WoSC'23 Keynote:
Short-lived Clouds

Intelligent Cloud Technologies Lab (ICTL)

Dr. Javier Picorel / Engineering Manager - Huawei Cloud R&D

12/2023

Security Level:

HUAWEI | MUNICH RESEARCH CENTER

Welcome & BIO

Dr. Javier Picorel

- EPFL — 2011-2017: PhD CS *"Compute Architecture/DBMS/OS/Cloud"*
- HUAWEI — 2018-2019: Senior/Principal Engineer – OS R&D Dept.
- HUAWEI CLOUD — 2019+: Engineering Manager – Huawei Cloud R&D

Find me @ Middlewar'23 and let's chat!

HUAWEI CLOUD: Empowering Applications and Harnessing the Value of Data for an Intelligent World

600+
e-Government clouds in China

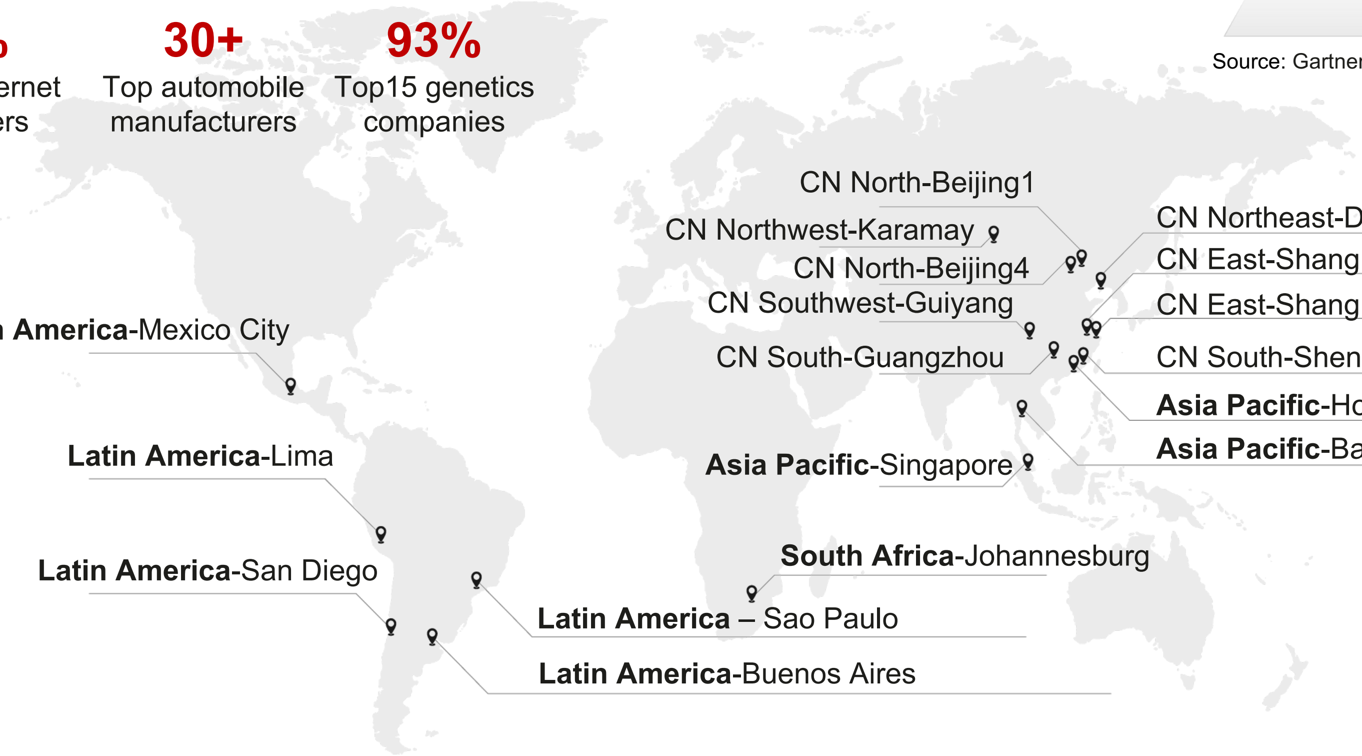
220+
Financial customers

300+
SAP on Cloud customers

70%
Top 50 Internet customers

30+
Top automobile manufacturers

93%
Top15 genetics companies



Fastest-growing cloud

No.2
China market

No.5
Global market

Source: Gartner: Market Share: IT Services, Worldwide 2020, IaaS market

23
Regions

45
AZs

2500+
CDN nodes

*Huawei partners' public clouds included

One-stop global services Relying on Huawei's **30 years** of experience in B2B services and global service teams

HUAWEI MUNICH RESEARCH CENTER

HUAWEI CLOUD Technology Stack: Continuously Innovating to Develop a Full Range of Products

220+ cloud services for **15** industries, and **210+** general and industrial solutions

Security system + Operations system

Marketplace
- Featured Products Mall
- SaaS applications
- Industry suites
- Professional services

Industry solutions
- Pan-government
- Pan-enterprise
- Pan-finance
- Pan-Internet

4 types of connections
- Connecting applications
- Connecting developers
- Connecting things
- Connecting organizations

3 types of technology enablement services

ROMA (application enablement)
Manage applications throughout lifecycle
- Industry application assets - ROMA Exchange
- Common services & capabilities - -ROMA Service Core
- E2E agile development - ROMA Factory
- Unified platform - ROMA Connect

ModelArts (AI enablement)
Accelerate AI adoption by industry
- Knowledge compute
- Industry Suite - ModelArts Pro
- AI development pipeline - ModelArts Fundamental

DAYU (data enablement)
Unleash the value of data
- Data asset center - DAYU Hub
- Data enablement suite - DAYU Ekit
- Intelligent data lake - FusionInsight

1 foundation: cloud infrastructure powered by the QingTian architecture
- Compute services
- Storage services
- Network services
- Security services
- Database services
- QingTian architecture

Edge
- CDN
- IEF
- IEC
- IES

Device
- Device IoT module
- IoT SDK

5 deployment scenarios
- Hybrid cloud
- Full-stack dedicated cloud
- Public cloud
- Edge data center
- Enterprise edge site

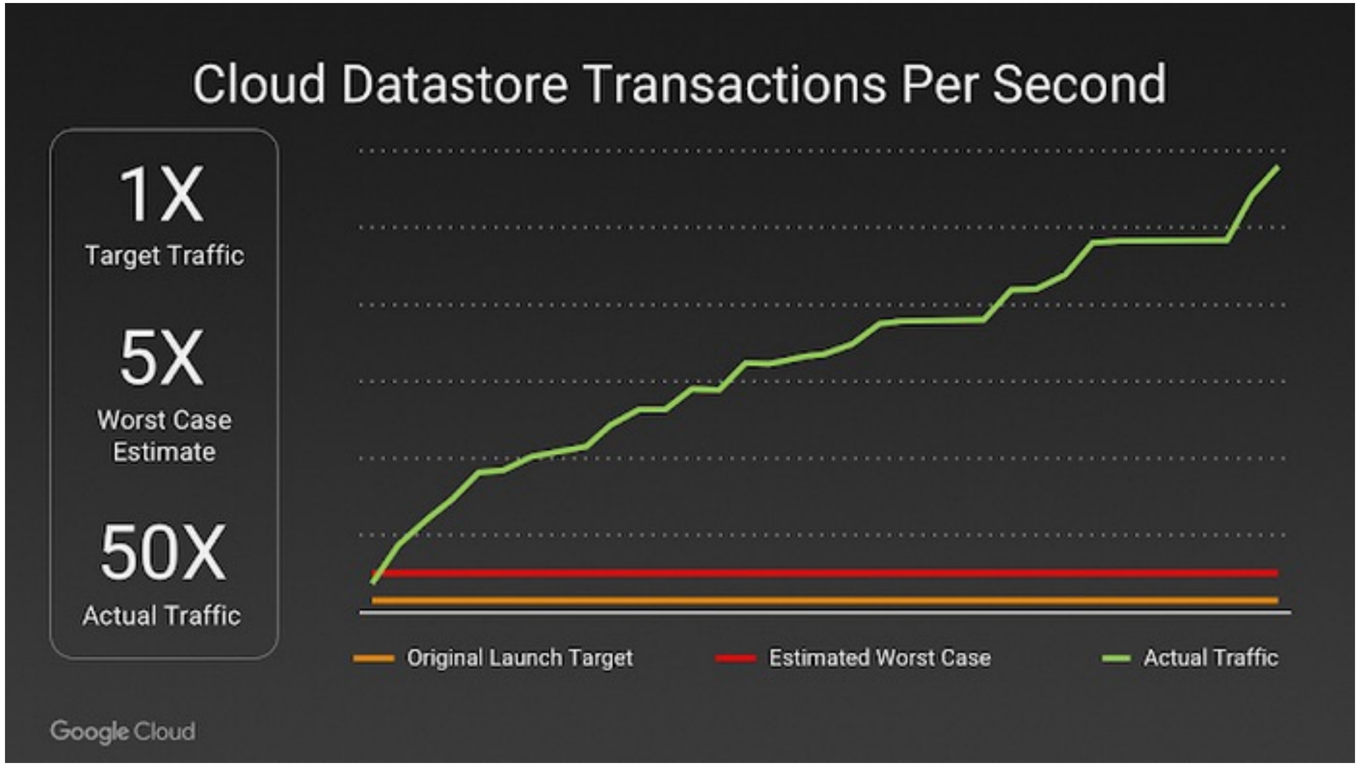
HUAWEI | MUNICH RESEARCH CENTER

Why Cloud Computing?

Google Cloud

Bringing Pokémon GO to life on Google Cloud

September 30, 2016



https://cloud.google.com/blog/products/containers-kubernetes/bringing-pokemon-go-to-life-on-google-cloud

When you cannot buy server blades fast enough (or you don't know how many to buy)

Why Serverless Computing?

DOI:10.1145/3406011

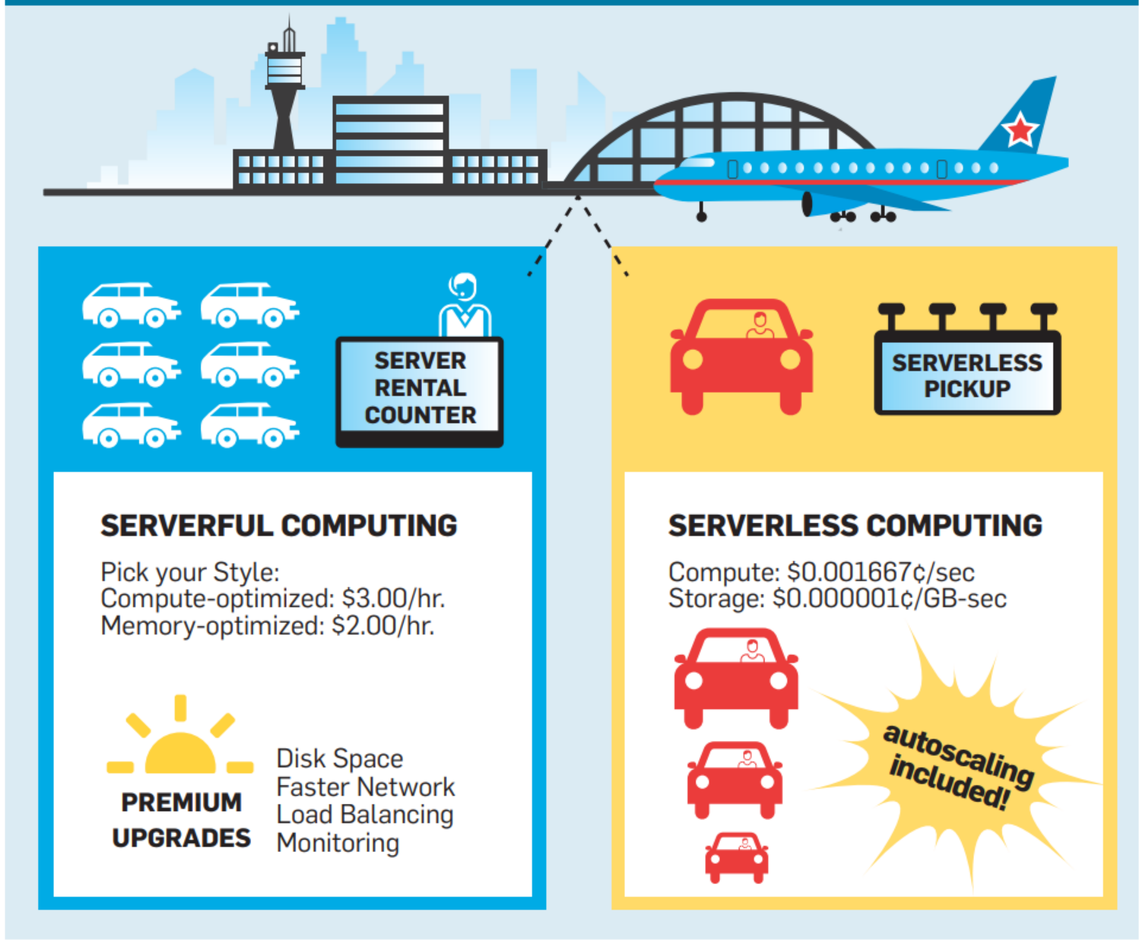
The evolution that serverless computing represents, the economic forces that shape it, why it could fail, and how it might fulfill its potential.

