Serverless Computing:
Challenges, Opportunities, and Beyond

Technology Innovation
Cloud BU
R&D Division

www.huawei.com

Dr. Javier Picorel, Engineering Manager

HUAWEI TECHNOLOGIES CO., LTD.

Serverless is Skyrocketing!

Why Serverless Computing Is The Fastest-Growing Cloud Services Segment

September 2, 2018

This is the first in our three part series on serverless computing. Check out how the major cloud providers are getting into the space here and take a look at the final installment on early-stage companies to watch here.

Cloud Enterprise IT Expert Intelligence Trends

WHERE IS THIS DATA COMING FROM?

The serverless market is expected to reach $7.7B by 2021, up from $1.9B in 2016.

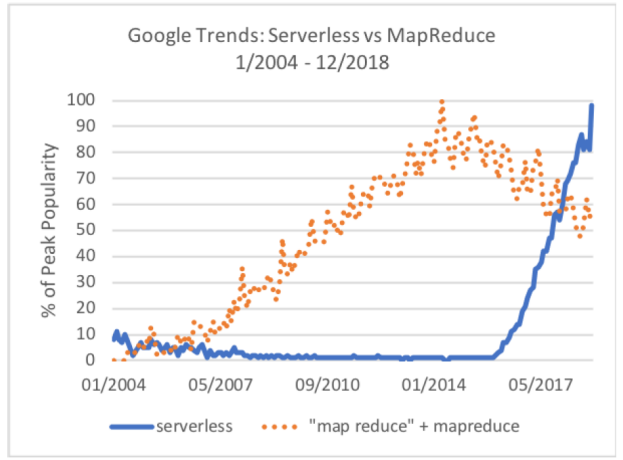


Figure 1: Google Trends for "Serverless" and "Map Reduce" from 2004 to time of publication.

Global Serverless Architecture Market to Reach $21.99 Billion by 2025 at 27.8% CAGR: Allied Market Research

Rapid rise of the app development market along with increase in demand for useful applications for different platforms such as Android and iOS have boosted the growth of the serverless architecture market. Growing shift from DevOps to serverless computing and rising need to eliminate server management challenges have supplemented to the growth of the market. The region of North America region accounted for nearly half of the market of global serverless architecture in terms of revenue in 2017.

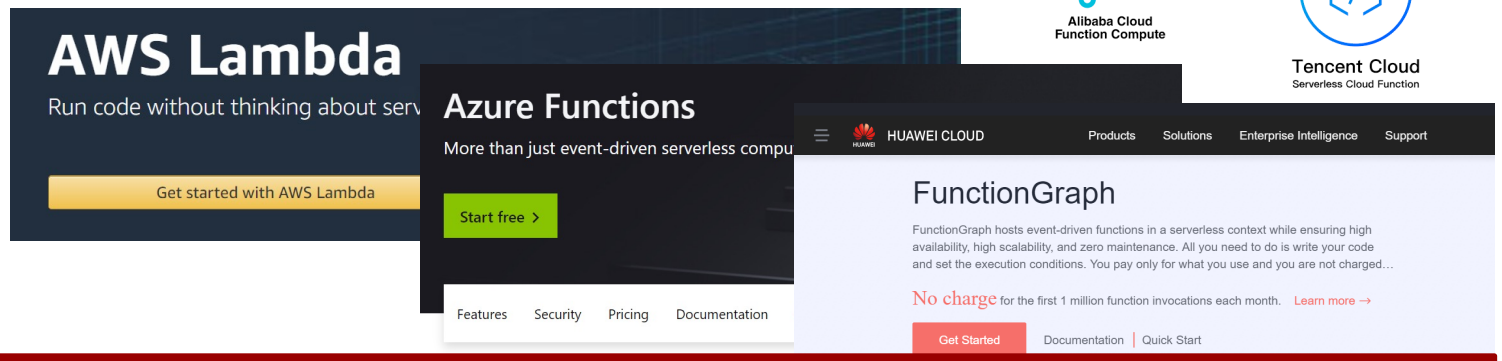
Email Print Friendly Share

Profile
Allied Market Research

PRESS RELEASE

Global Serverless Architecture Market was valued at USD 3.86 Billion in 2018 and is projected to reach USD 26.44 Billion by 2026, growing at a CAGR of 27.17% from 2019 to 2026

