MARCIN COPIK, TORSTEN HOEFLER

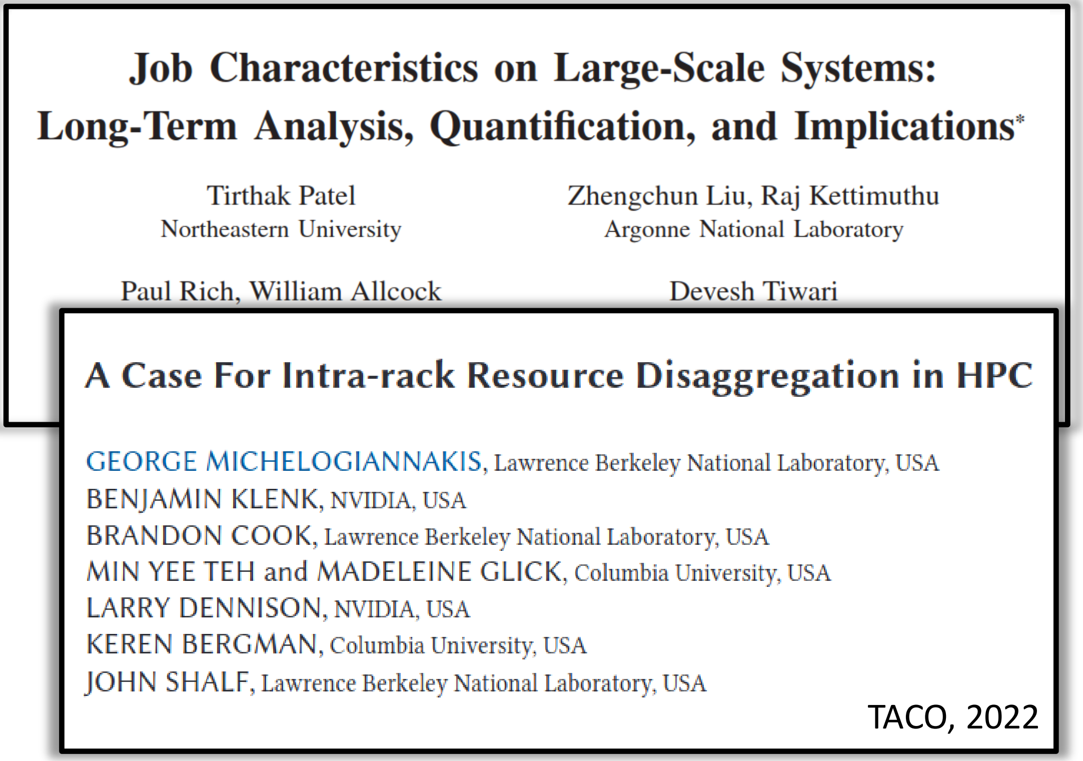
High Performance Serverless for HPC and Clouds

SPCL

SC23
Denver, CO | i am hpc.

Tracking Wasted Money

Tracking Wasted Money



Job Characteristics on Large-Scale Systems: Long-Term Analysis, Quantification, and Implications*

Tirthak Patel
Northeastern University

Zhengchun Liu, Raj Kettimuthu
Argonne National Laboratory

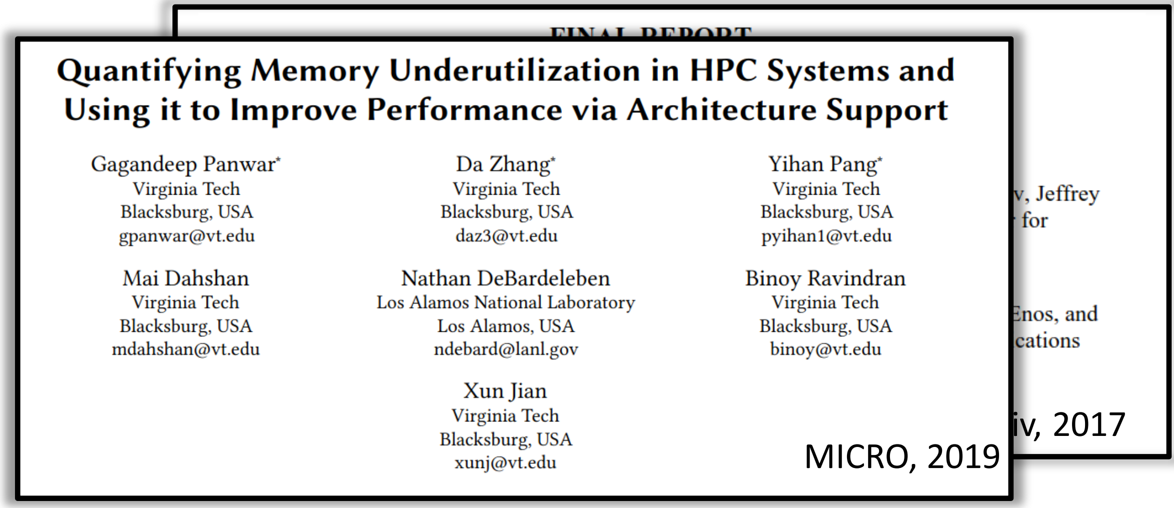
Paul Rich, William Allcock

Devesh Tiwari

A Case For Intra-rack Resource Disaggregation in HPC

GEORGE MICHELOGIANNAKIS, Lawrence Berkeley National Laboratory, USA
BENJAMIN KLENK, NVIDIA, USA
BRANDON COOK, Lawrence Berkeley National Laboratory, USA
MIN YEE TEH and MADELEINE GLICK, Columbia University, USA
LARRY DENNISON, NVIDIA, USA
KEREN BERGMAN, Columbia University, USA
JOHN SHALF, Lawrence Berkeley National Laboratory, USA

TACO, 2022



FINAL REPORT

v, Jeffrey
for

Enos, and
cations

iv, 2017

Quantifying Memory Underutilization in HPC Systems and Using it to Improve Performance via Architecture Support

Gagandeep Panwar*
Virginia Tech
Blacksburg, USA
gpanwar@vt.edu

Da Zhang*
Virginia Tech
Blacksburg, USA
daz3@vt.edu

Yihan Pang*
Virginia Tech
Blacksburg, USA
pyihan1@vt.edu

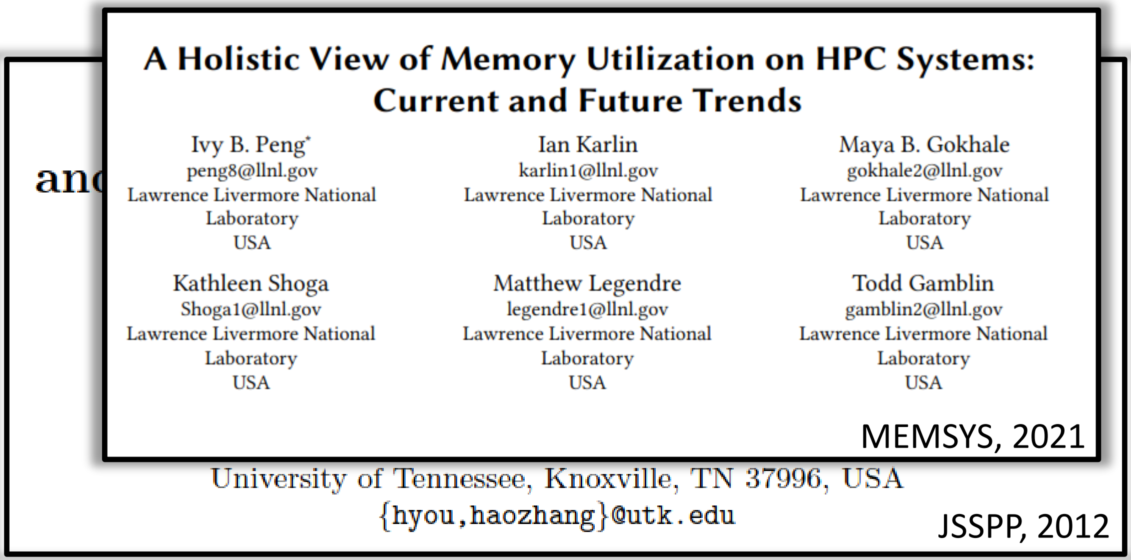
Mai Dahshan
Virginia Tech
Blacksburg, USA
mdahshan@vt.edu

Nathan DeBardeleben
Los Alamos National Laboratory
Los Alamos, USA
ndebard@lanl.gov

Binoy Ravindran
Virginia Tech
Blacksburg, USA
binoy@vt.edu

Xun Jian
Virginia Tech
Blacksburg, USA
xunj@vt.edu

MICRO, 2019



and

A Holistic View of Memory Utilization on HPC Systems: Current and Future Trends

Ivy B. Peng*
peng8@llnl.gov
Lawrence Livermore National Laboratory
USA

Ian Karlin
karlin1@llnl.gov
Lawrence Livermore National Laboratory
USA

Maya B. Gokhale
gokhale2@llnl.gov
Lawrence Livermore National Laboratory
USA

Kathleen Shoga
Shoga1@llnl.gov
Lawrence Livermore National Laboratory
USA

Matthew Legendre
legendre1@llnl.gov
Lawrence Livermore National Laboratory
USA

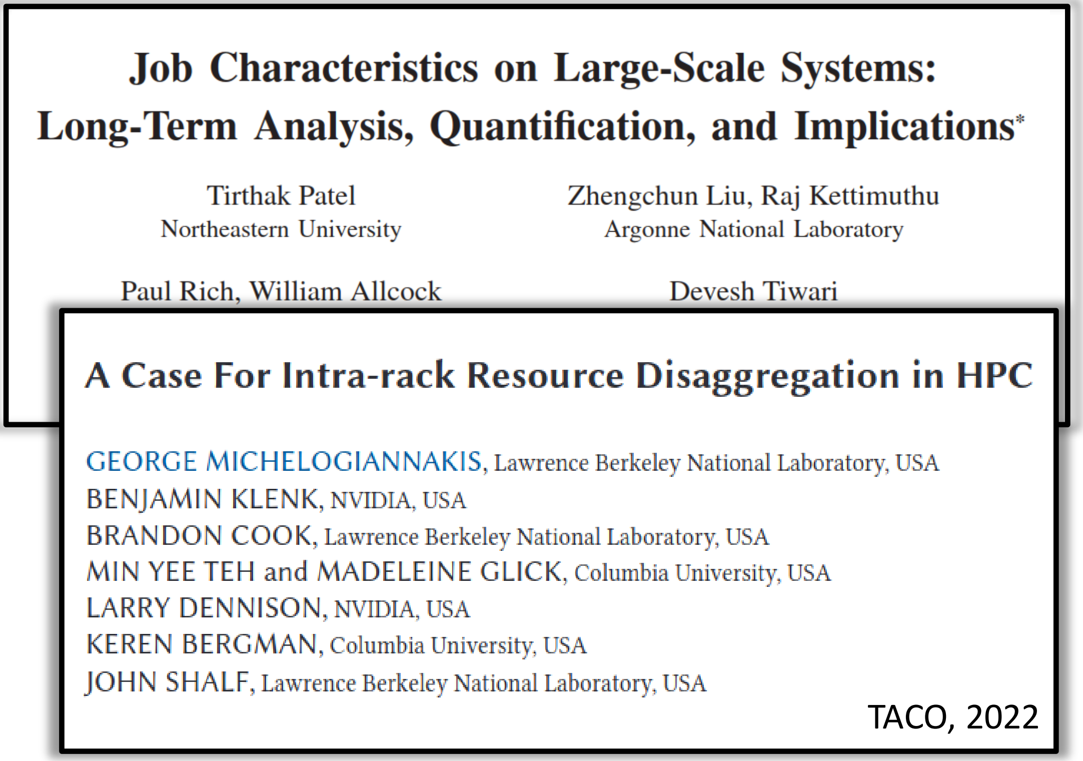
Todd Gamblin
gamblin2@llnl.gov
Lawrence Livermore National Laboratory
USA

MEMSYS, 2021

University of Tennessee, Knoxville, TN 37996, USA
{hyou,haozhang}@utk.edu

JSSPP, 2012

Tracking Wasted Money



Job Characteristics on Large-Scale Systems: Long-Term Analysis, Quantification, and Implications[*]

Tirthak Patel
Northeastern University

Zhengchun Liu, Raj Kettimuthu
Argonne National Laboratory

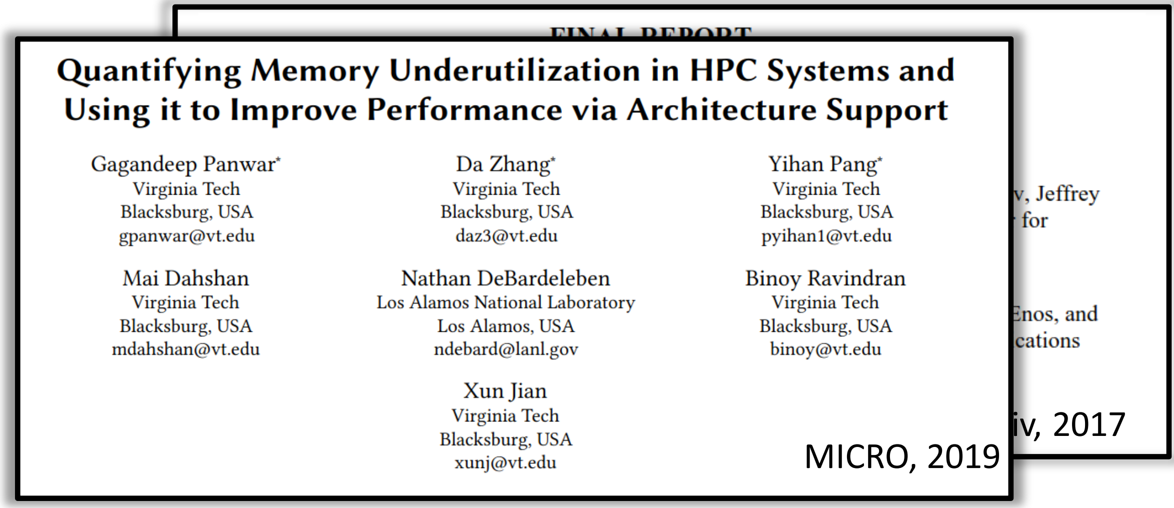
Paul Rich, William Allcock

Devesh Tiwari

A Case For Intra-rack Resource Disaggregation in HPC

GEORGE MICHELOGIANNAKIS, Lawrence Berkeley National Laboratory, USA
BENJAMIN KLENK, NVIDIA, USA
BRANDON COOK, Lawrence Berkeley National Laboratory, USA
MIN YEE TEH and MADELEINE GLICK, Columbia University, USA
LARRY DENNISON, NVIDIA, USA
KEREN BERGMAN, Columbia University, USA
JOHN SHALF, Lawrence Berkeley National Laboratory, USA

TACO, 2022



FINAL REPORT

Quantifying Memory Underutilization in HPC Systems and Using it to Improve Performance via Architecture Support

Gagandeep Panwar[*]
Virginia Tech
Blacksburg, USA
gpanwar@vt.edu

Da Zhang[*]
Virginia Tech
Blacksburg, USA
daz3@vt.edu

Yihan Pang[*]
Virginia Tech
Blacksburg, USA
pyihan1@vt.edu

Mai Dahshan
Virginia Tech
Blacksburg, USA
mdahshan@vt.edu

Nathan DeBardeleben
Los Alamos National Laboratory
Los Alamos, USA
ndebard@lanl.gov

Binoy Ravindran
Virginia Tech
Blacksburg, USA
binoy@vt.edu

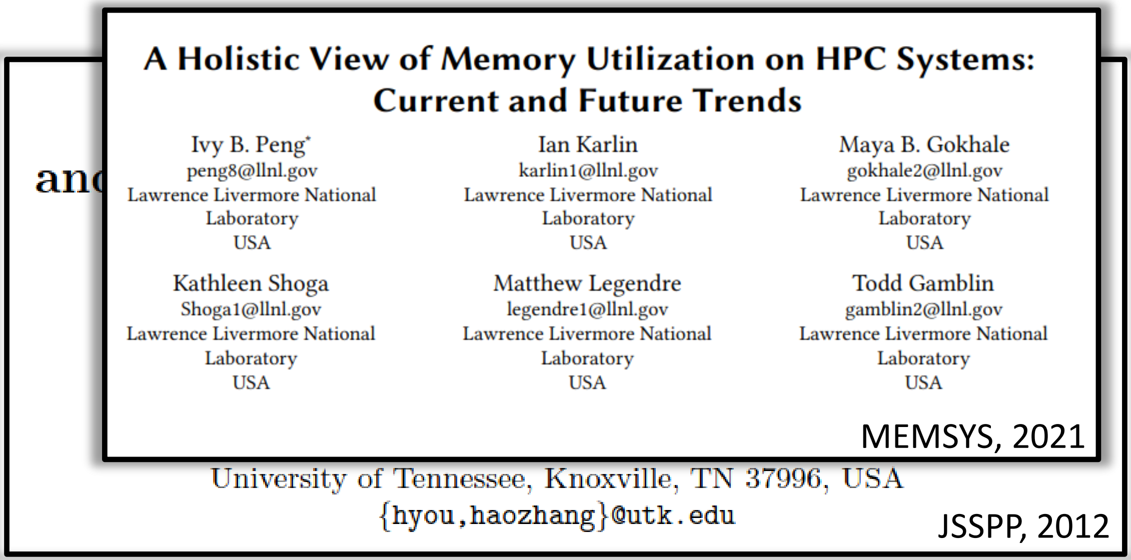
Xun Jian
Virginia Tech
Blacksburg, USA
xunj@vt.edu

MICRO, 2019

v, Jeffrey
for

Enos, and
cations

iv, 2017



A Holistic View of Memory Utilization on HPC Systems: Current and Future Trends

Ivy B. Peng[*]
peng8@llnl.gov
Lawrence Livermore National Laboratory
USA

Ian Karlin
karlin1@llnl.gov
Lawrence Livermore National Laboratory
USA

Maya B. Gokhale
gokhale2@llnl.gov
Lawrence Livermore National Laboratory
USA

Kathleen Shoga
Shoga1@llnl.gov
Lawrence Livermore National Laboratory
USA

Matthew Legendre
legendre1@llnl.gov
Lawrence Livermore National Laboratory
USA

Todd Gamblin
gamblin2@llnl.gov
Lawrence Livermore National Laboratory
USA

MEMSYS, 2021

an

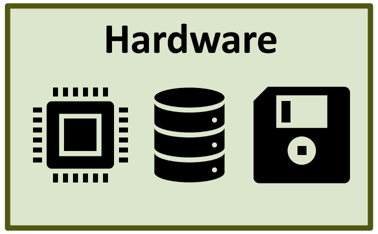
University of Tennessee, Knoxville, TN 37996, USA
{hyou,haozhang}@utk.edu

JSSPP, 2012

Can we solve underutilization with sharing and fine-grained allocations?