BY JOHANN SCHLEIER-SMITH, VIKRAM SREEKANTI, ANURAG KHANDELWAL, JOAO CARREIRA, NEERAJA J. YADWADKAR, RALUCA ADA POPA, JOSEPH E. GONZALEZ, ION STOICA, AND DAVID A. PATTERSON

What Serverless Computing Is and Should Become: The Next Phase of Cloud Computing

https://cacm.acm.org/magazines/2021/5/252179-what-serverless-computing-is-and-should-become/

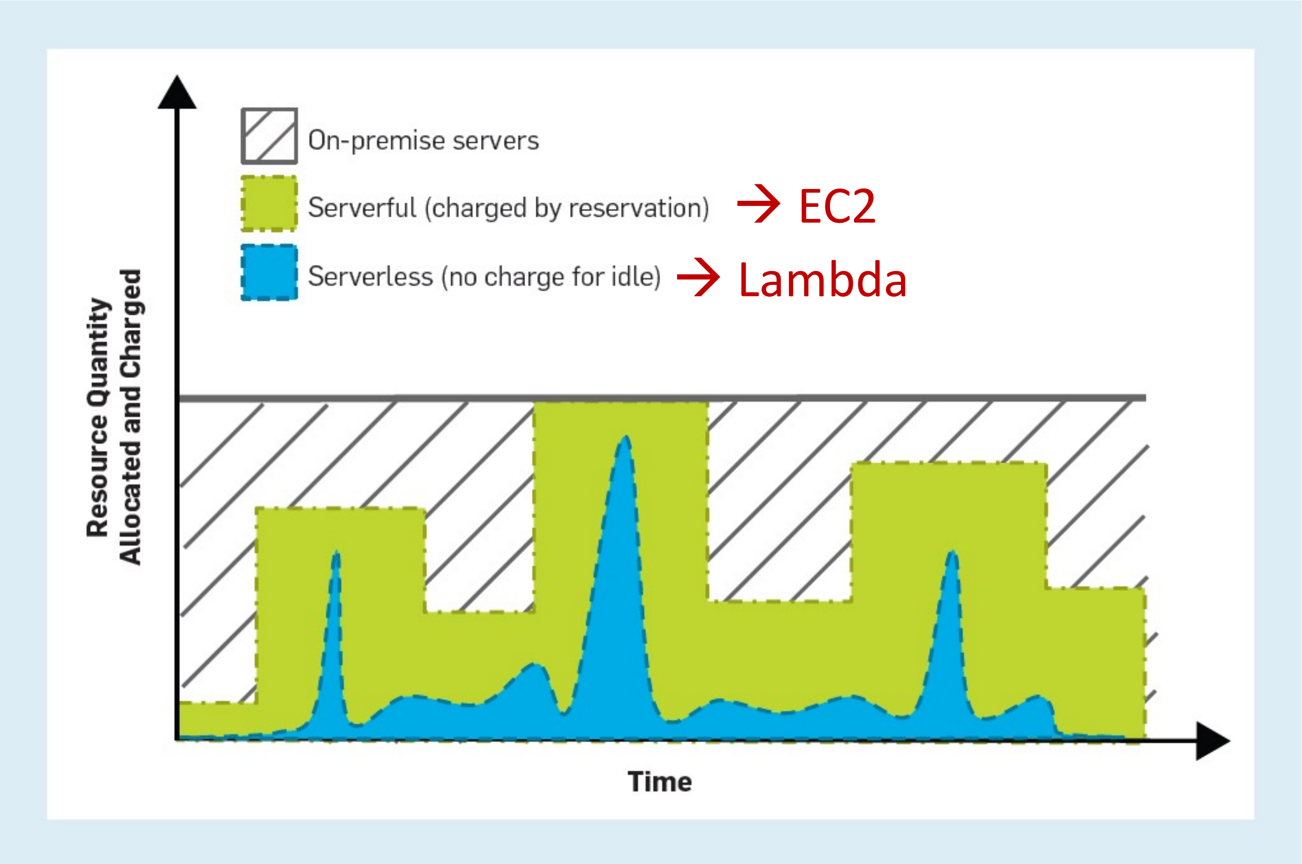
Figure 1. Cloud computing approaches compared to rides from an airport: Serverful as renting a car and serverless as taking a taxi ride.



Just pay for the ride and forget about operating, driving, and maintaining the vehicle

Cost of Serverless Computing

What do you pay for?



https://cacm.acm.org/magazines/2021/5/252179-what-serverless-computing-is-and-should-become/

How much do you pay for it?

Table 1. Resource unit prices in AWS (as of September 2022).

Resource type	Lambda		EC2 on-demand	EC2 spot
CPU (¢/core-h)	10	~2x →	4.8	1.1
RAM (¢/GB-h)	6	~5x →	1.2	0.27
Network (¢/Gbps-h)	85.71	~5x →	15.36	3.53

Using Cloud Functions as Accelerator for Elastic Data Analytics, Haoqiong et al.

You only pay for what you use…but cost is *~2x-~5x more expensive given same time*

Outline

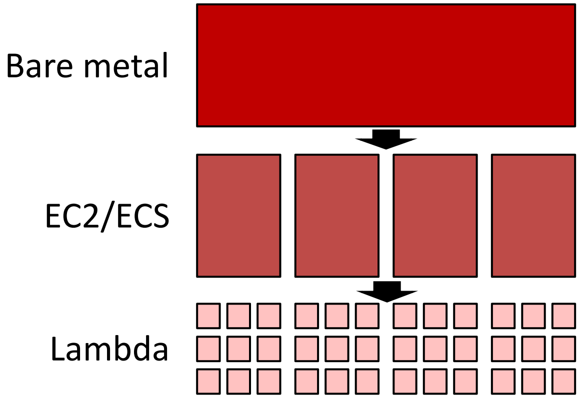
- Introduction
- The good, the bad, and the ugly
- Toward short-lived clouds
- Serverless computing in AI-centric clouds
- Conclusion

The Good, the Bad, and the Ugly

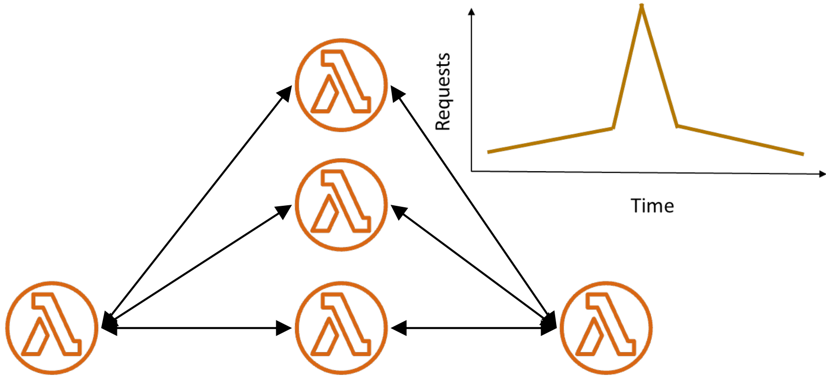
Good:

- Zero provisioning
- Autoscaling & Fast start-up times
- Fine-grained pricing
- Fine-grained resource allocation

Resource Allocation

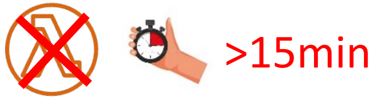


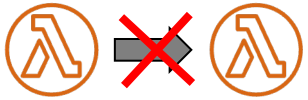
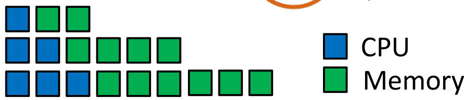
Autoscaling

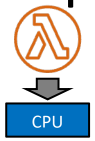


"From zero to infinite, no provisioning required"

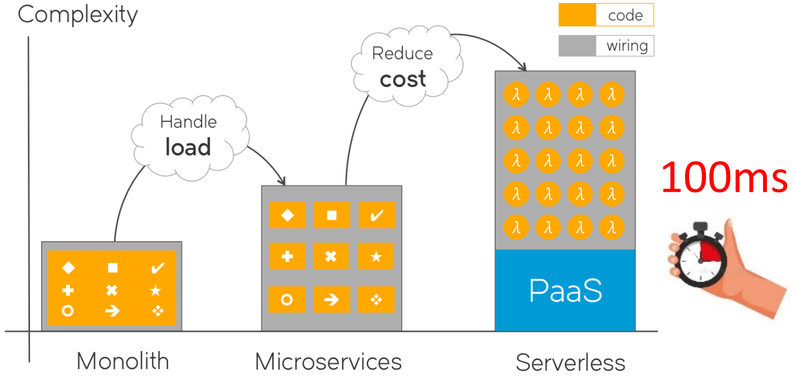
Bad:

- Limited execution time 

- Lack of lambda-to-lambda communication 
- Fixed memory-to-compute ratios 

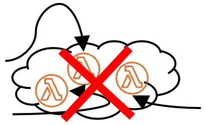
- Restricted to CPUs 

Pay-as-you-go



"You only pay for every 100ms that code is running and number of times it's triggered"

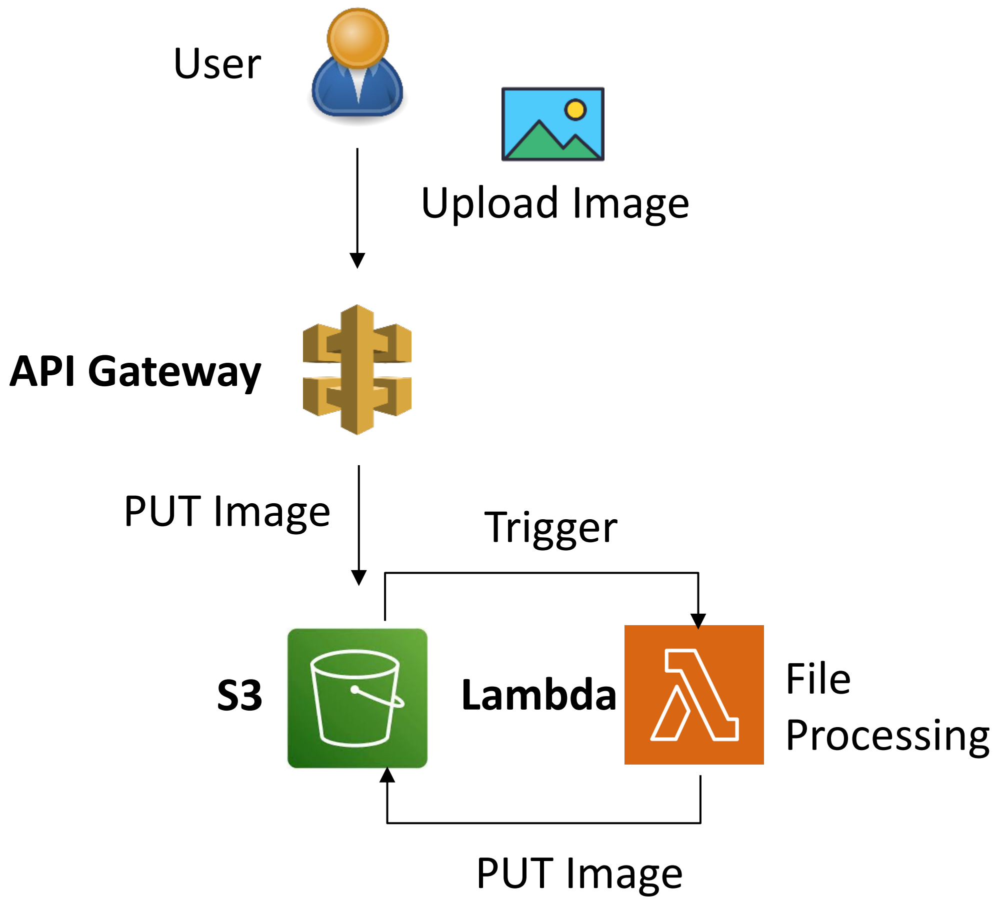
Ugly:

