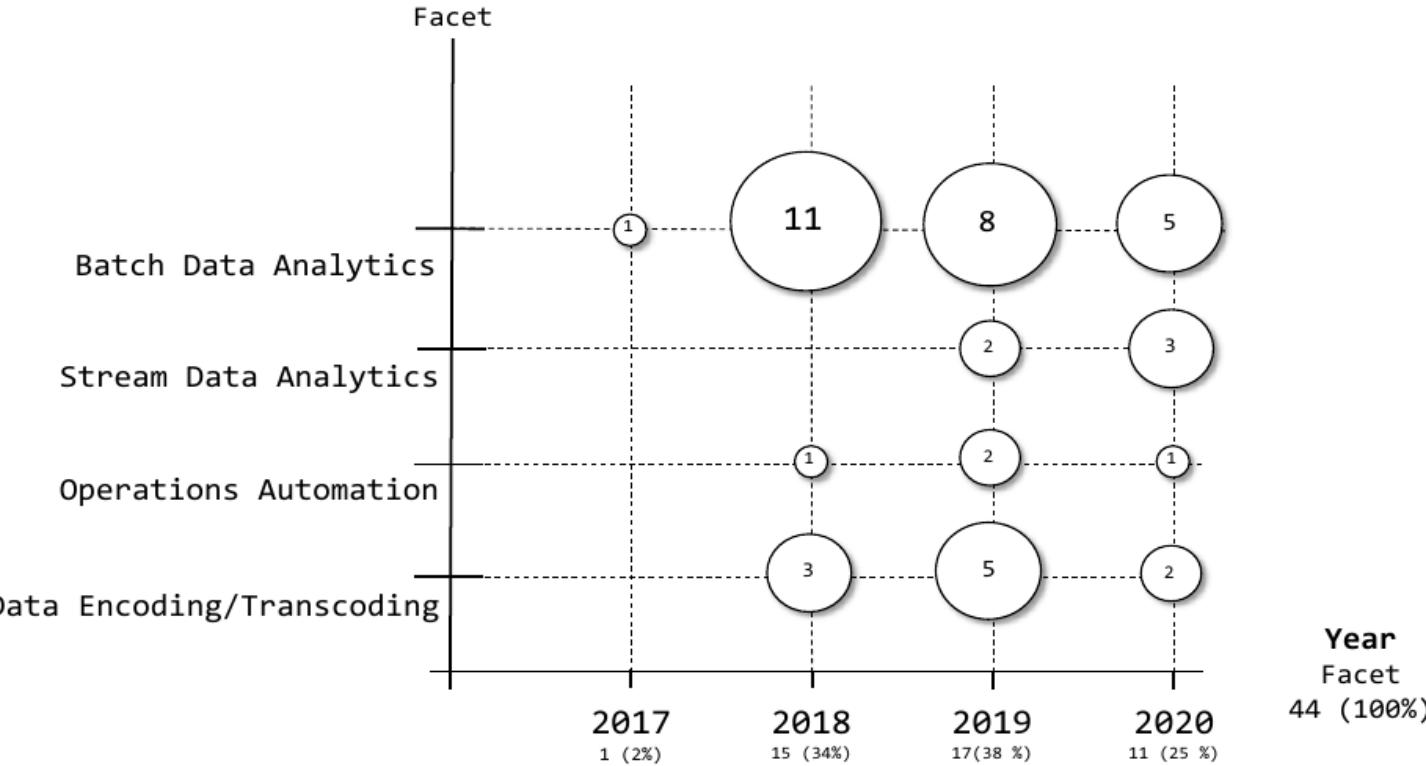
Q2: What use-cases and purposes for serverless data processing are investigated?