Published: Aug 26, 2019 4:26 p.m. ET



Google Cloud Functions

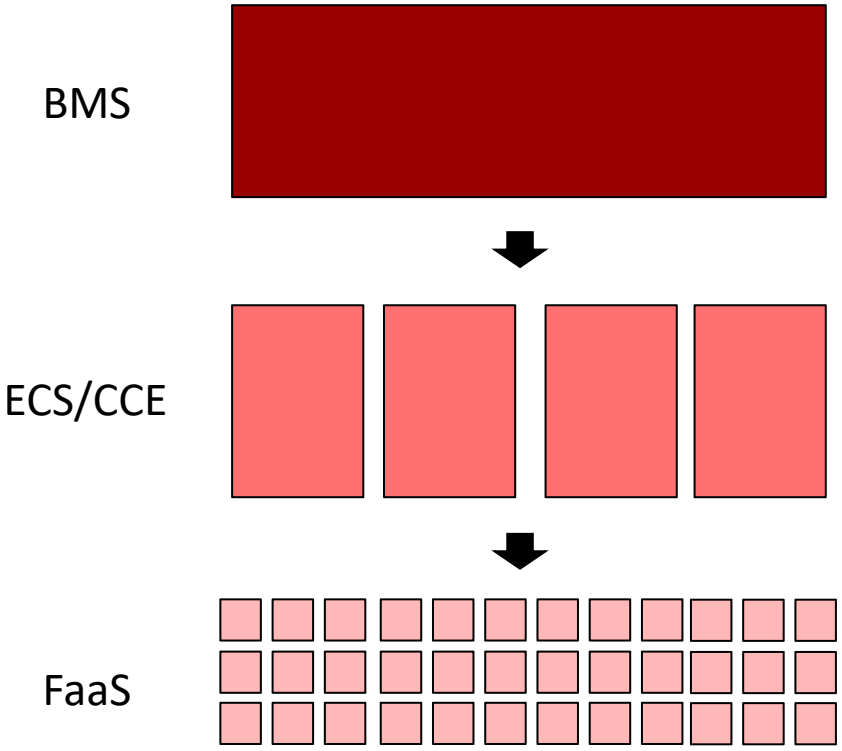
Alibaba Cloud Function Compute

Tencent Cloud Serverless Cloud Function

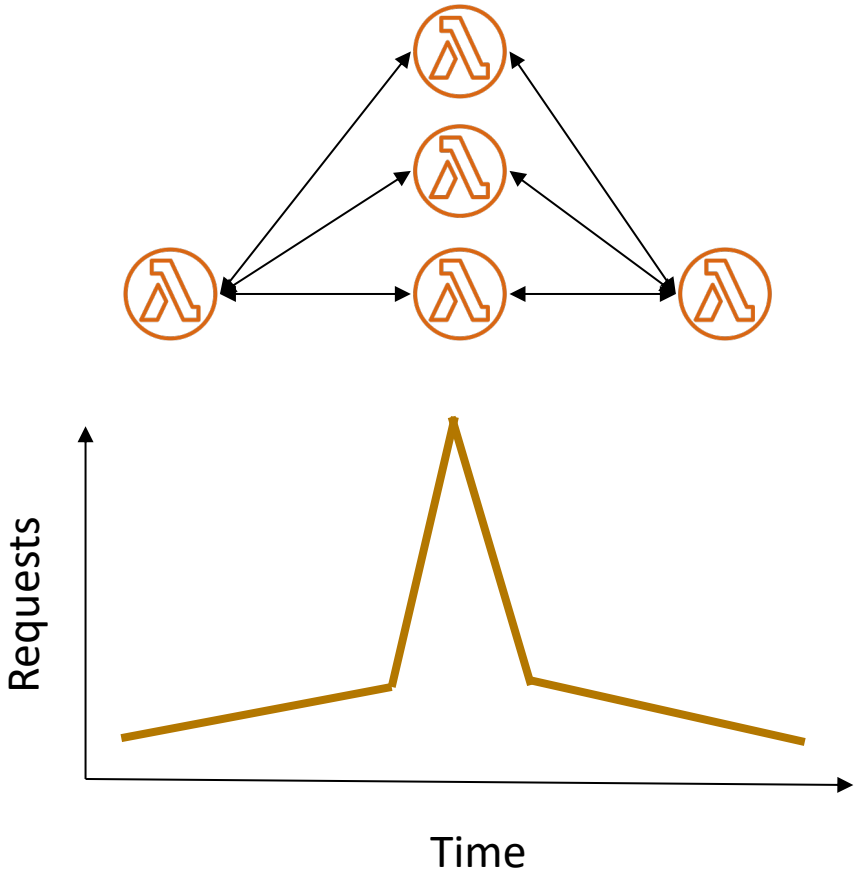
Serverless computing market growing at unprecedented pace

Advantages of Serverless Computing

- Fine-grained resource provisioning



- On-demand scaling



High-density multi-tenant resource sharing & elasticity

Cloud-Native: Storage Disaggregation

Recent trend on separating compute & storage:

- Load scales differently
- Differences in device lifecycles
- Different requirements for reliability & availability

→ Overall **TCO** (cost) reduction

Gartner report "Data Distribution & Complexity Drive Information Infrastructure Modernization":

- By 2019, 90% of cloud DBMS architectures will support separation of compute & storage
- Future cloud architectures centered around object storages