- No control of execution and deployment 

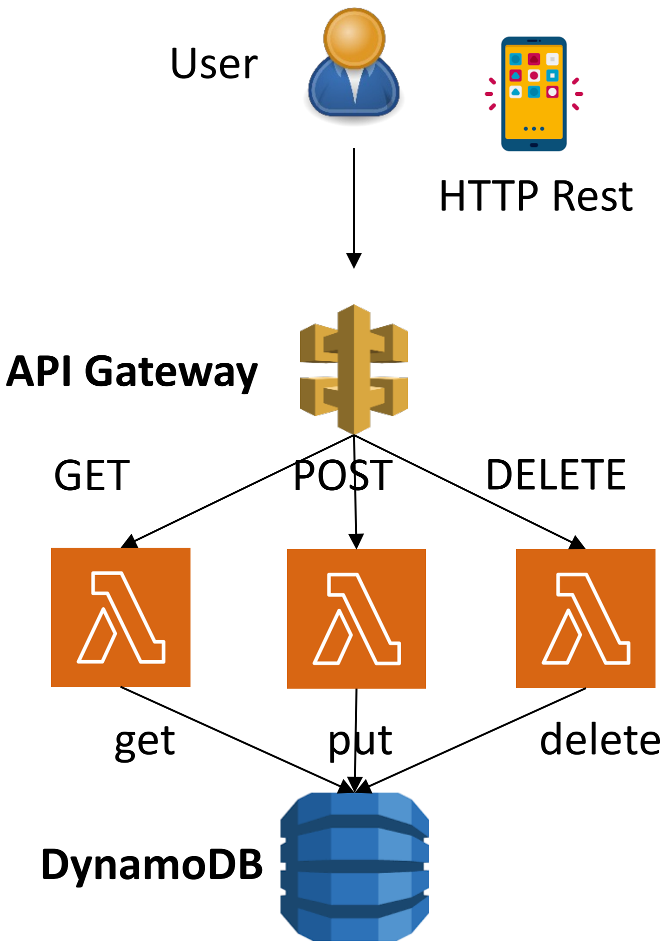
What are the implications to real-world applications?

Mostly Simple Applications Benefit from Serverless Today

File Processing



Web Application



Serverless computing exhibits several limitations but…which of these are really fundamental?

Bad: Limited Execution Time

Release of AWS Lambda

2014

Max. Execution Time:
5 min

Update on AWS Lambda Features

2018

Max. Execution Time:
15 min

Configuring Lambda function options

PDF | RSS

Configuring function timeout (console)

Lambda runs your code for a set amount of time before timing out. *Timeout* is the maximum amount of time in seconds that a Lambda function can run. The default value for this setting is 3 seconds, but you can adjust this in increments of 1 second up to a maximum value of 15 minutes.

https://docs.aws.amazon.com/lambda/latest/dg/configuration-function-common.html#configuration-timeout-console

Not fundamental → Arbitrary decision (some average) on current limited execution time

Bad: Lack of Lambda-to-Lambda Communication

Boxer [Wawrzoniak'23]

Relies on NAT punching

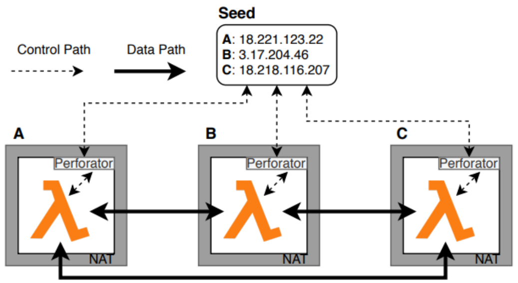


Figure 1: Networked serverless functions use a Seed process to connect functions during the startup. After the startup phase, the seed process is no longer needed.

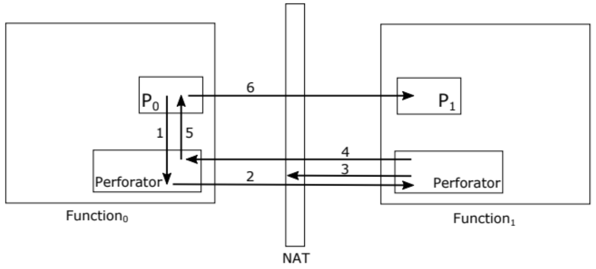


Figure 3: Opening TCP connection by process P_0 to process P_1 running in a remote function.

https://arxiv.org/pdf/2202.06646.pdf

XDT [Ustiugov'23]

Extends Knative Queue/Proxy Component

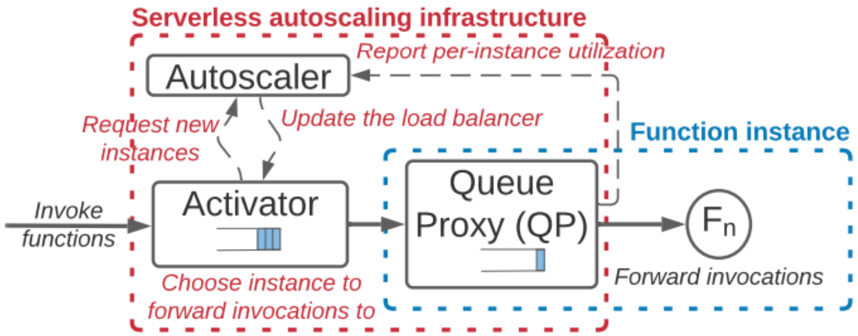


Figure 1: Operation of serverless autoscaling infrastructure.

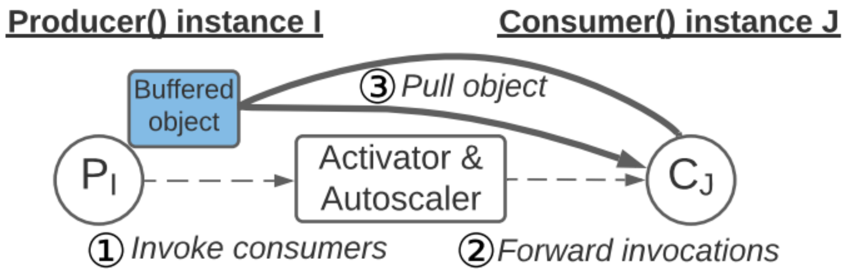


Figure 3: XDT architecture overview.

https://arxiv.org/pdf/2309.14821v1.pdf

Not fundamental → Published and on-going work shows it's possible

Bad: Fixed Memory-to-Compute Ratios

Memory and computing power

Memory is the principal lever available to Lambda developers for controlling the performance of a function. You can configure the amount of memory allocated to a Lambda function, between 128 MB and 10,240 MB. The Lambda console defaults new functions to the smallest setting and many developers also choose 128 MB for their functions.

The amount of memory also determines the amount of virtual CPU available to a function. Adding more memory proportionally increases the amount of CPU, increasing the overall computational power available. If a function is CPU-, network- or memory-bound, then changing the memory setting can dramatically improve its performance.

https://docs.aws.amazon.com/lambda/latest/operatorguide/computing-power.html

With Great Freedom Comes Great Opportunity: Rethinking Resource Allocation for Serverless Functions

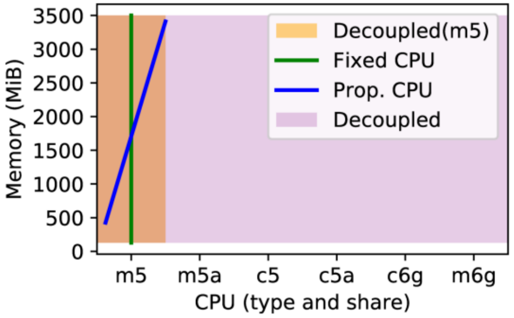
Muhammad Bilal*
IST(ULisboa)/INESC-ID and UCLouvain

Marco Canini
KAUST

Rodrigo Fonseca
Azure Systems Research

Rodrigo Rodrigues
IST(ULisboa)/INESC-ID

https://sands.kaust.edu.sa/papers/serverless.eurosys23.pdf



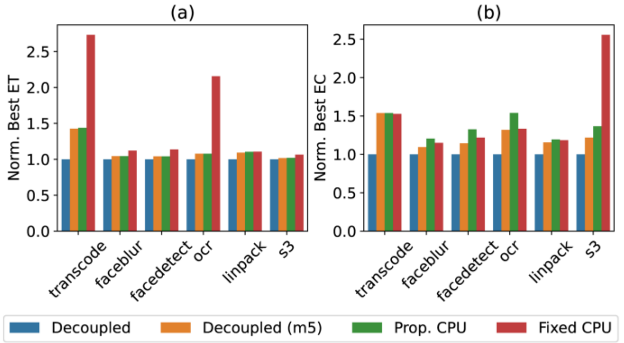


Figure 3. Potential gains within each search space. The graphs show the best (a) Execution Time (ET) and (b) Execution Cost (EC) of each function across different search spaces, normalized to the overall best configuration.

Not fundamental → Arbitrary decision (some average) on current ratios

Bad: Restricted to Off-the-shelf CPUs

aws

AWS Lambda Functions Powered by AWS Graviton2 Processor – Run Your Functions on Arm and Get Up to 34% Better Price Performance

https://aws.amazon.com/blogs/aws/aws-lambda-functions-powered-by-aws-graviton2-processor-run-your-functions-on-arm-and-get-up-to-34-better-price-performance/

Architecture Info
Choose the instruction set architecture you want for your function code.
- x86_64
- arm64

Alibaba Cloud

Introduction to serverless GPUs

Updated at: 2023-10-24 11:52

"Serverless GPU" is an emerging cloud-based GPU service. Serverless GPUs provide on-demand GPU computing resources for you and you do not have to worry about the underlying infrastructure such as servers. Compared with resident GPU computing resources, serverless GPUs improve the resource utilization and elasticity and reduce costs. This topic describes the features and benefits of serverless GPUs.

https://www.alibabacloud.com/help/en/fc/use-cases/introduction-to-serverless-gpus

HUAWEI

HUAWEI CLOUD

GPU Functions

Updated on 2023-05-29 GMT+08:00

View PDF

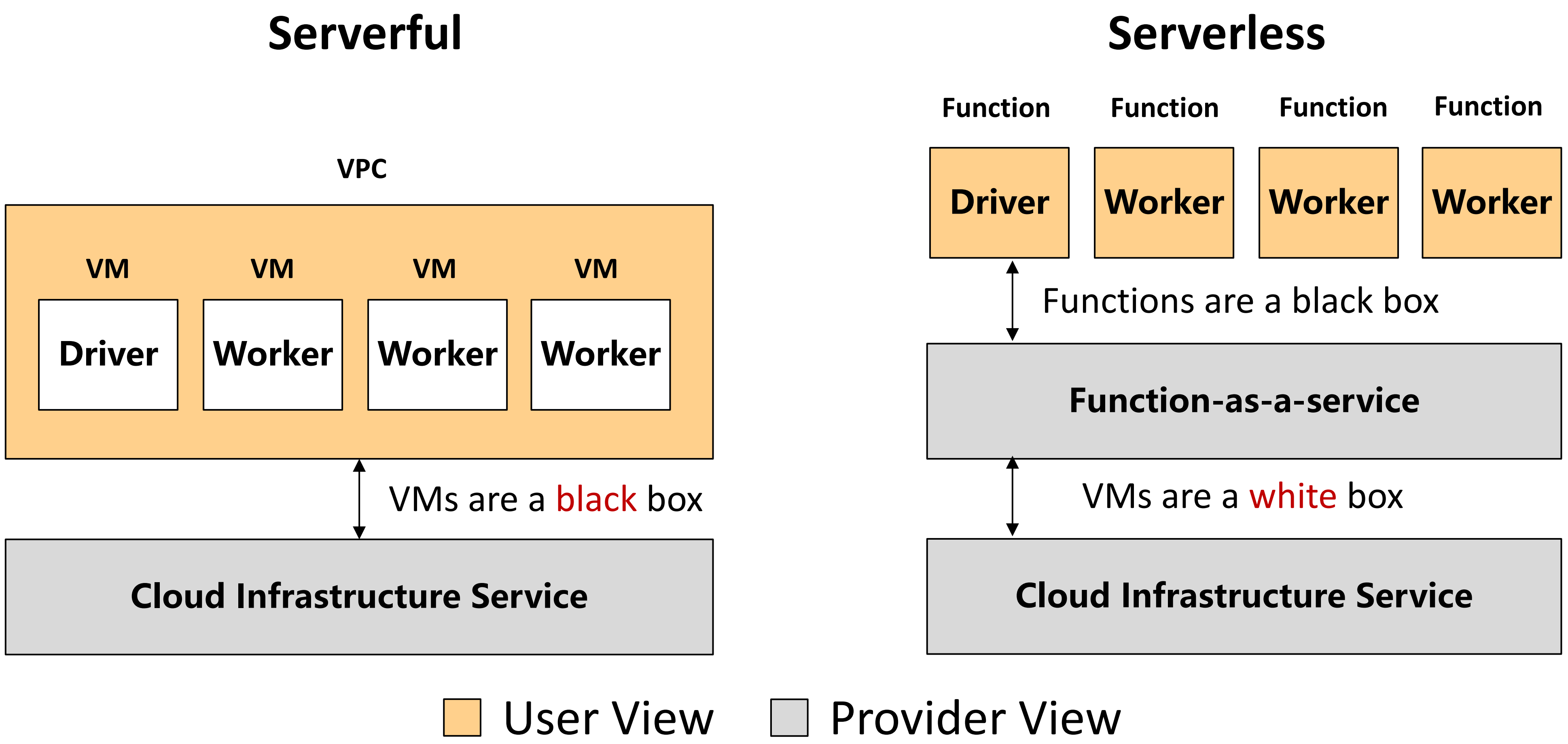
GPU functions provide GPU hardware acceleration for simulation, scientific computing, audio/videos, AI, and image processing to improve service efficiency.