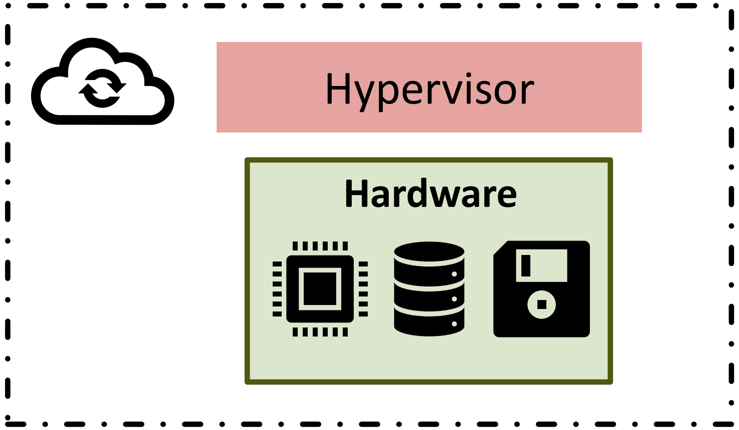
Cloud and Serverless

Cloud and Serverless

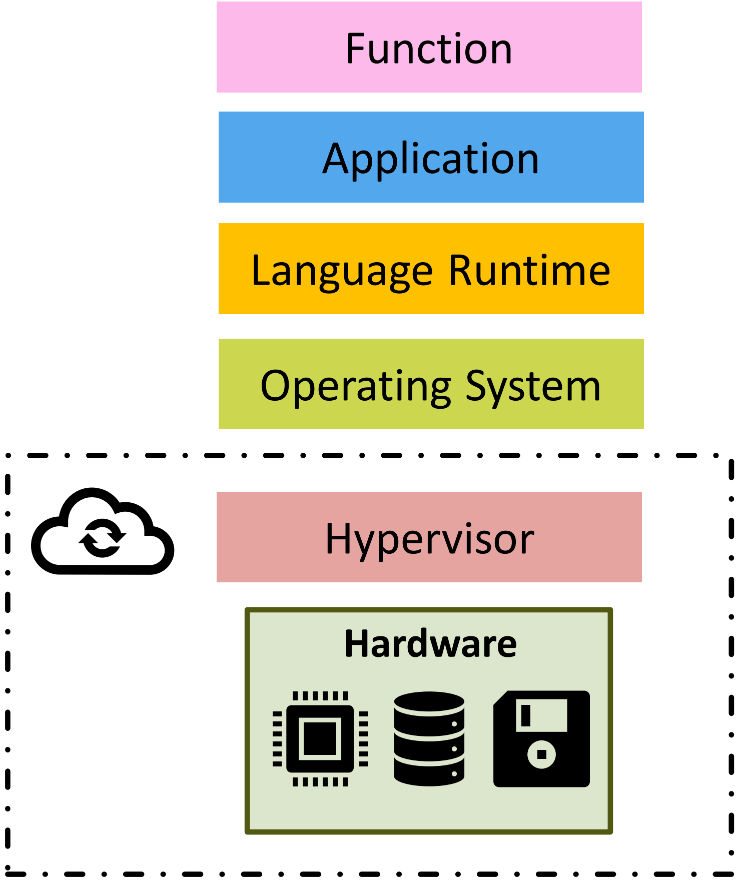


Hardware

Cloud and Serverless

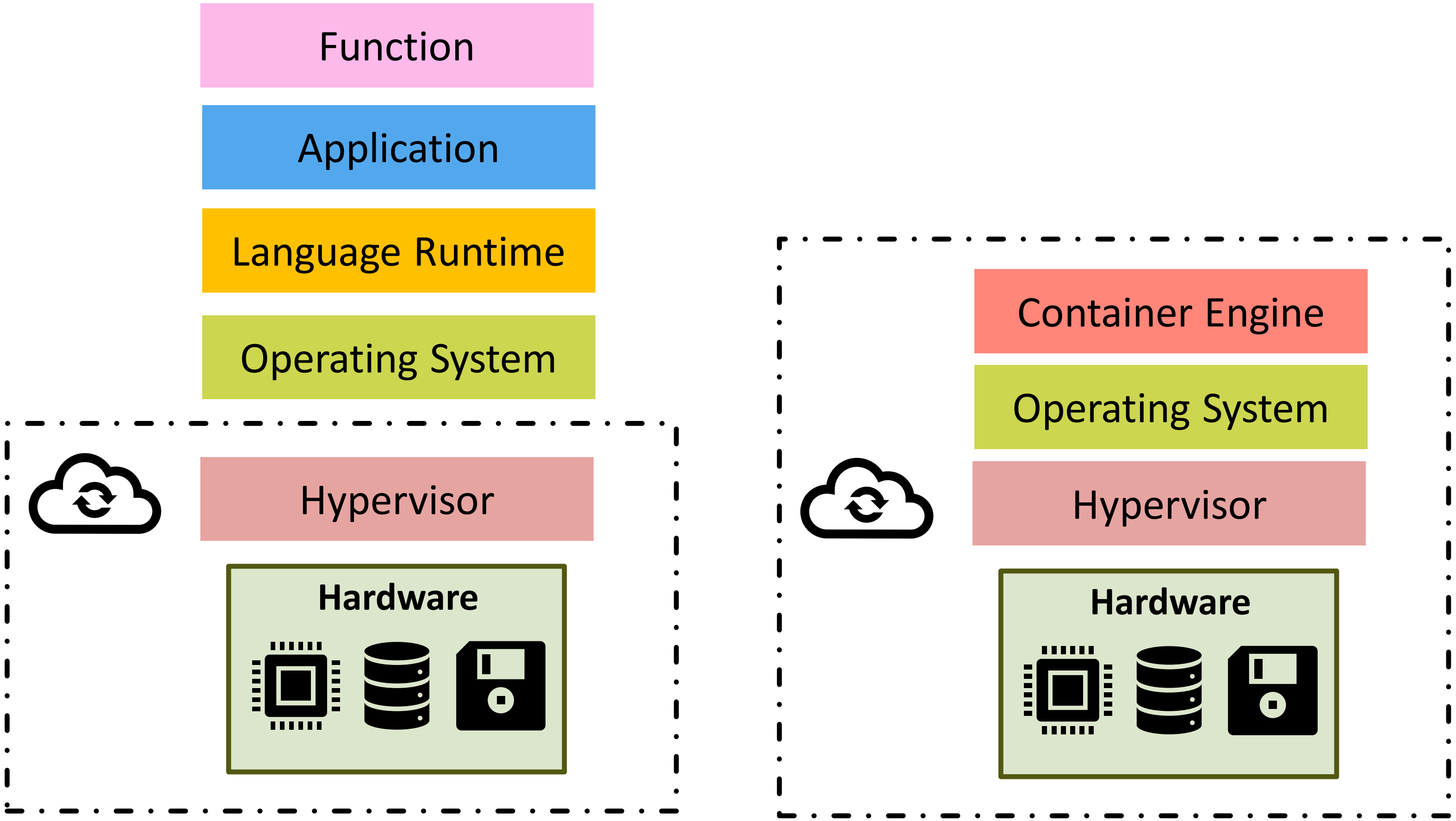


Cloud and Serverless



Virtual Machine

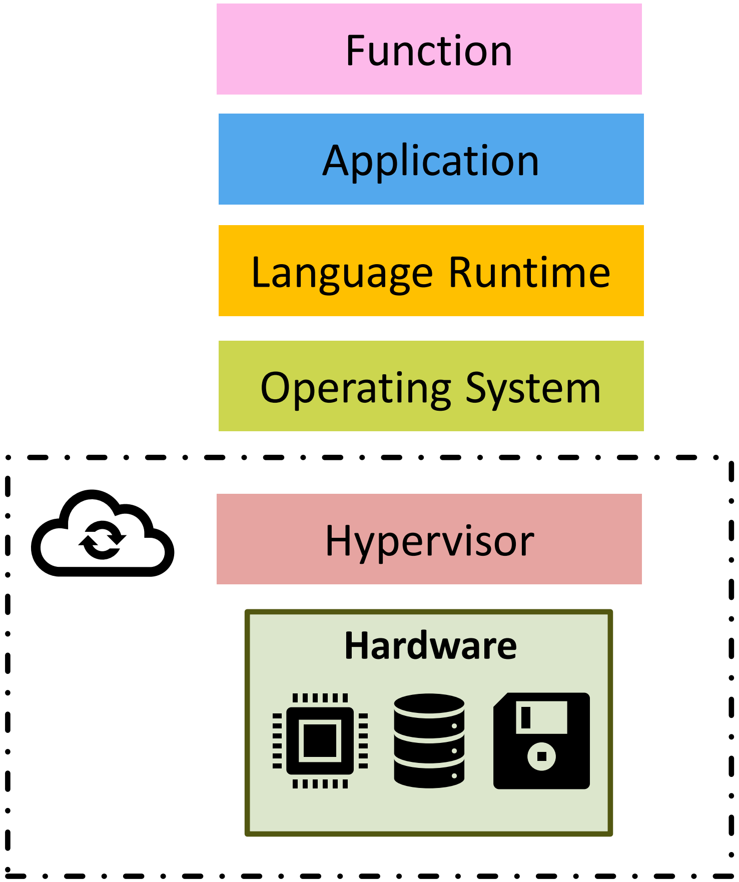
Cloud and Serverless



Virtual Machine

Containers

Cloud and Serverless



Function

Application

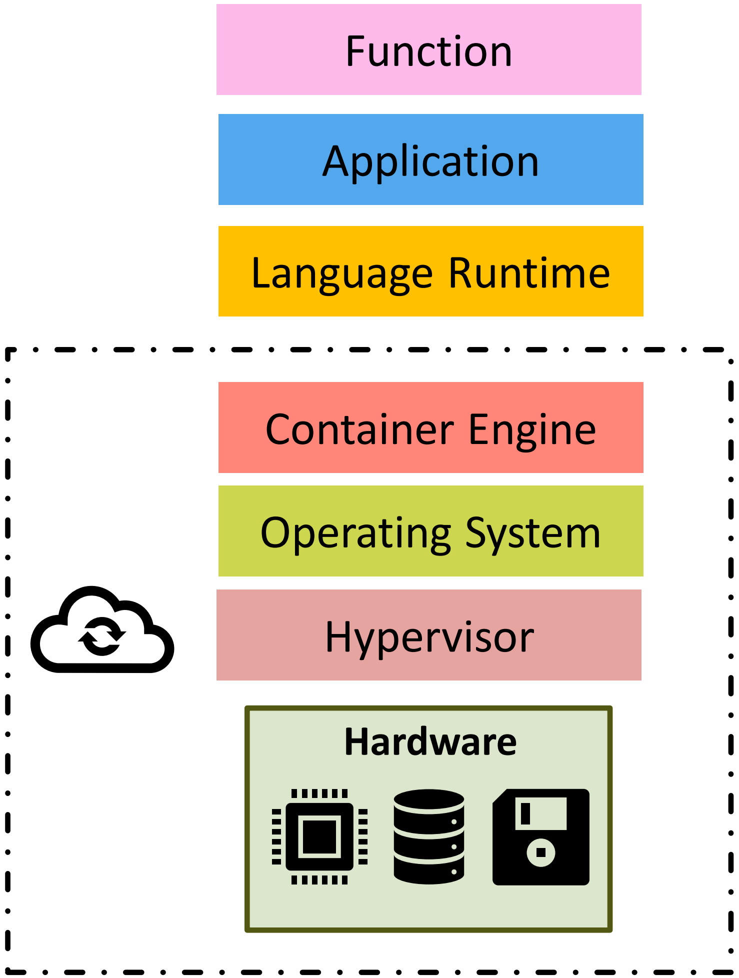
Language Runtime

Operating System

Hypervisor

Hardware

Virtual Machine



Function

Application

Language Runtime

Container Engine

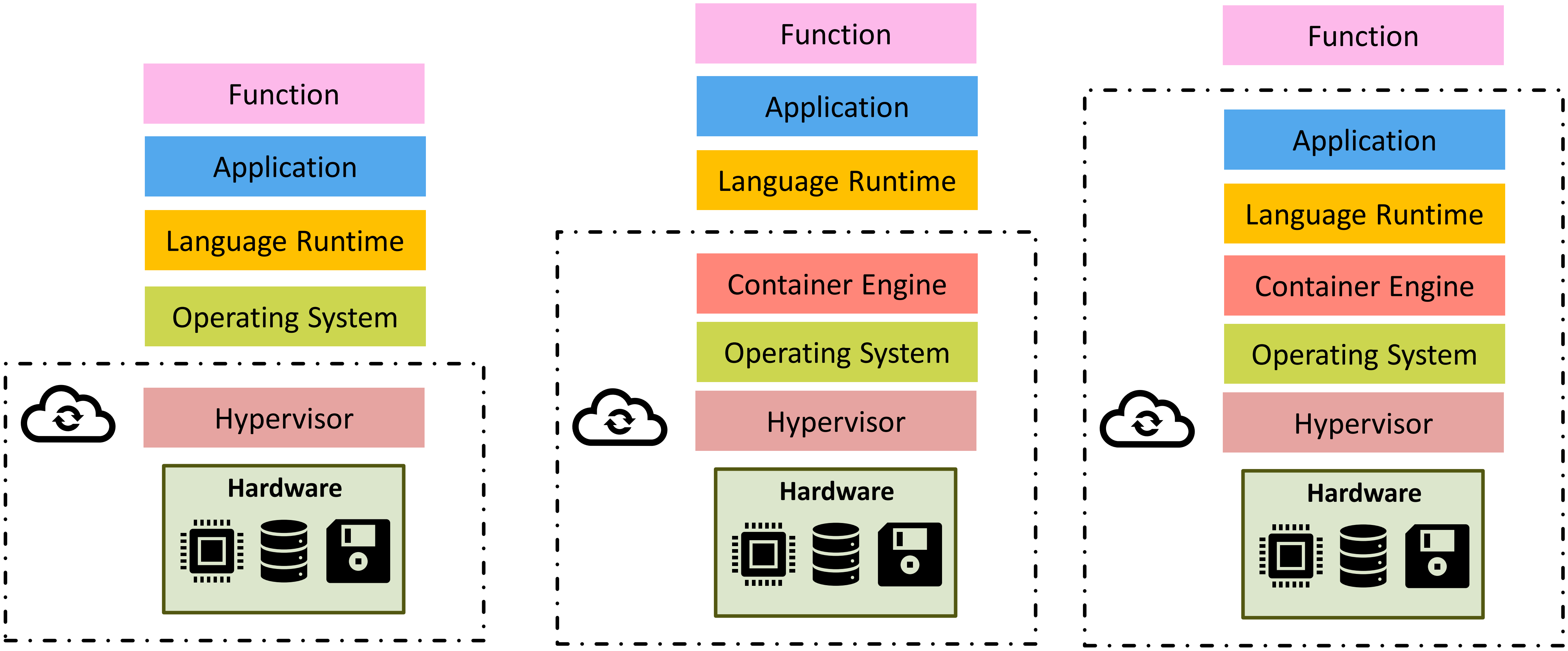
Operating System

Hypervisor

Hardware

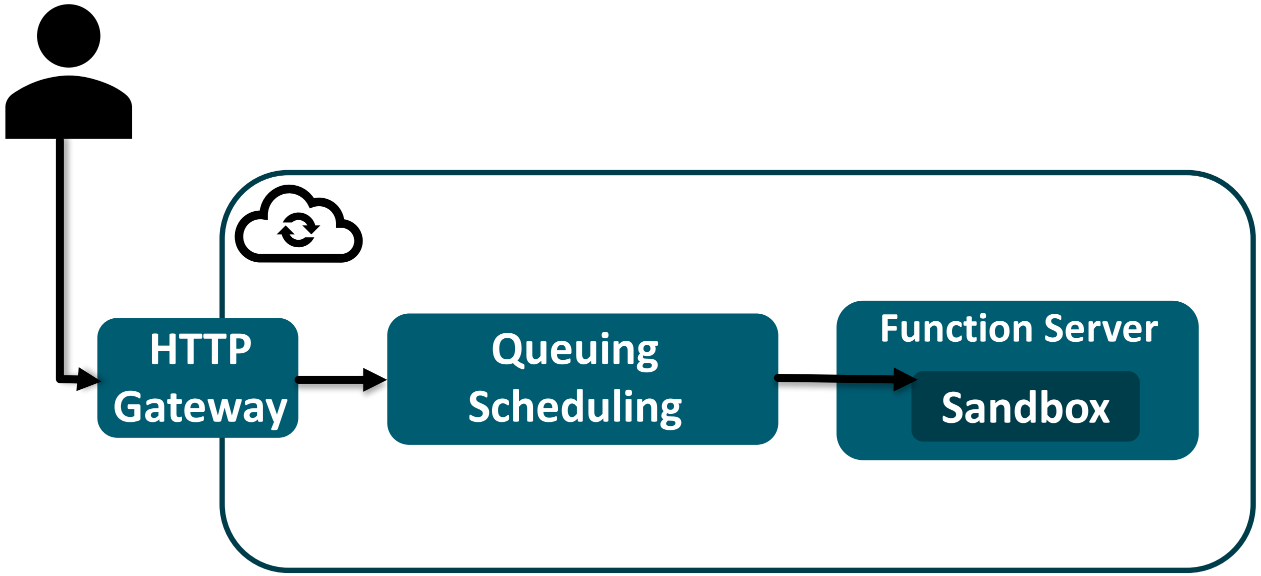
Containers

Cloud and Serverless

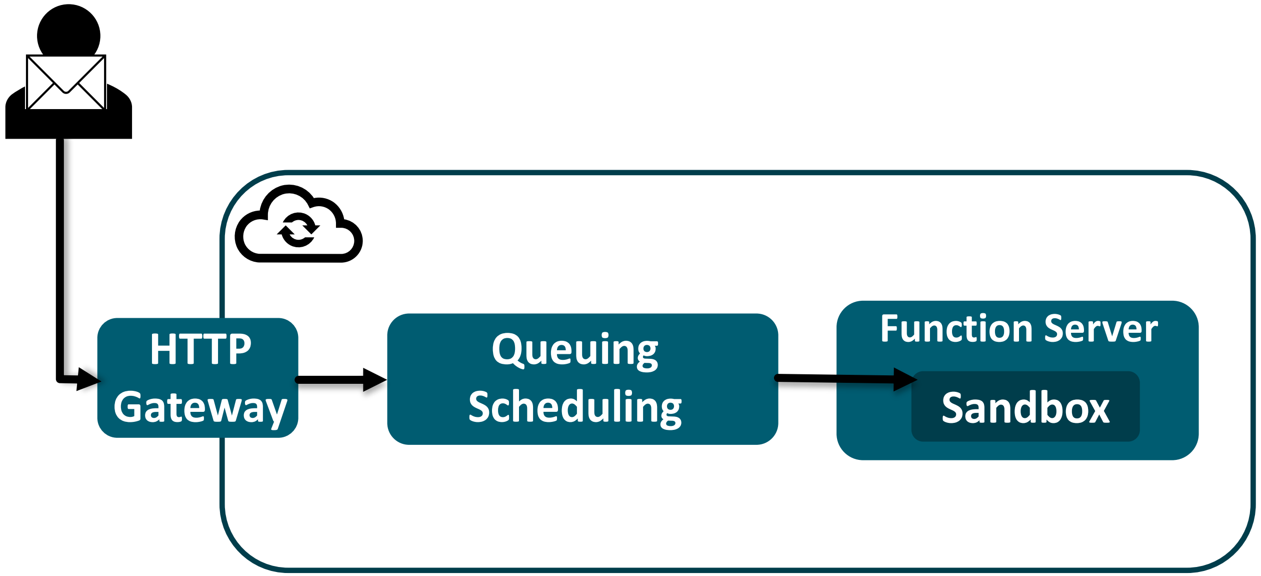


Virtual Machine **Containers** **Functions**

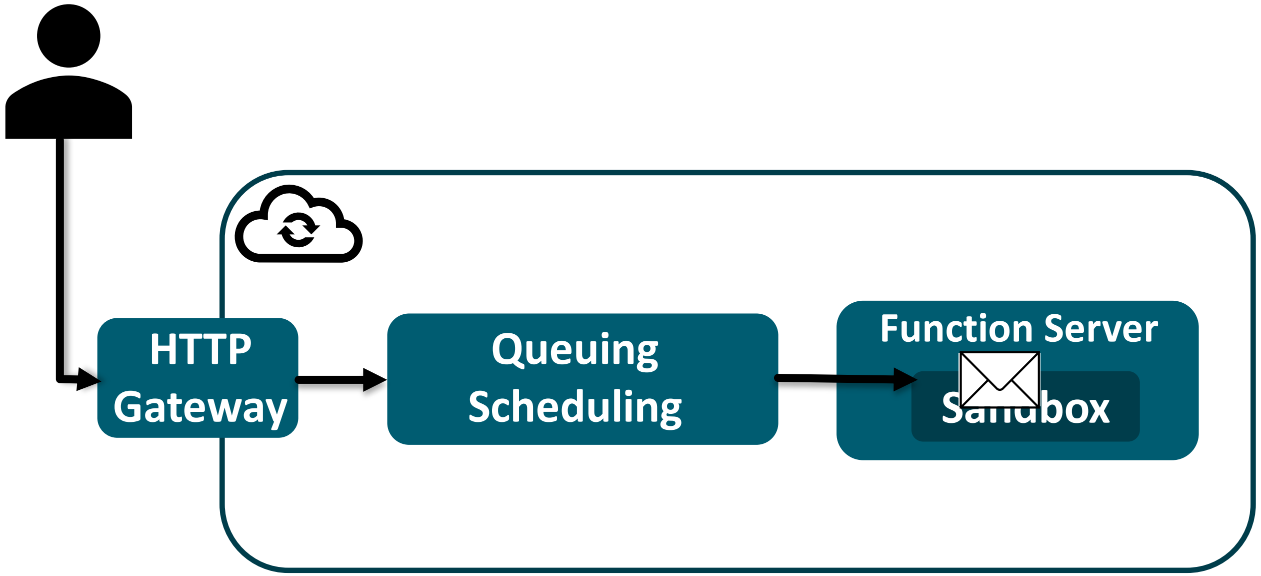
How does Function-as-a-Service (FaaS) work?



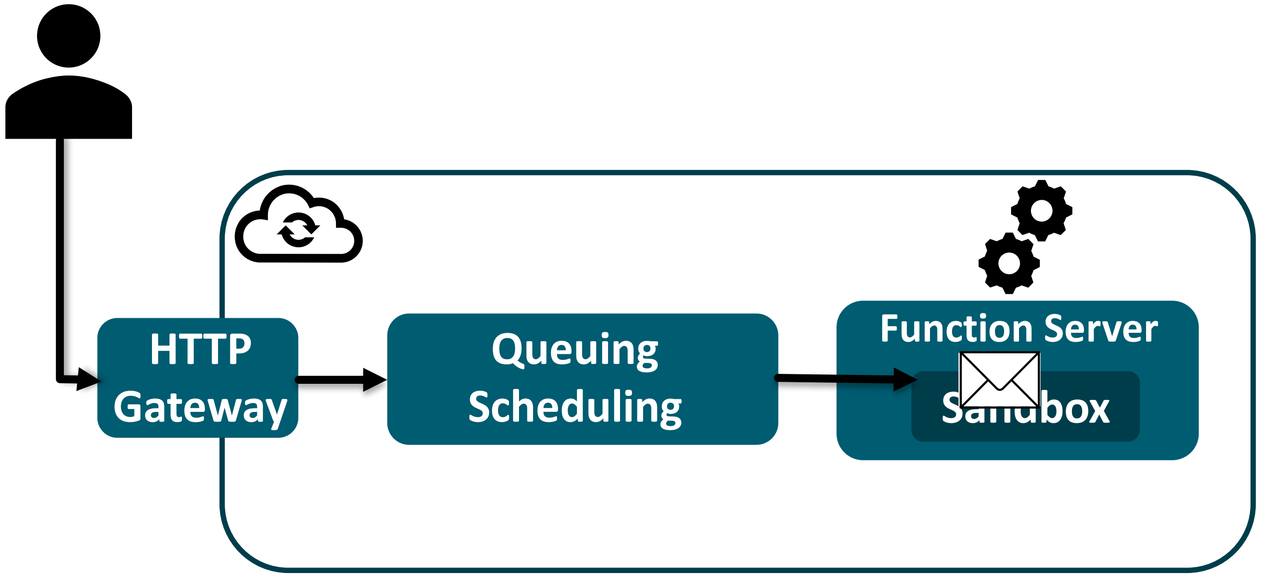
How does Function-as-a-Service (FaaS) work?



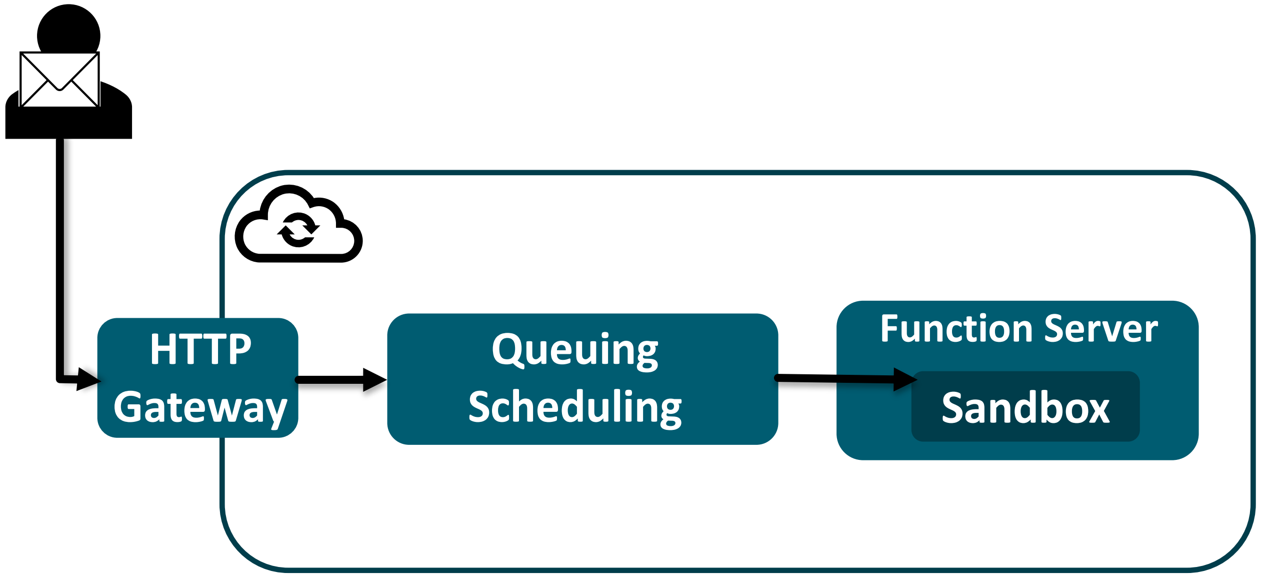
How does Function-as-a-Service (FaaS) work?