451 Group Report "Separation of compute and storage drives analytics in the cloud"

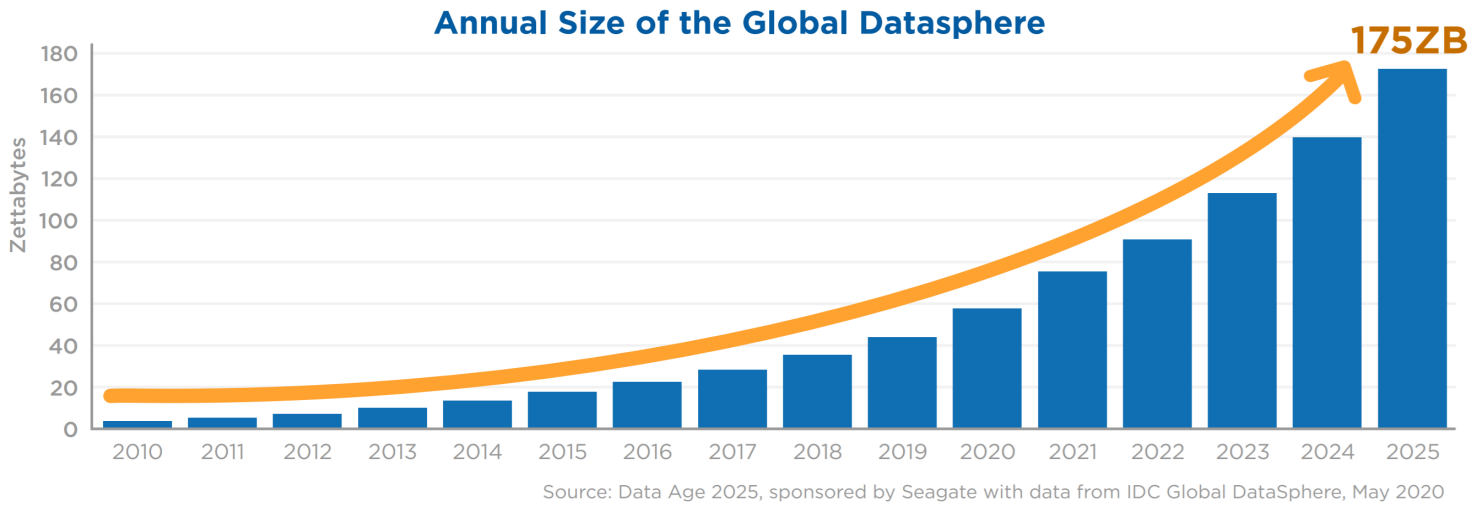
	Storage	Computing		
Vendor	**High scalability Unified storage**	**Autoscaling Complex computing capability**	**Autoscaling Preprocessing computing capability**	**Autoscaling Arbitrary functions capability**
AWS Redshift Spectrum	S3	Redshift	Spectrum layer	AWS Lambda
Microsoft Azure Data Lake	Azure Storage (OBJ/HDFS)	Azure SQL DW/Polybase	Azure Data Lake Analytic(U-SQL)	Azure Functions
Google GCP BigQuery	GFS/CFS	NA	BigQuery (Dremel)	Cloud Functions

* Even Oracle dug a large number of high-end experts from AWS and Microsoft to build a bare metal cloud storage system

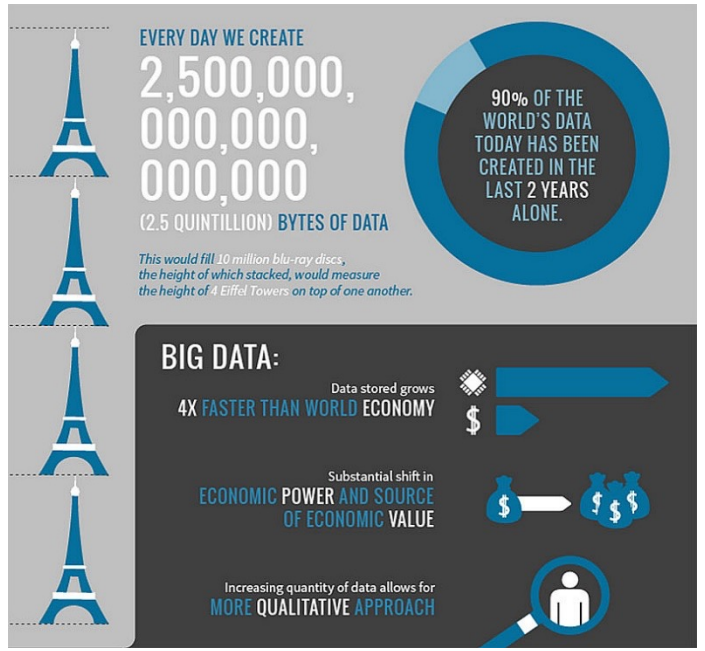
Elastic and independent scaling of storage & compute

https://www.seagate.com/files/www-content/our-story/trends/files/dataage-idc-report-final.pdf

It's all About Data!



IDC predicts that the Global Datasphere will grow from **45 Zettabytes** in 2019 to **175 Zettabytes** by 2025



In 2025 IDC predicts that **46%** of the world's stored data will reside in public cloud environments

"Much of today's economy relies on data, and this reliance will only increase in the future as companies capture, catalog, and cash in on data in every step of their supply

Managing data lifecycle the next "killer app" in the cloud

Recap

- Tons of use cases for serverless computing (& growing)
- Storage and compute disaggregation as the new norm
- Tons of data (& growing)

First challenge of serverless computing?

