Marcin Copik, Alexandru Calotoiu, Torsten Hoefler, Michał Podstawski, Laurin Brandner, Larissa Schmid, Nico Graf, Grzegorz Kwaśniewski, Kacper Janda, Mateusz Knapik, Jakub Czerski, Mahla Sarifi, Paweł Żuk, Sascha Kehrli, Abhishek Kumar, Prajin Khadka, Horia Mercan, and many others!

Demystifying Serverless Performance:
A Hands-on Tutorial with Serverless Benchmark Suite SeBS

HiPEAC 2026
Cracow, Poland

Agenda

SeBS Slack

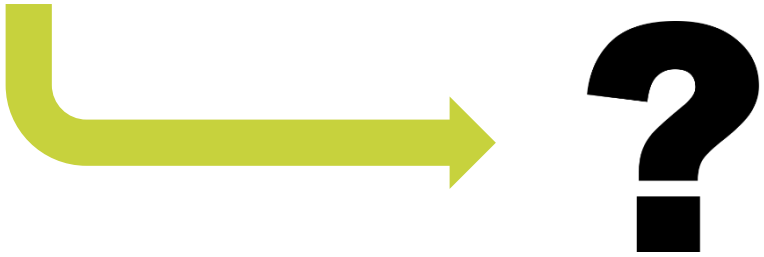
- ✓ Part I
  - ✓ What is Serverless?
  - ✓ Benchmarking Suite SeBS
  - ✓ Working with SeBS
- ✓ Hands-on I: Local Deployment & Storage
- ✓ Part II
  - ✓ Communication and Data
  - ✓ Serverless Workflows
  - ✓ Experiments in SeBS
- ✓ Hands-on II: FaaS Platforms & Experiments
- ✓ **Part III**
  - ✓ **Research Directions in Serverless & Performance**
  - ✓ Development of SeBS

slack **Join SeBS Slack channel: https://serverlessbenchmark.slack.com**

What's Next for Serverless?

"Toy" Serverless

General-Purpose Serverless



?

What's Next for Serverless?

"Toy" Serverless

General-Purpose Serverless

What's Next for Serverless?

"Toy" Serverless

Vercel @vercel

Serverless servers: Node.js with in-function concurrency

?

What Comes Next for Serverless?

What Comes Next for Serverless?

What will be the runtime of the future?

What Comes Next for Serverless?

What will be the runtime of the future?

Where are limits of scalability and resource allocation?

What Comes Next for Serverless?

What will be the runtime of the future?

Where are limits of scalability and resource allocation?

Are we going to break free from the vendor lock-in?

What Comes Next for Serverless?

What will be the runtime of the future?

Where are limits of scalability and resource allocation?

Are we going to break free from the vendor lock-in?

What will be the next serverless programming model?

Serverless was never designed for HPC

AWS Lambda turns 10: A rare look at the doc that

Introducing AWS Lambda

Posted On: Nov 13, 2014

AWS Lambda is a compute service that runs your code in response to events and automatically manages the compute resources for you, making it easy to build applications that respond quickly to new information. AWS Lambda starts running your code within milliseconds of an event such as an image upload, in-app activity, website click, or output from a connected device. You can also use AWS Lambda to create new back-end services where compute resources are automatically triggered based on custom requests. With AWS Lambda you pay only for the requests served and the compute time required to run your code. Billing is metered in increments of 100 milliseconds, making it cost-effective and easy to scale automatically from a few requests per day to thousands per second.

AWS Lambda is available in Preview. Learn more at http://aws.amazon.com/lambda.

Serverless was never designed for HPC

AWS Lambda turns 10: A rare look at the doc that started it

November 14, 2024 • 5460 words

Serverless was never designed for HPC

31. How does Lambda support parallel processing?

Developers can run multiple applications and/or ***multiple copies of the same application simultaneously****. They can also access Lambda APIs programmatically from within applications, using the AWS client SDK, which allows them to delegate and orchestrate work by running other applications.*

Serverless was never designed for HPC

31. How does Lambda support parallel processing?

Developers can run multiple applications and/or ***multiple copies of the same application simultaneously****. They can also access Lambda APIs programmatically from within applications, using the AWS client SDK, which allows them to delegate and orchestrate work by running other applications.*

Burst Launches, Colocation Policies
Bulk Synchronous Parallel Model
Communicators, Message Passing, Collectives
…

Serverless was never designed for HPC

31. How does Lambda support parallel processing?

Developers can run multiple applications and/or ***multiple copies of the same application simultaneously****. They can also access Lambda APIs programmatically from within applications, using the AWS client SDK, which allows them to delegate and orchestrate work by running other applications.*

Burst Launches, Colocation Policies
Bulk Synchronous Parallel Model
Communicators, Message Passing, Collectives
…

Embarrassingly Parallel

Research into Serverless Performance

Research into Serverless Performance

Fast and Lightweight Sandboxes
(gVisor, Catalyzer, SEUSS, Photon)

Research into Serverless Performance

Fast and Lightweight Sandboxes
(gVisor, Catalyzer, SEUSS, Photon)

Stateful Functions
(Cloudburst, Faasm, Crucial, PraaS)

Research into Serverless Performance

Fast and Lightweight Sandboxes
(gVisor, Catalyzer, SEUSS, Photon)

Stateful Functions
(Cloudburst, Faasm, Crucial, PraaS)

Networking and Communication
(Boxer, FMI, rFaaS)

Research into Serverless Performance

Fast and Lightweight Sandboxes
(gVisor, Catalyzer, SEUSS, Photon)

Stateful Functions
(Cloudburst, Faasm, Crucial, PraaS)

Networking and Communication
(Boxer, FMI, rFaaS)

HPC FaaS
(Globus Compute, rFaaS, Lithops)

Research into Serverless Performance

Fast and Lightweight Sandboxes
(gVisor, Catalyzer, SEUSS, Photon)

Stateful Functions
(Cloudburst, Faasm, Crucial, PraaS)

Networking and Communication
(Boxer, FMI, rFaaS)

HPC FaaS
(Globus Compute, rFaaS, Lithops)

New HPC Workloads
(Cirrus, LambdaML, Mashup, DayDream)

Research into Serverless Performance

Fast and Lightweight Sandboxes
(gVisor, Catalyzer, SEUSS, Photon)

Stateful Functions
(Cloudburst, Faasm, Crucial, PraaS)

Networking and Communication
(Boxer, FMI, rFaaS)

HPC FaaS
(Globus Compute, rFaaS, Lithops)

New HPC Workloads
(Cirrus, LambdaML, Mashup, DayDream)

Improved Cold Starts
(RainbowCake, IceBreaker, Medes)

Research into Serverless Performance

Fast and Lightweight Sandboxes
(gVisor, Catalyzer, SEUSS, Photon)

Stateful Functions
(Cloudburst, Faasm, Crucial, PraaS)

Networking and Communication
(Boxer, FMI, rFaaS)

HPC FaaS
(Globus Compute, rFaaS, Lithops)

New HPC Workloads
(Cirrus, LambdaML, Mashup, DayDream)

Improved Cold Starts
(RainbowCake, IceBreaker, Medes)

Improved Scheduling (Wukong, Palette, PraaS, ProPack, Pheromone)

Research into Serverless Performance

Fast and Lightweight Sandboxes
(gVisor, Catalyzer, SEUSS, Photon)

Stateful Functions
(Cloudburst, Faasm, Crucial, PraaS)

Networking and Communication
(Boxer, FMI, rFaaS)

HPC FaaS
(Globus Compute, rFaaS, Lithops)

New HPC Workloads
(Cirrus, LambdaML, Mashup, DayDream)

Improved Cold Starts
(RainbowCake, IceBreaker, Medes)

Improved Scheduling (Wukong, Palette, PraaS, ProPack, Pheromone)

HPC Utilization
(HPC-Whisk, Serverless Disaggregation)

Research into Serverless Performance

Fast and Lightweight Sandboxes
(gVisor, Catalyzer, SEUSS, Photon)

Stateful Functions
(Cloudburst, Faasm, Crucial, PraaS)

Networking and Communication
(Boxer, FMI, rFaaS)

HPC FaaS
(Globus Compute, rFaaS, Lithops)

New HPC Workloads
(Cirrus, LambdaML, Mashup, DayDream)

Improved Cold Starts
(RainbowCake, IceBreaker, Medes)

Improved Scheduling (Wukong, Palette, PraaS, ProPack, Pheromone)

HPC Utilization
(HPC-Whisk, Serverless Disaggregation)

Benchmark Suites
(SeBS, Serverlessbench, FaaSDom)

Research into Serverless Performance

Fast and Lightweight Sandboxes
(gVisor, Catalyzer, SEUSS, Photon)

Stateful Functions
(Cloudburst, Faasm, Crucial, PraaS)