The following table lists the GPU function specifications.

https://support.huaweicloud.com/intl/en-us/usermanual-functiongraph/functiongraph_01_2002.html

Not fundamental → Cloud providers have started offering serverless computing in GPUs

Ugly: No Control of Execution and Deployment



- Lack of control could indeed lead to terrible performance (e.g., locality, overlapping)

Fundamental → It's the contract between user and provider…but can I use it to my advantage?

Recap: The Good, the Bad, and the Ugly

Good:

- Zero provisioning
- Autoscaling & Fast start-up times
- Fine-grained pricing
- Fine-grained resource allocation

Bad:

- Limited execution time → Not fundamental
- Lack of lambda-to-lambda communication → Not fundamental
- Fixed memory-to-compute ratios → Not fundamental
- ~~Restricted to CPUs~~

Ugly:

- No control of execution and deployment → Fundamental, but can I use to my advantage?

What are the implications?

Outline

- Introduction
- The good, the bad, and the ugly
- Toward short-lived clouds
- Serverless computing in AI-centric clouds
- Conclusion

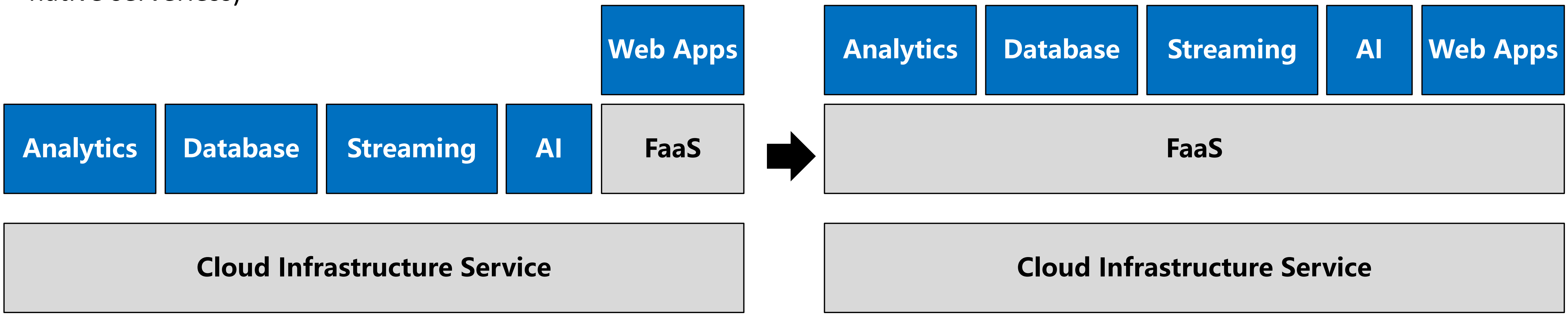
Our Vision: From Provisioned to Short-lived Clouds

Provisioned Cloud

Serverless version of these are based on provisioned software stacks just deployed by the provider (not native serverless)

Short-lived Cloud

Serverless-native services built from the ground up around serverless computing



Provisioned services reserve resources

- Regardless of demand or utilization

Zero provisioning → Lowest TCO

- Both providers & users benefit

Short-lived Clouds: Almost all cloud services built around ephemeral functions

Recipe to Achieve Short-lived Clouds

Good:
- Zero provisioning
- Autoscaling & Fast start-up times
- Fine-grained pricing
- Fine-grained resource allocation

Actually exploit the benefits of serverless computing!
- **Do what provisioned services cannot do**

Bad:
- Limited execution time
- Lack of lambda-to-lambda communication
- Fixed memory-to-compute ratios
- ~~Restricted to CPUs~~

Mitigate the limitations (for now)
- **Show providers that it is worth it**

Ugly:
- No control of execution and deployment

Turn the lack of control from the user side
Into an advantage for the provider
- **Expose relationships between functions**

Short-lived Clouds: Almost all cloud services built around ephemeral functions

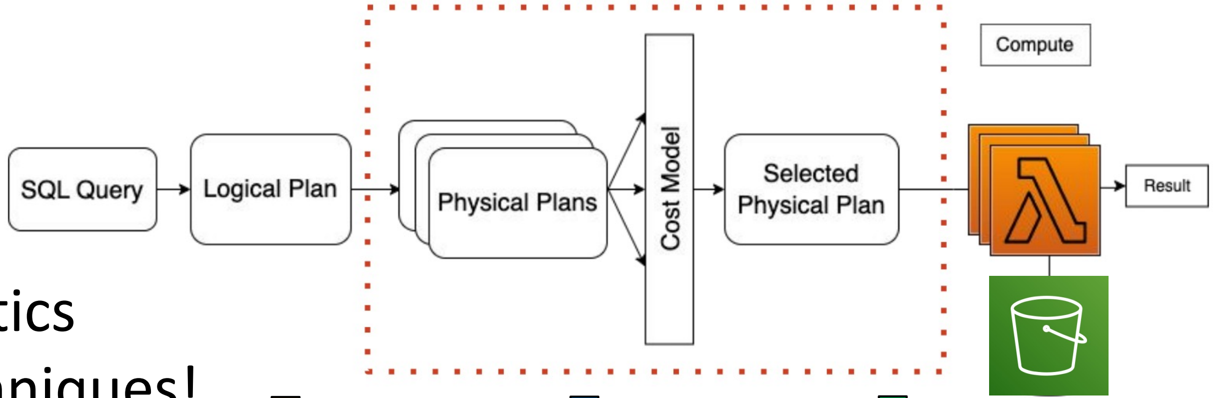
Exploit autoscaling

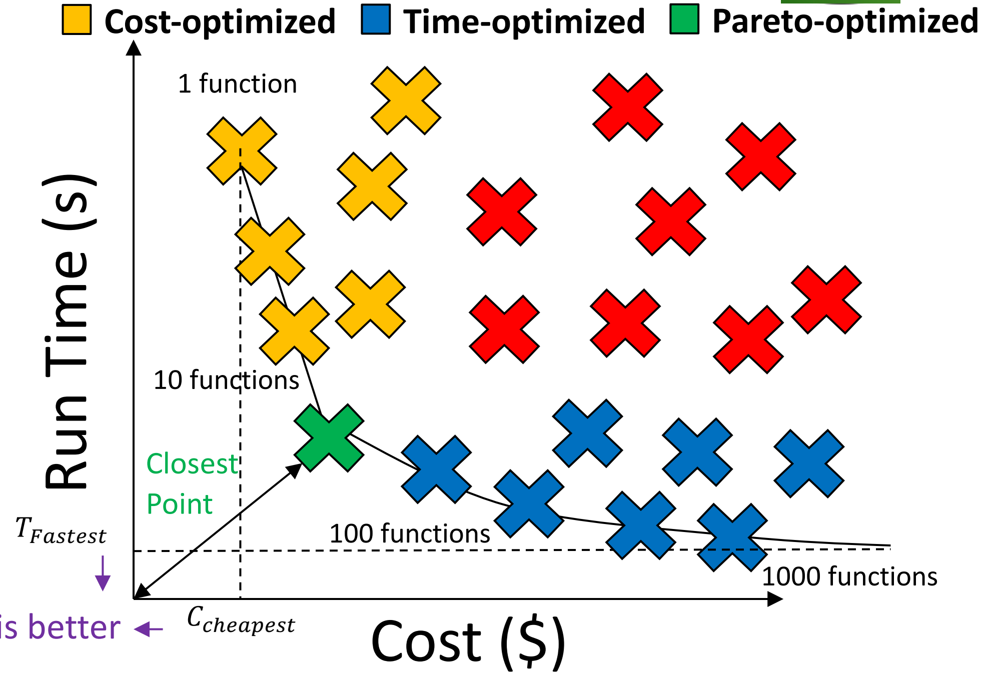
“Starling: A Scalable Query Engine on Cloud Function Services”, Perron et al.
“Lambada: Interactive Data Analytics on Cold Data Using Serverless Cloud Infrastructure”, Muller et al.
“Using Cloud Functions as Accelerator for Elastic Data Analytics, Bian et al.
“Resource Allocation in Serverless Query Processing”, Kassing et al.

Today’s provisioned platforms (VM-based):
- Slow to scale → Difficult to absorb bursts
- Always on → Expensive ($$$)

Insight: Serverless-native analytics are still analytics
- Can apply established query optimization techniques!

Serverless analytics benefits:
- (Almost) no cold start (100ms to a few seconds)
- Elasticity to visit entire Pareto frontier
  - Per query and per stage
- Able to match the resources to query/dataset
  - Able to achieve sweet spot



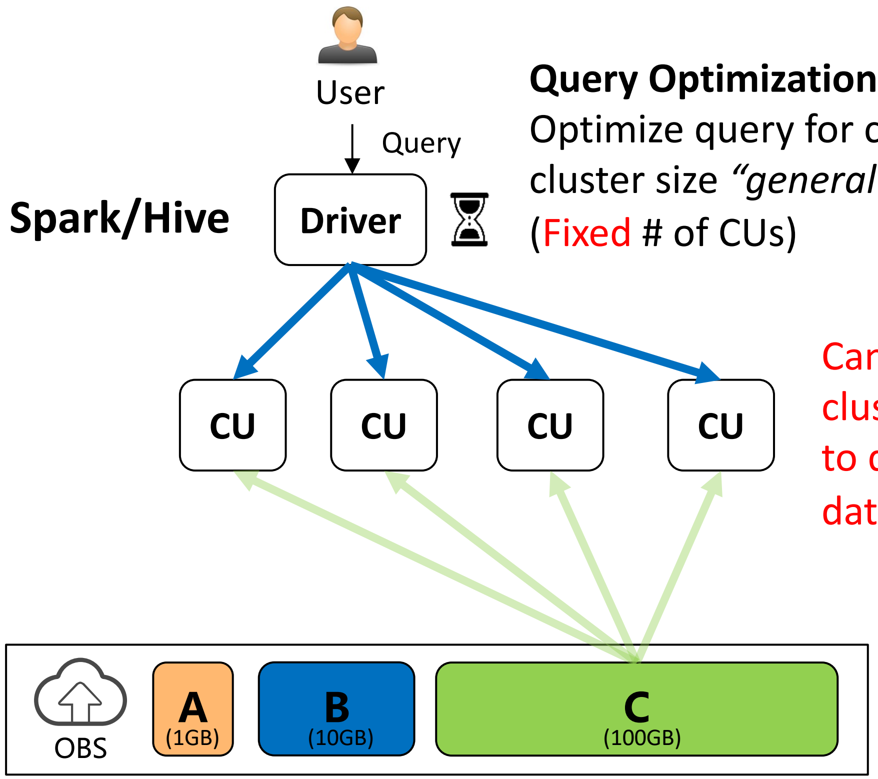


Serverless-native analytics allows to tailor resources to each query and dataset size

Exploit autoscaling

"*Starling: A Scalable Query Engine on Cloud Function Services*", Perron et al.
"Lambada: Interactive Data Analytics on Cold Data Using Serverless Cloud Infrastructure", Muller et al.
"Using Cloud Functions as Accelerator for Elastic Data Analytics, Bian et al.
"Resource Allocation in Serverless Query Processing", Kassing et al.