How does Function-as-a-Service (FaaS) work?



How does Function-as-a-Service (FaaS) work?



SeBS: The Serverless Benchmark Suite

"SeBS: a Serverless Benchmark Suite for Function-as-a-Service Computing", ACM/IFIP Middleware 2021

SeBS: The Serverless Benchmark Suite

Serverless Platforms

APACHE OpenWhisk

“SeBS: a Serverless Benchmark Suite for Function-as-a-Service Computing”, ACM/IFIP Middleware 2021

SeBS: The Serverless Benchmark Suite

Serverless Platforms

APACHE OpenWhisk

Benchmarks

python™

node JS

“SeBS: a Serverless Benchmark Suite for Function-as-a-Service Computing”, ACM/IFIP Middleware 2021

SeBS: The Serverless Benchmark Suite

Serverless Platforms

APACHE OpenWhisk

Benchmarks

python™

node JS

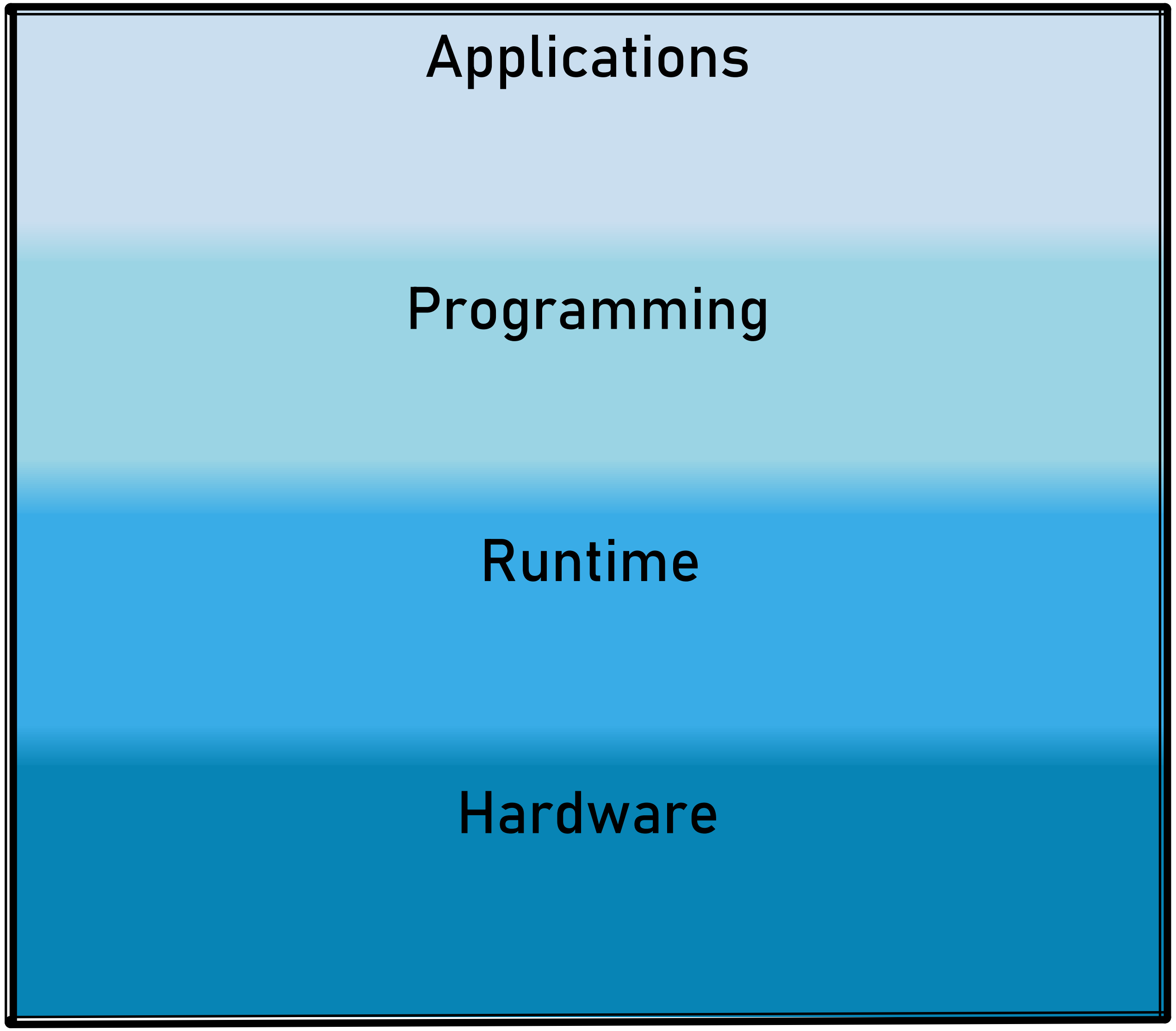
Experiments

Performance & Cost
Invocation Overhead
Container Eviction
Serverless Communication
Serverless Workflows

“SeBS: a Serverless Benchmark Suite for Function-as-a-Service Computing”, ACM/IFIP Middleware 2021

Serverless for High-Performance Applications

Serverless for High-Performance Applications



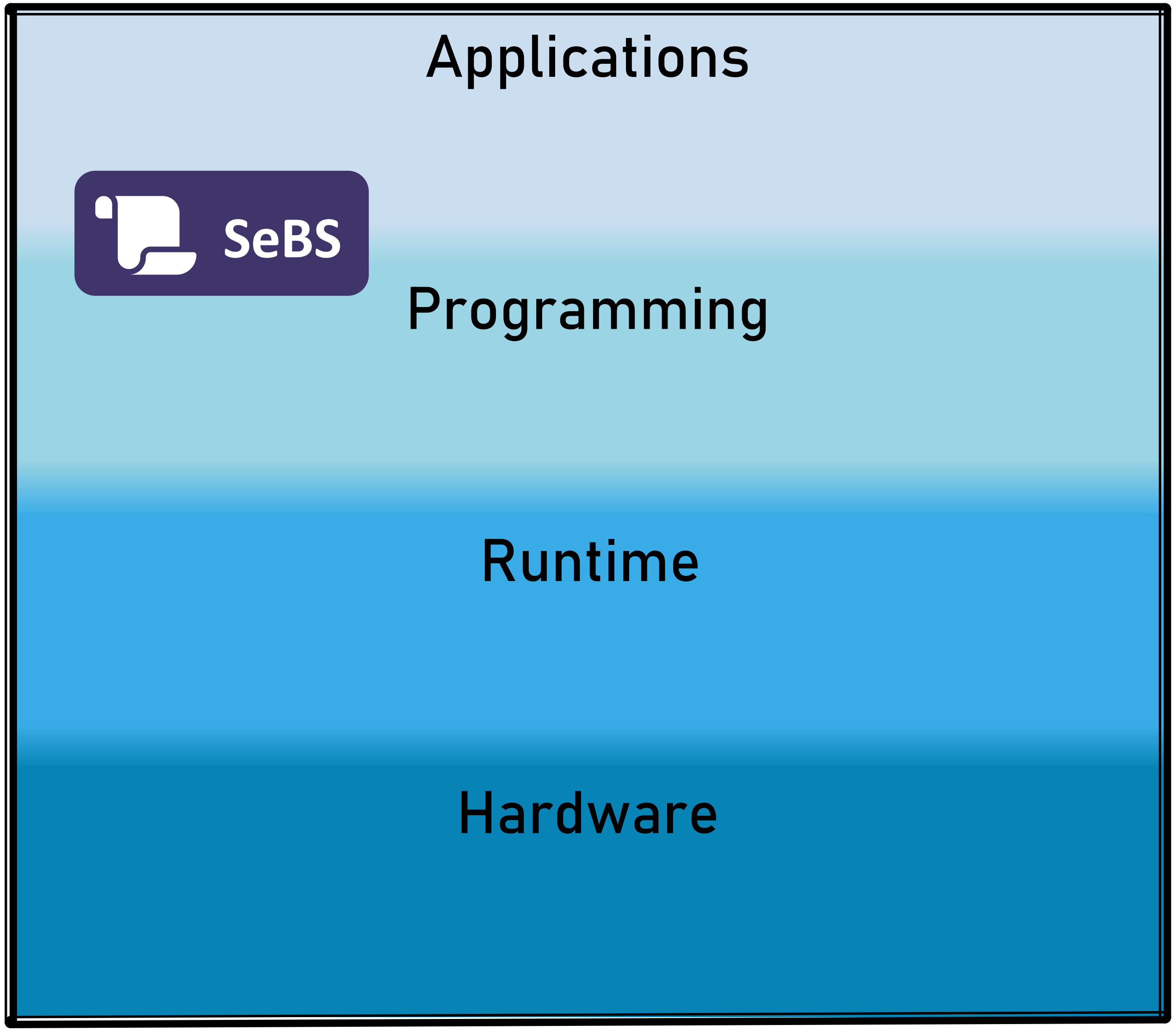
Applications

Programming

Runtime

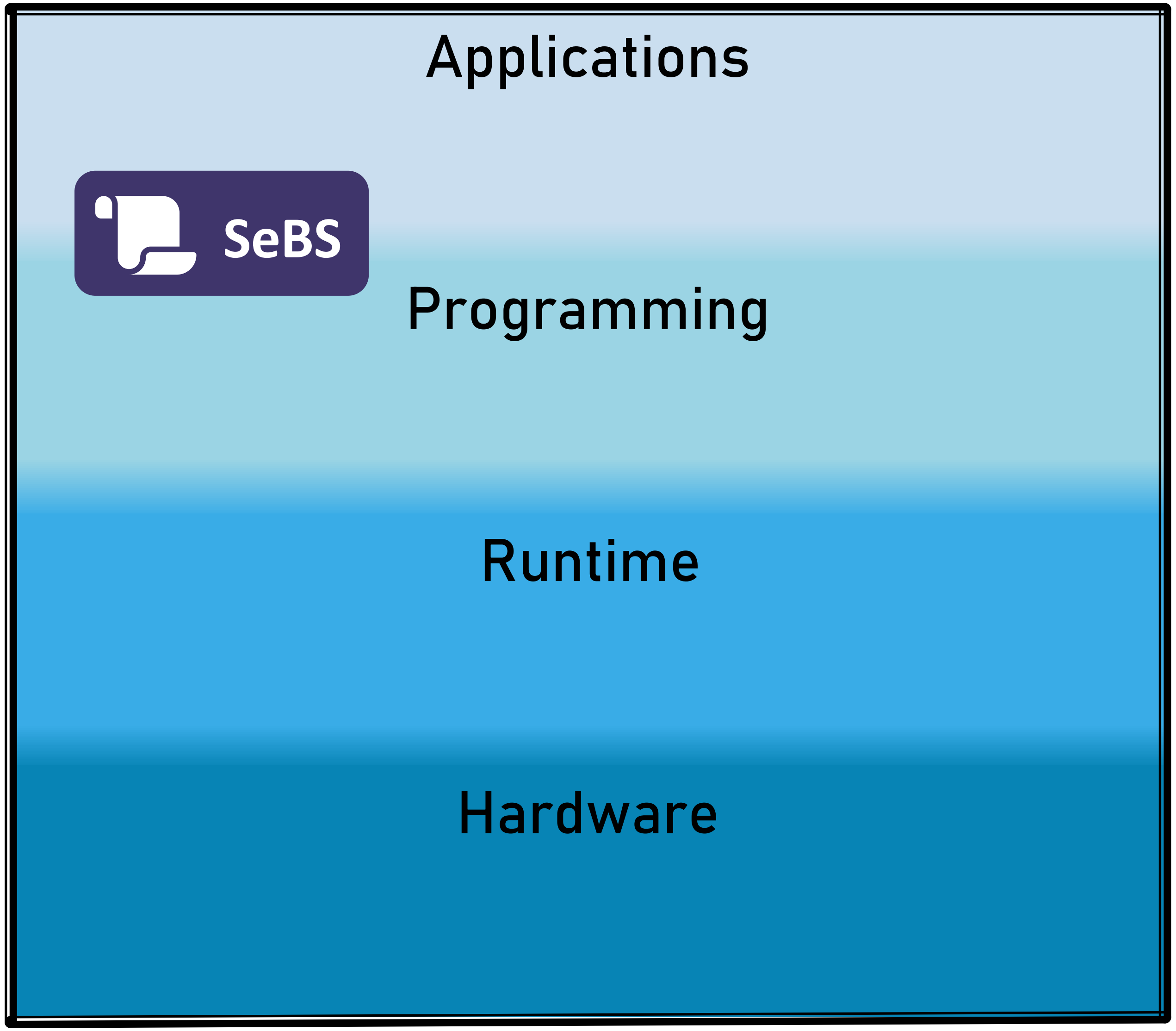
Hardware

Serverless for High-Performance Applications



Serverless for High-Performance Applications

Functions are expensive to invoke.



Applications

SeBS

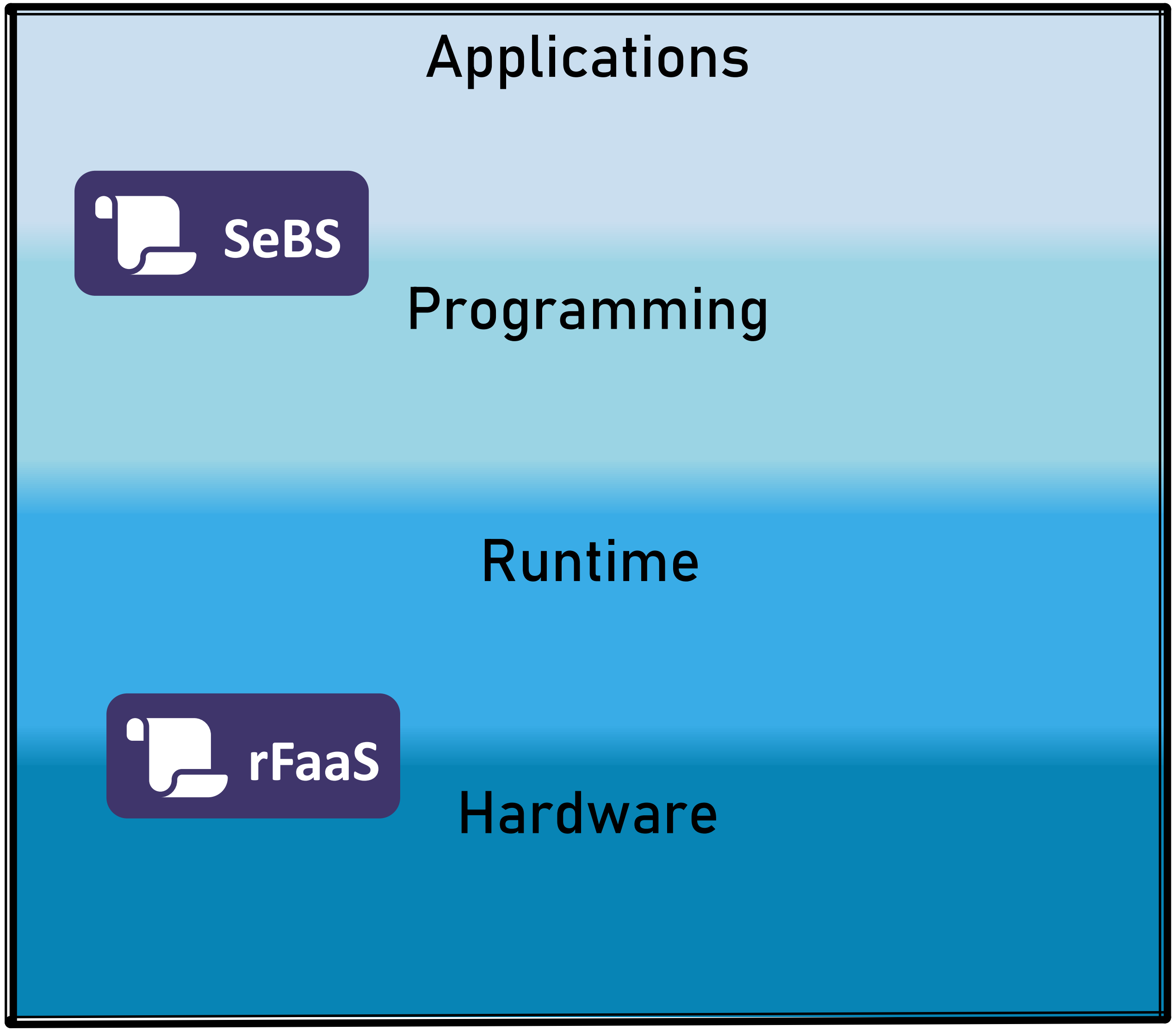
Programming

Runtime

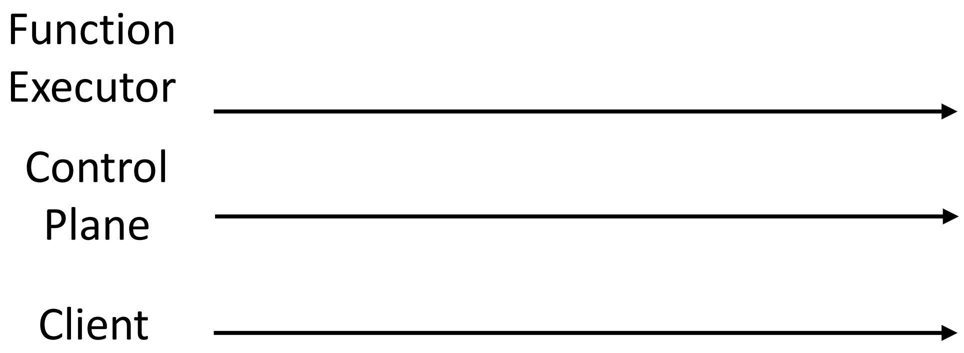
Hardware

Serverless for High-Performance Applications

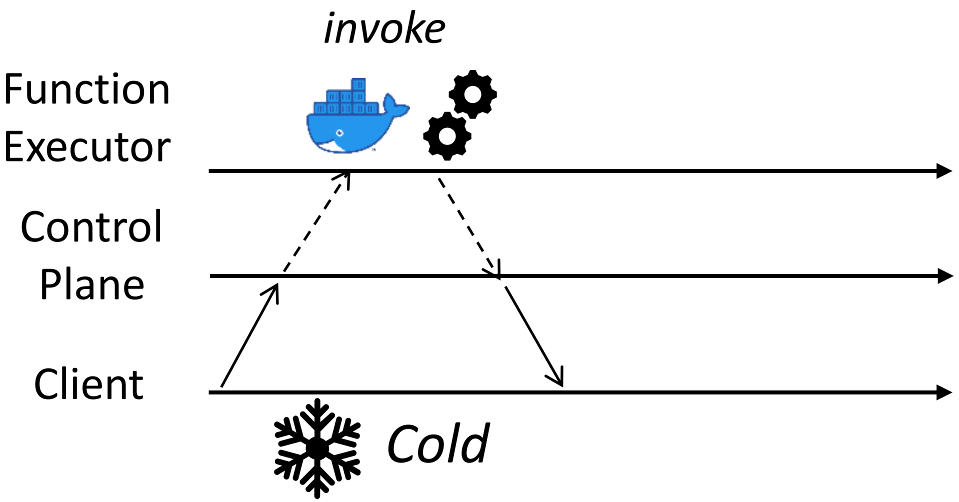
Functions are expensive to invoke.