Challenge of Serverless Computing

Serverless Computing: One Step Forward, Two Steps Back

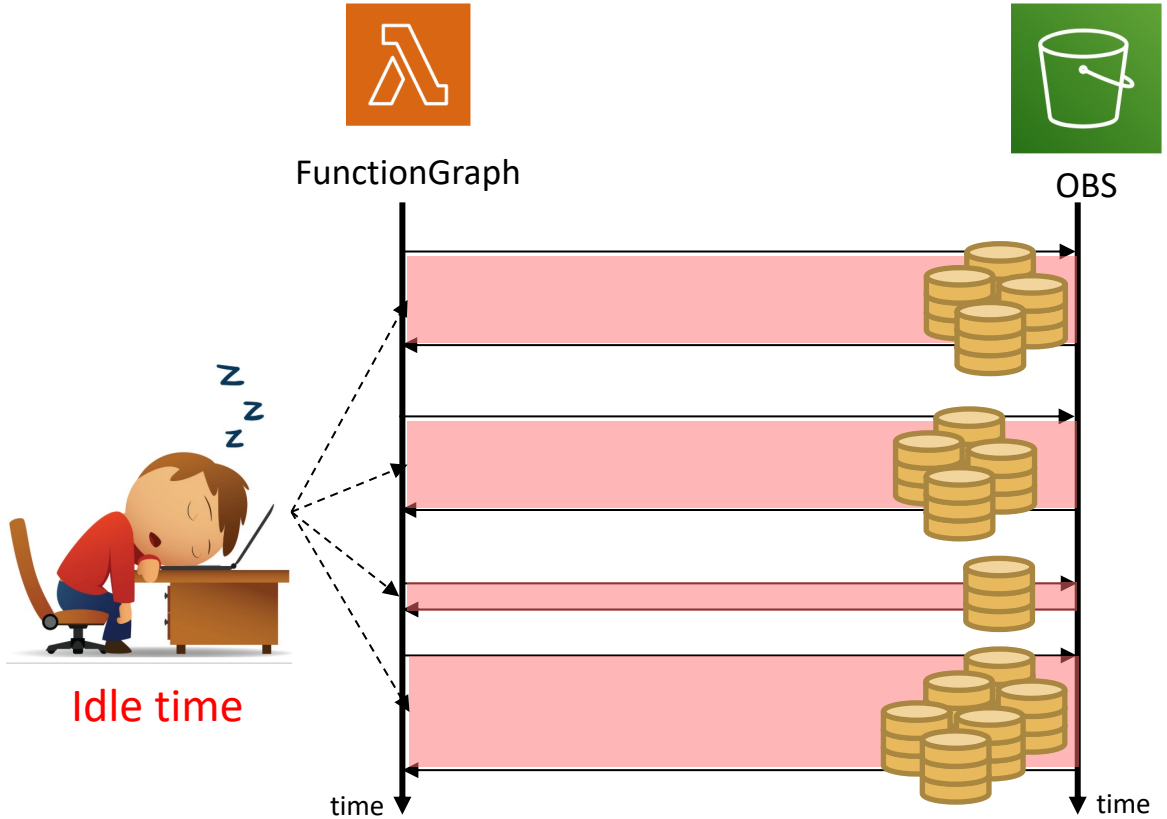
Joseph M. Hellerstein, Jose Faleiro, Joseph E. Gonzalez, Johann Schleier-Smith, Vikram Sreekanti, Alexey Tumanov and Chenggang Wu
UC Berkeley
{hellerstein,jmfaleiro,jegonzal,jssmith,vikrams,atumanov,cgwu}@berkeley.edu

CIRD'19

ABSTRACT
Serverless computing offers the potential to program the cloud in ... offers the attractive notion of a platform in the cloud where developers simply upload their code, and the platform executes it on