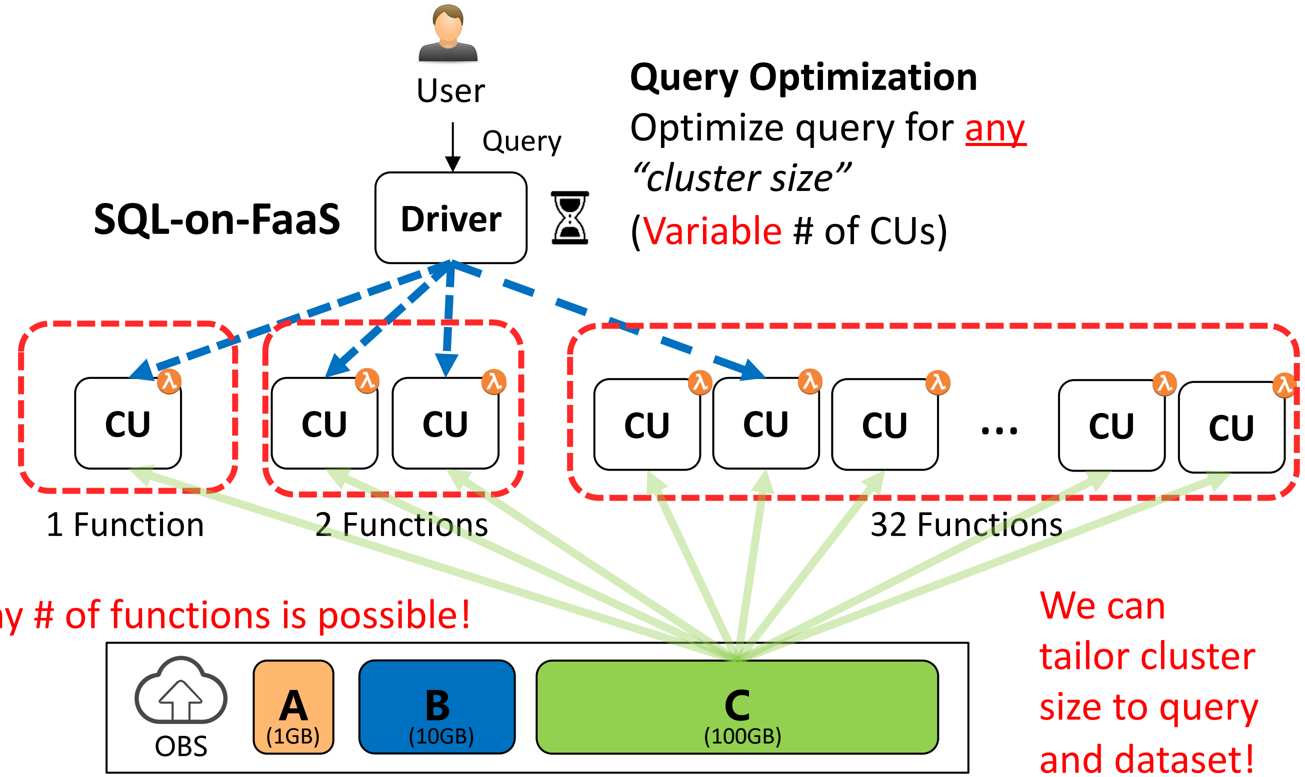
Provisioned (user or provider)



Query Optimization
Optimize query for current cluster size *"general purpose"* (Fixed # of CUs)

Cannot tailor cluster size to query and dataset!

Serverless-native



Query Optimization
Optimize query for any *"cluster size"* (Variable # of CUs)

Any # of functions is possible!

We can tailor cluster size to query and dataset!

Don't be shy, you can visit the entire Pareto prontier!

State Checkpointing for Timeout-resilient Functions

Checkpoint state for resuming function after timeout

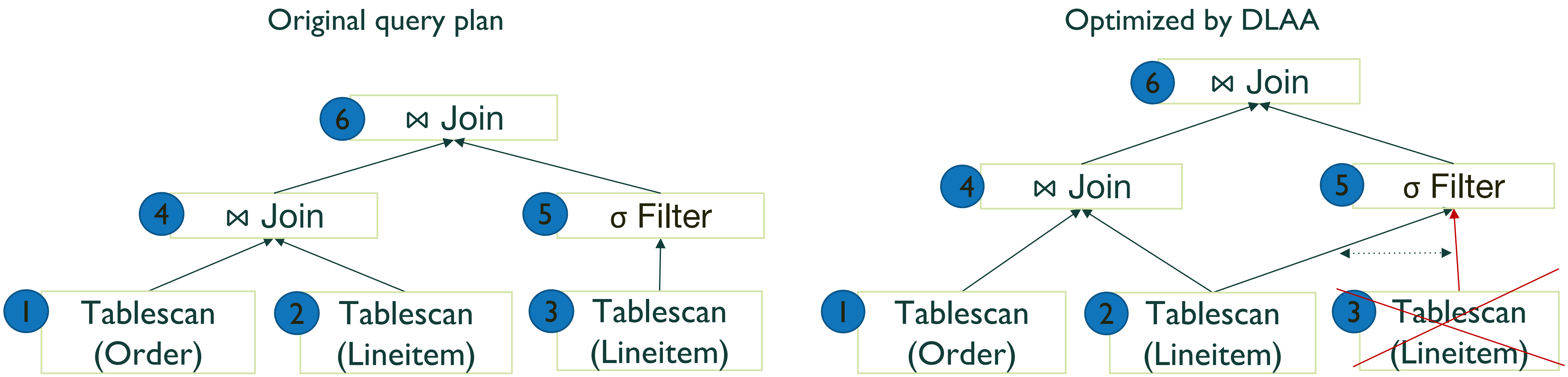
- Data-plane functions can have huge states (e.g., GBs)
- Control-plane functions tend to have small states (e.g., KBs)
- Timeout of 15 min coarse enough for low overhead (in most cases)

Timeout-resilient functions via user-level checkpointing

Create Locality to Reduce Communication

Logical Operator Fusion: Data Locality Aware Algorithm (DLAA)

- Group vertices read the same(similar) data content.
- Transfer all the edges in the group to one single vertex

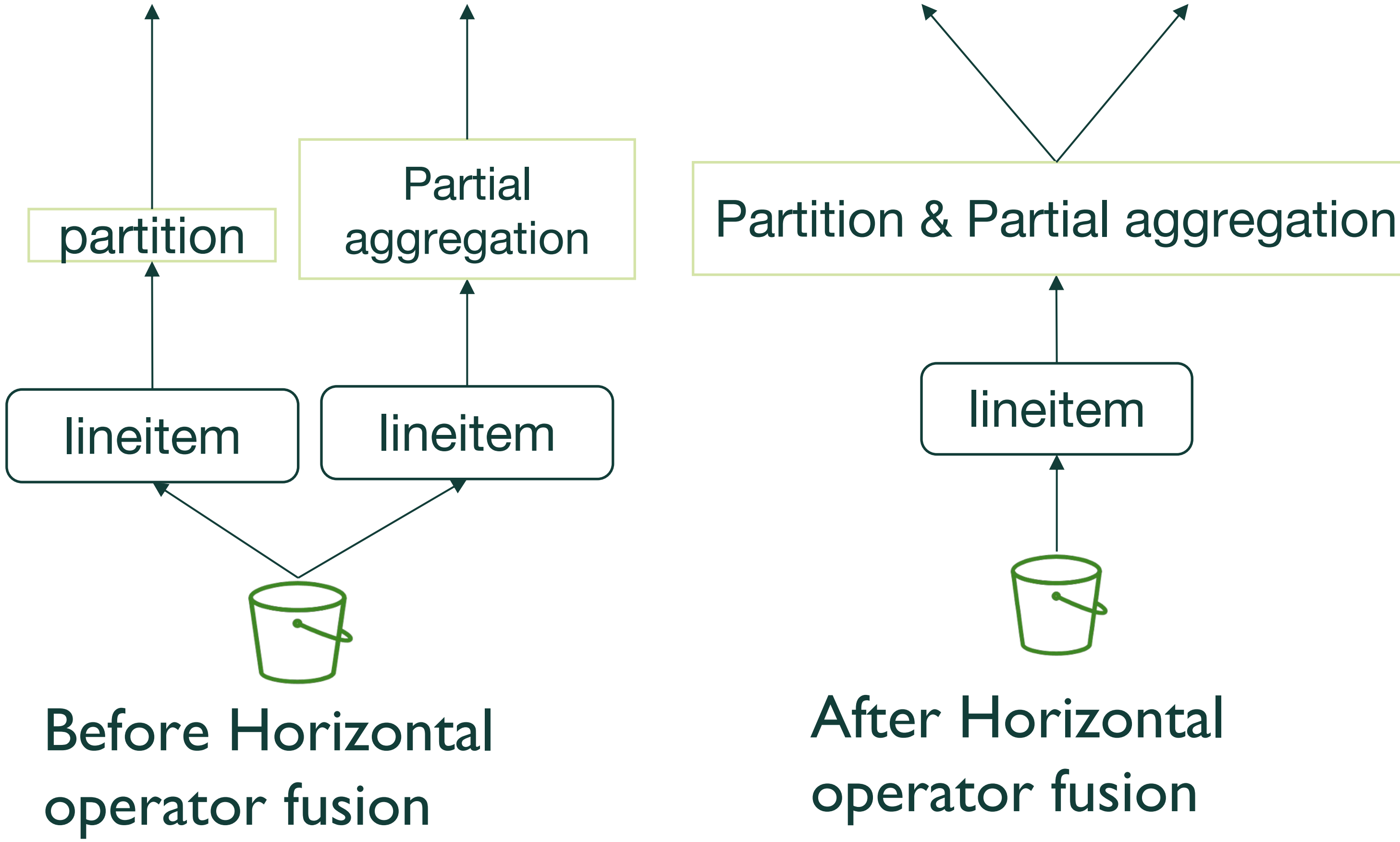


Avoid redundant reads

Create Locality to Reduce Communication

<u>Horizontal Operator Function:</u>

- The invoked function will run Different operators in parallel
- Data save can be huge when table is big!

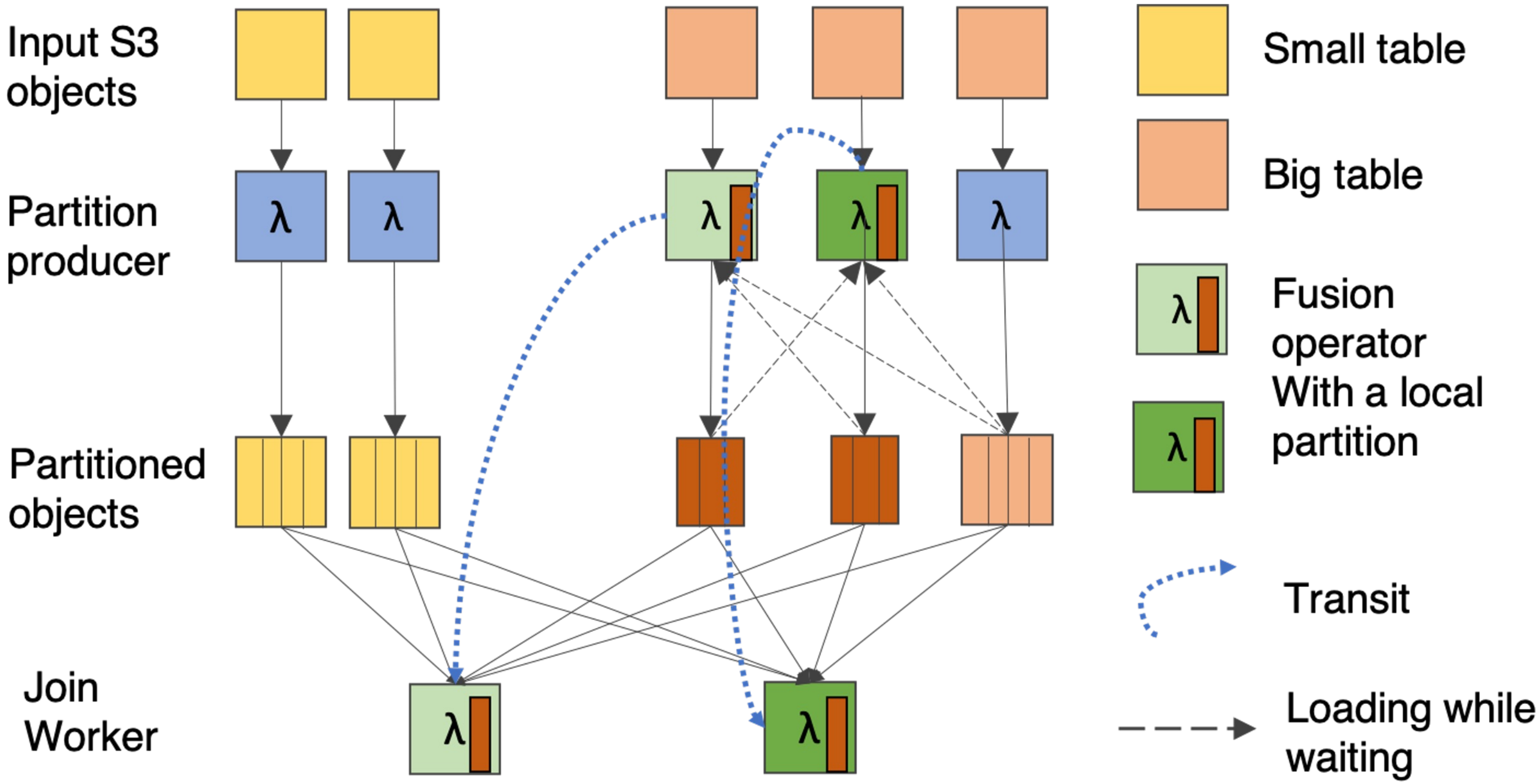


Single read, multi thread, multiple write

Create Locality to Reduce Communication

Vertical Operator Fusion:

- Partition worker will transit to Join worker
- The fusion worker retain one Partition doesn't writes to S3
- Load the ready partition into Local disk



Join example of horizontal operator fusion version I Big table retain one partition not writing to S3

Hide read latency inside the waiting time

Locality is much "easier" to create when you tailor query to functions

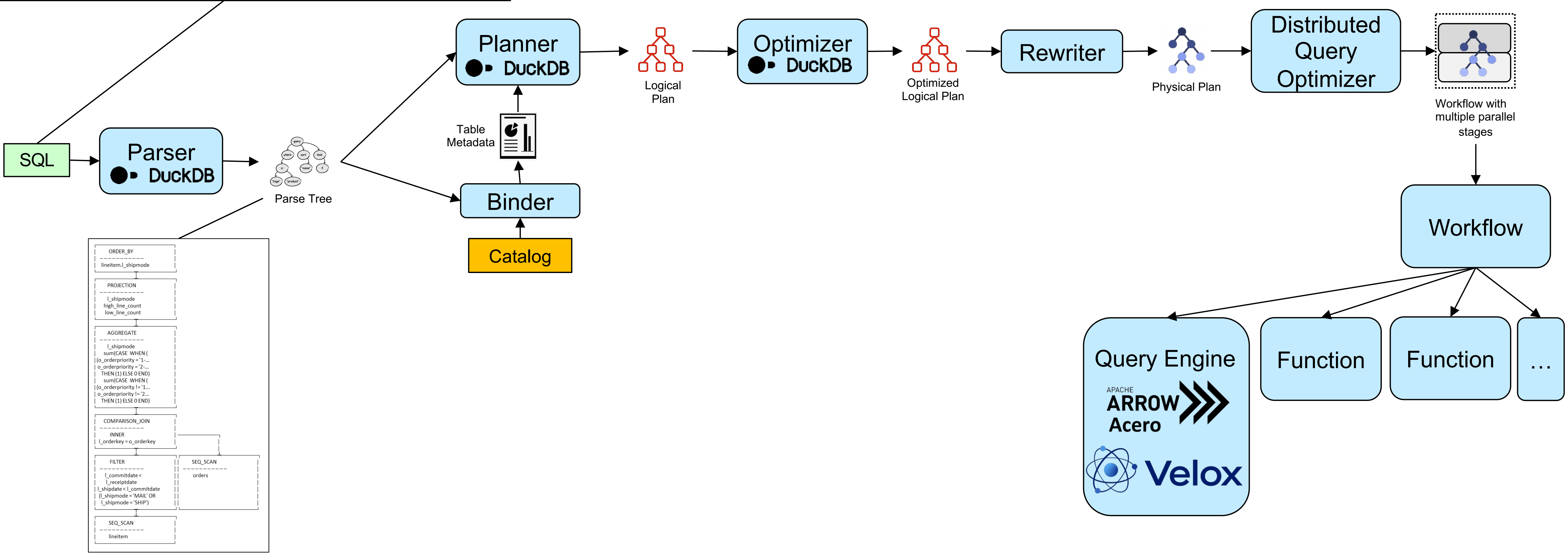
Decompose, Decompose, and Decompose

```
SELECT l_shipmode,
       SUM(CASE o_orderpriority IN ('1','2') THEN 1 ELSE 0) AS high_line_count,
       SUM(CASE o_orderpriority NOT IN ('1','2') THEN 1 ELSE 0) AS
low_line_count,
  FROM lineitem INNER JOIN orders ON l_orderkey = o_orderkey
 WHERE l_commitdate < l_receiptdate
   AND l_shipdate < l_commitdate
   AND l_shipmode IN ('MAIL', 'SHIP')
 GROUP BY l_shipmode, line_count
 ORDER BY l_shipmode
```

Decompose user logic into well-defined Lego blocks

- Blocks can be glued together differently later

c



Decomposability allows for fine-grained resource allocation and increases elasticity

Decompose, Decompose, and Decompose

Snowflake released *Snowset* dataset w/ statistics of real-world customer queries [1]

- Dataset contains statistics of 70M queries for 14-day period (21/02/2018—07/03/2018)

Several insights in [2] benchmarking Snowset:

Most queries complete within few seconds
- Median: 2.2 s; 2.8% of queries run > 1 min
- **Implication:** Cold-start must be < 100ms (<5%)

Still most time & CPU is spent in long-running queries
- **Implication:** Elasticity required to scale-out as needed

Most queries just touch a few MBs of data
- Median: 5.3 MB; 0.1% of queries read > 1TB
- **Implication:** Engine must be lightweight (compute efficient)

Still most of data is read by few data-hungry queries
- **Implication:** Reducing data movement is necessary

Database size per customer varies a lot but usually less 100 GBs
- **Implication:** Need to accommodate many different sizes

Still most data belongs to few customers w/ DB sizes of TBs/PBs
- **Implication:** Need to consider biggest customers too

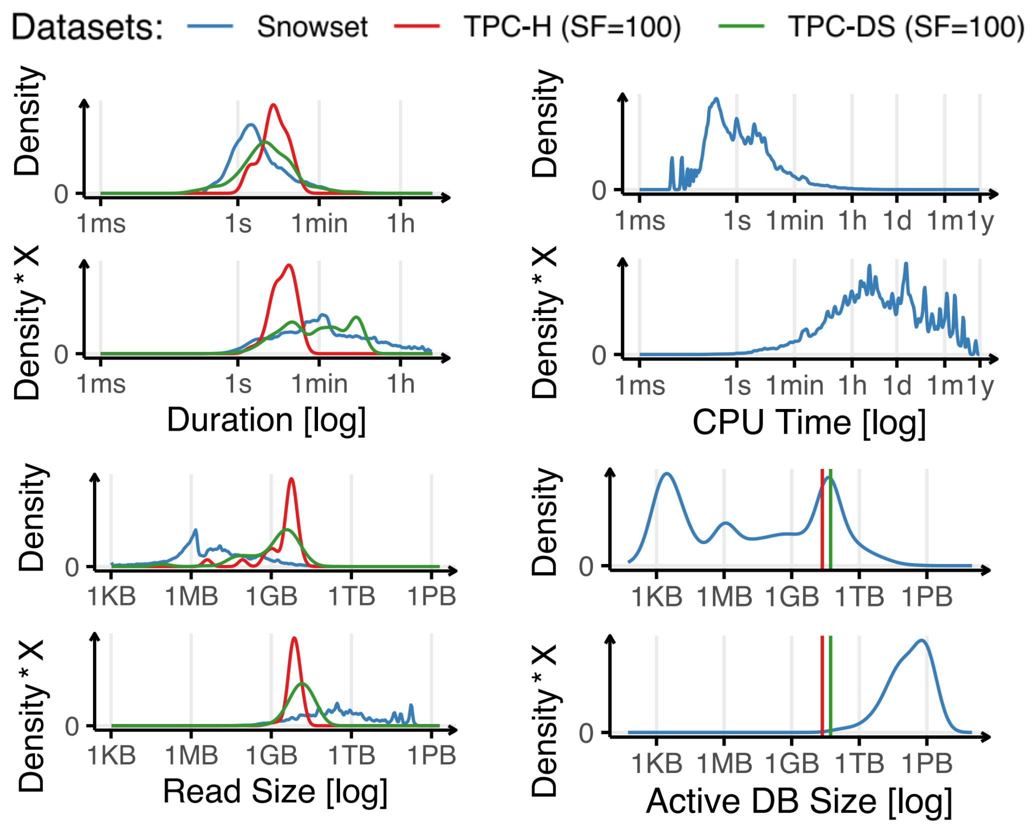


Figure 1. Density function of the duration, CPU time, read bytes per query, as well as database size per customer. Weight density function by query importance (its value from the x-axis)

[1] https://github.com/resource-disaggregation/snowset [2] Cloud Analytics Benchmark VLDB'23

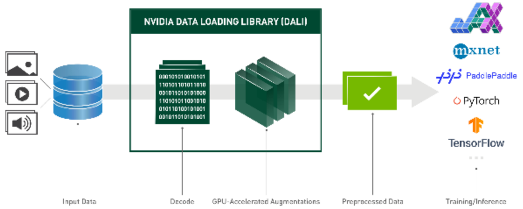
Extreme elasticity goal: queries running for a few sec. touching MBs & few hours touching PBs

GPU Functions Enabler of Broader Workloads

Myriad of GPU pre-processing libraries

Nvidia DALI

https://docs.nvidia.com/deeplearning/dali/user-guide/docs/index.html



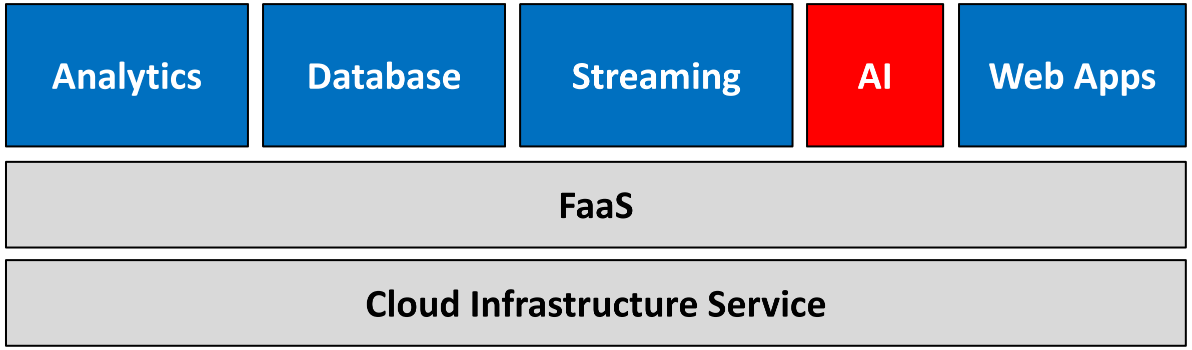
cuDF

https://github.com/rapidsai/cudf

RAPIDS cuDF - GPU DataFrames

cuDF can now be used as a no-code-change accelerator for pandas! To learn more, see here!

cuDF is a GPU DataFrame library for loading joining, aggregating, filtering, and otherwise manipulating data. cuDF leverages libcudf, a blazing-fast C++/CUDA dataframe library and the Apache Arrow columnar format to provide a GPU-accelerated pandas API.



Pre-processing benefits from disaggregation

Google tf.data service

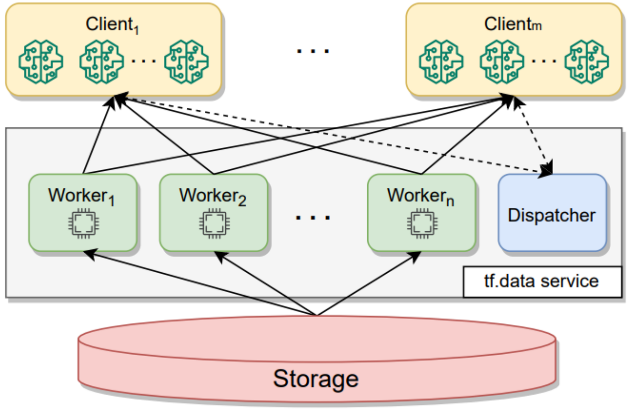
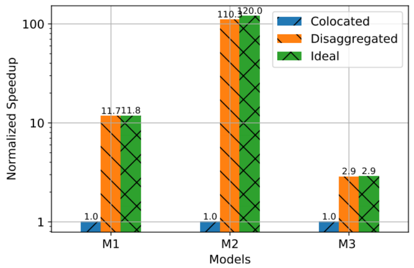
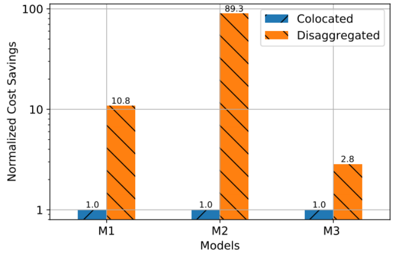


Figure 5: tf.data service architecture. Solid lines correspond to the data path, dashed lines correspond to the data path, dashed lines correspond to the control path.

https://dl.acm.org/doi/abs/10.1145/3620678.3624666





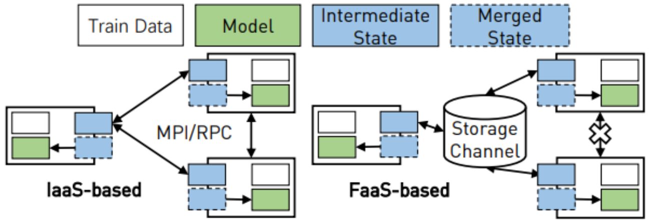


Figure 1: IaaS vs. FaaS-based ML system architectures.