Invocations in FaaS and rFaaS

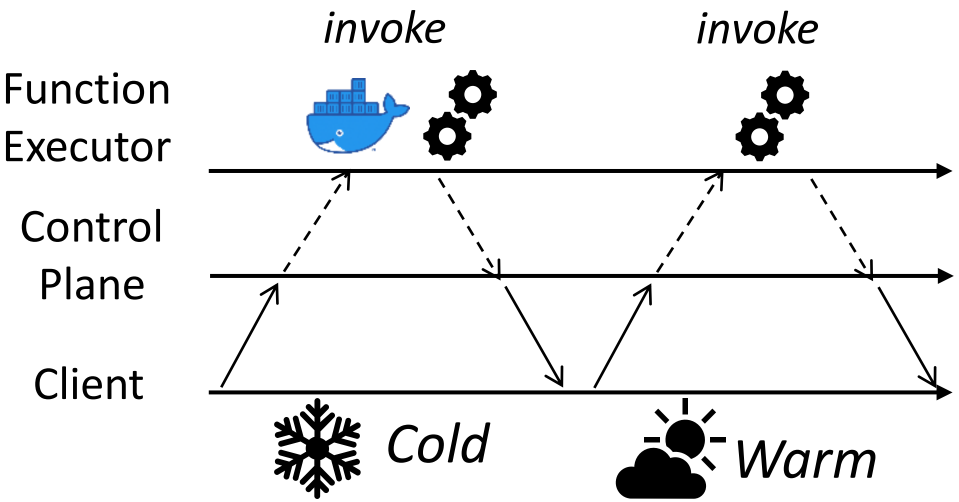


Invocations in FaaS and rFaaS

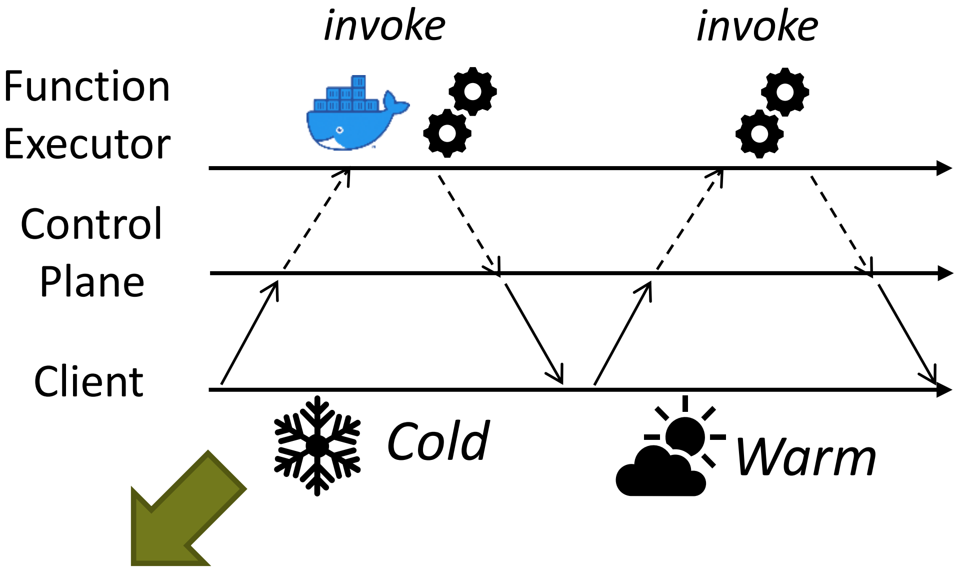


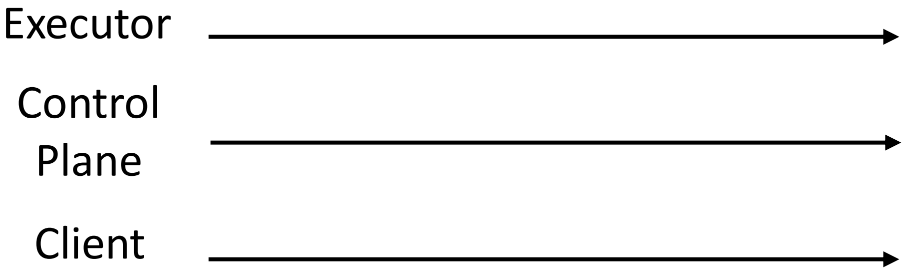
“rFaaS: Enabling High Performance Serverless with RDMA and Leases”, IPDPS’23

Invocations in FaaS and rFaaS



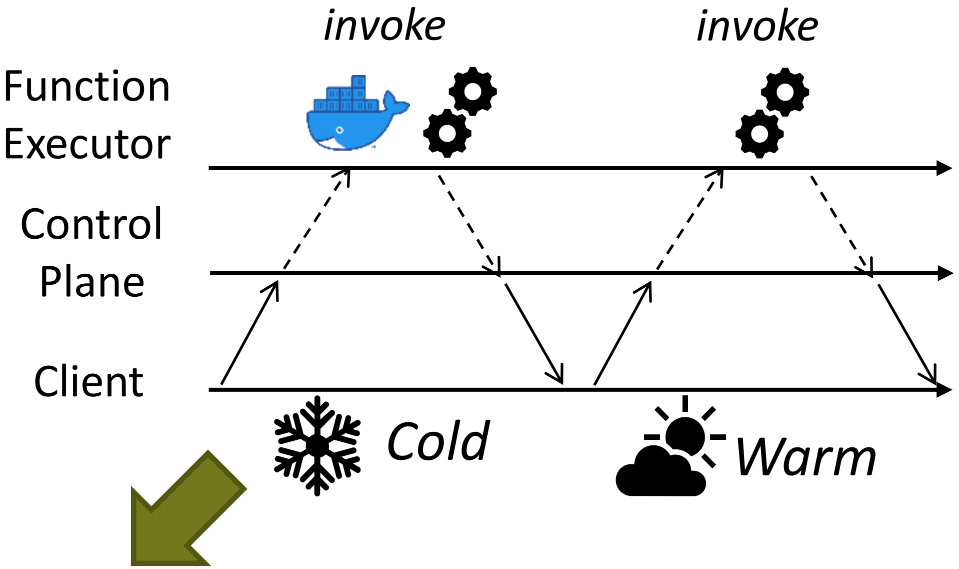
Invocations in FaaS and rFaaS

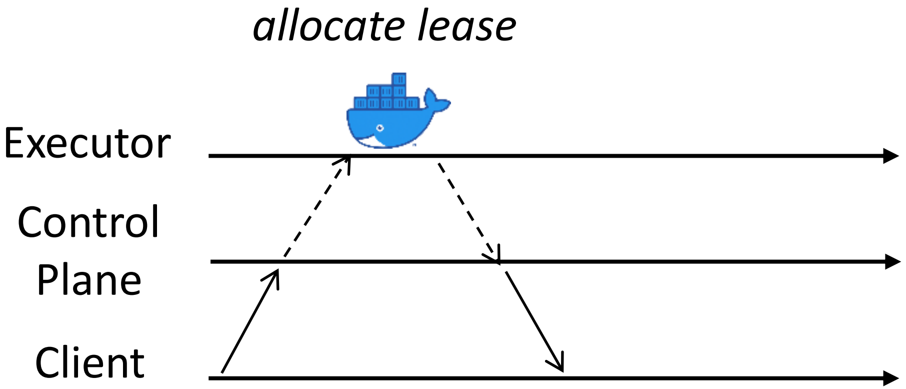




"rFaaS: Enabling High Performance Serverless with RDMA and Leases", IPDPS'23

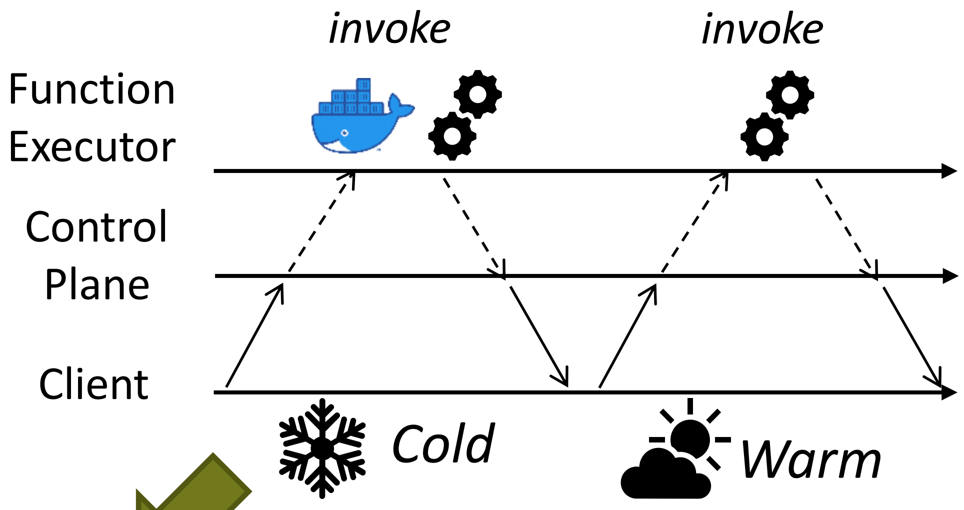
Invocations in FaaS and rFaaS

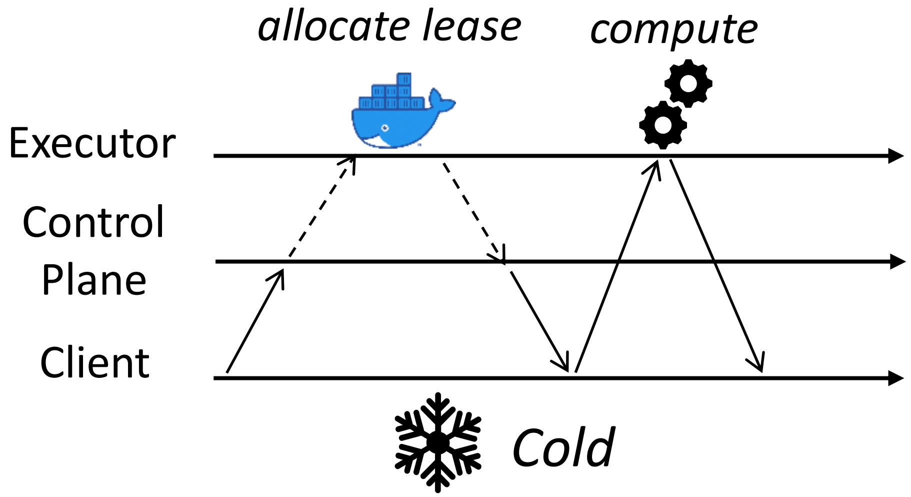




“rFaaS: Enabling High Performance Serverless with RDMA and Leases”, IPDPS’23

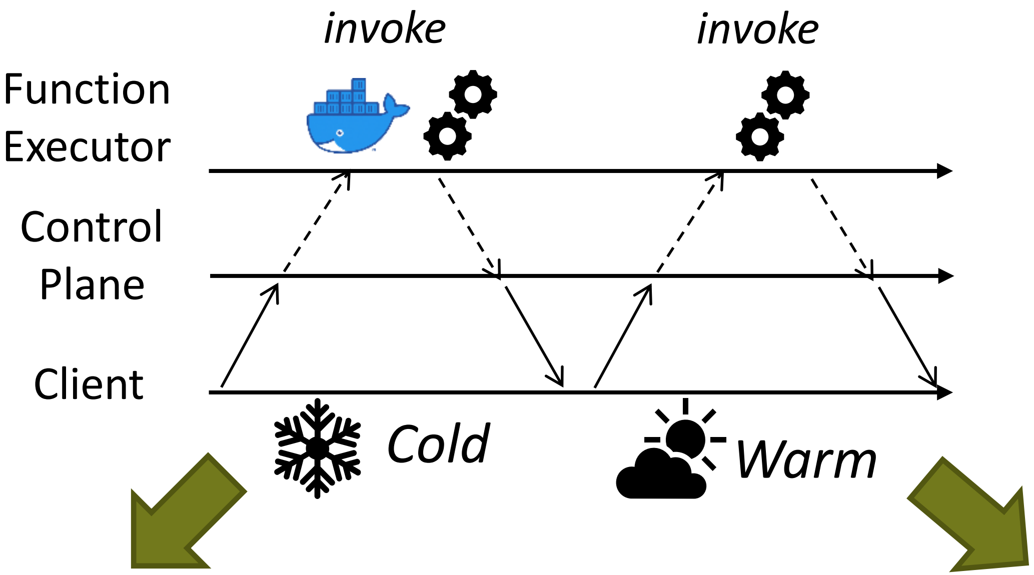
Invocations in FaaS and rFaaS

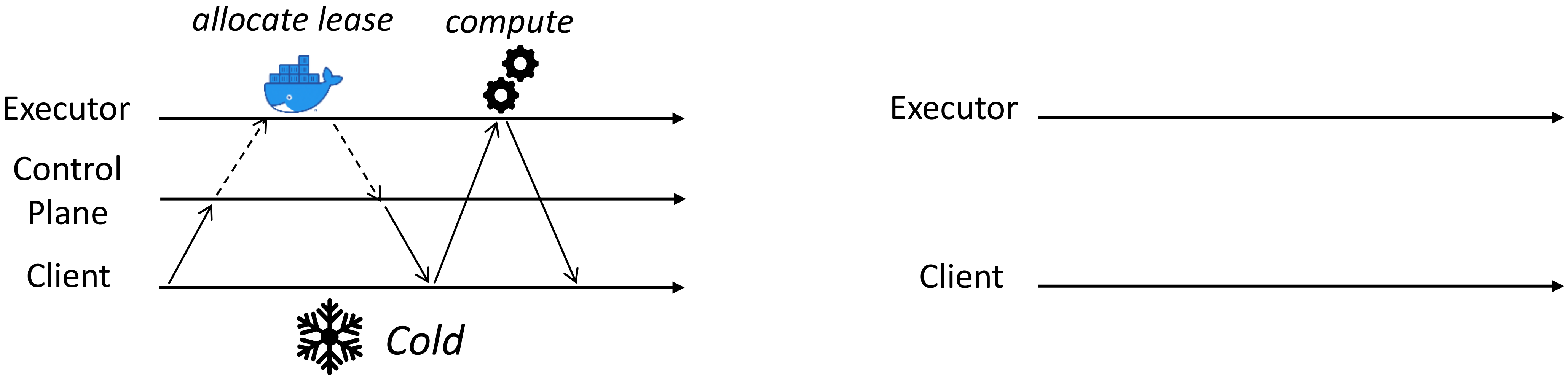




"rFaaS: Enabling High Performance Serverless with RDMA and Leases", IPDPS'23

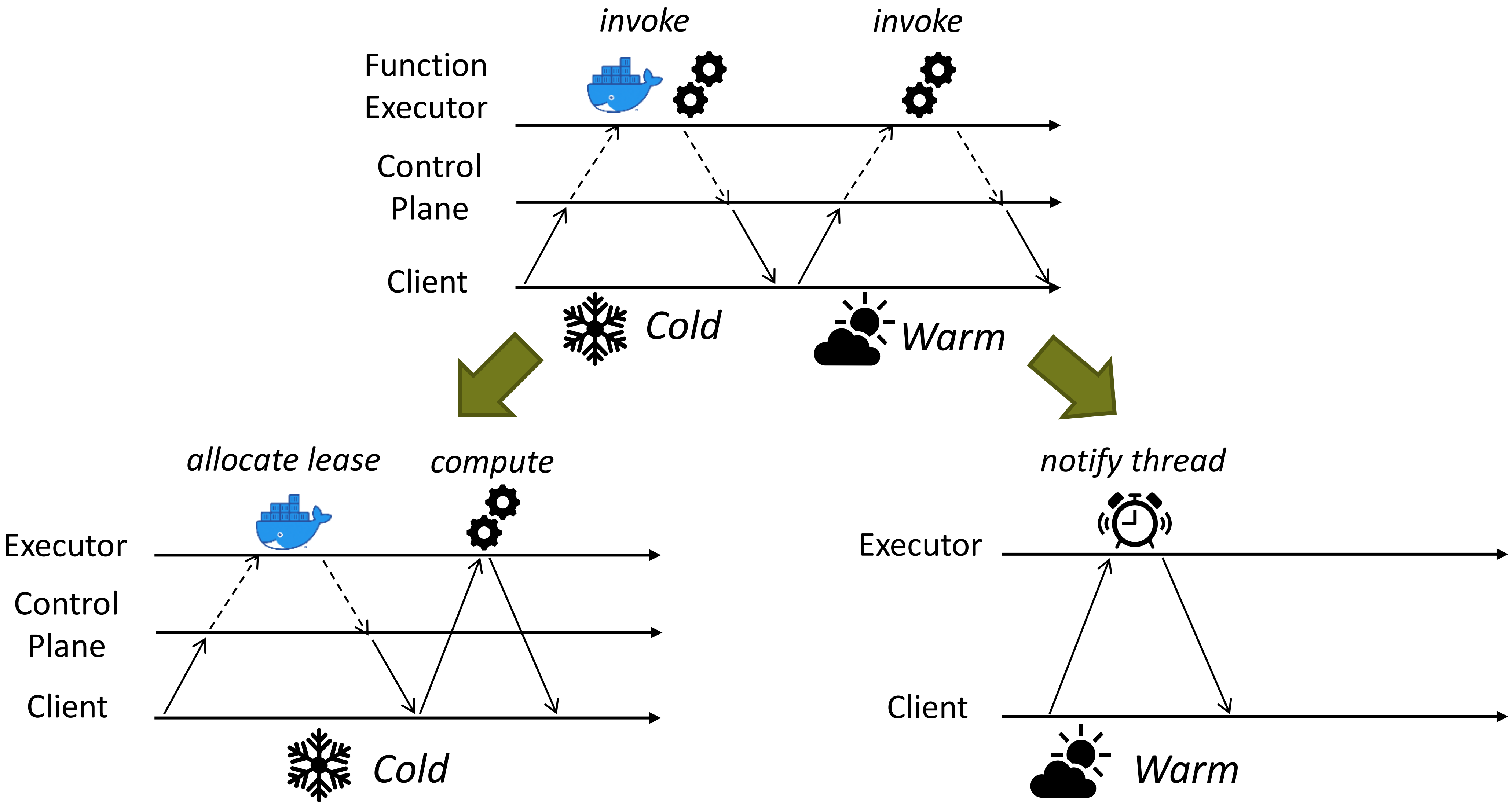
Invocations in FaaS and rFaaS





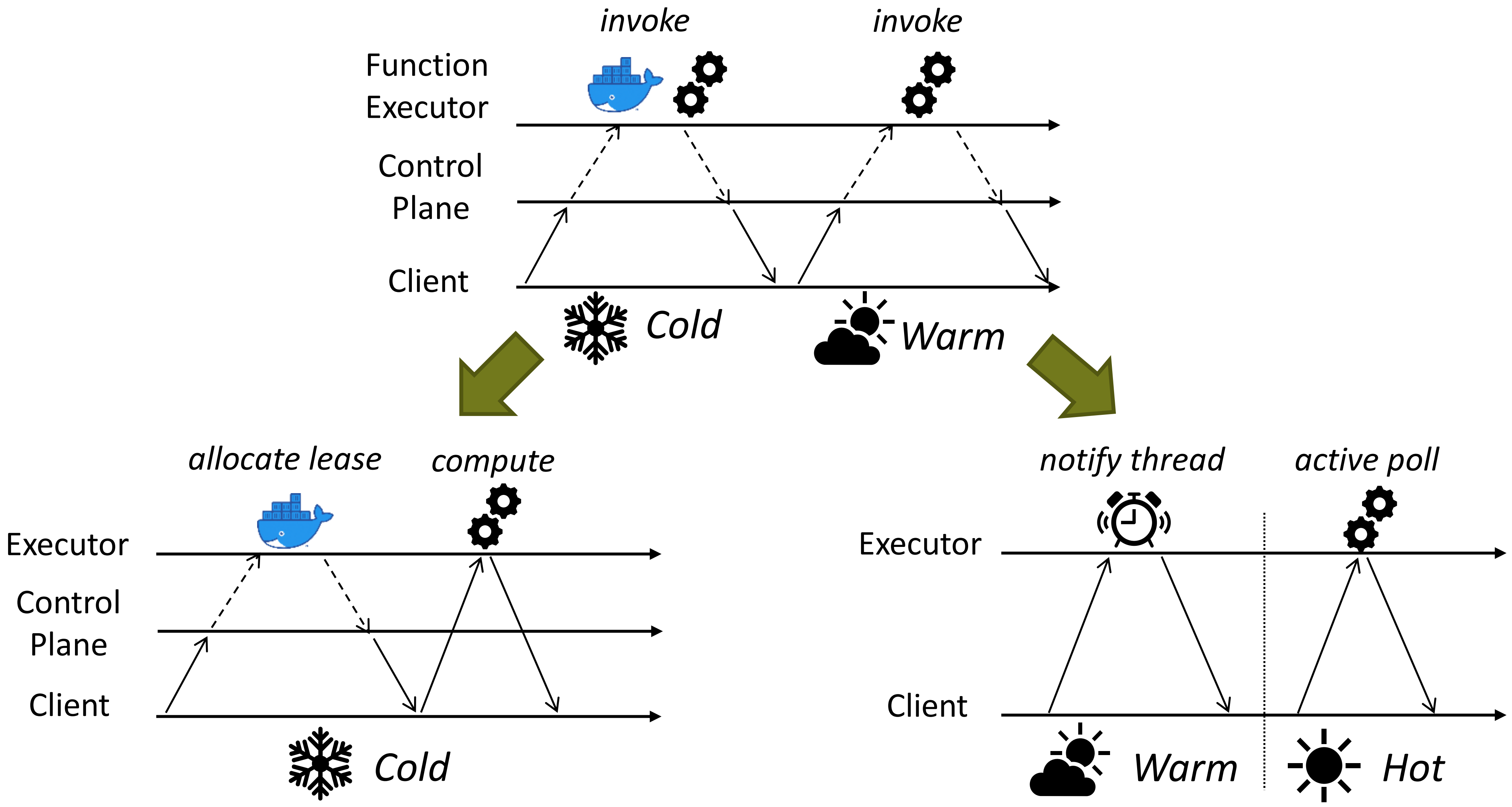
"rFaaS: Enabling High Performance Serverless with RDMA and Leases", IPDPS'23

Invocations in FaaS and rFaaS



“rFaaS: Enabling High Performance Serverless with RDMA and Leases”, IPDPS’23

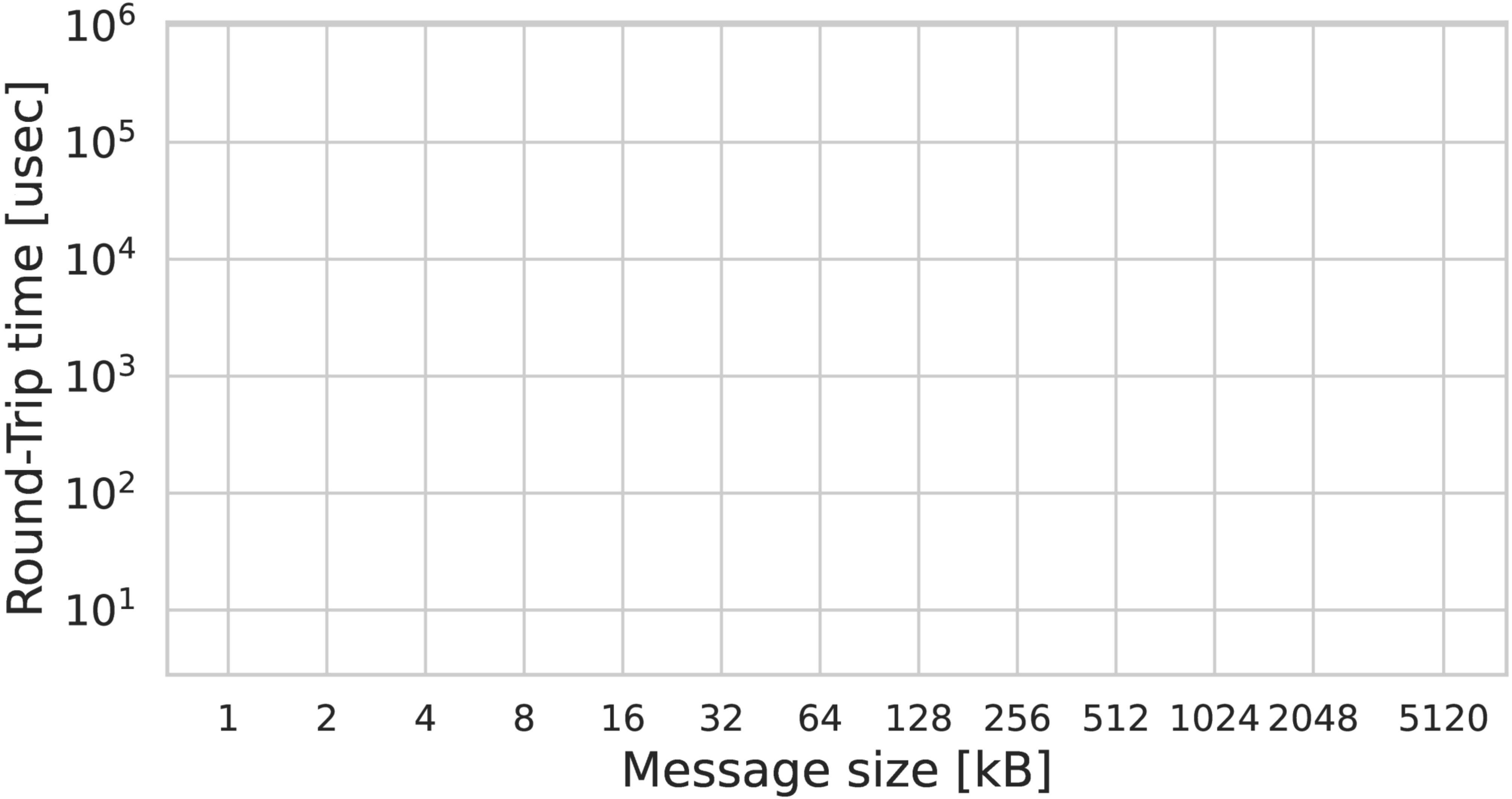
Invocations in FaaS and rFaaS



"rFaaS: Enabling High Performance Serverless with RDMA and Leases", IPDPS'23

How fast are invocations in FaaS?