- Shipping data (state) to code (logic) paradigm

- User pays for additional idle time



Serverless computing forces function to access all data remotely

Use Case 1: Multimedia Processing

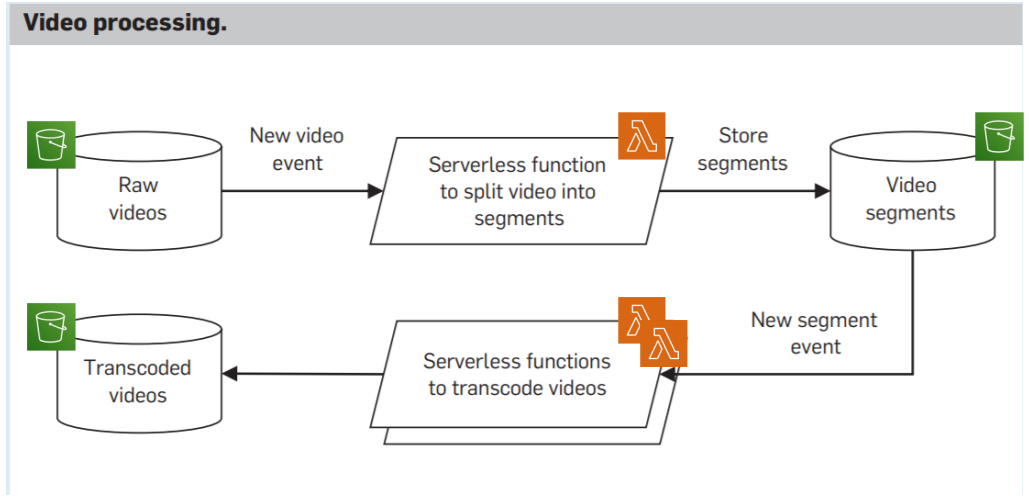
NETFLIX

Netflix & AWS Lambda Case Study

2014

Netflix is one of the world's largest online media streaming providers delivering almost 7 billion hours of videos to nearly 50 million customers in 60 countries per quarter. The company is planning to use AWS Lambda to build rule-based self-managing infrastructure and replace inefficient processes to reduce the rate of errors and save valuable time. Watch Neil Hunt, Netflix's Chief Product Officer, explain how the company can use event-based triggers to help automate the encoding process of media files, the validation of backup completions and instance deployments at scale, and the monitoring of AWS resources used by the organization.

Netflix uses serverless functions to process video files. The videos are uploaded to Amazon S3, which emits events that trigger Lambda functions that split the video and transcode them in parallel to different formats.



ATC USENIX'18

Understanding Ephemeral Storage for Serverless Analytics

Ana Klimovic[1], Yawen Wang[1], Christos Kozyrakis[1], Patrick Stuedi[2], Jonas Pfefferle[2], and Animesh Trivedi[2]

[1]Stanford University
[2]IBM Research

Abstract

Serverless computing frameworks allow users to launch thousands of concurrent tasks with high elasticity and fine-grain resource billing without explicitly managing data analytics. Several frameworks are being developed which leverage serverless computing to exploit high degrees of parallelism in analytics workloads and achieve near real-time performance [13, 17, 10].

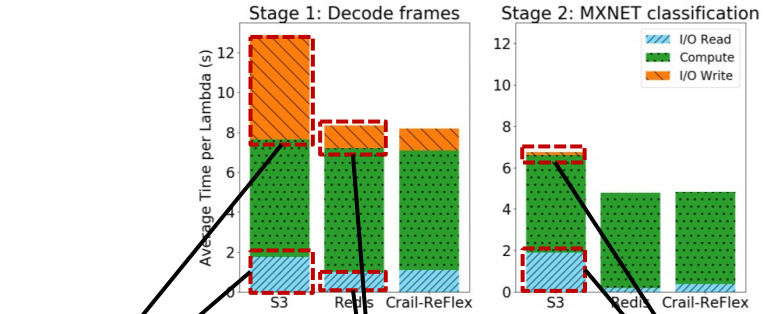
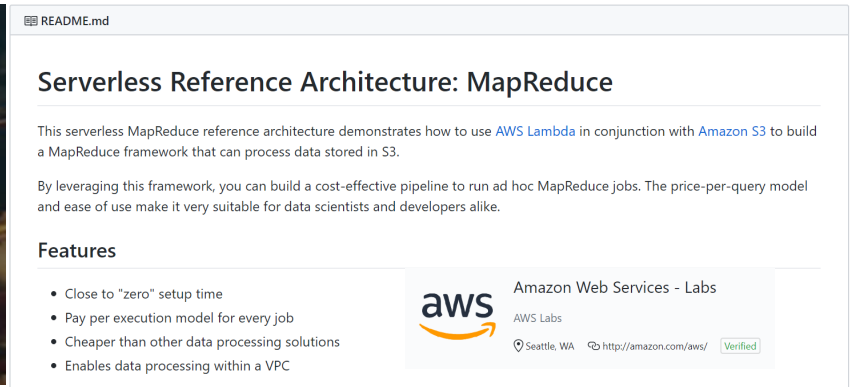


Figure 7: Video analytics I/O vs. compute breakdown, storing ephemeral data in S3, Redis and Crail-ReFlex.

~53% time S3 I/O ~25% time I/O ~33% time S3 I/O

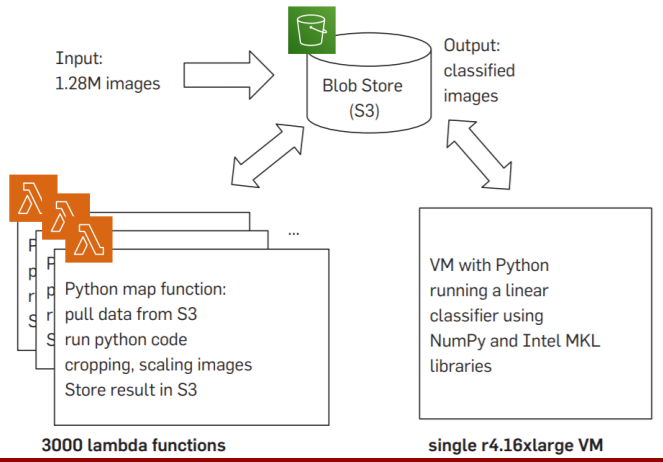
Serverless functions (video processing) spend up to ~50% of execution time in S3

Use Case 2: MapReduce Analytics



PyWren is a Python-based system that utilizes the serverless framework to avoid development and management overhead of running MapReduce jobs. It is able to get up to 40TFLOPS peak performance from AWS Lambda, using AWS S3 for storage and caching. A similar reference architecture has been proposed by AWS Labs. PyWren exemplifies a class of use cases that uses a serverless platform for highly parallel analytics workloads.