Networking and Communication
(Boxer, FMI, rFaaS)

HPC FaaS
(Globus Compute, rFaaS, Lithops)

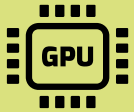
New HPC Workloads
(Cirrus, LambdaML, Mashup, DayDream)

Improved Cold Starts
(RainbowCake, IceBreaker, Medes)

Improved Scheduling (Wukong, Palette, PraaS, ProPack, Pheromone)

HPC Utilization
(HPC-Whisk, Serverless Disaggregation)

Benchmark Suites
(SeBS, Serverlessbench, FaaSDom)



Accelerated Functions
(DGSF, KaaS, MIGnificient)

Serverless: One Step Toward HPC – Cloud Convergence

Infrastructure

Serverless: One Step Toward HPC – Cloud Convergence

Infrastructure

Cloud

Virtual Machines



Object Storage



Virtual Networks

HPC

Serverless: One Step Toward HPC – Cloud Convergence

Infrastructure

Cloud

Virtual Machines

Object Storage

Virtual Networks

HPCaaS

HPC

HPC in the cloud

Serverless: One Step Toward HPC – Cloud Convergence

Infrastructure

Cloud

Virtual Machines

Object Storage

Virtual Networks

HPC

HPCaaS

HPC in the cloud

IaaS, SaaS, PaaS

GPU

Cloud in HPC

Serverless: One Step Toward HPC – Cloud Convergence

Infrastructure

Deployment

Cloud

Virtual Machines

Object Storage

Virtual Networks

Containers

(Managed) Kubernetes

Services

HPC

HPCaaS

HPC in the cloud

Fugaku

IaaS, SaaS, PaaS

GPU

Cloud in HPC

Serverless: One Step Toward HPC – Cloud Convergence

Infrastructure

Deployment

Cloud

- Virtual Machines
- Object Storage
- Virtual Networks

- Containers
- (Managed) Kubernetes
- Services

HPC

HPCaaS

- HPC in the cloud
- Cloud in HPC

- HPC
- Containers
- Flux
- XaaS

Serverless: One Step Toward HPC – Cloud Convergence

	Infrastructure	Deployment	Compute
Cloud	Virtual Machines	Containers	Serverless Functions
	Object Storage	(Managed) Kubernetes	Serverless Containers
	Virtual Networks	Services	
HPC	HPCaaS	HPC	
	HPC in the cloud	Containers	
		Flux	
	Cloud in HPC	XaaS	

IaaS, SaaS, PaaS

Serverless: One Step Toward HPC – Cloud Convergence

	Infrastructure	Deployment	Compute
Cloud	Virtual Machines	Containers	Serverless Functions
	Object Storage	(Managed) Kubernetes	Serverless Containers
	Virtual Networks	Services	
HPC	HPCaaS	HPC Containers	Globus Compute (funcX)
	HPC in the cloud	Flux	rFaaS
	Cloud in HPC	XaaS	Lithops

Serverless: One Step Toward HPC – Cloud Convergence

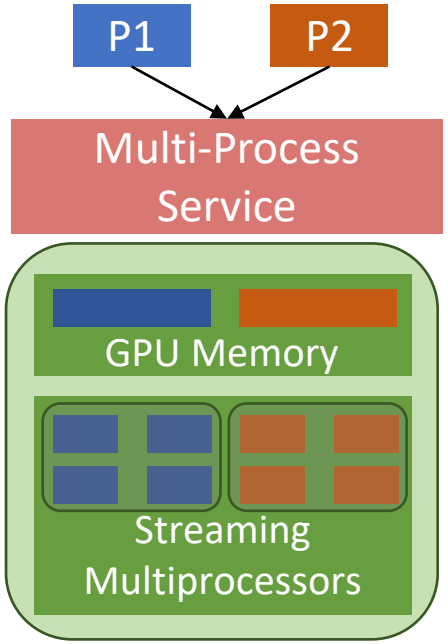
	Infrastructure	Deployment	Compute	Applications
Cloud	Virtual Machines	Containers	Serverless Functions	Dask, Spark Ray
	Object Storage	(Managed) Kubernetes	Serverless Containers	Serverless Workflows
	Virtual Networks	Services		
HPC	HPCaaS	HPC Containers	Globus Compute (funcX)	
	HPC in the cloud	Flux	RDMA rFaaS	
	Cloud in HPC	XaaS	Lithops	

IaaS, SaaS, PaaS

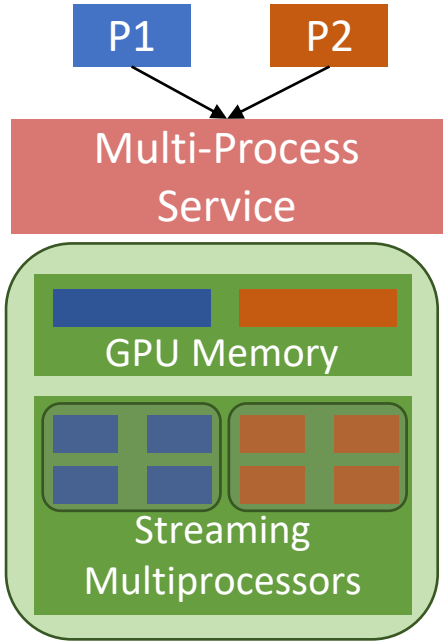
Serverless: One Step Toward HPC – Cloud Convergence

	Infrastructure	Deployment	Compute	Applications
Cloud	Virtual Machines	Containers	Serverless Functions	Dask, Spark Ray
	Object Storage	(Managed) Kubernetes	Serverless Containers	Serverless Workflows
	Virtual Networks	Services		
HPC	HPCaaS	HPC Containers	Globus Compute (funcX)	
	HPC in the cloud	Flux	RDMA rFaaS	
	Cloud in HPC	XaaS	Lithops	

Serverless GPUs: Multi-Process Service (MPS) to the Rescue?

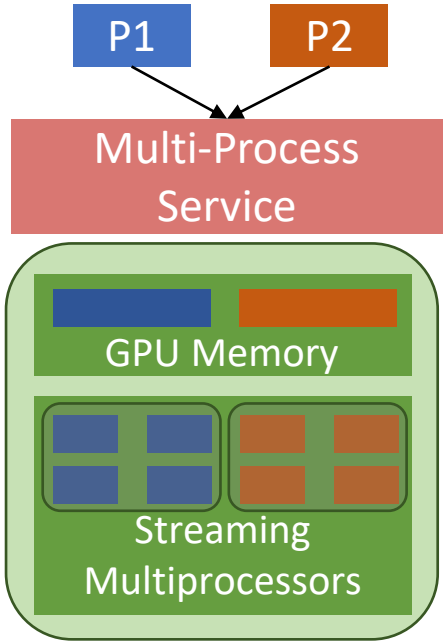


Serverless GPUs: Multi-Process Service (MPS) to the Rescue?



"MPS is only recommended for running **cooperative processes** effectively acting as a **single application**, such as multiple ranks of the same MPI job, such that the severity of the following memory protection and error containment limitations is acceptable."

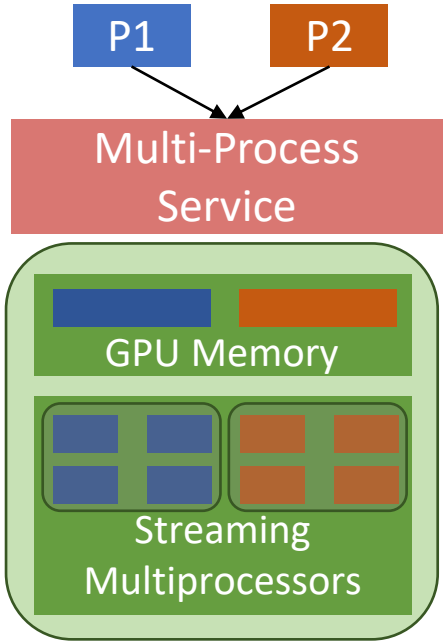
Serverless GPUs: Multi-Process Service (MPS) to the Rescue?



"MPS is only recommended for running **cooperative processes** effectively acting as a **single application**, such as multiple ranks of the same MPI job, such that the severity of the following memory protection and error containment limitations is acceptable."

⚠ Serverless is multi-tenant and executes arbitrary user code.

Serverless GPUs: Multi-Process Service (MPS) to the Rescue?