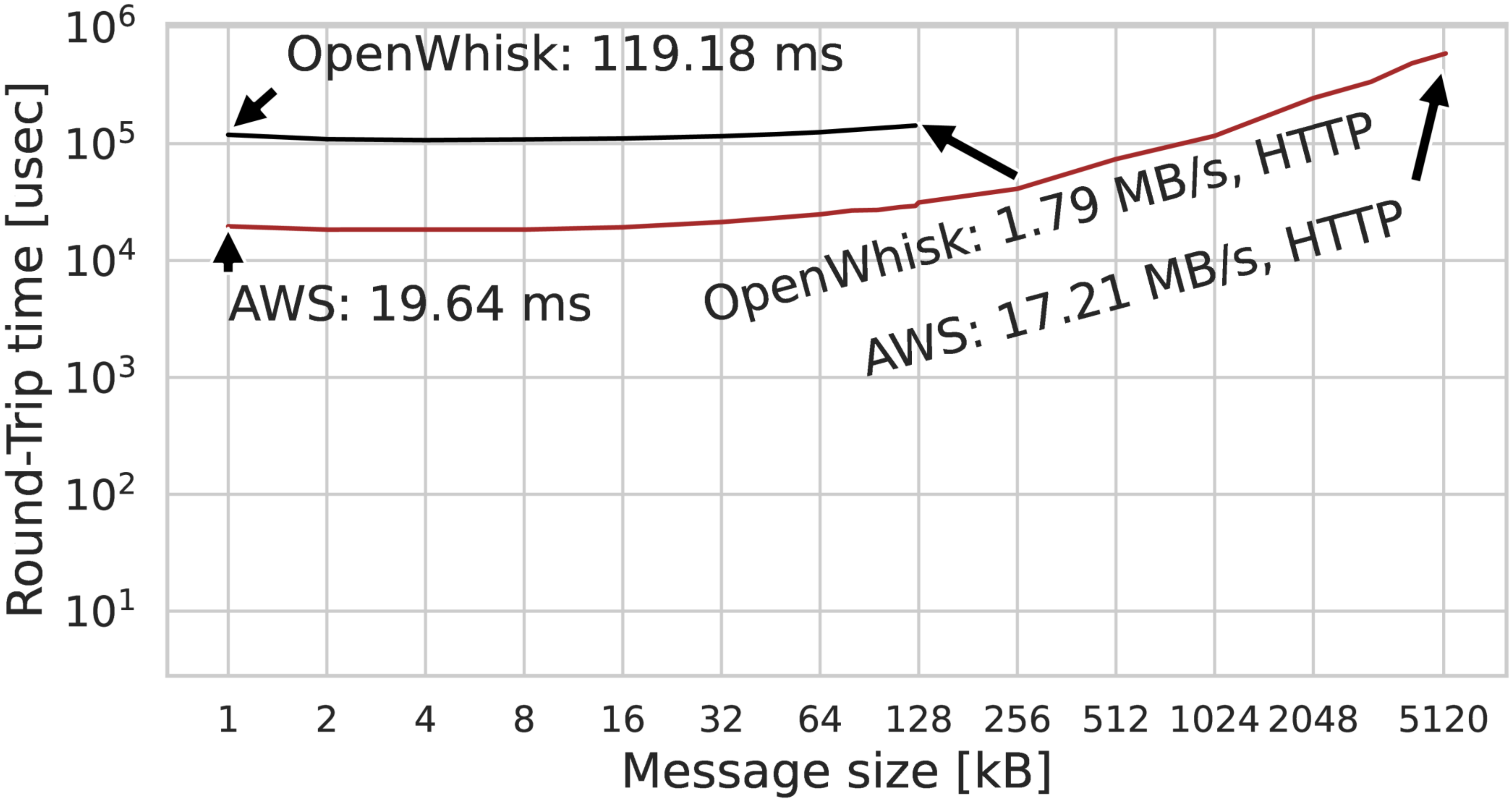
36 CPU cores, 377 GB memory.
100 Gbps Ethernet with RoCEv2 support.



"rFaaS: Enabling High Performance Serverless with RDMA and Leases", IPDPS'23

How fast are invocations in FaaS?

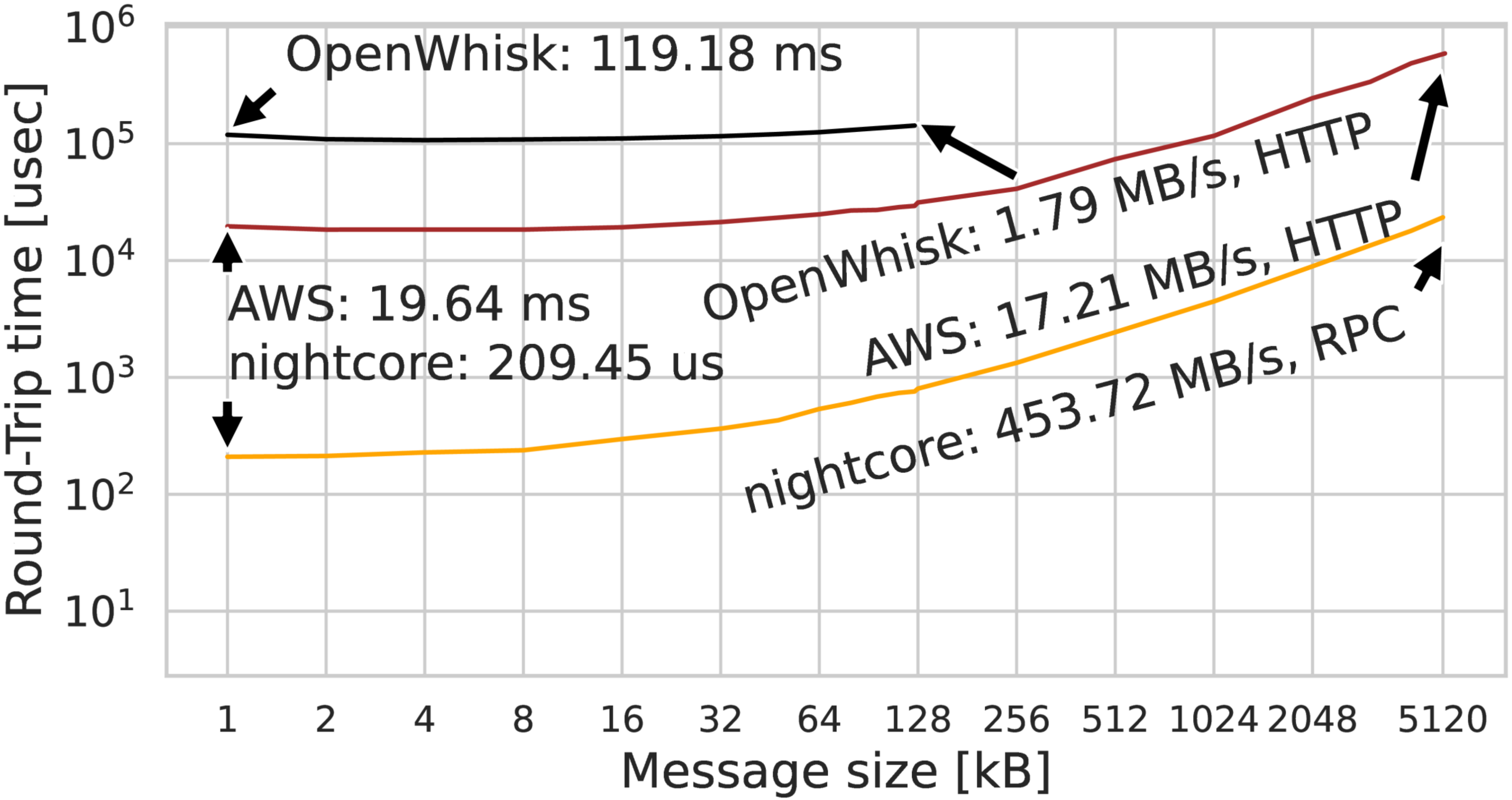
36 CPU cores, 377 GB memory.
100 Gbps Ethernet with RoCEv2 support.



"rFaaS: Enabling High Performance Serverless with RDMA and Leases", IPDPS'23

How fast are invocations in FaaS?

36 CPU cores, 377 GB memory.
100 Gbps Ethernet with RoCEv2 support.



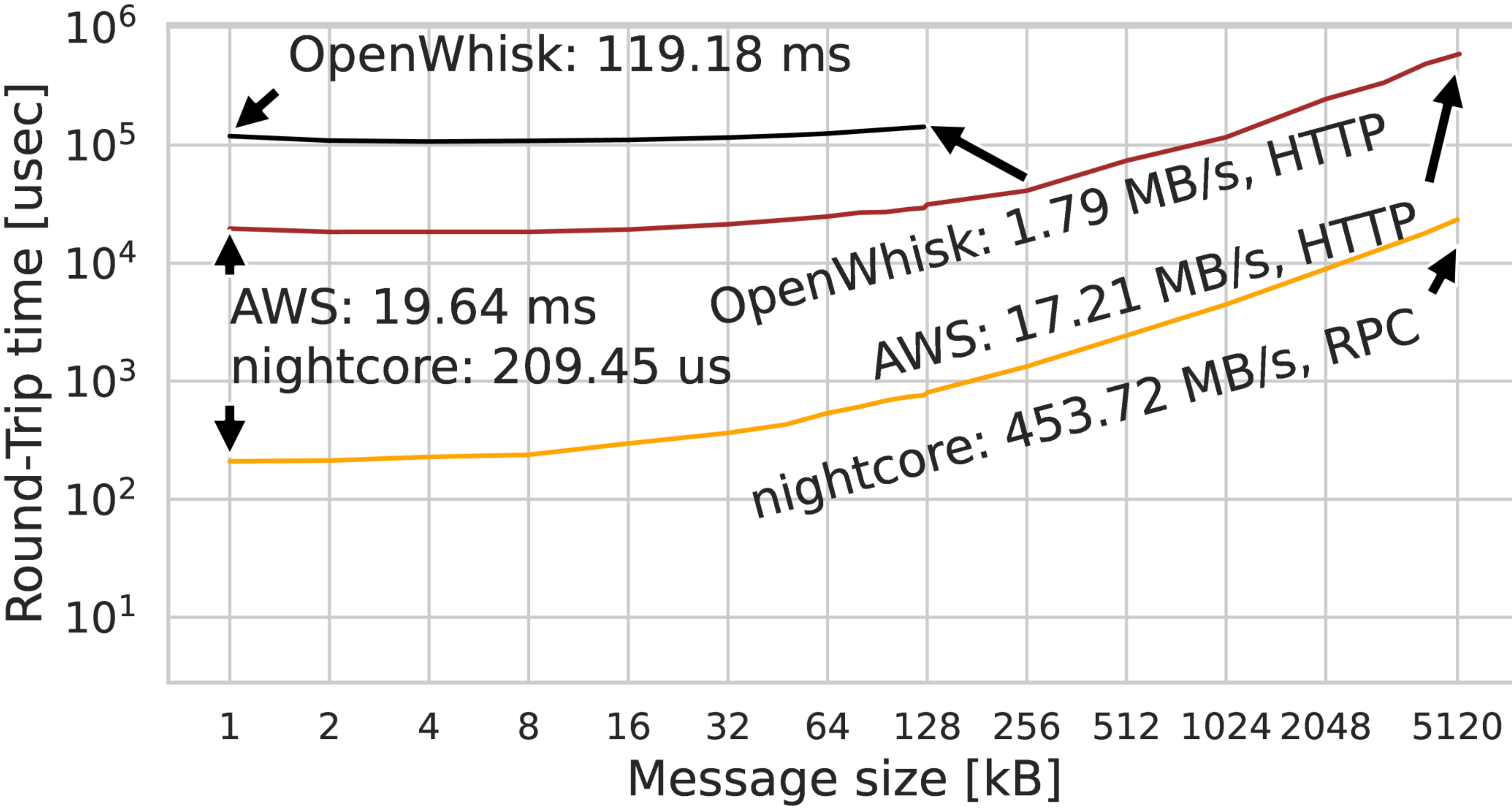
"rFaaS: Enabling High Performance Serverless with RDMA and Leases", IPDPS'23

How fast are invocations in FaaS?

36 CPU cores, 377 GB memory.
100 Gbps Ethernet with RoCEv2 support.

Reduced invocation critical path

Zero-copy RDMA networking



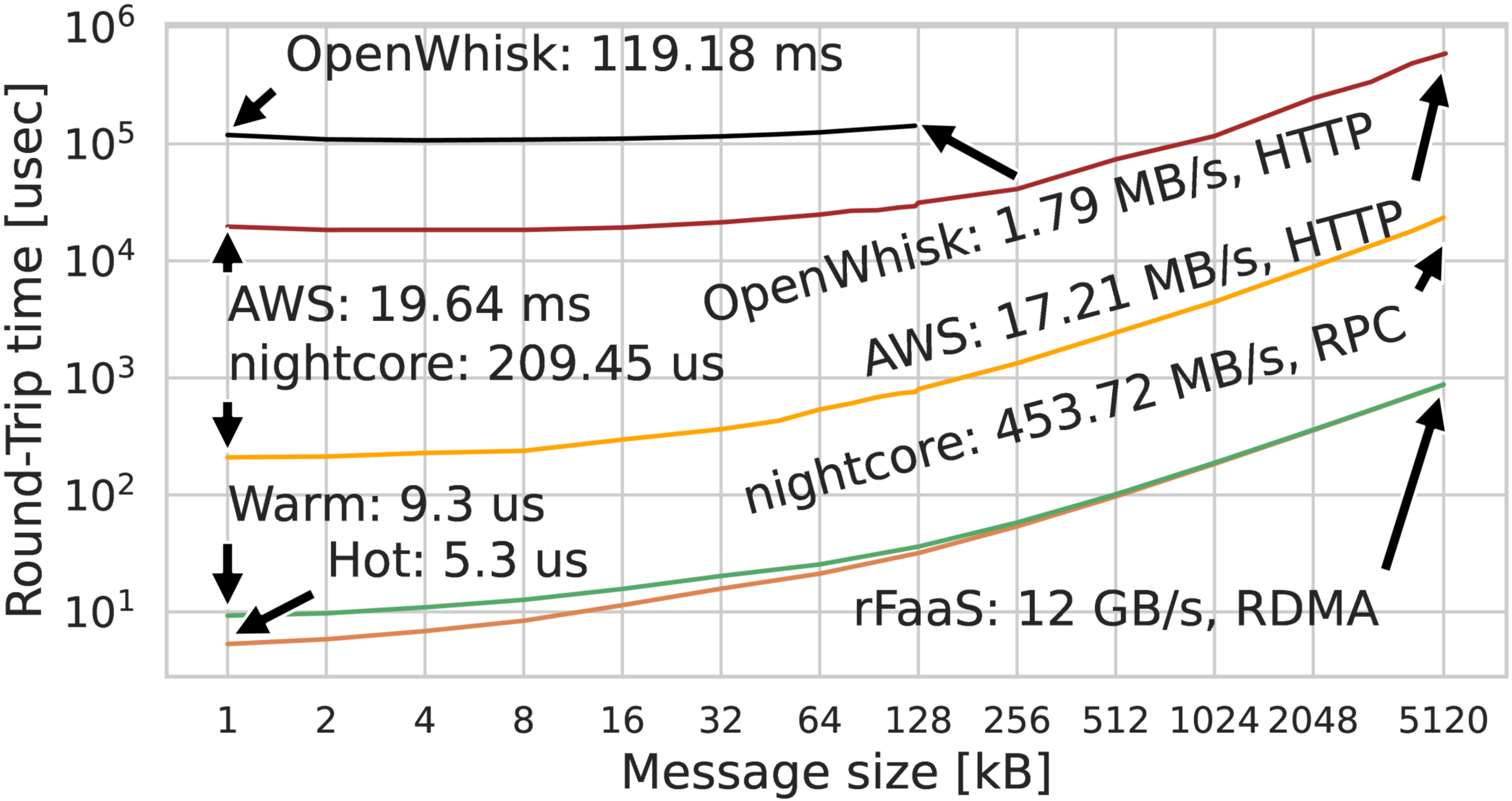
"rFaaS: Enabling High Performance Serverless with RDMA and Leases", IPDPS'23

How fast are invocations in FaaS?

36 CPU cores, 377 GB memory.
100 Gbps Ethernet with RoCEv2 support.

Reduced invocation critical path

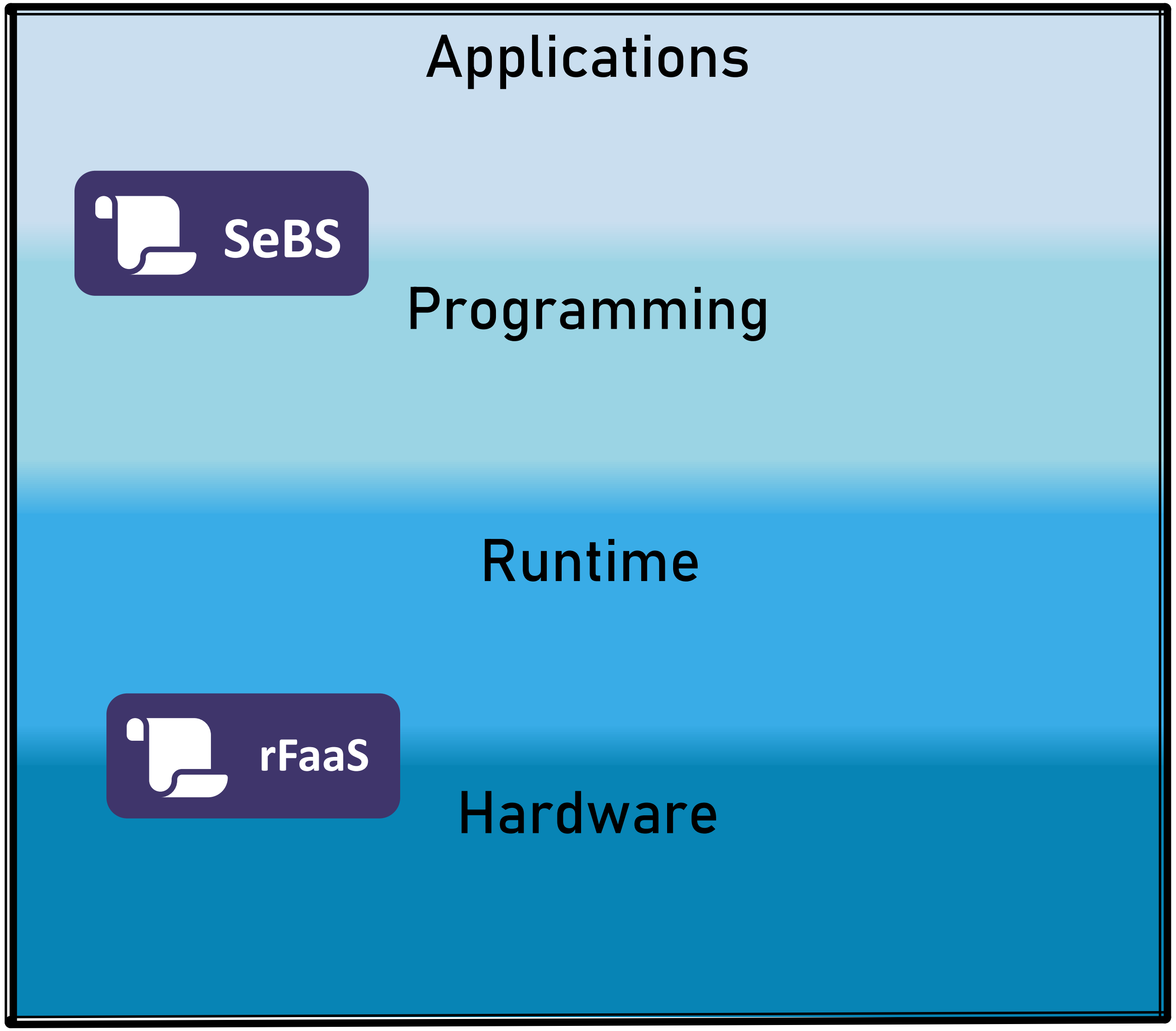
Zero-copy RDMA networking



“rFaaS: Enabling High Performance Serverless with RDMA and Leases”, IPDPS’23

Serverless for High-Performance Applications

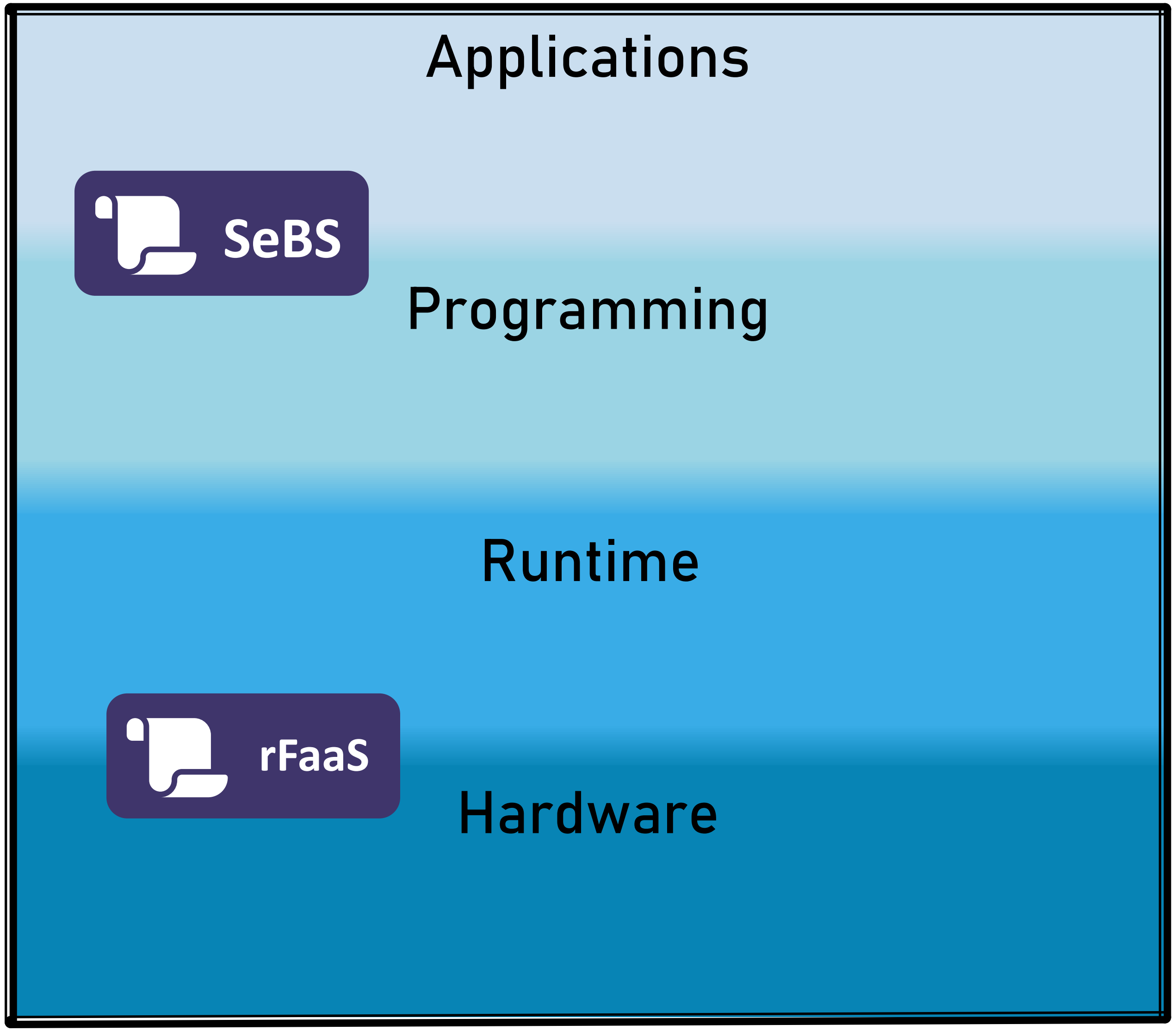
Functions are expensive to invoke.