Map + monolithic Reduce PyWren example implementing ImageNet Large Scale Visual Recognition Challenge.



ATC USENIX'18

Understanding Ephemeral Storage for Serverless Analytics

Ana Klimovic[1], Yawen Wang[1], Christos Kozyrakis[1], Patrick Stuedi[2], Jonas Pfefferle[2], and Animesh Trivedi[2]

[1]Stanford University
[2]IBM Research

Abstract

Serverless computing frameworks allow users to launch thousands of concurrent tasks with high elasticity and fine-grain resource billing without explicitly managing data analytics. Several frameworks are being developed which leverage serverless computing to exploit high degrees of parallelism in analytics workloads and achieve near real-time performance [13, 17, 10].

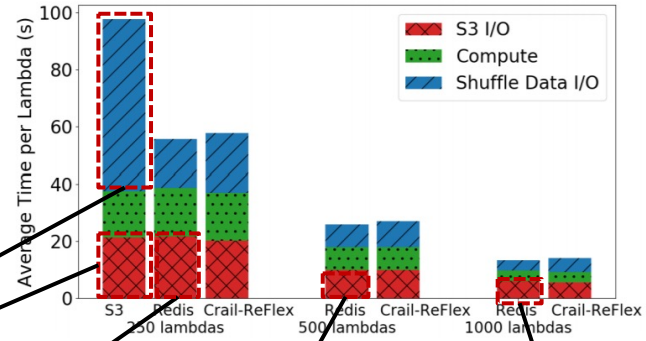
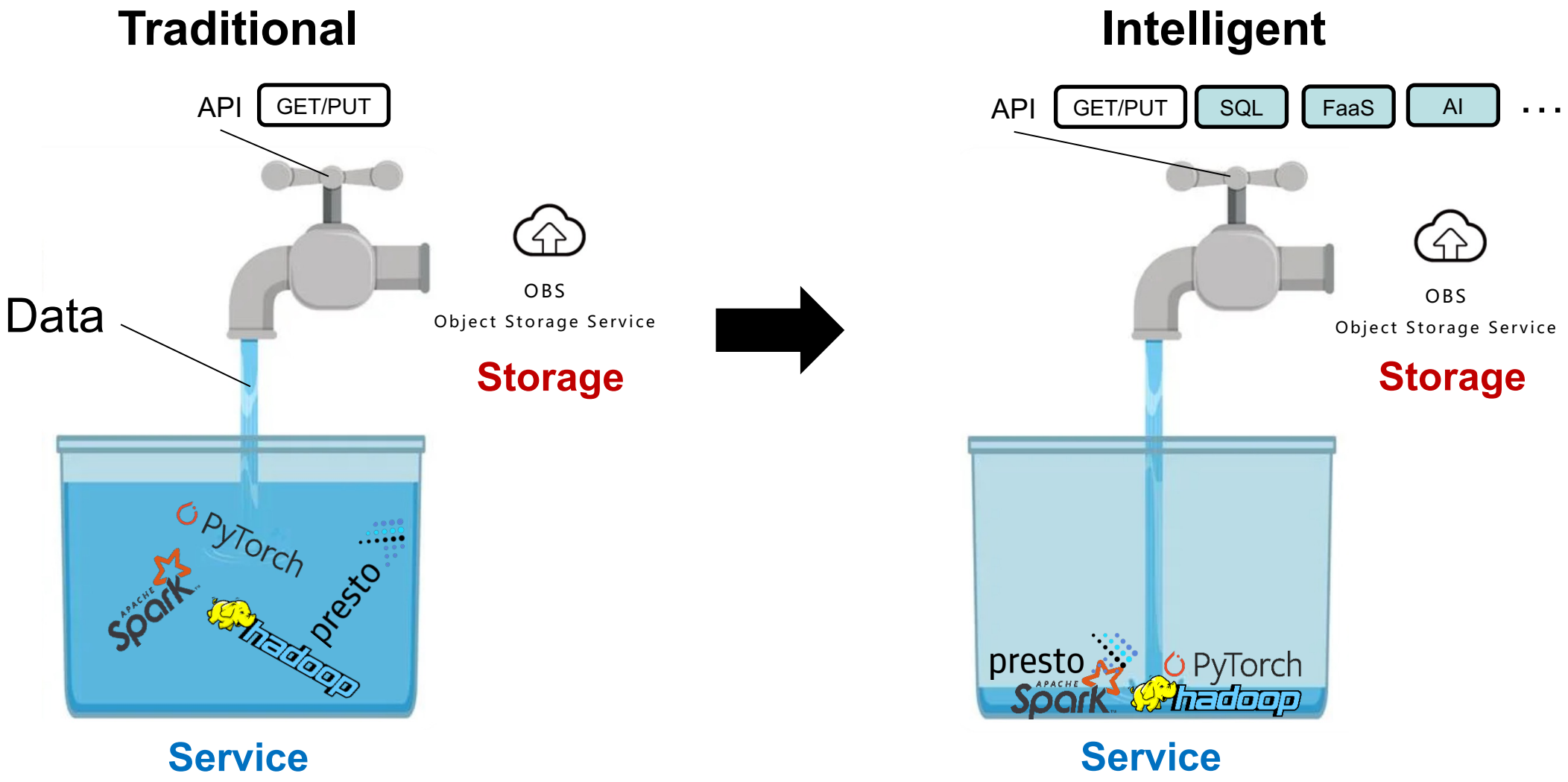