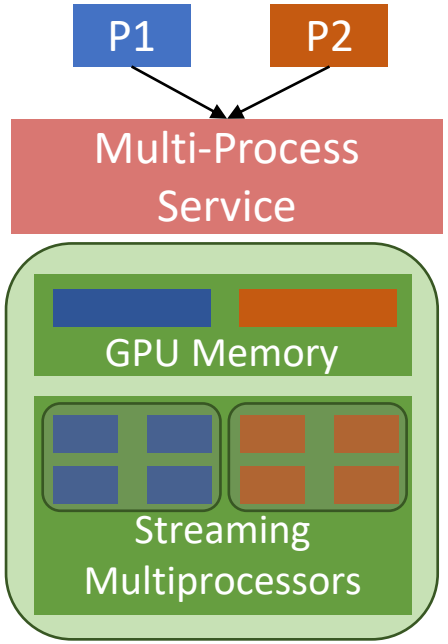


"MPS is only recommended for running **cooperative processes** effectively acting as a **single application**, such as multiple ranks of the same MPI job, such that the severity of the following memory protection and error containment limitations is acceptable."

Serverless is multi-tenant and executes arbitrary user code.

Limited security!
Function can conduct side-channel attack.

Serverless GPUs: Multi-Process Service (MPS) to the Rescue?



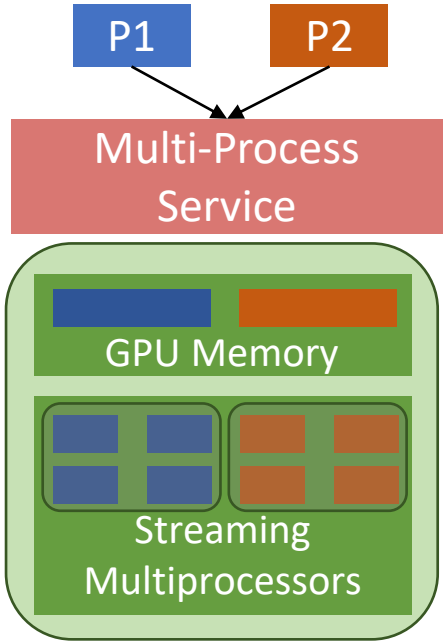
"MPS is only recommended for running **cooperative processes** effectively acting as a **single application**, such as multiple ranks of the same MPI job, such that the severity of the following memory protection and error containment limitations is acceptable."

⚠ Serverless is multi-tenant and executes arbitrary user code.

Limited security!
Function can conduct side-channel attack.

No performance isolation!
Function can hog the memory bandwidth.

Serverless GPUs: Multi-Process Service (MPS) to the Rescue?



"MPS is only recommended for running **cooperative processes** effectively acting as a **single application**, such as multiple ranks of the same MPI job, such that the severity of the following memory protection and error containment limitations is acceptable."

⚠ Serverless is multi-tenant and executes arbitrary user code.

Limited security!
Function can conduct side-channel attack.

No performance isolation!
Function can hog the memory bandwidth.

No error containment!
Function can maliciously kill GPU contexts.

Serverless GPUs: Building from Blocks with Multi-Instance GPU (MIGs)

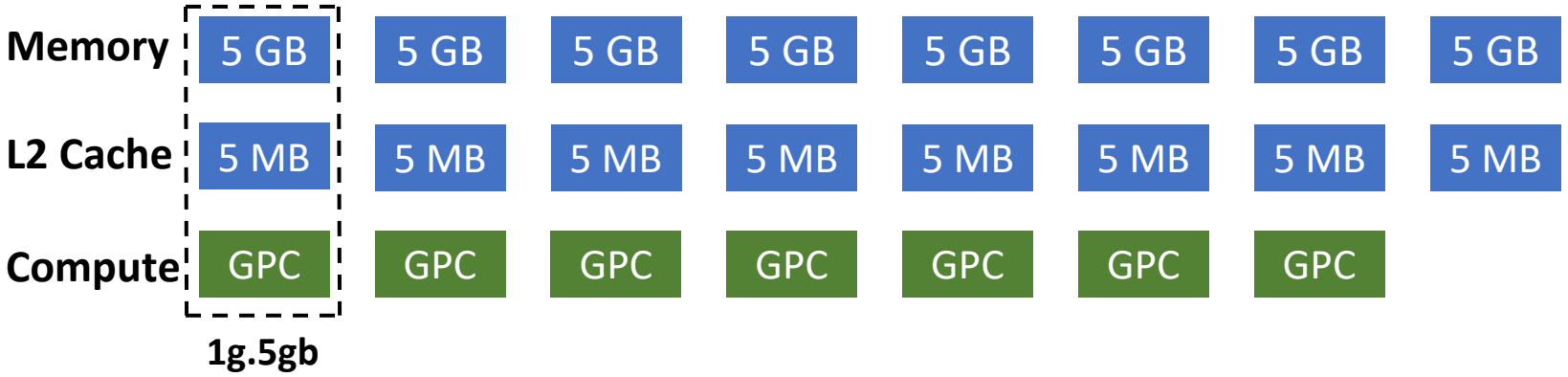
Serverless GPUs: Building from Blocks with Multi-Instance GPU (MIGs)

Memory	5 GB	5 GB	5 GB	5 GB	5 GB	5 GB	5 GB	5 GB
L2 Cache	5 MB	5 MB	5 MB	5 MB	5 MB	5 MB	5 MB	5 MB

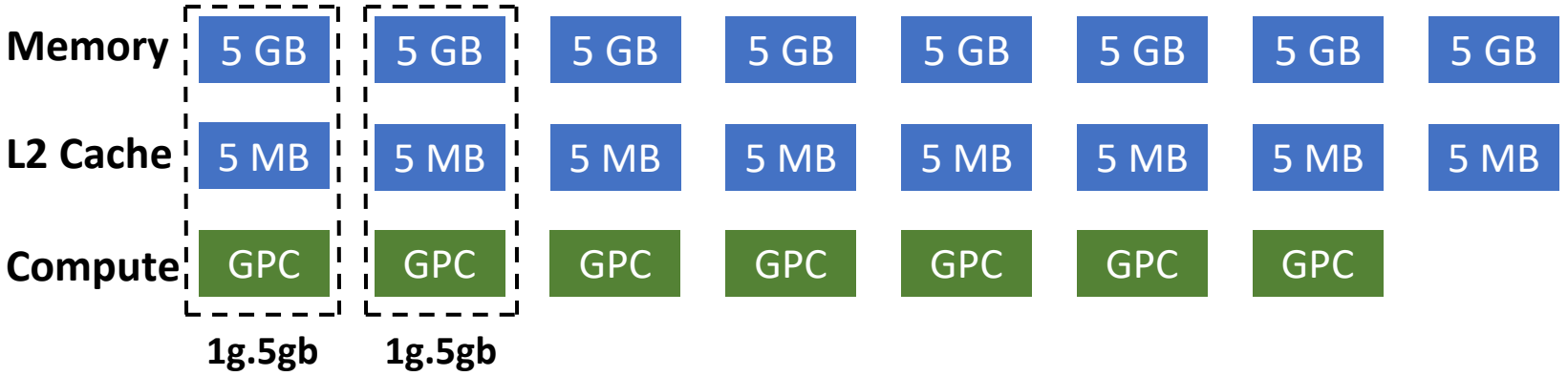
Serverless GPUs: Building from Blocks with Multi-Instance GPU (MIGs)

Memory	5 GB	5 GB	5 GB	5 GB	5 GB	5 GB	5 GB	5 GB
L2 Cache	5 MB	5 MB	5 MB	5 MB	5 MB	5 MB	5 MB	5 MB
Compute	GPC	GPC	GPC	GPC	GPC	GPC	GPC	

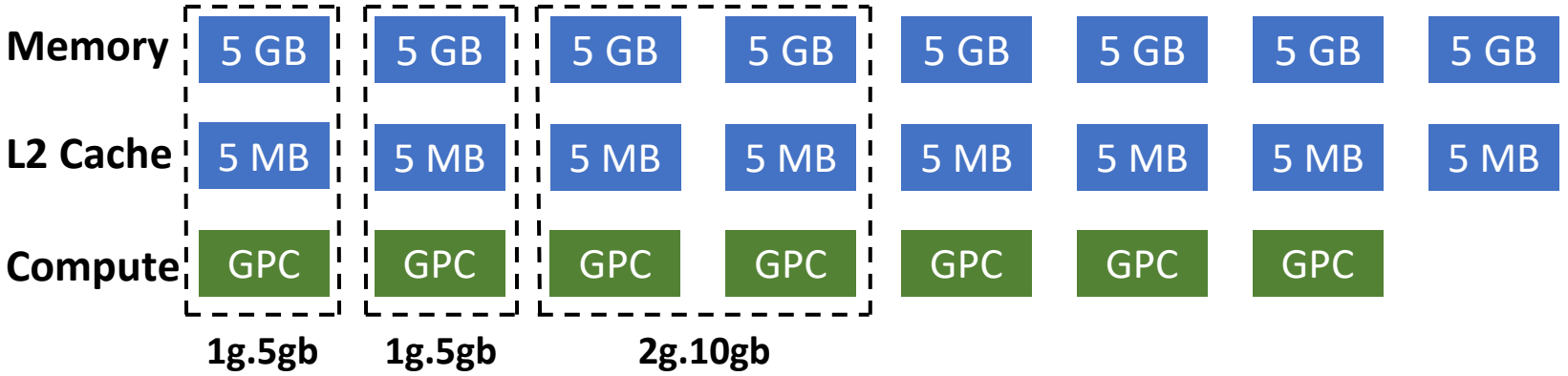
Serverless GPUs: Building from Blocks with Multi-Instance GPU (MIGs)