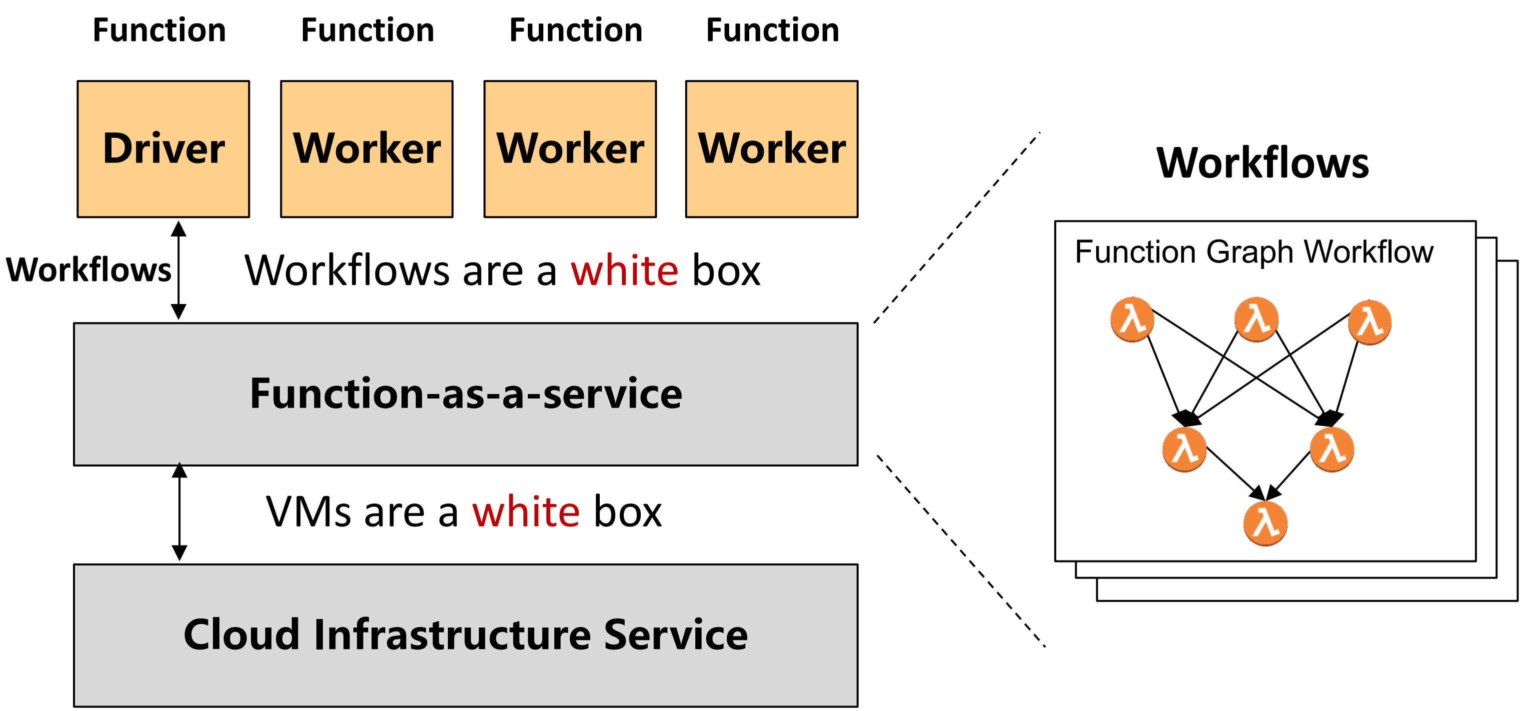
Towards Demystifying Serverless Machine Learning Training, Jiang et al.

Explore broader class of workloads as heterogeneity enters the space

Expose the Workflow Abstraction

Expose the workflow abstraction as a native entity in the serverless computing service

- A workflow is a DAG of functions
- Provider is able to understand the relationships between workers (not possible w/ IaaS)



How are Workflows Useful?

- Increases locality (co-location)
- Overlaps computation/communication
- Adapts to changes
- Permits higher function density
- …

→ Overall better execution & deployment

Native workflows allows providers to “see” the relationship among workers

Recipe to Achieve Short-lived Clouds

Good:

- Zero provisioning
- Autoscaling & Fast start-up times
- Fine-grained pricing
- Fine-grained resource allocation

→ Exploit autoscaling and the Pareto curve

Bad:

- Limited execution time → State checkpointing
- Lack of lambda-to-lambda communication → Create locality
- Fixed memory-to-compute ratios → Decompose
- ~~Restricted to CPUs~~ → ML-on-FaaS

Ugly:

- No control of execution and deployment → Workflows as first-class citizens

Receipt toward achieving short-lived clouds

Outline

- Introduction
- The good, the bad, and the ugly
- Toward short-lived clouds
- Serverless computing in AI-centric clouds
- Conclusion

"It's the Memory Stupid", Richard Sites

RICHARD SITES

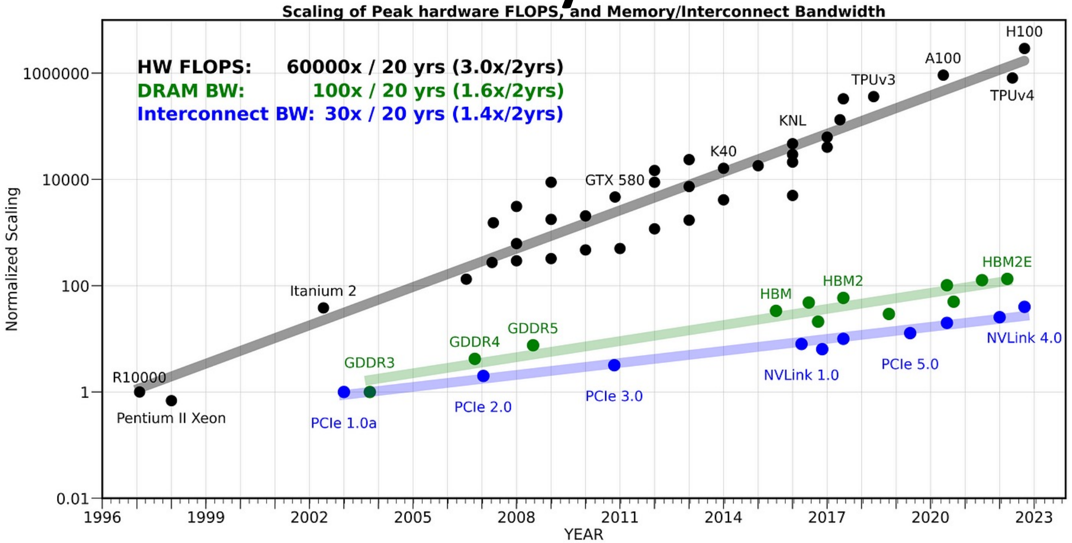
It's the Memory, Stupid!

Microprocessor Report

"I expect that over the coming decades memory subsystem design will be the only important design issue for microprocessors."
Richard Sites [MPR'96]

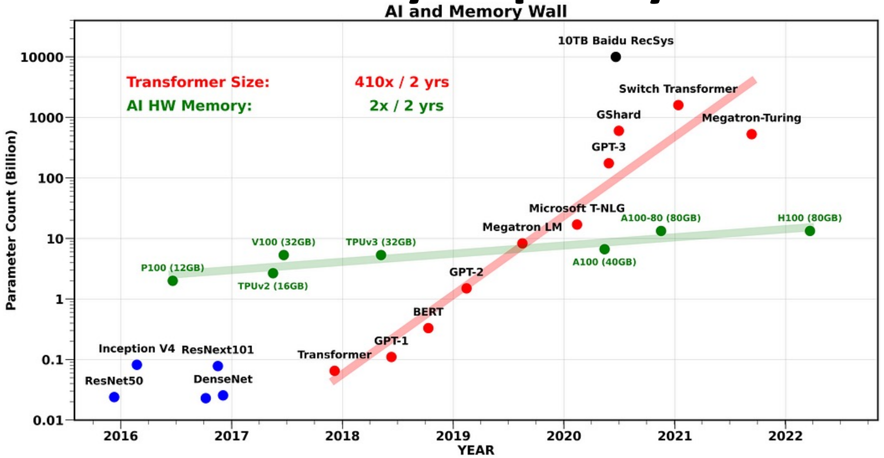
http://cva.stanford.edu/classes/cs99s/papers/architects_look_to_future.pdf

Memory BW Wall



https://medium.com/riselab/ai-and-memory-wall-2cb4265cb0b8

Memory Capacity Wall



Memory $ Wall

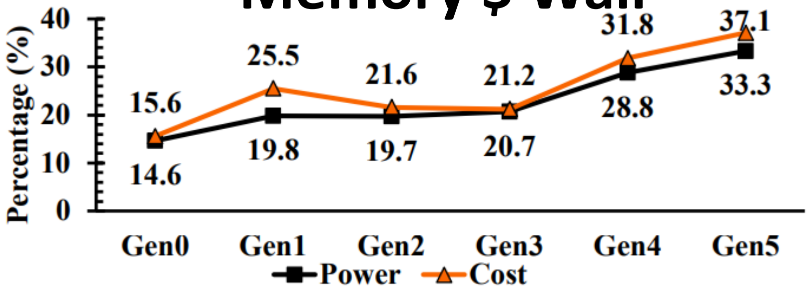
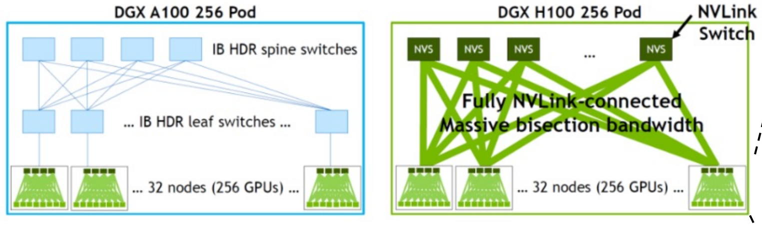


Figure 3: *Memory as a percentage of rack TCO and power across different hardware generations of Meta.*

Memory has become the only issue in Datacenter design in AI era

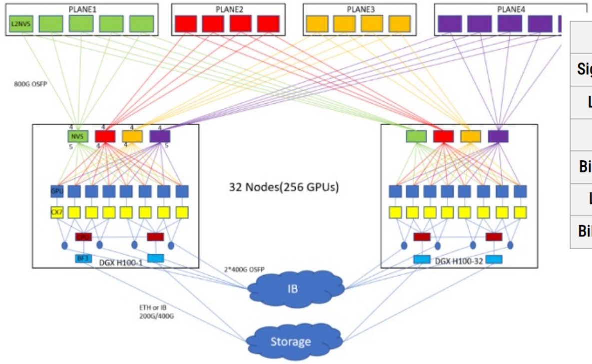
AI-centric Supercomputer Clusters

https://www.nvidia.com/en-us/data-center/dgx-superpod/

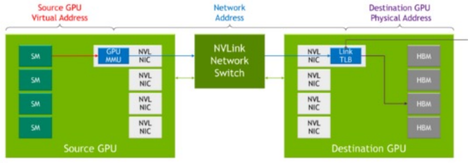


	A100 SuperPod			H100 SuperPod			Speedup	
	Dense PFLOP/s	Bisection [GB/s]	Reduce [GB/s]	Dense PFLOP/s	Bisection [GB/s]	Reduce [GB/s]	Bisection	Reduce
1 DGX / 8 GPUs	2.5	2,400	150	16	3,600	450	1.5x	3x
32 DGXs / 256 GPUs	80	6,400	100	512	57,600	450	9x	4.5x

NVLink Fabric



	NVLink 1.0	NVLink 2.0	NVLink 3.0	NVlink 4.0
Signaling Rate	20 GT/s	25 GT/s	50 GT/s	100 GT/s
Lanes/Link	8	8	4	2
Rate/Link	20 GB/s	25 GB/s	25 GB/s	25 GB/s
BiDir BW/Link	40 GB/s	50 GB/s	50 GB/s	50 GB/s
Links/Chip	4 (P100)	6 (V100)	12 (A100)	18 (H100)
BiDir BW/Chip	160 GB/s (P100)	300 GB/s (V100)	600 GB/s (A100)	900 GB/s (H100)



https://cloud.google.com/blog/topics/systems/tpu-v4-enables-performance-energy-and-co2e-efficiency-gains ISCA'23

Google's Cloud TPU v4 provides exaFLOPS-scale ML with industry-leading efficiency

Google Cloud

Figure 3: Eight of 64 racks for one 4096-chip supercomputer.