Usage

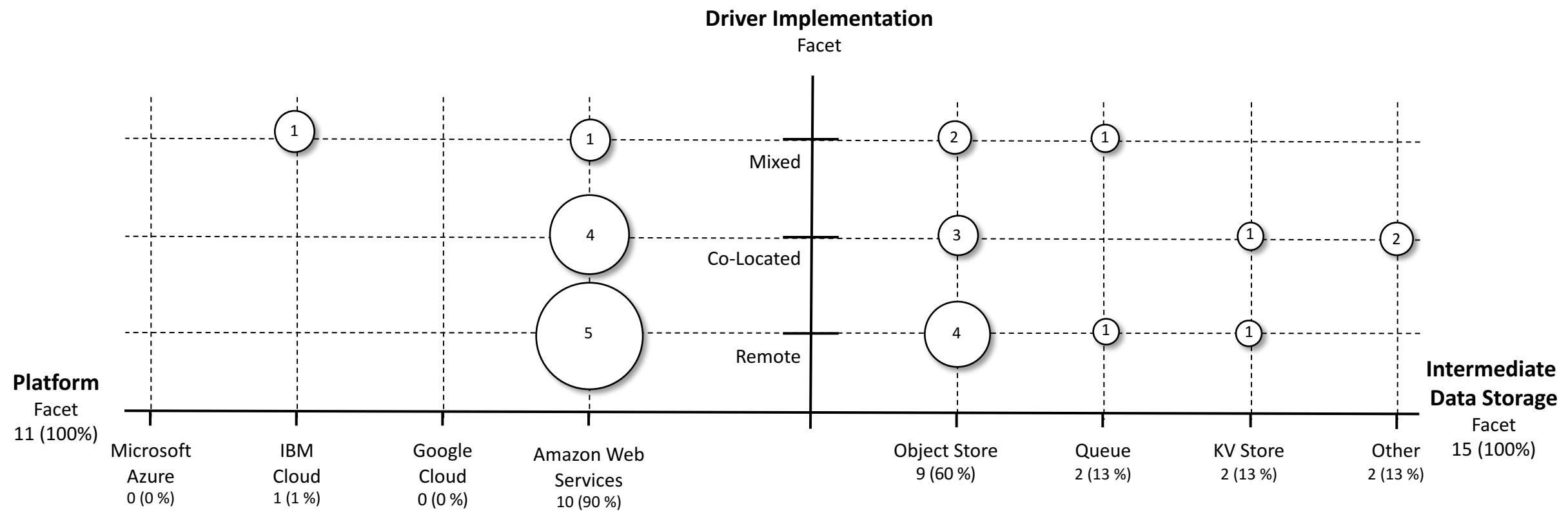
- 18 serverless data processing frameworks and platforms
- 11 allow general purpose computation and use cloud resources
- Most target “Batch Data Analytics”
- Most development in 2018 and 2019

FaaS-based Data Processing Prototype Categories*



Architectural Overview

- All use a driver program for job orchestration and monitoring
- Most use object stores for intermediate data
- AWS most targeted platform for data processing



Frameworks

- 63%⁴ have available source code for independent testing
- 54% offer Map-Reduce Programming Model
- 18% use high-level abstractions like Apache Spark