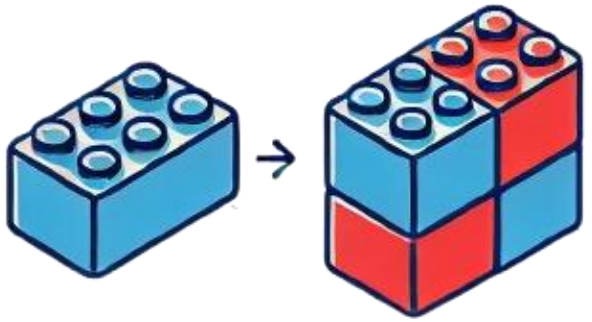


Serverless GPUs: Building from Blocks with Multi-Instance GPU (MIGs)

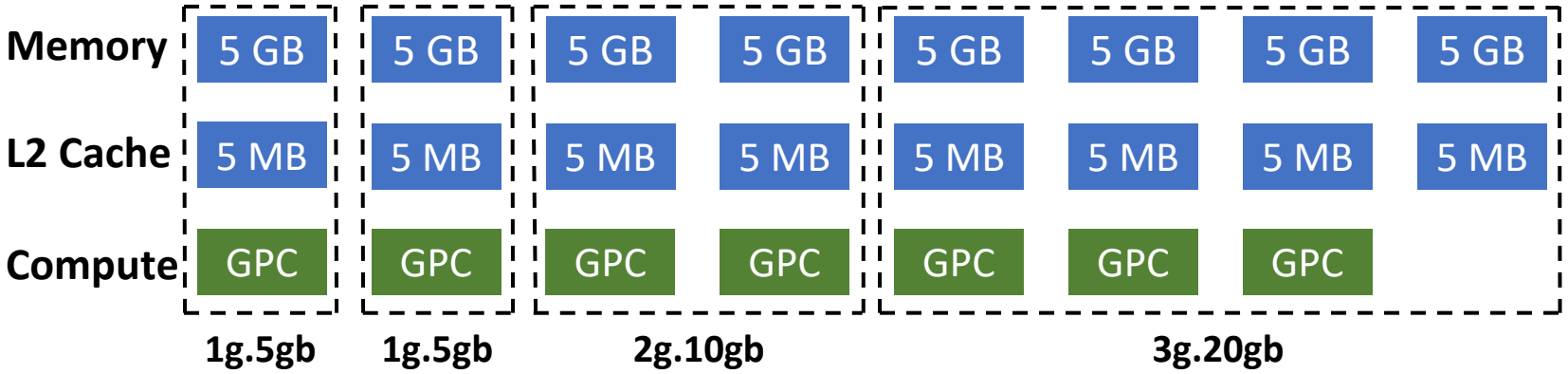


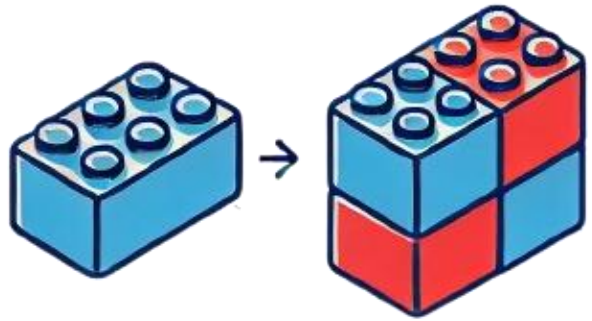
Serverless GPUs: Building from Blocks with Multi-Instance GPU (MIGs)



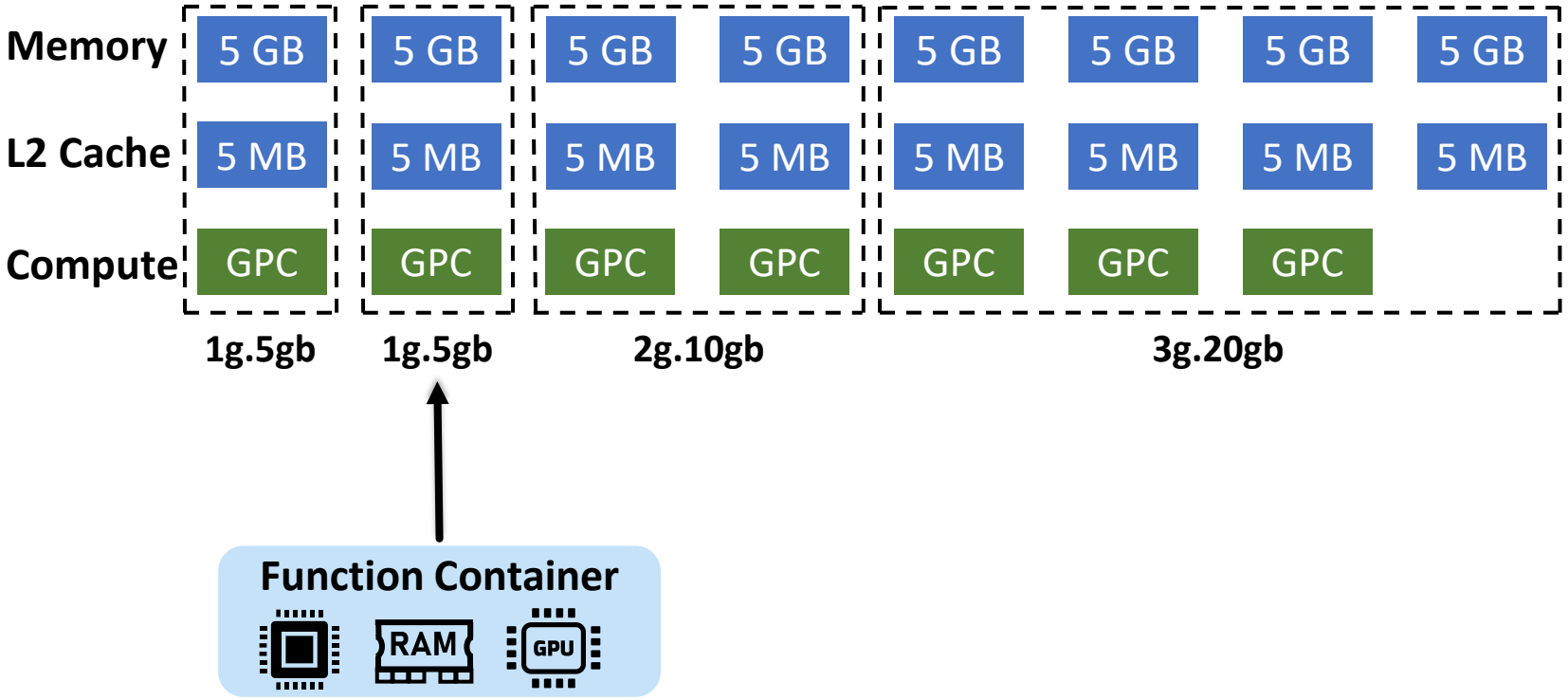


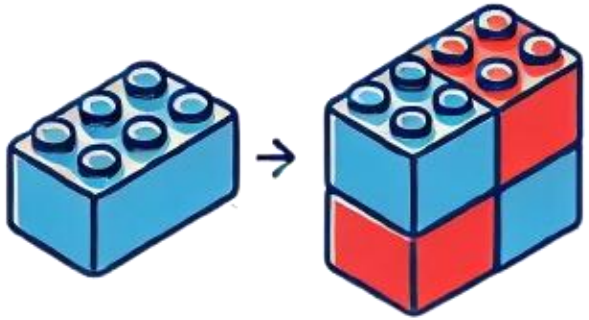
Serverless GPUs: Building from Blocks with Multi-Instance GPU (MIGs)