Serverless for High-Performance Applications

Functions are expensive to invoke.

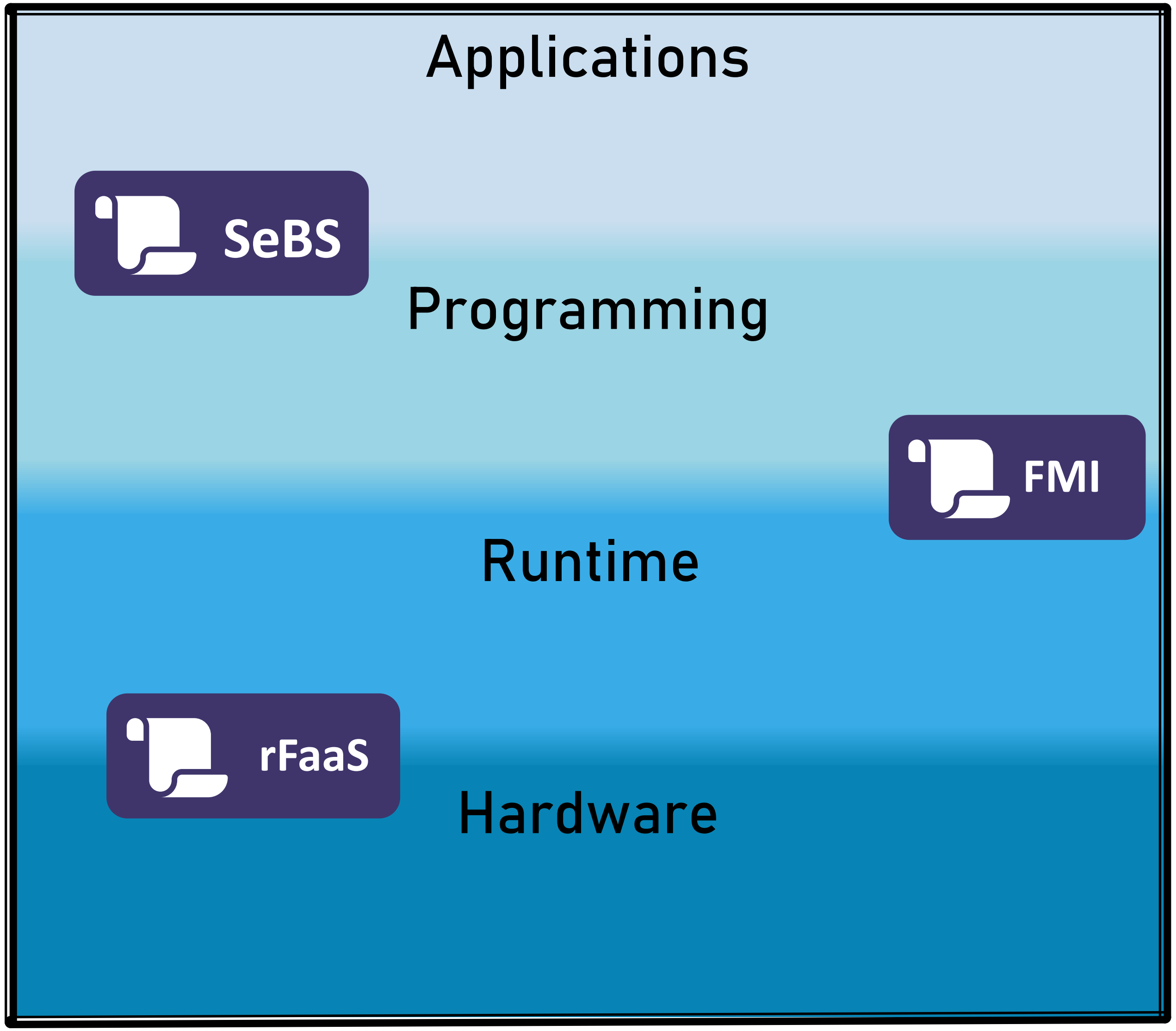
Communication is slow and restricted.



Serverless for High-Performance Applications

Functions are expensive to invoke.

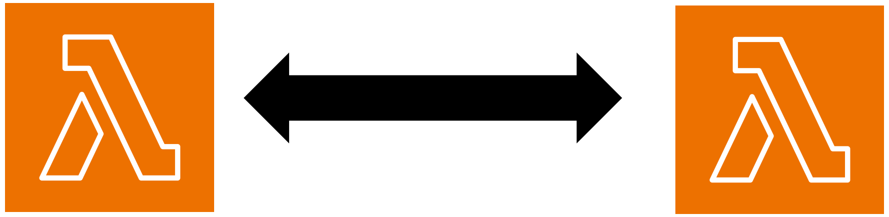
Communication is slow and restricted.



Communication in serverless

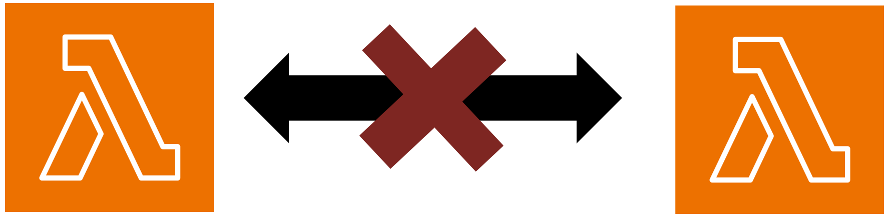
“FMI: Fast and Cheap Message Passing for Serverless Functions”, ICS’23

Communication in serverless



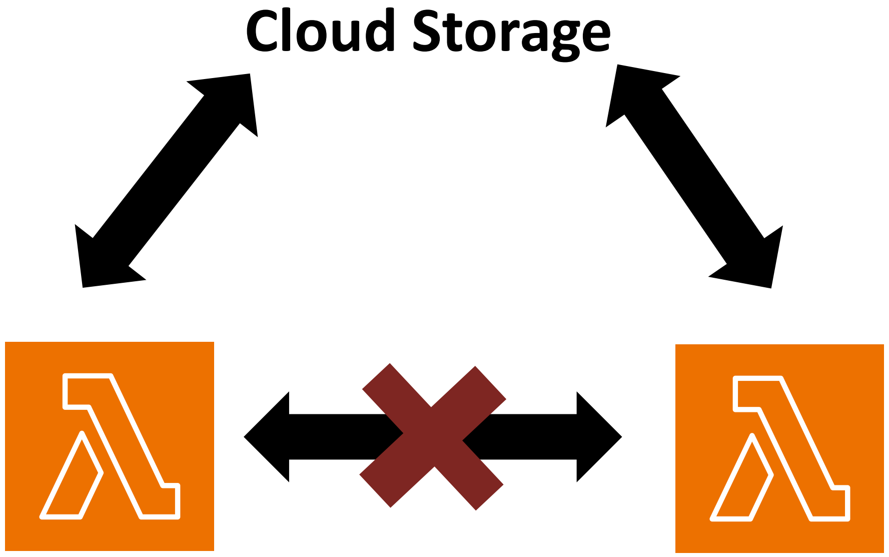
“FMI: Fast and Cheap Message Passing for Serverless Functions”, ICS’23

Communication in serverless



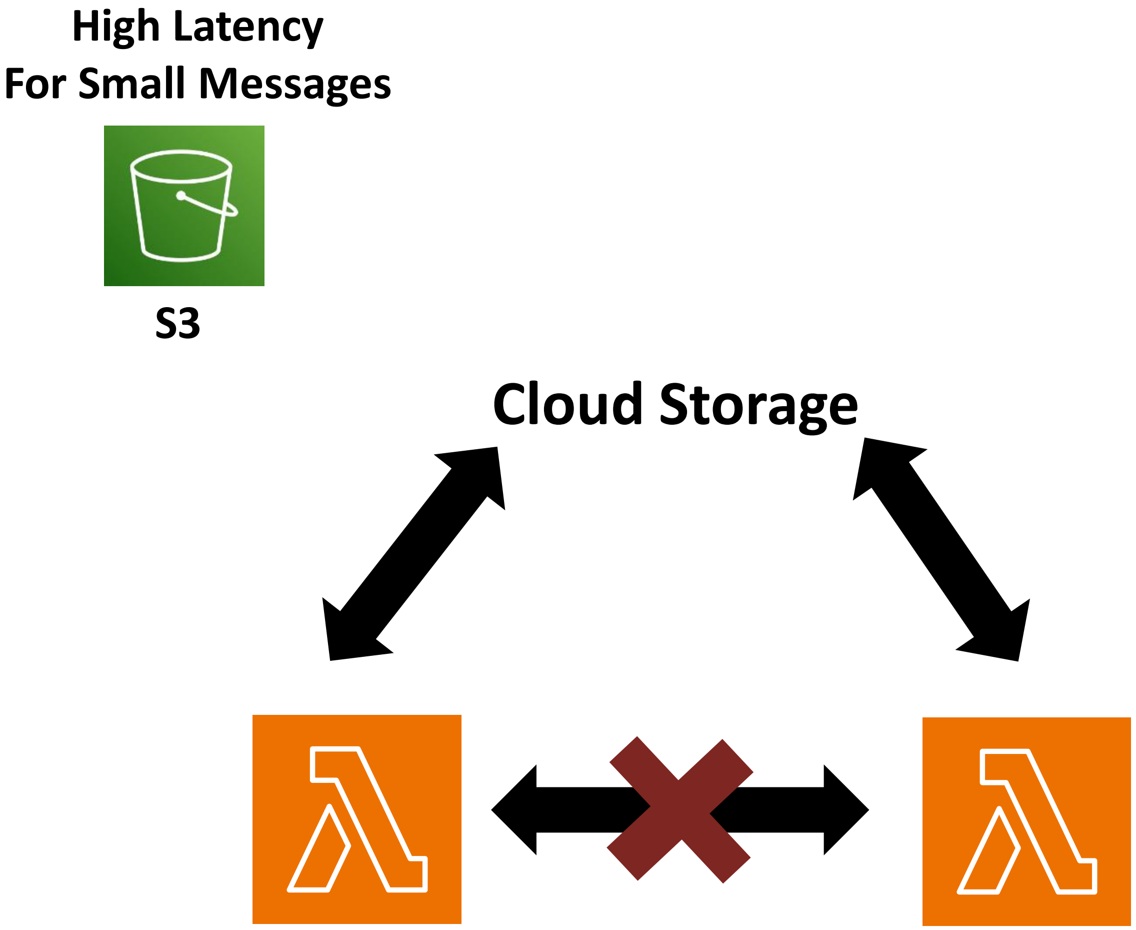
“FMI: Fast and Cheap Message Passing for Serverless Functions”, ICS’23

Communication in serverless



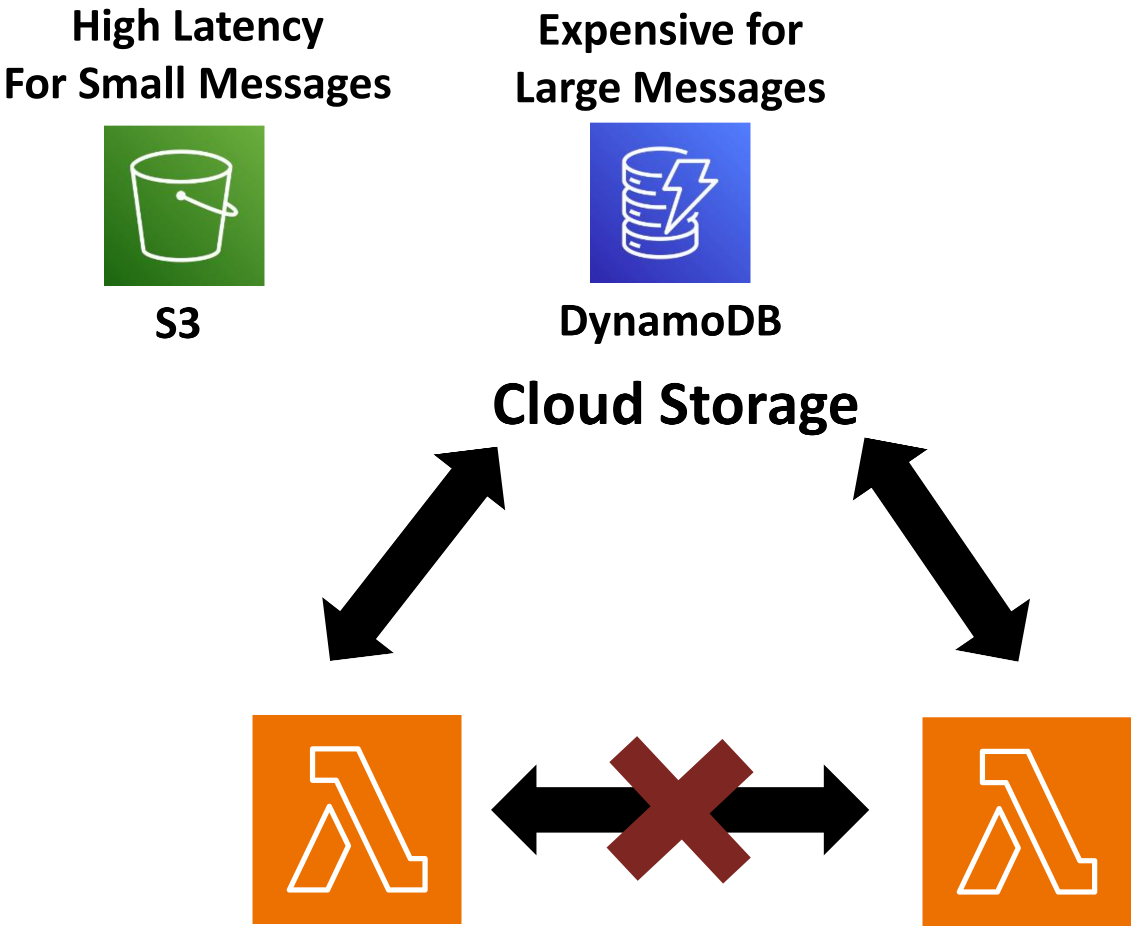
“FMI: Fast and Cheap Message Passing for Serverless Functions”, ICS’23