Figure 6: Average time per lambda for 100GB sort. S3 gives I/O rate limit errors with over 250 lambdas.

~80% time S3 I/O ~35% time I/O ~33% time I/O ~50% time I/O

Serverless functions (MapReduce Analytics) spend up to ~80% of execution time in S3

R&D Direction: Intelligent Object Storage



In-object storage FaaS capabilities (among others)

Business Layout: Competitive Analysis

Myriad of intelligent features in object storage ready for **commercial use today**:

Features	Amazon S3	ABS	Alibaba OSS	Huawei OBS
Key-Value API	Yes	Yes	Yes	Yes
Index API		ABS Blob Index	Data Indexing	
SQL API	S3 Select	Query Acceleration	OSS Select	OBS Select
Image API			OSS IMG	OBS Image Processing
Video API			OSS Capture video snapshots	
Document API			OSS IMM Document transform	
AI/ML API			OSS IMM Facial recognition OSS IMM Image recognition	
Lambda API	S3 Object Lambda			OBS Lambda

Amazon Simple Storage Service

Microsoft Azure Blob Storage

OSS Object Storage Service

OBS Object Storage Service

All cloud providers moving into intelligent object storage

OBS Lambda: Function Templates

We define template functions and workflows

- Templates encapsulate functionality common of object types (e.g., documents, images)

Common:

- Compress
- Decompress
- Encrypt
- Decrypt
- ACL
- …

Documents:

- Wordcount
- Linecount
- Replace
- Copy
- Getline
- …

Media:

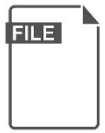
- Framecount
- Segmentsplit
- Scale
- Encode
- Decode
- …

Image:

- Scale
- Crop
- Encode
- Decode
- Copy
- …

ML:

- Inference
- Training
- Feature Extraction
- …

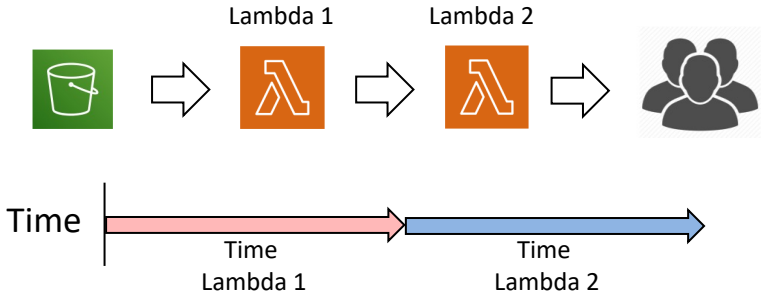


Templates ease programmability and raise level of abstraction of object storage

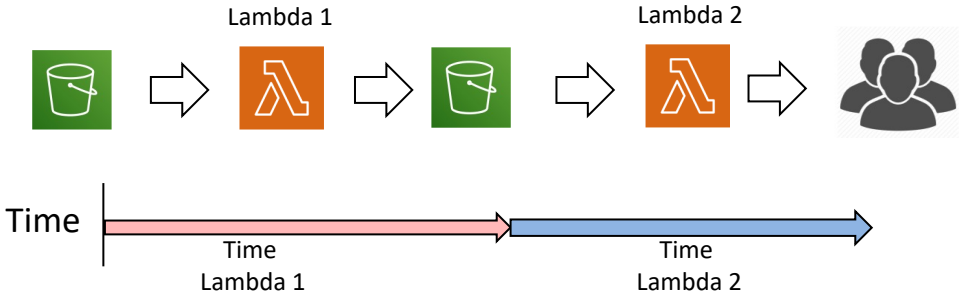
Further Challenges of Serverless Computing (1)

FunctionGraph

- No pipelining

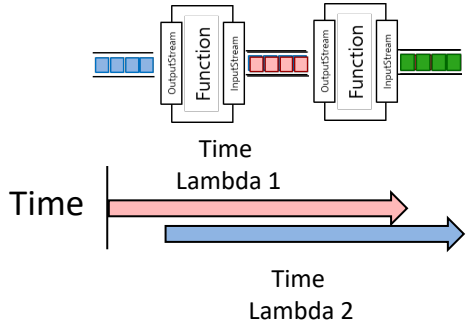


- Payload limit (> 6MB)