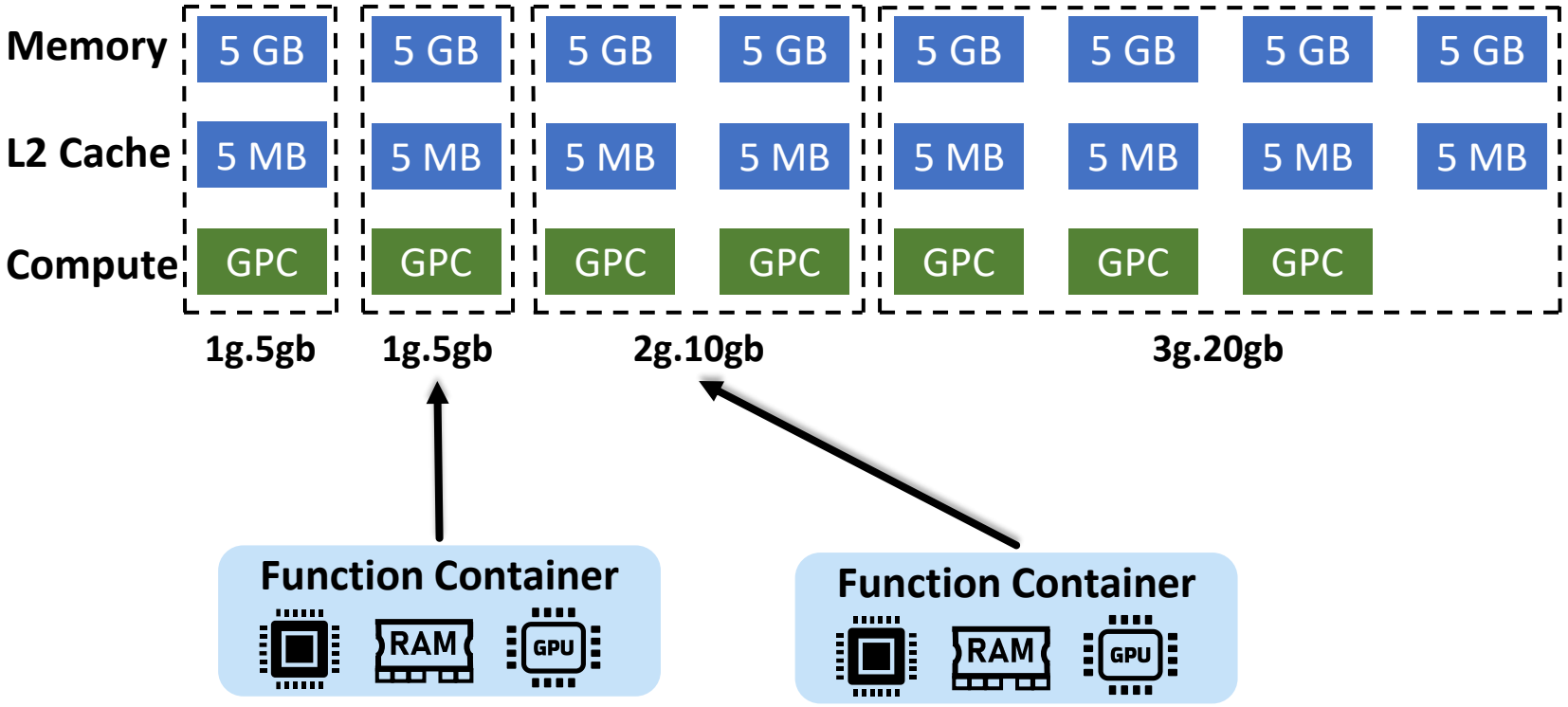


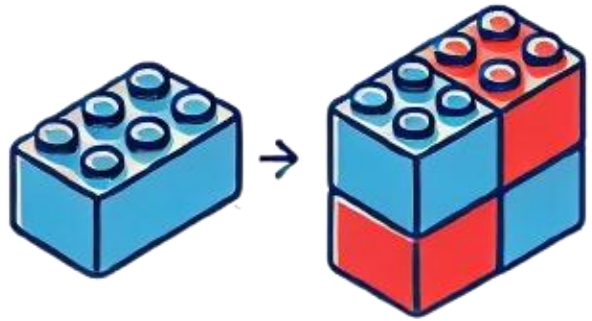
Serverless GPUs: Building from Blocks with Multi-Instance GPU (MIGs)



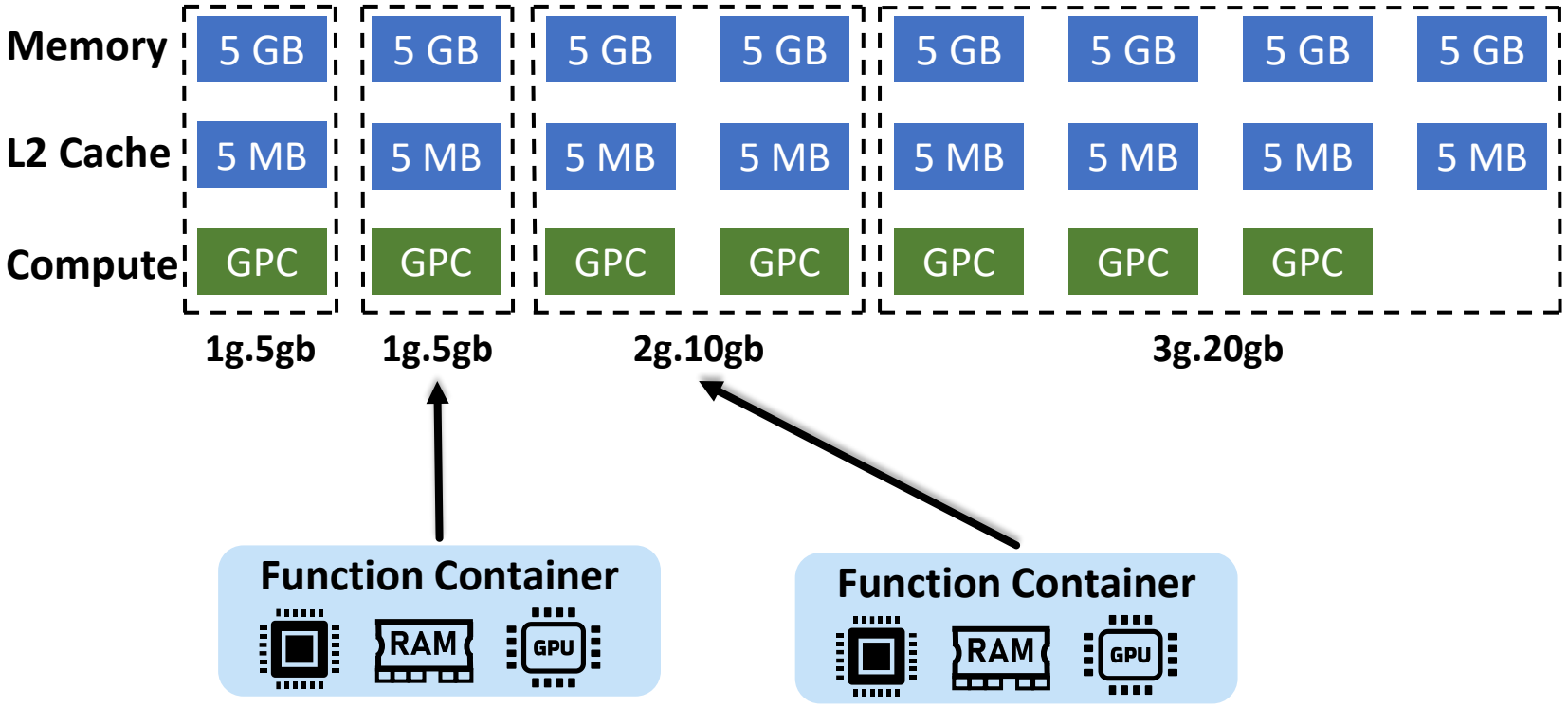


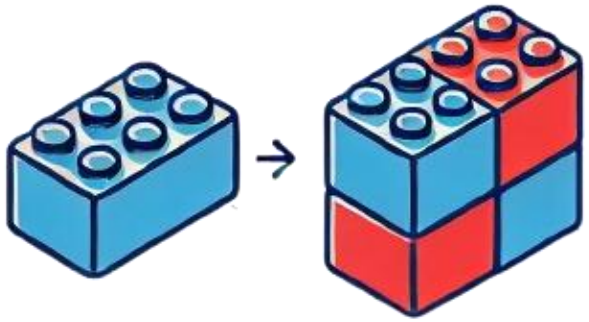
Serverless GPUs: Building from Blocks with Multi-Instance GPU (MIGs)