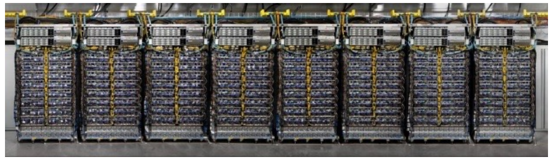


Table 1: Workloads by DNN model type (% TPUs used).

DNN Model	*TPU v1 7/2016 (Inference)*	*TPU v3 4/2019 (Training & Inference)*	*TPU v4 Lite 2/2020 (Inference)*	*TPU v4 10/2022 (Training)*
MLP/DLRM	61%	27%	25%	24%
RNN	29%	21%	29%	2%
CNN	5%	24%	18%	12%
Transformer	--	21%	28%	57%
(BERT)	--	--	*(28%)*	*(26%)*
(LLM)	--	--	--	*(31%)*

Recommenders ~25% / Transformers ~60%

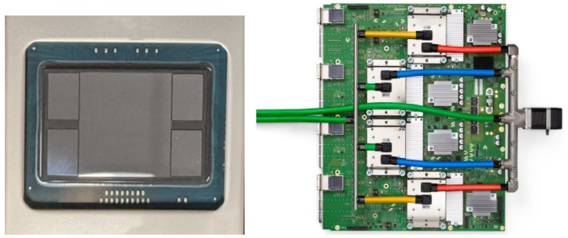
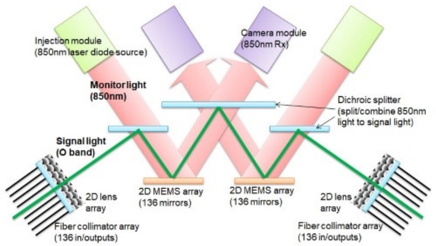


Figure 2: The TPU v4 package (ASIC in center plus 4 HBM stacks) and printed circuit board with 4 liquid-cooled packages. The board's front panel has 4 top-side PCIe connectors and 16 bottom-side OSFP connectors for inter-tray ICI links.

Reconfigurable optical switch



https://arxiv.org/abs/2304.01433

Reconfiguration allows for more efficient all-to-all patterns due to the enabler of twisted torus topologies (2D/3D torus are ok for all-reduce patterns)

SparseCore: Hardware Support for Embeddings

Optimizes for low-arithmetic intensity operations (sparse)

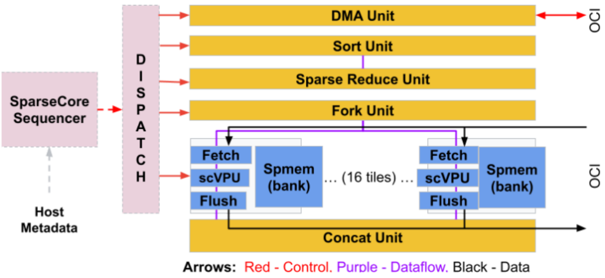


Figure 7: SparseCore (SC) Hardware Architecture.

Clusters of XPUs interconnect via high-speed networks

From Compute- to Memory-centric Clouds

State-of-the-art WSC [~2000s-2020s]

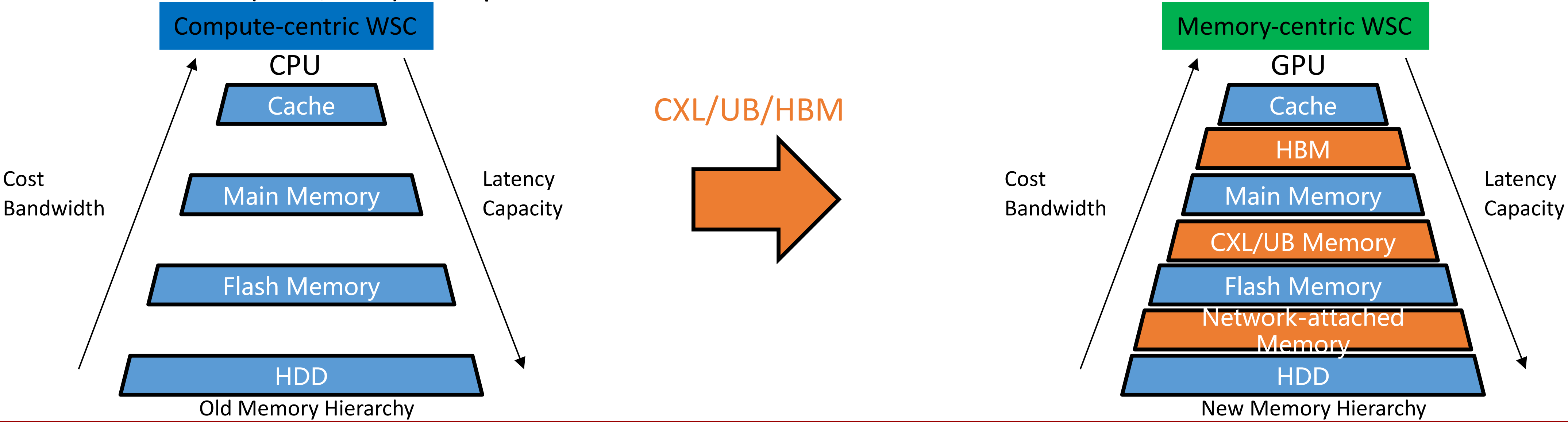
Killer App
Web Search for a Planet

Technological assumptions
- Most of the cost goes to CPU
- Network fabrics are slow (100s-10s us)
- Accelerators (GPU/TPU) are optional

Emerging WSC [Post 2020s]

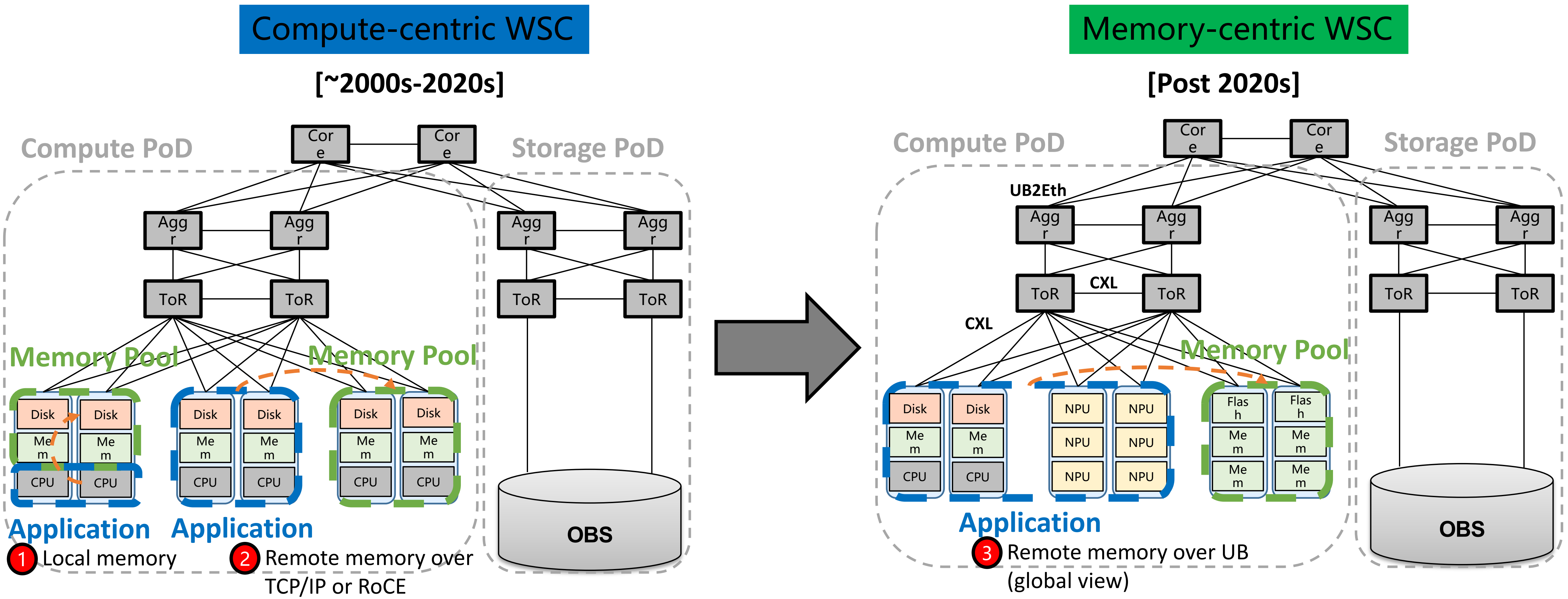
Killer App
Generative AI for a planet

Technological assumptions
- Most of the cost goes to Memory
- Network fabrics are fast (100s ns)
- Accelerators (GPU/TPU) are a necessity



New cloud design built around memory

What are the Implications to Serverless Computing?



Not clear how to build FaaS in this cloud...food for thought!

Conclusion

- Serverless computing embodies the original goal of “pay-as-you-go” in the cloud
- Serverless computing exhibits unprecedented elasticity but a few key limitations
- Luckily most (if not old) limitations are not inherently fundamental
- Short-lived clouds builds (almost) all cloud applications around serverless computing
- Cookbook for achieving short-lived clouds:
    - Exploit autoscaling (per request)
    - Checkpointing state
    - Increase locality
    - Broader application space with heterogeneity
    - Expose workflows as first-class citizens

Research Team

Internal | External

Team Lead

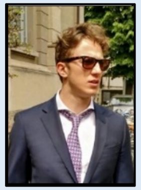
Dr. Javier Picorel
Cloud Computing and Systems expert

Architect

Norbert Martinez
Expert in Data Warehouses

R&D

Dr. Jeyhun Karimov
Cloud Streaming and Big Data expert



Dr. Lorenzo Affetti
Cloud Streaming and Big Data Expert

Plamen Petrov
Expert in serverless computing

Kushagra Shah
MSc Data Management

Dr. Vishal Boddu
Systems and HPC expert

Alexandrina Panfil
Computer Systems

Jingrong Chen
Data Management and serverless

Pure Research

Dr. Florin Dinu
Cloud Scheduling and Systems expert

Dr. Diego Didona
Expert in Transactional KVs

Interns

Bernhard Linn
(UoE)

Tong Xing
(UoE)

Faculty Collaborators

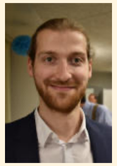
Prof. David Atienza
(IEEE Fellow)

Prof. Pascal Frossard
(IEEE Fellow)

Prof. Rachid Guerraoui
(ACM Fellow)

Prof. An-marie Kermarrec
(ACM Fellow)

EPFL



Prof. Zsolt István

TECHNISCHE UNIVERSITÄT DARMSTADT

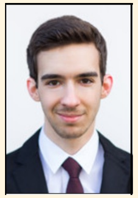
Prof. Boris Grot

THE UNIVERSITY OF EDINBURGH

PhD Collaborators

Diana Petrescu
(EPFL)

Arsany Xygkis
(EPFL)



Antoine Murat
(EPFL)

Rafael Pires
(EPFL)

Darong Huang
(EPFL)

Amirhossein Shahbazinia
(EPFL)

Shyam Jesalpura
(UoE)

Join Us!

- Very competitive salary and welfare!
- Change the world with your research!
- Work and collaborate with global top talents!

HUAWEI CLOUD is a leading cloud service provider, which brings Huawei's 30-plus years of expertise together in ICT infrastructure products and solutions. We are committed to providing reliable, secure, and cost-effective cloud services to empower applications, harness the power of data, and help organizations of all sizes grow in today's intelligent world. HUAWEI CLOUD is also committed to bringing affordable, effective, and reliable cloud and AI services through technological innovation.