Communication in serverless



"FMI: Fast and Cheap Message Passing for Serverless Functions", ICS'23

Communication in serverless



High Latency
For Small Messages

S3

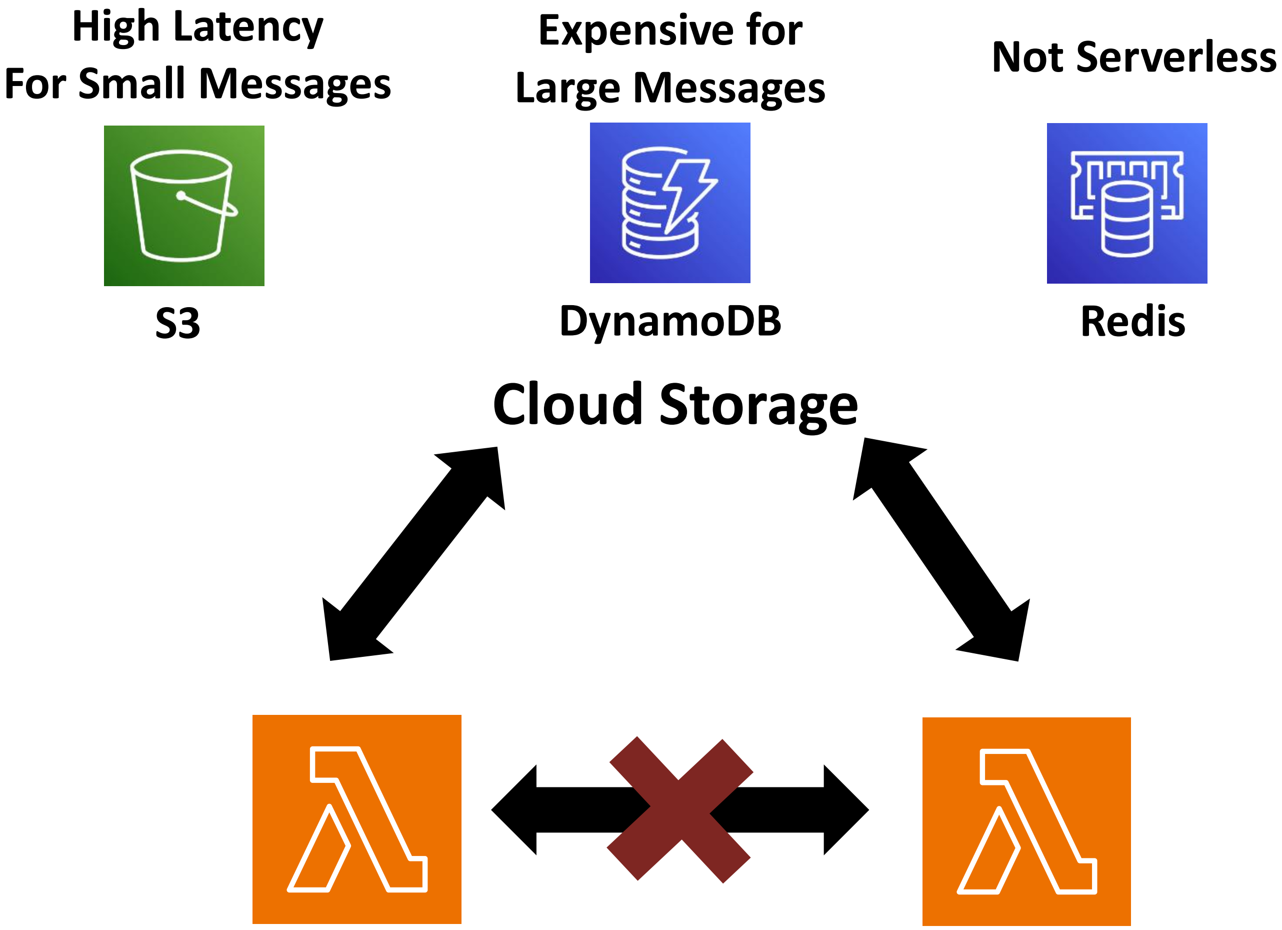
Expensive for
Large Messages

DynamoDB

Cloud Storage

"FMI: Fast and Cheap Message Passing for Serverless Functions", ICS'23

Communication in serverless



High Latency For Small Messages

Expensive for Large Messages

Not Serverless

S3

DynamoDB

Redis

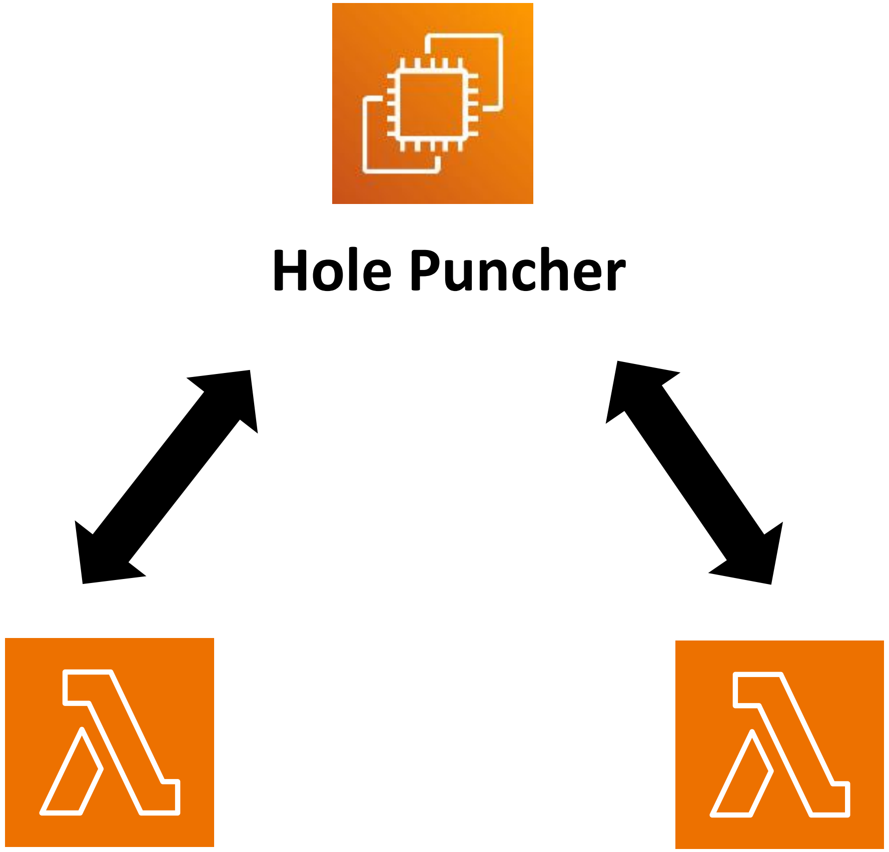
Cloud Storage

“FMI: Fast and Cheap Message Passing for Serverless Functions”, ICS’23

Communication in serverless

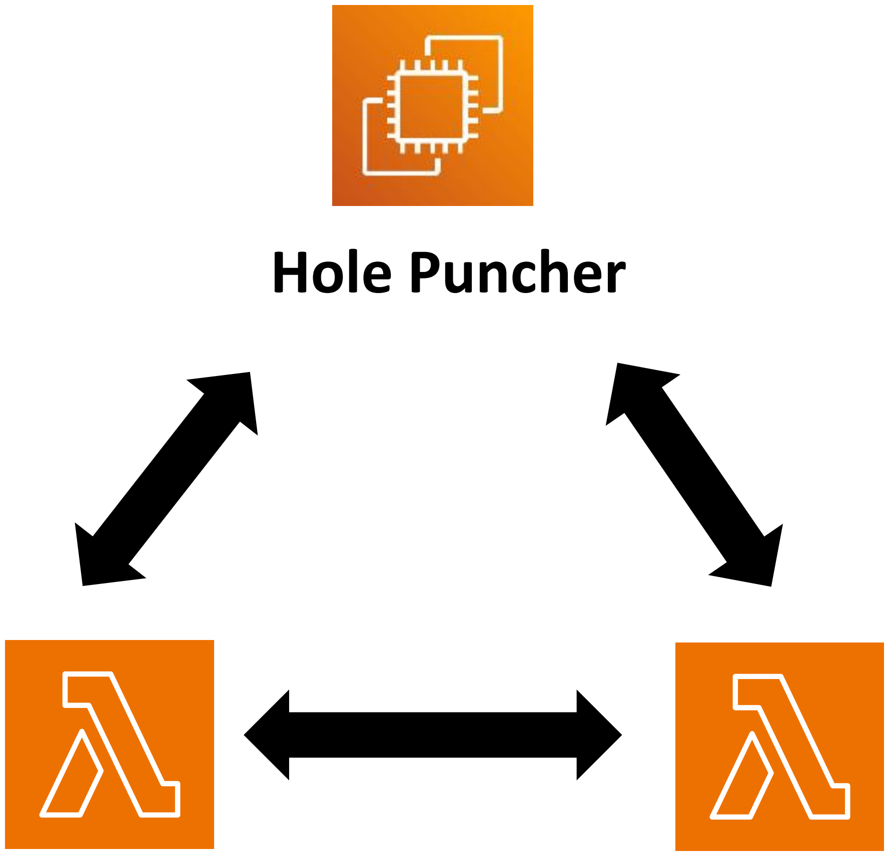
“FMI: Fast and Cheap Message Passing for Serverless Functions”, ICS’23

Communication in serverless



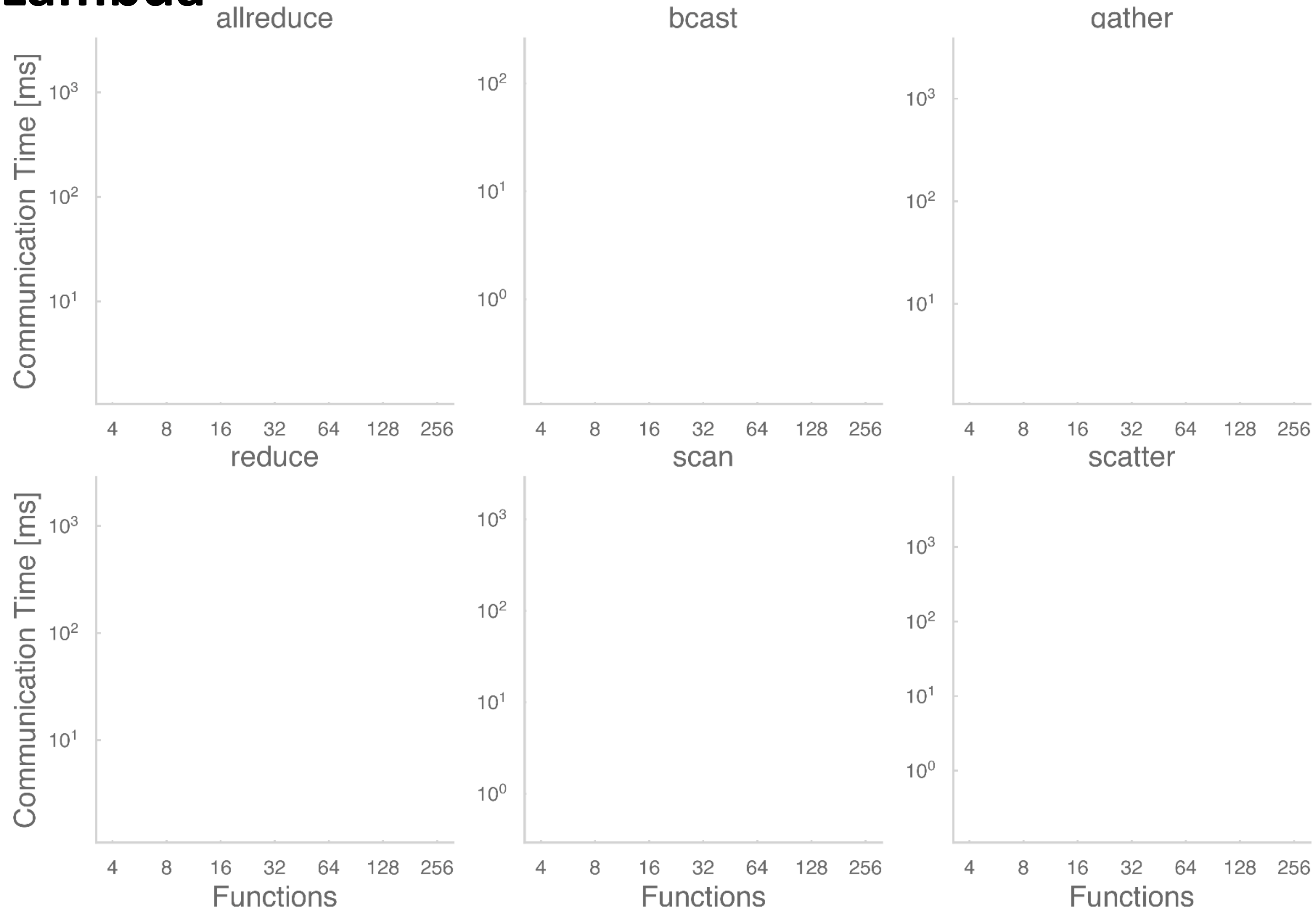
“FMI: Fast and Cheap Message Passing for Serverless Functions”, ICS’23

Communication in serverless



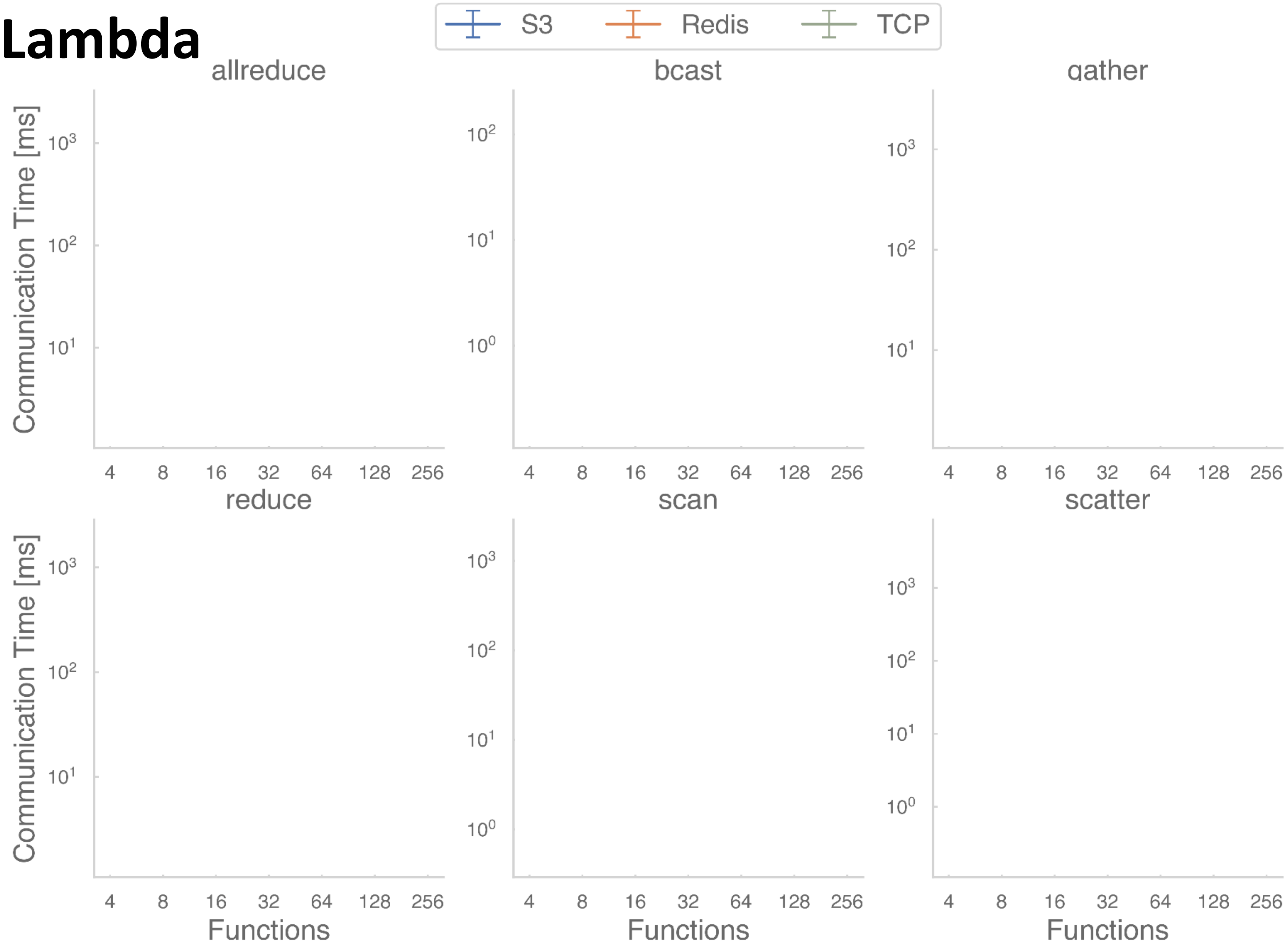
“FMI: Fast and Cheap Message Passing for Serverless Functions”, ICS’23

FMI on AWS Lambda



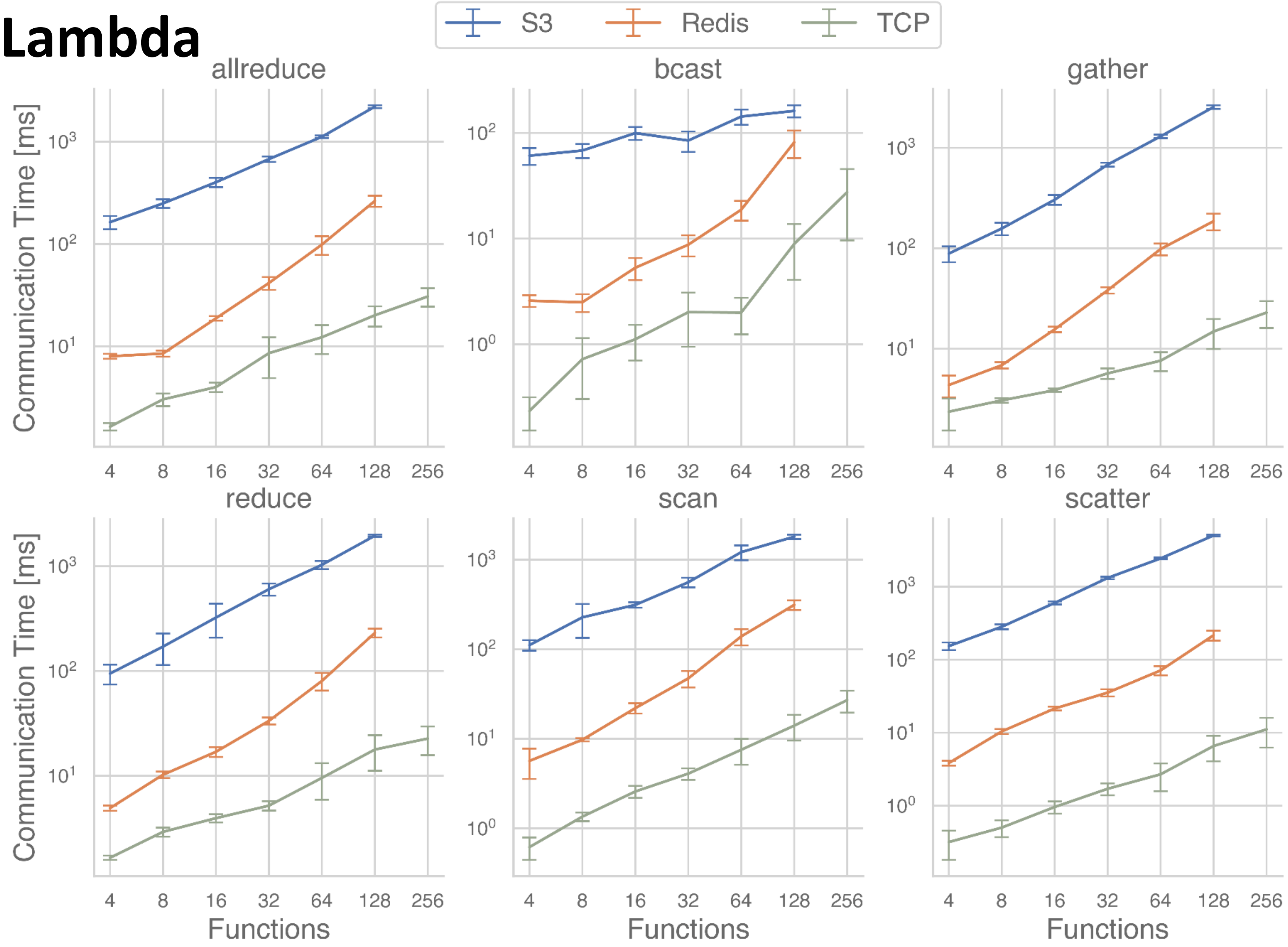
“FMI: Fast and Cheap Message Passing for Serverless Functions”, ICS’23

FMI on AWS Lambda



“FMI: Fast and Cheap Message Passing for Serverless Functions”, ICS’23

FMI on AWS Lambda

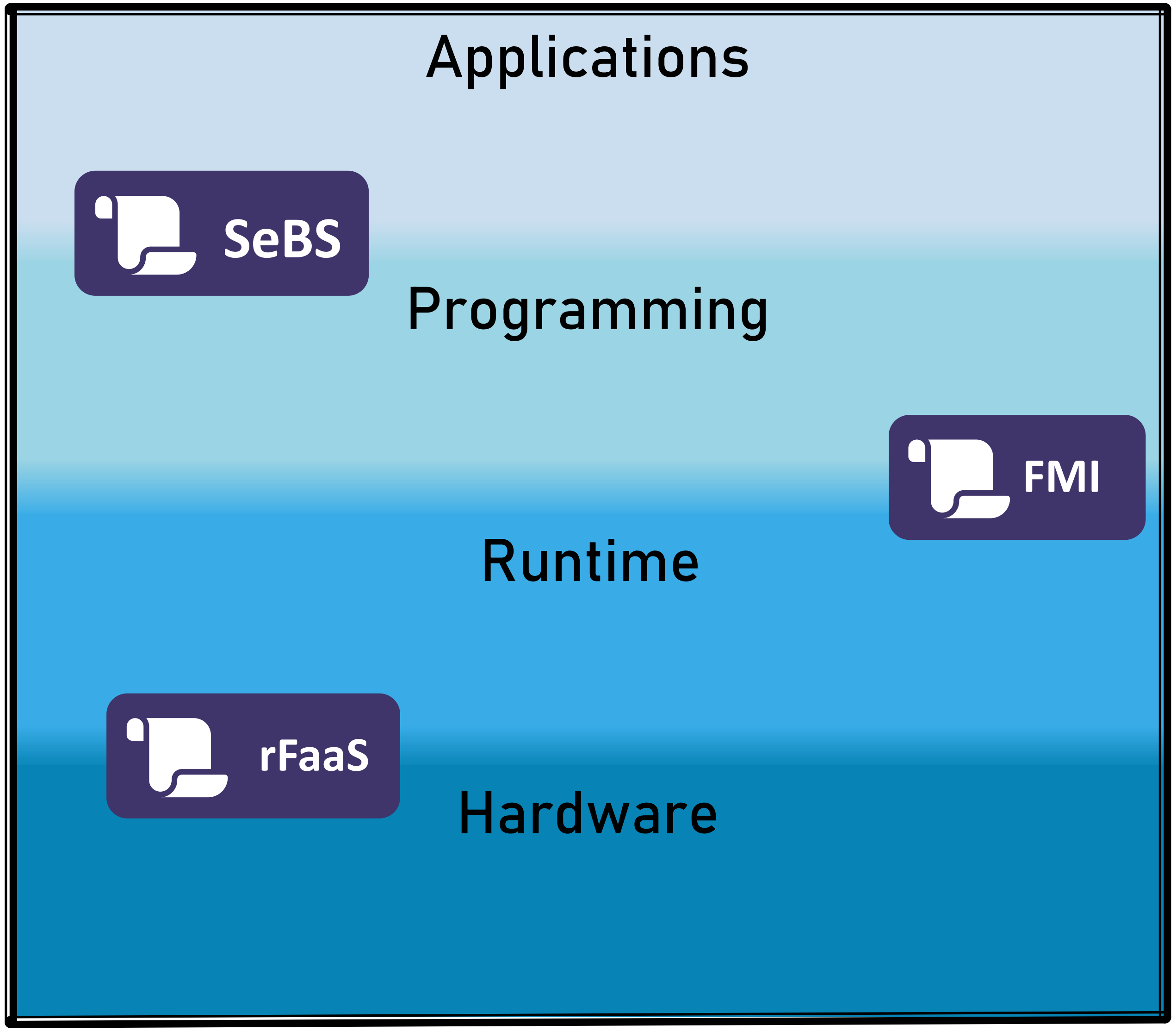


“FMI: Fast and Cheap Message Passing for Serverless Functions”, ICS’23

Serverless for High-Performance Applications

Functions are expensive to invoke.

Communication is slow and restricted.

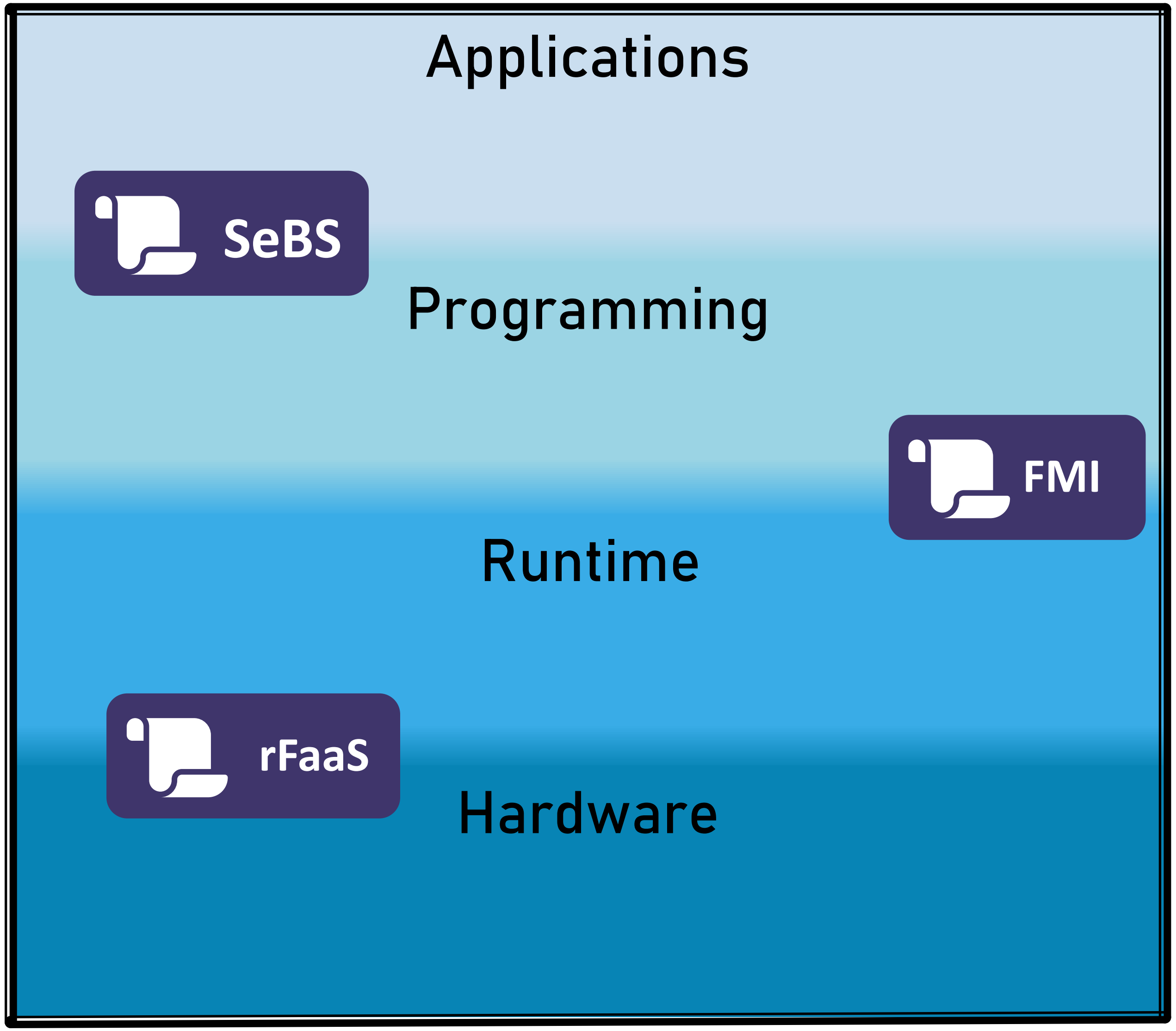


Serverless for High-Performance Applications

Functions are expensive to invoke.