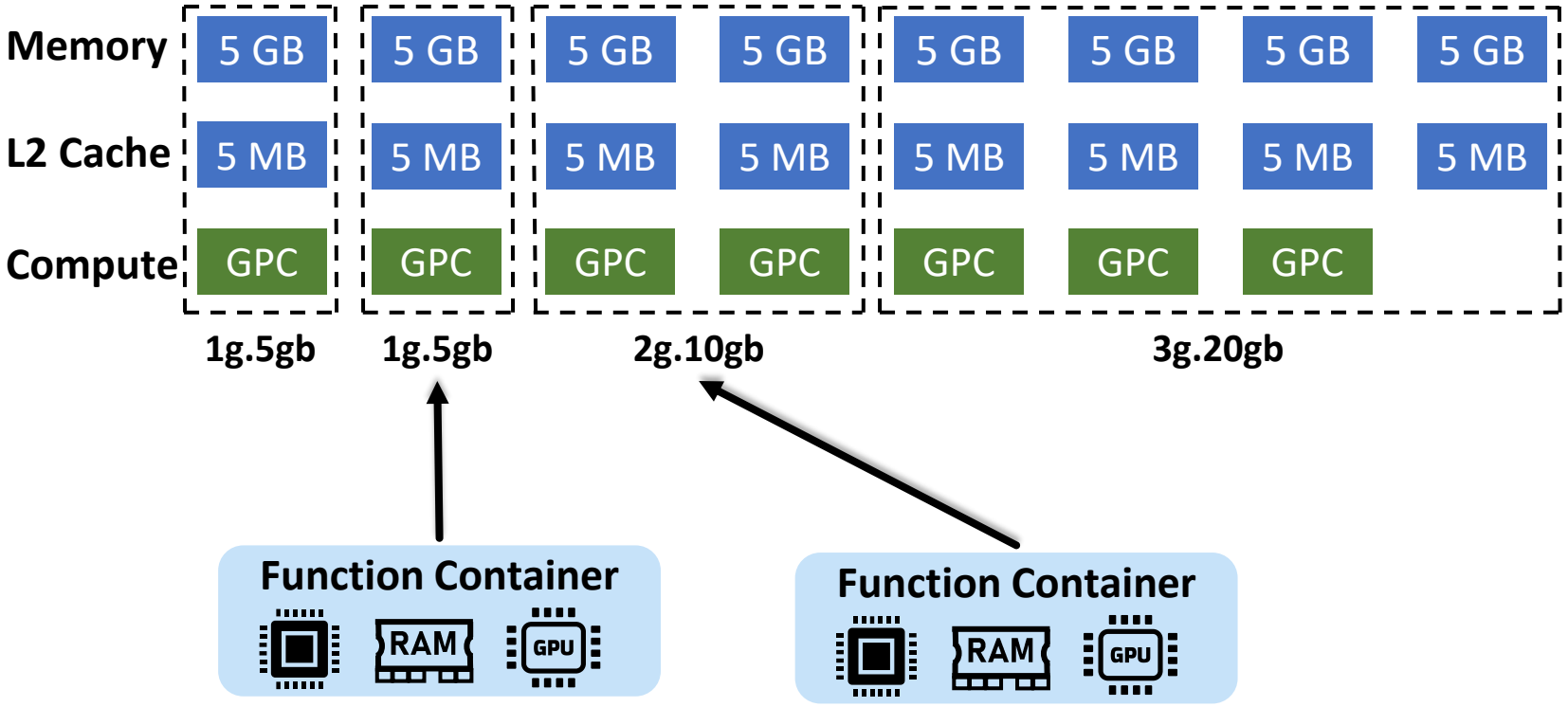
Serverless GPUs: Building from Blocks with Multi-Instance GPU (MIGs)

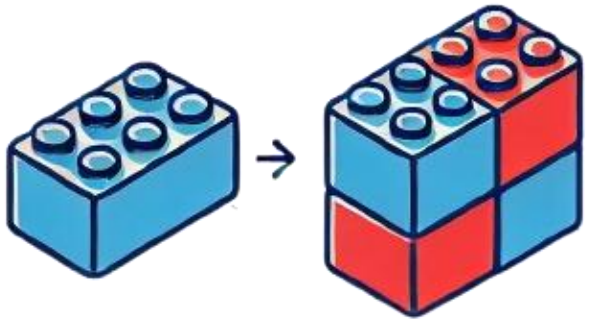




Not elastic!
Partitioning is quite static.

Serverless GPUs: Building from Blocks with Multi-Instance GPU (MIGs)





Not elastic!
Partitioning is quite static.

Static underutilization!
Difficult to migrate between partitions.

Agenda

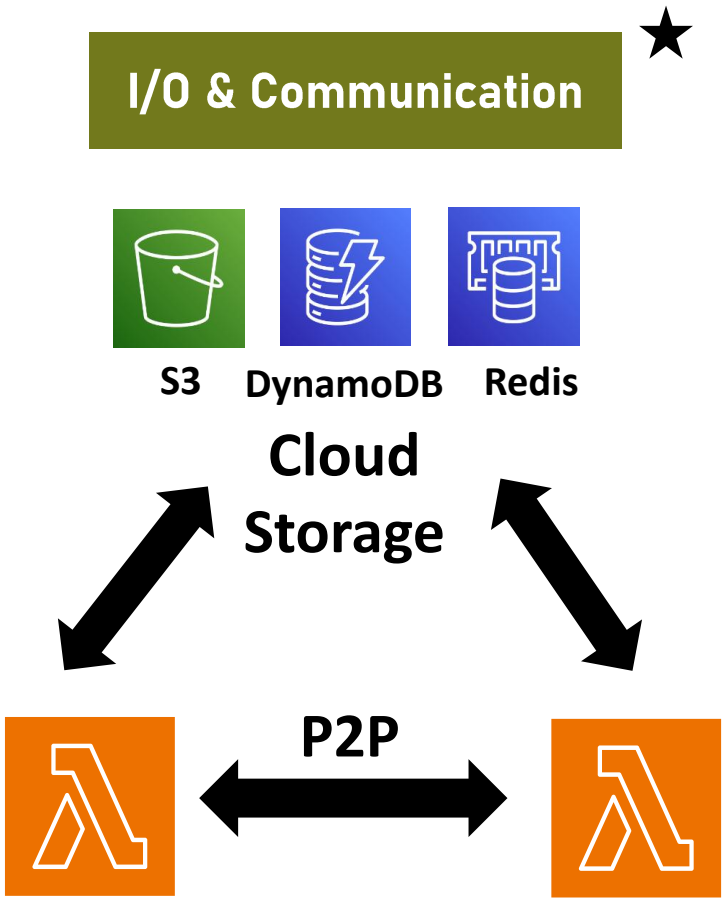
SeBS Slack

- ✓ Part I
  - ✓ What is Serverless?
  - ✓ Benchmarking Suite SeBS
  - ✓ Working with SeBS
- ✓ Hands-on I: Local Deployment & Storage
- ✓ Part II
  - ✓ Communication and Data
  - ✓ Serverless Workflows
  - ✓ Experiments in SeBS
- ✓ Hands-on II: FaaS Platforms & Experiments
- ✓ **Part III**
  - ✓ Research Directions in Serverless & Performance
  - ✓ **Development of SeBS**

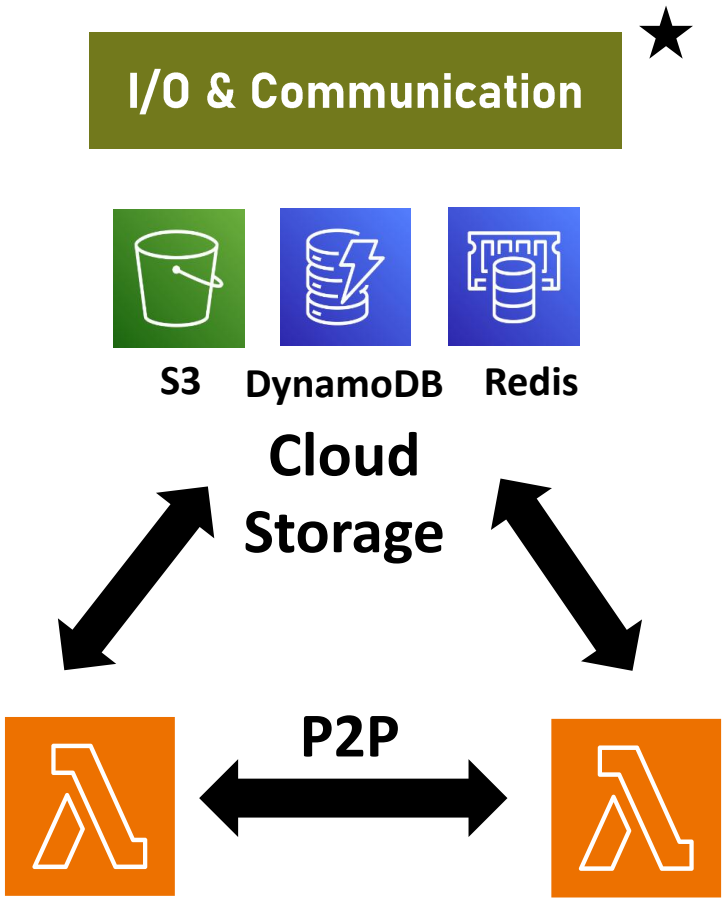
slack **Join SeBS Slack channel: https://serverlessbenchmark.slack.com**

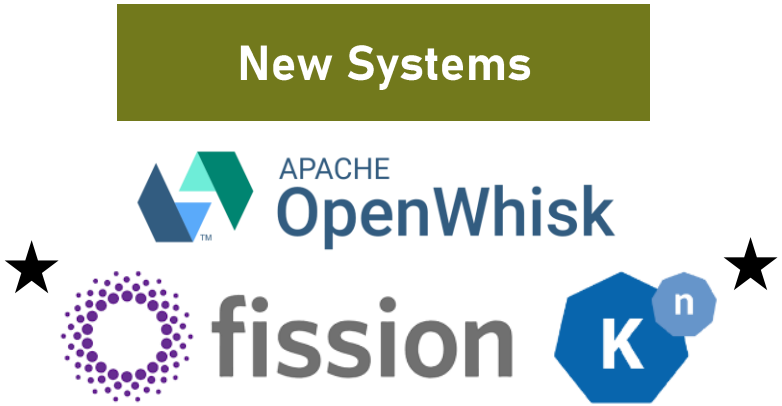
Serverless has been changing – and so did we!