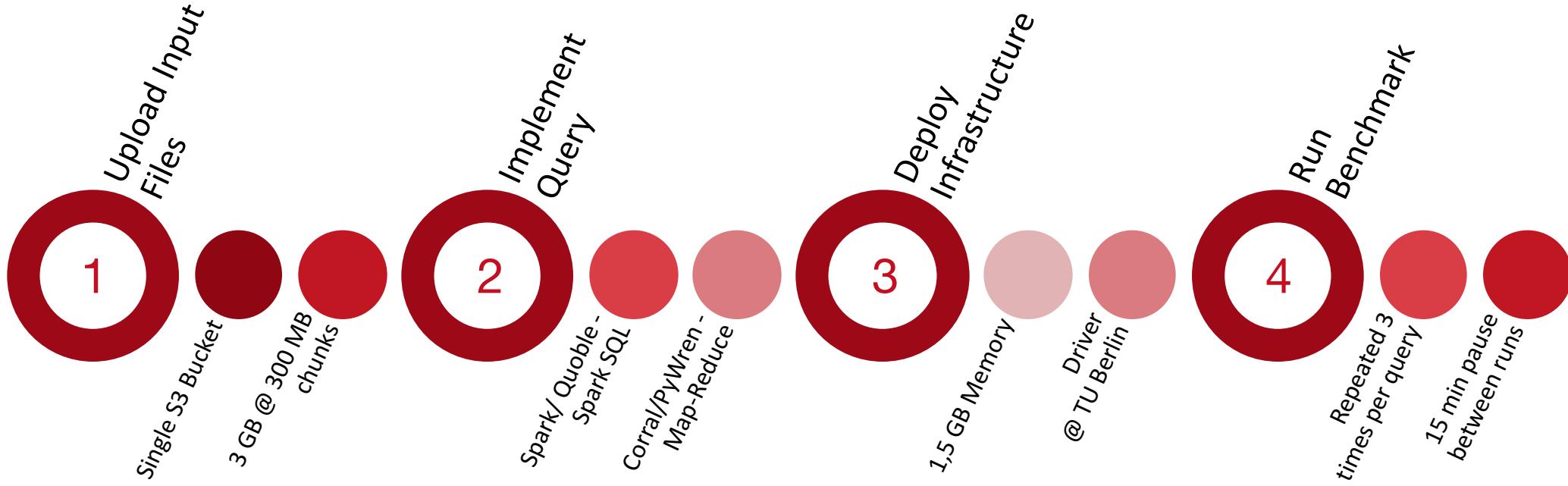
Framework Name	Source Available	Programming Model
PyWren	Yes	Map ²
IBM PyWren	Yes	Map-Reduce
gg	Yes	Map ²
Flint	No	Map-Reduce
Lambada	No	Map-Reduce
Starling	No	Map-Reduce
Corral	Yes	Map-Reduce
Quoble ¹	Yes	Map-Reduce
Crucial	No	N.A.
WuKong	Yes	Dask ³
Marla	Yes	Map-Reduce

1: Quoble - Spark on Lambda 2: Map like processing abstraction, or single stage parallel execution

3: <http://dask.org> 4: at the time of writing the paper we did not discover WuKong and Marla

Experiment Design

We selected, **Quoble**, **Corral** and **PyWren** on AWS Lambda and **Apache Spark** on AWS EMR. We selected the TPC-H benchmark for the comparison.