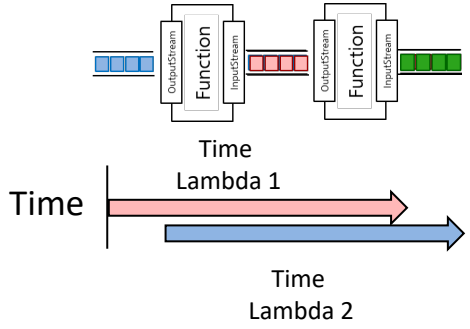


Ideal

- Pipelining



- Intercepts flow any object size



Ideal: Data-driven serverless computing (i.e., functions in the data path)

Further Challenges of Serverless Computing (2)

Performance and cost trade-offs of decomposing 1 big lambda into several smaller ones

Small lambdas

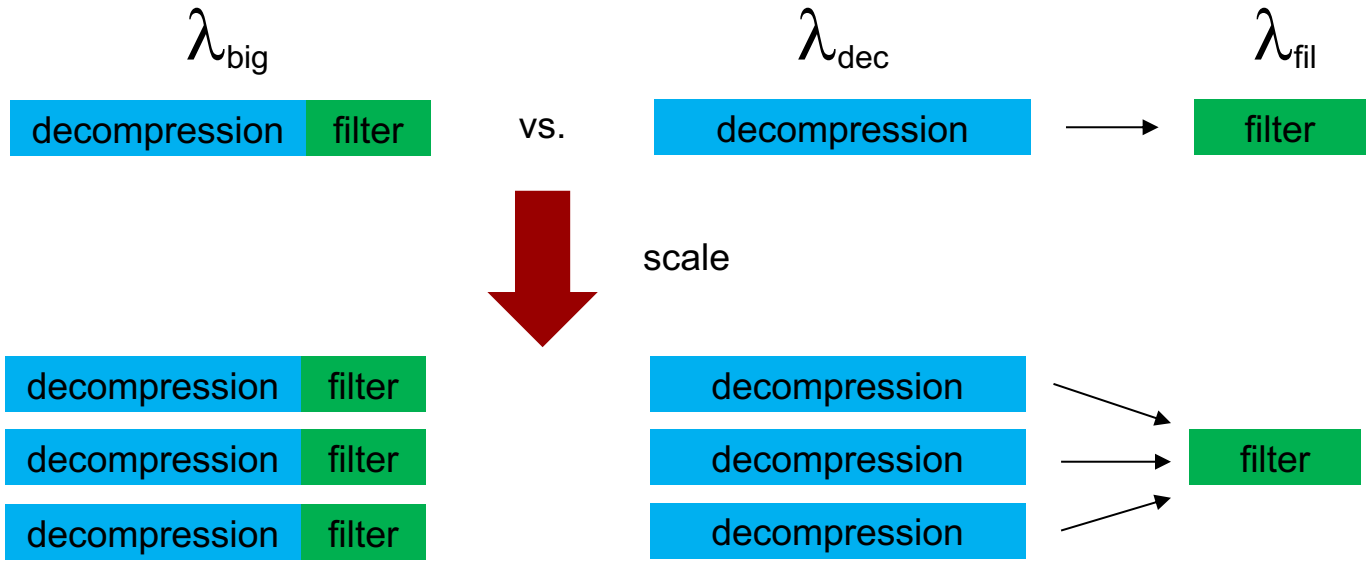
Pros:
- Start faster
- More predictable latency
- Better tail latency
- Can be scaled independently
- Pipeline output from one lambda to another

Cons:
- Unwieldy to write applications with
- Require communication between lambdas

Big lambdas

Pros:
- Easy to write

Cons:
- Different scaling requirements of the components
- Idle resources / wasteful duplication of lambdas



Research Challenges

- Decide when it is “better” to split lambdas
- Automatically split big lambdas
- Low latency communication between lambdas
- Pipeline output from one lambda as input to another in stream processing

Ideal: Efficient lambda decomposition

Beyond

Serverless computing great match for exploiting heterogenous hardware due to state-less nature:

- Compute (e.g., GPUs, FPGAs) or IO hardware (e.g., SmartNICs, SmartSSDs)

Integration of serverless computing in conventional provisioned services

- Redshift Lambda UDF support [2020] or Snowflake External Functions [2020]

Serverless computing great match for “pay-as-you-go” cloud services

- “SQL-on-FaaS” [SIGMOD’20] or “ML-on-FaaS” [SIGMOD’21]

State-less and serverless nature good opportunity for “running-anywhere”

- Idea of “CloudButton”

Conclusion

- Serverless Computing here to stay
- Serverless computing still has many challenges to solve
- Serverless computing service backbone of future cloud services

Thank you

www.huawei.com

Pannel

How do you see Serverless Computing in five years?

- Heterogeneous support
- Building block of other cloud services (e.g., SQL-on-FaaS, ML-on-FaaS)
- Integrated in provisioned cloud services
- Pushing towards computing in de-centralize clouds

Propose a technical challenge to solve in this field in the next years

- How to “compile” an existing application into FaaS?

What question would you want to ask another participant?

- Can FaaS become de-facto building of cloud services (e.g., Data Warehouses, ML, …)