Serverless has been changing – and so did we!



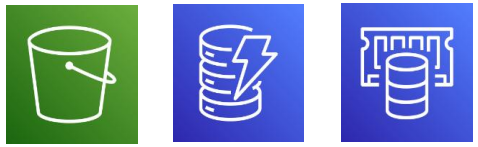
Serverless has been changing – and so did we!

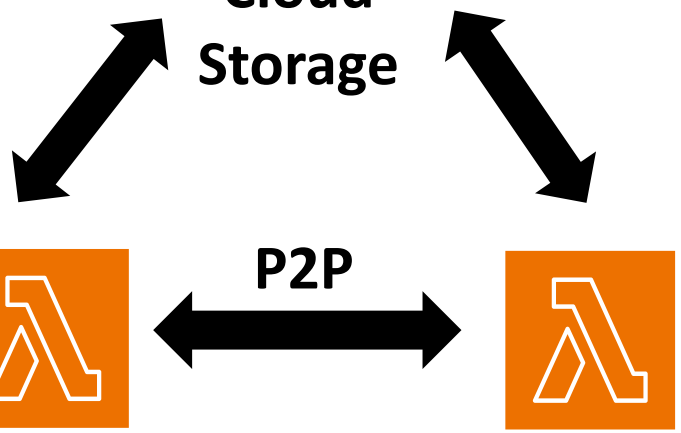




Serverless has been changing – and so did we!

I/O & Communication ★





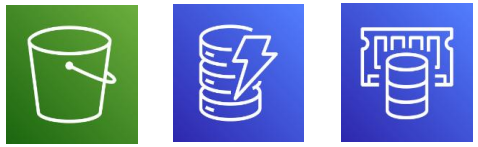
New Systems

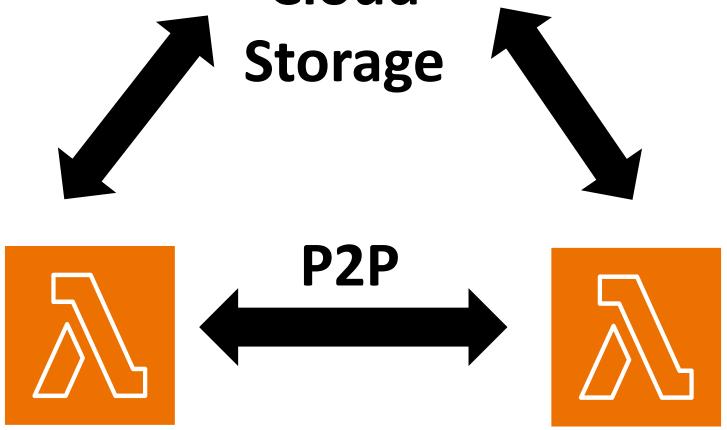
★ ★

NoSQL Storage

Serverless has been changing – and so did we!

I/O & Communication ★

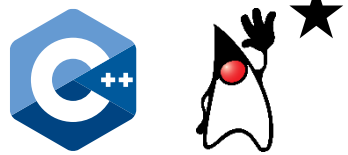




New Systems

★ ★

New Features ★



Cloud Queues
New Language Runtimes
Support for ARM

NoSQL Storage

AWS DynamoDB

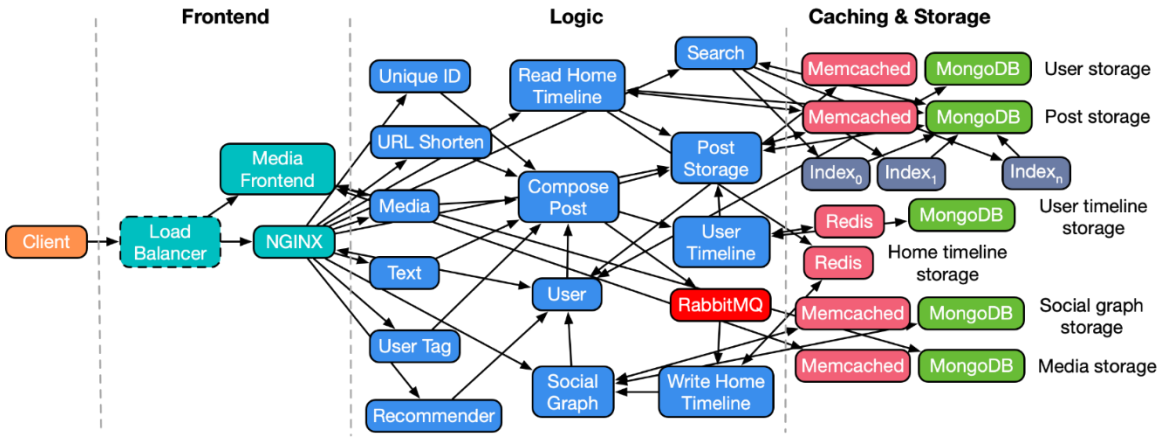
Azure CosmosDB

Google Cloud Datastore

Serverless is changing – and so will we!

Serverless is changing – and so will we!

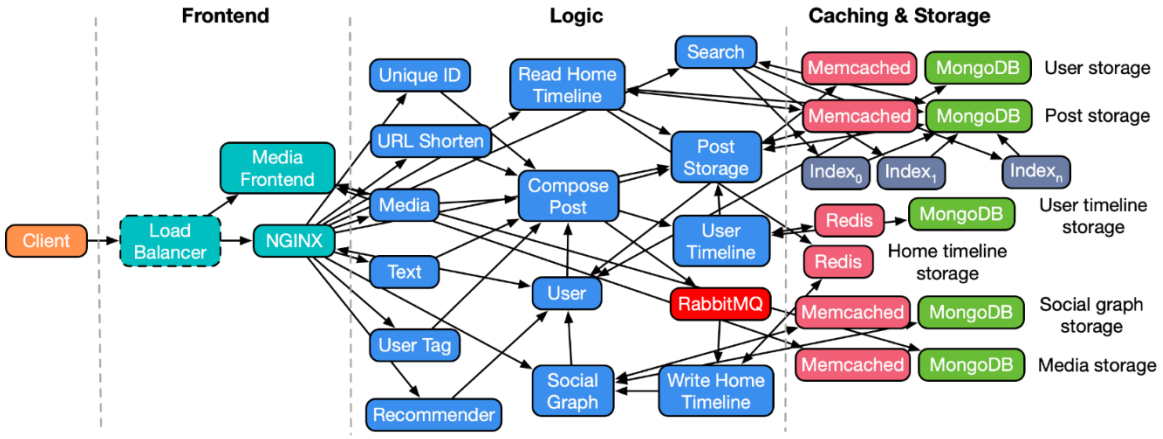
New Workloads



Source: https://github.com/delimitrou/DeathStarBench/tree/master/socialNetwork

Serverless is changing – and so will we!

New Workloads

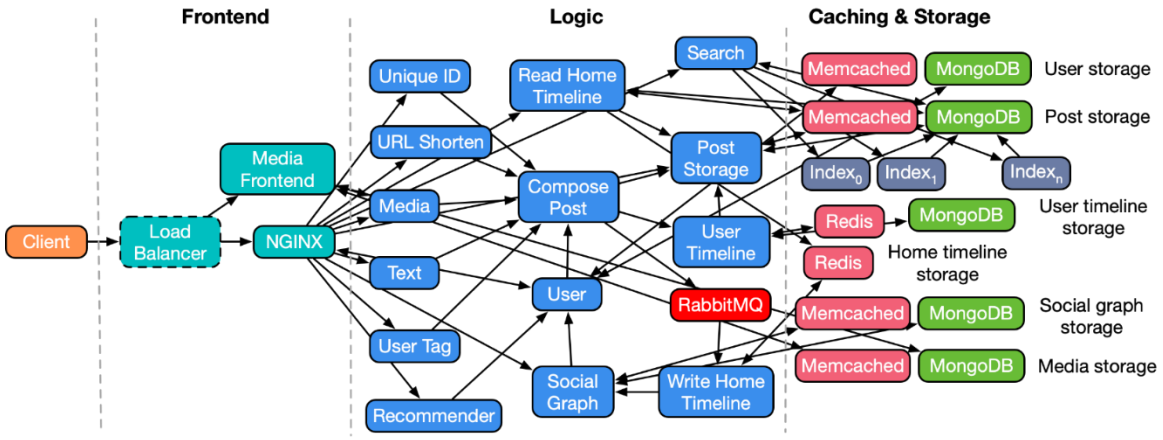


Realistic Invocations

Source: https://github.com/delimitrou/DeathStarBench/tree/master/socialNetwork

Serverless is changing – and so will we!