Experiment

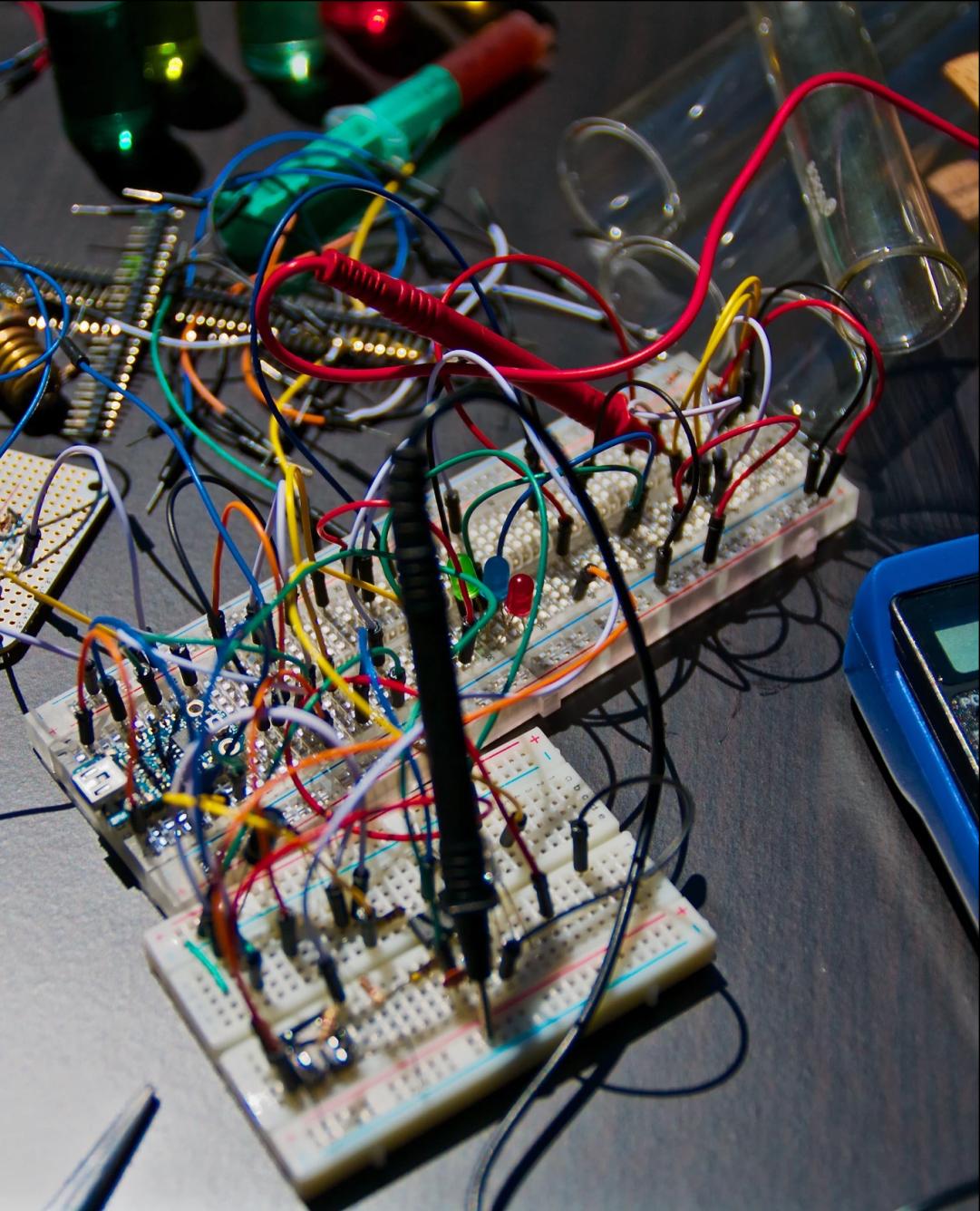
Workload:

- TPC-H-queries 1 and 6
- Q1 – single group by, no joins → Simple Implementation
- Q6 – no joins, no group by → Easy to paralyze

Metrics:

- Total Runtime (RT [s])
- Total Cost (Cost [μ \$])
- Invocations (Worker [#])
- S3 Operations (S3 Ops [#])
- Queries per Hour (QphH)
- Cost per Hour (CpH)
- Setup & Implementation Time (sT [h] | iT [min])

<https://github.com/ISE-SMILE/framework-experiments>



Cost and Performance Results

Framework	TPC-H Query	S3 Ops.	QphH [#]	CpH [\$]	Mean RT [s]	Mean Cost [μ \$]
Quoble	1	15000	58	4.30	62	1840
	6	3000	80	1.20	45	1250
Corral	1	470	49	0.10	73	77
	6	410	143	0.20	23	75
PyWren	1	205	51	0.10	70	72
	6	191	52	0.10	69	75
EMR	1	36	40	1.90	91	18700
	6	32	40	1.90	90	18700

Maturity and Developer Experience

Framework	Setup/Deployment Time [h]	Query Implementation Time [min]	Commits [#]	Last Commit [year]
Quoble	20	5	8	2017
Corral	3	15	82	2019
PyWren	2	15	51	2018
EMR (Spark 2.6)	½	5	3659	2020

Conclusion

- Only a few frameworks are publicly available
- AWS is the data processing infrastructure of choice
- Serverless computing well suited for ad-hoc query processing

Future Work

- Driver implementation and intermediate storage strongly influence performance and cost
- Extended experiment design

Paper:



<https://www.serverlesscomputing.org/wosc6/#p4>

Contact



sw@ise.tu-berlin.de



jk@ise.tu-berlin.de



rg@tu-berlin.de

ISE



www.ise.tu-berlin.de



<https://www.ise.tu-berlin.de/youtube>



<https://ise-smile.github.io/>