New Workloads



Realistic Invocations

Serverless in the Wild: Characterizing and Optimizing the Serverless Workload at a

How Does It Function? Characterizing Long-term Trends in Production Serverless Workloads

Pau

The globus compute dataset: An open function-as-a-service dataset from the edge to the cloud

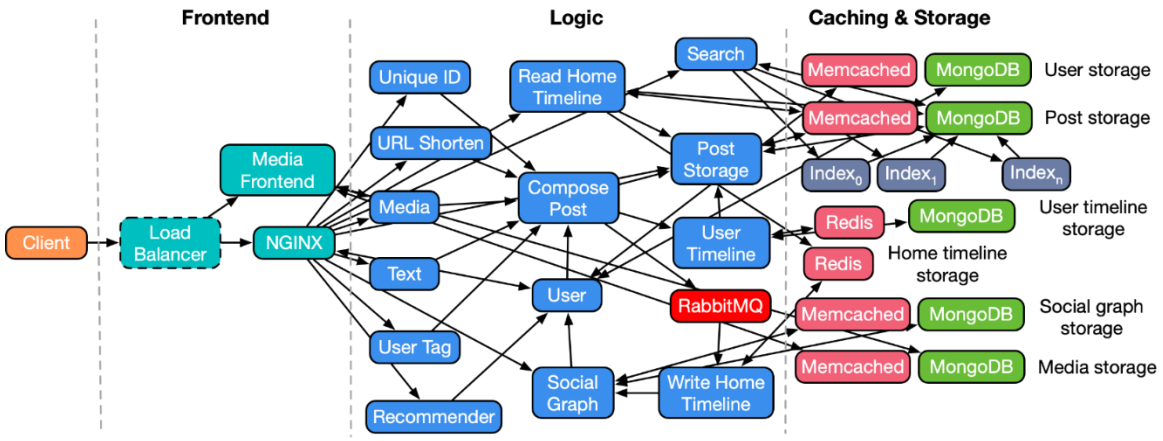
André Bauer [a,b,*], Haochen Pan [a], Ryan Chard [b], Yadu Babuji [a], Josh Bryan [a], Devesh Tiwari [c], Ian Foster [b,a], Kyle Chard [a,b]

[a] University of Chicago, United States
[b] Argonne National Laboratory, United States
[c] Northeastern University, United States

Source: https://github.com/delimitrou/DeathStarBench/tree/master/socialNetwork

Serverless is changing – and so will we!

New Workloads



Realistic Invocations

Serverless in the Wild: Characterizing and Optimizing the Serverless Workload at a

How Does It Function? Characterizing Long-term Trends in Production Serverless Workloads

Pau

The globus compute dataset: An open function-as-a-service dataset from the edge to the cloud

André Bauer [a,b,*], Haochen Pan [a], Ryan Chard [b], Yadu Babuji [a], Josh Bryan [a], Devesh Tiwari [c], Ian Foster [b,a], Kyle Chard [a,b]

[a] University of Chicago, United States
[b] Argonne National Laboratory, United States
[c] Northeastern University, United States

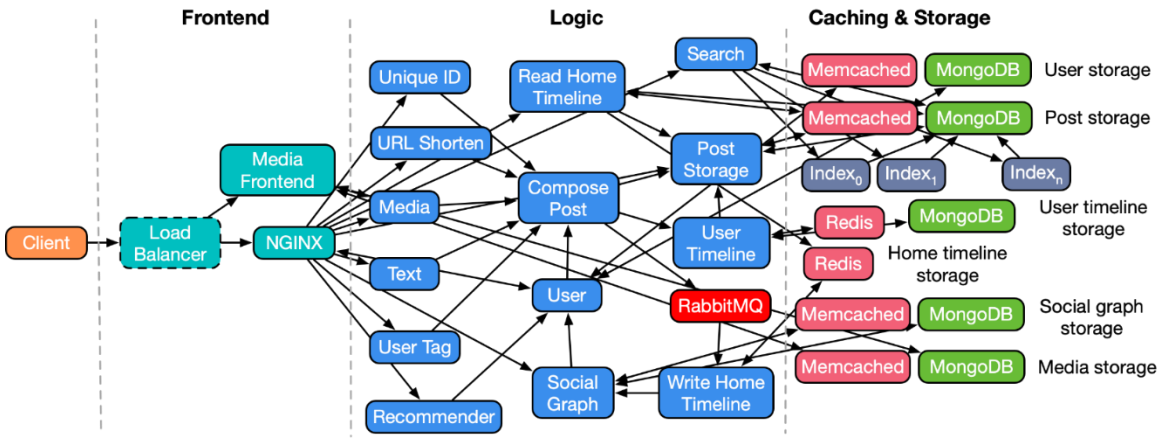
Heterogeneous Serverless

AI/ML is Difficult Without GPUs
How to share GPUs efficiently?

Source: https://github.com/delimitrou/DeathStarBench/tree/master/socialNetwork

Serverless is changing – and so will we!

New Workloads



Realistic Invocations

Serverless in the Wild: Characterizing and Optimizing the Serverless Workload at a

How Does It Function? Characterizing Long-term Trends in Production Serverless Workloads

Pau

The globus compute dataset: An open function-as-a-service dataset from the edge to the cloud

André Bauer [a,b,*], Haochen Pan [a], Ryan Chard [b], Yadu Babuji [a], Josh Bryan [a], Devesh Tiwari [c], Ian Foster [b,a], Kyle Chard [a,b]

[a] University of Chicago, United States
[b] Argonne National Laboratory, United States
[c] Northeastern University, United States

Heterogeneous Serverless

AI/ML is Difficult Without GPUs
How to share GPUs efficiently?

Usability

Bring-your-own-function
Custom deployment, automatic experiments

Source: https://github.com/delimitrou/DeathStarBench/tree/master/socialNetwork

High-Performance Solutions for Serverless

High-Performance Solutions for Serverless

spcl/serverless-benchmarks

spcl/rFaaS

spcl/PraaS

spcl/FaaSKeeper

spcl/FMI

spcl/XaaS

spcl/serverless-benchmarks

More of SPCL's research:

youtube.com/@spcl — 240+ Talks

twitter.com/spcl_eth — 1.7K+ Followers

github.com/spcl — 7.2K+ Stars

... or spcl.ethz.ch

Research Credits Support:

aws

Google Cloud

Google Summer of Code

SeBS Paper

SeBS-Flow Paper

Serverless Projects

spcl/serverless-benchmarks

With contributions from: Michał Podstawski, Laurin Brandner, Larissa Schmid, Nico Graf, Grzegorz Kwaśniewski, Kacper Janda, Mateusz Knapik, Jakub Czerski, Mahla Sarifi, Paweł Żuk, Sascha Kehrli, Oana Rosca, Abhishek Kumar, Prajin Khadka, Horia Mercan, and many others!

More of SPCL's research:

- youtube.com/@spcl — 240+ Talks
- twitter.com/spcl_eth — 1.7K+ Followers
- github.com/spcl — 7.2K+ Stars

… or spcl.ethz.ch

Research Credits Support:

aws

Google Cloud

Google Summer of Code

SeBS Paper

SeBS-Flow Paper

Serverless Projects

spcl/serverless-benchmarks

With contributions from: Michał Podstawski, Laurin Brandner, Larissa Schmid, Nico Graf, Grzegorz Kwaśniewski, Kacper Janda, Mateusz Knapik, Jakub Czerski, Mahla Sarifi, Paweł Żuk, Sascha Kehrli, Oana Rosca, Abhishek Kumar, Prajin Khadka, Horia Mercan, and many others!

Do you have an interesting, cutting-edge serverless workload?

Research Credits Support:

aws

Google Cloud

Google Summer of Code

SeBS Paper

SeBS-Flow Paper

Serverless Projects

More of SPCL's research:

youtube.com/@spcl — 240+ Talks

twitter.com/spcl_eth — 1.7K+ Followers

github.com/spcl — 7.2K+ Stars

... or spcl.ethz.ch