Communication is slow and restricted.

Serverless is hard to implement in practice.

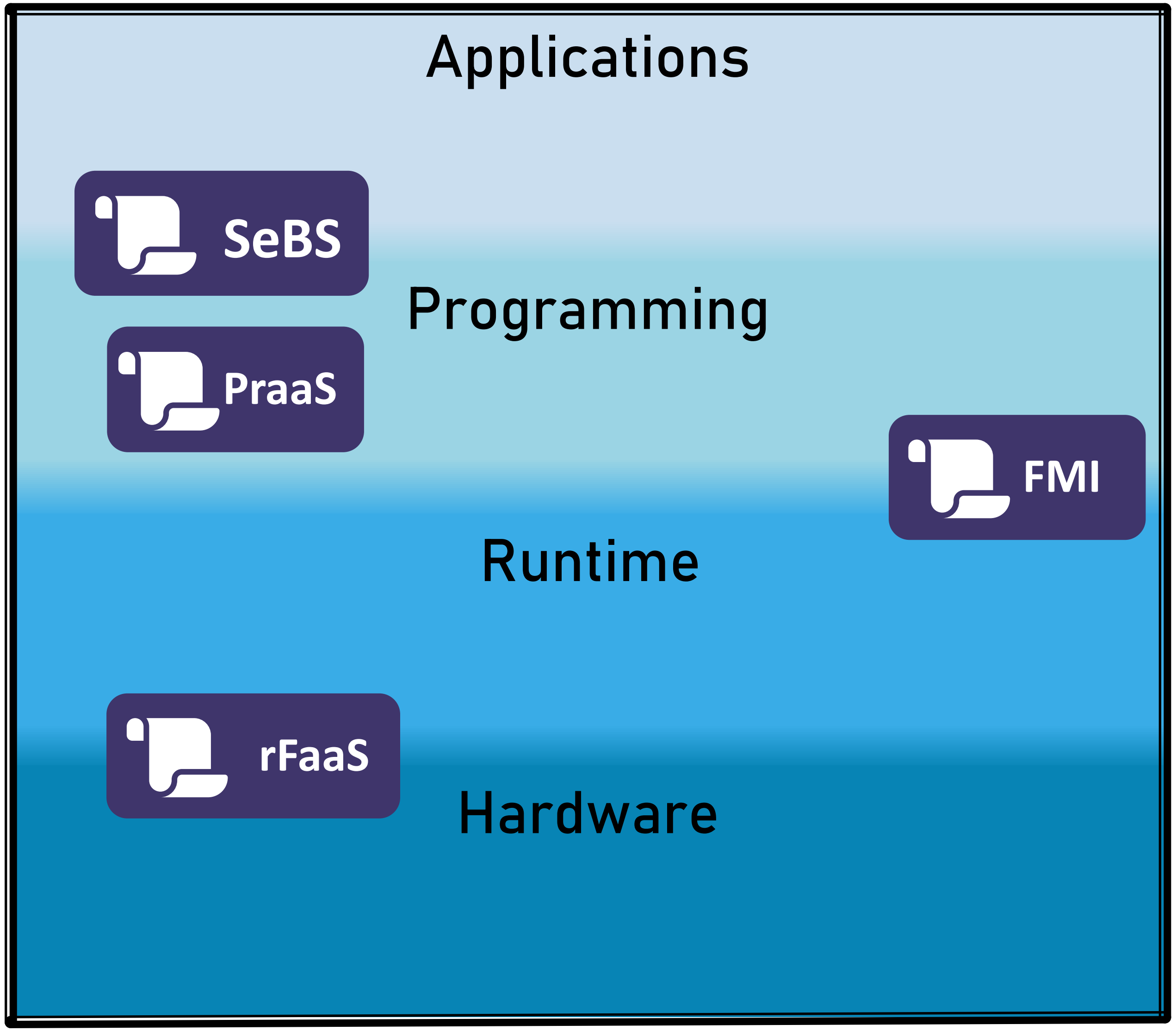


Serverless for High-Performance Applications

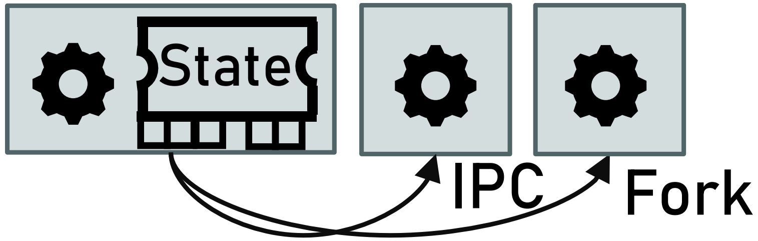
Functions are expensive to invoke.

Communication is slow and restricted.

Serverless is hard to implement in practice.



Serverless Process

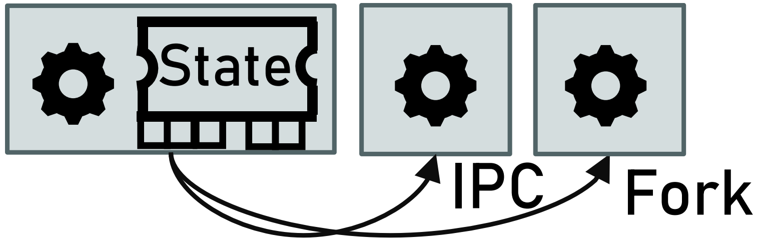


OS Process

Nano- and micro-second latency of OS primitives.

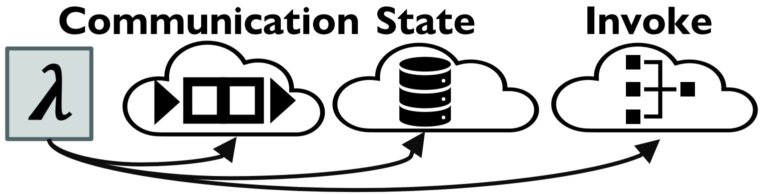
“Process-as-a-Service: Elastic and Stateful Serverless with Cloud Processes”, paper preprint.

Serverless Process



OS Process

Nano- and micro-second latency of OS primitives.

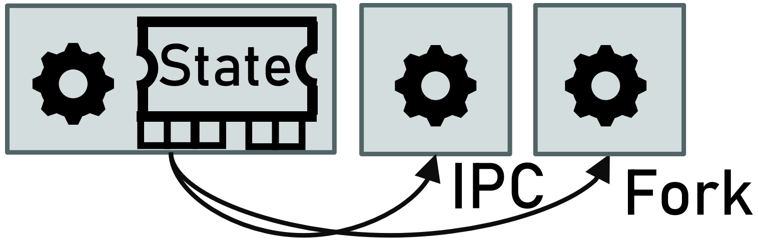


Serverless Function

Millisecond latency of cloud proxies.

"Process-as-a-Service: Elastic and Stateful Serverless with Cloud Processes", paper preprint.

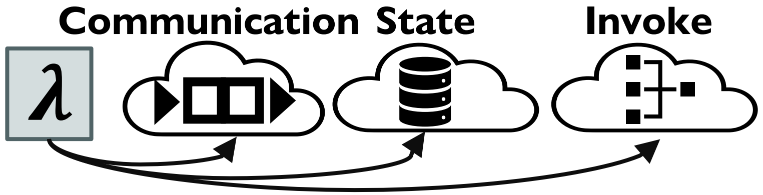
Serverless Process



OS Process

Nano- and micro-second latency of OS primitives.

+

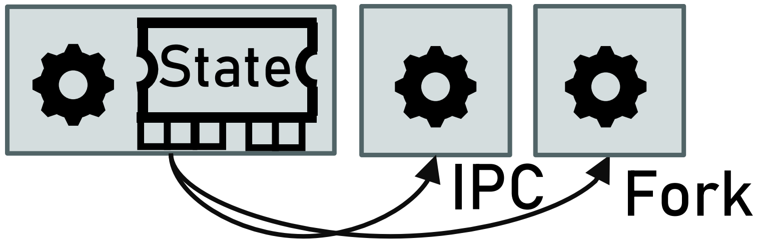


Serverless Function

Millisecond latency of cloud proxies.

“Process-as-a-Service: Elastic and Stateful Serverless with Cloud Processes”, paper preprint.

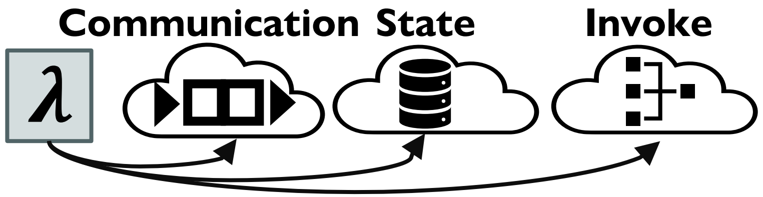
Serverless Process



OS Process

Nano- and micro-second latency of OS primitives.

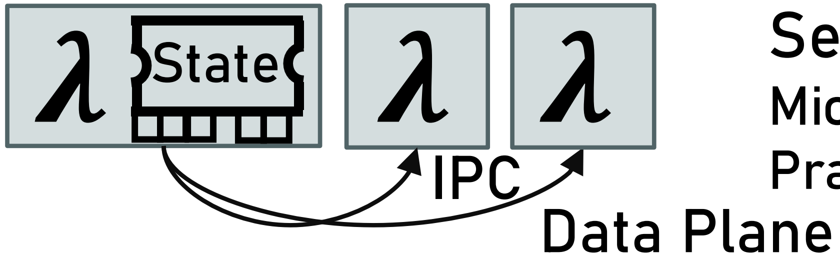
+



Serverless Function

Millisecond latency of cloud proxies.

=

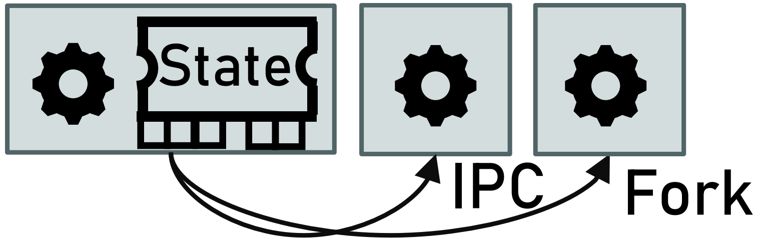


Serverless Process

Microsecond latency of PraaS backend.

“Process-as-a-Service: Elastic and Stateful Serverless with Cloud Processes”, paper preprint.

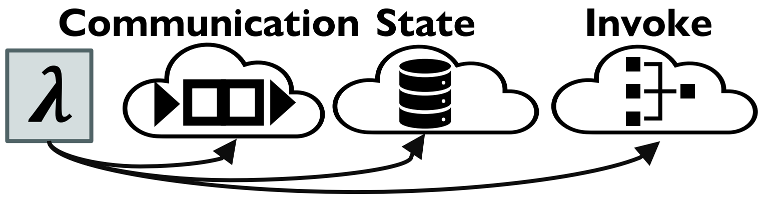
Serverless Process



OS Process

Nano- and micro-second latency of OS primitives.

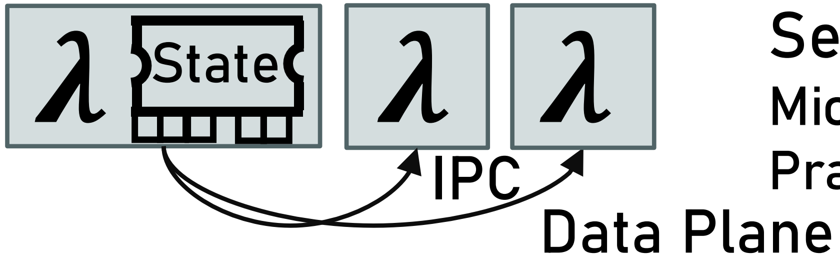
+



Serverless Function

Millisecond latency of cloud proxies.

=



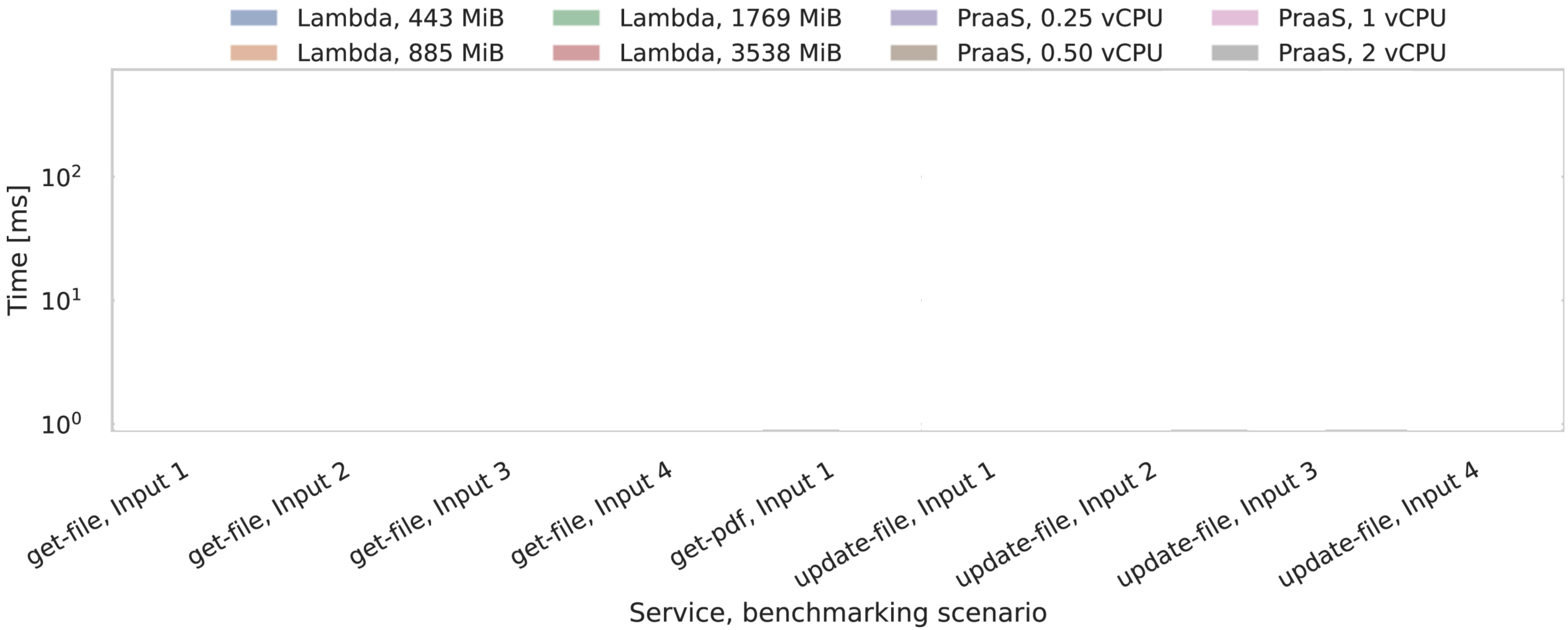
Serverless Process

Microsecond latency of PraaS backend.

Works on AWS Fargate, Knative, Kubernetes.

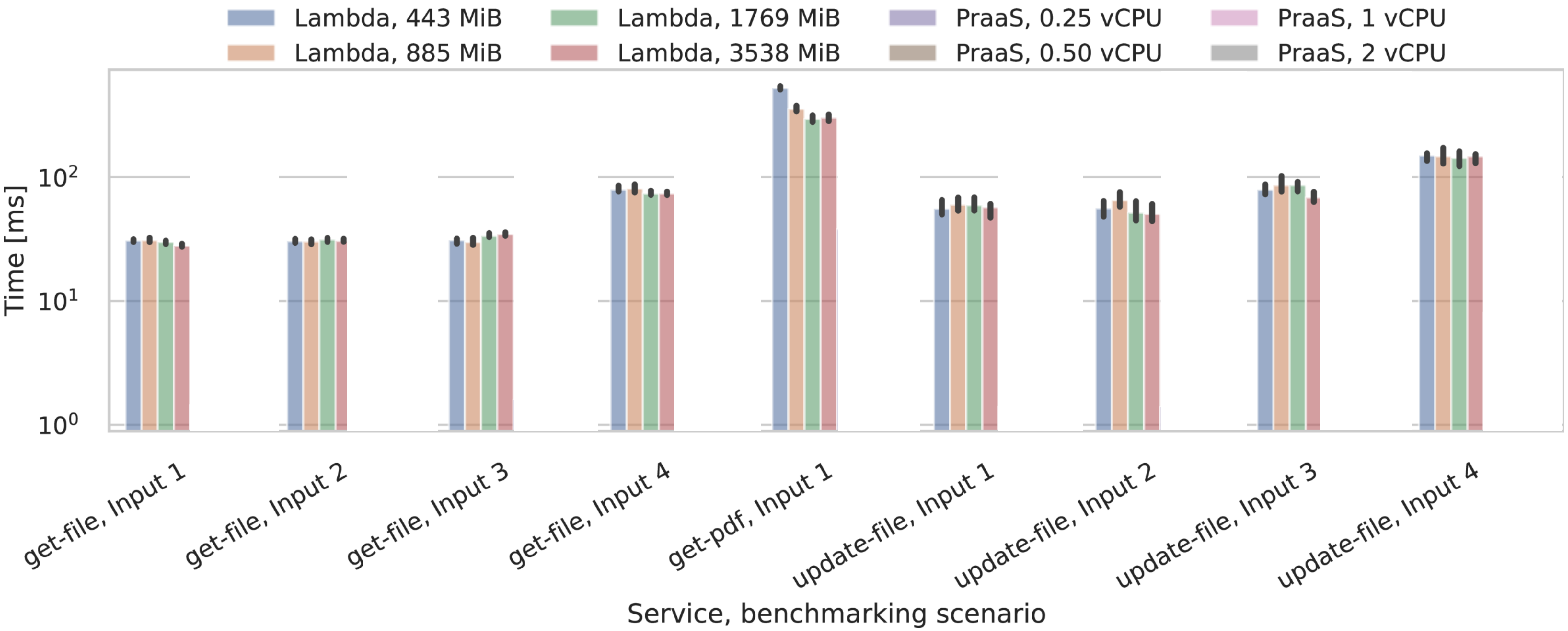
“Process-as-a-Service: Elastic and Stateful Serverless with Cloud Processes”, paper preprint.

Benchmark: LaTeX Microservice



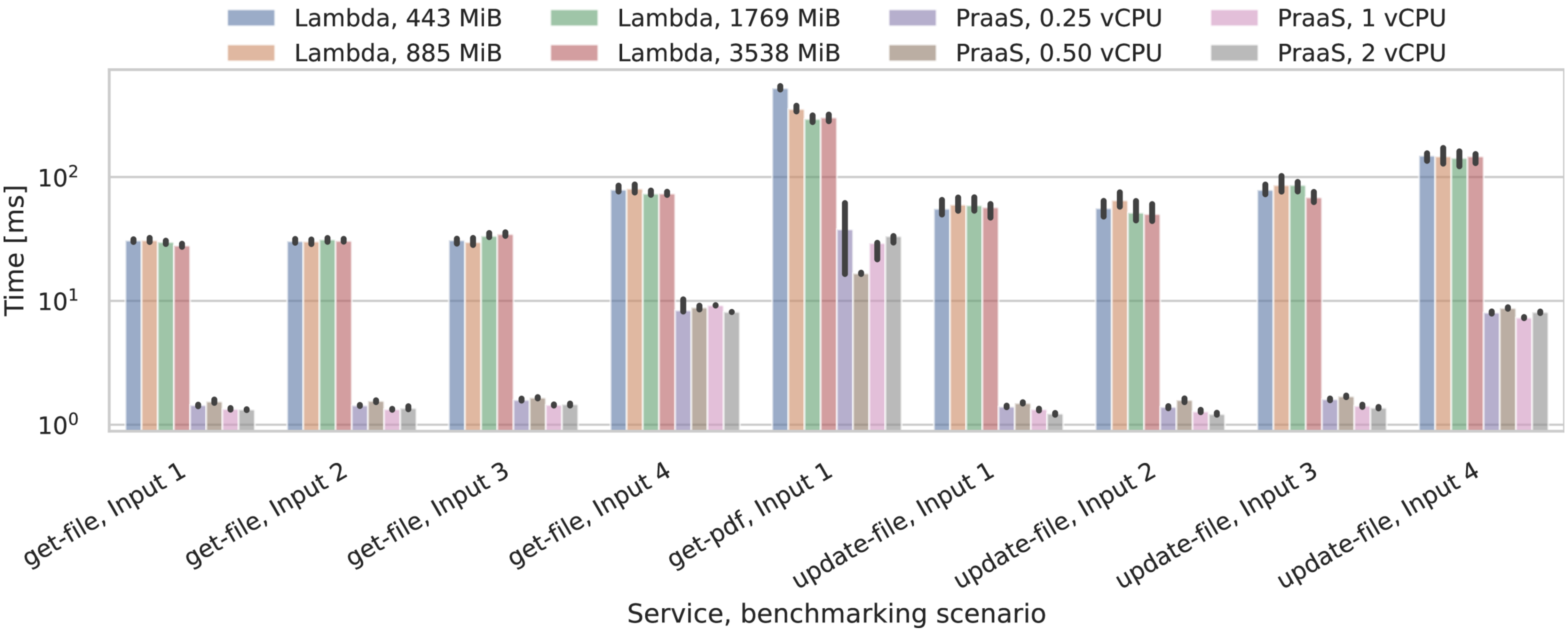
“Process-as-a-Service: Elastic and Stateful Serverless with Cloud Processes”, paper preprint.

Benchmark: LaTeX Microservice



“Process-as-a-Service: Elastic and Stateful Serverless with Cloud Processes”, paper preprint.

Benchmark: LaTeX Microservice



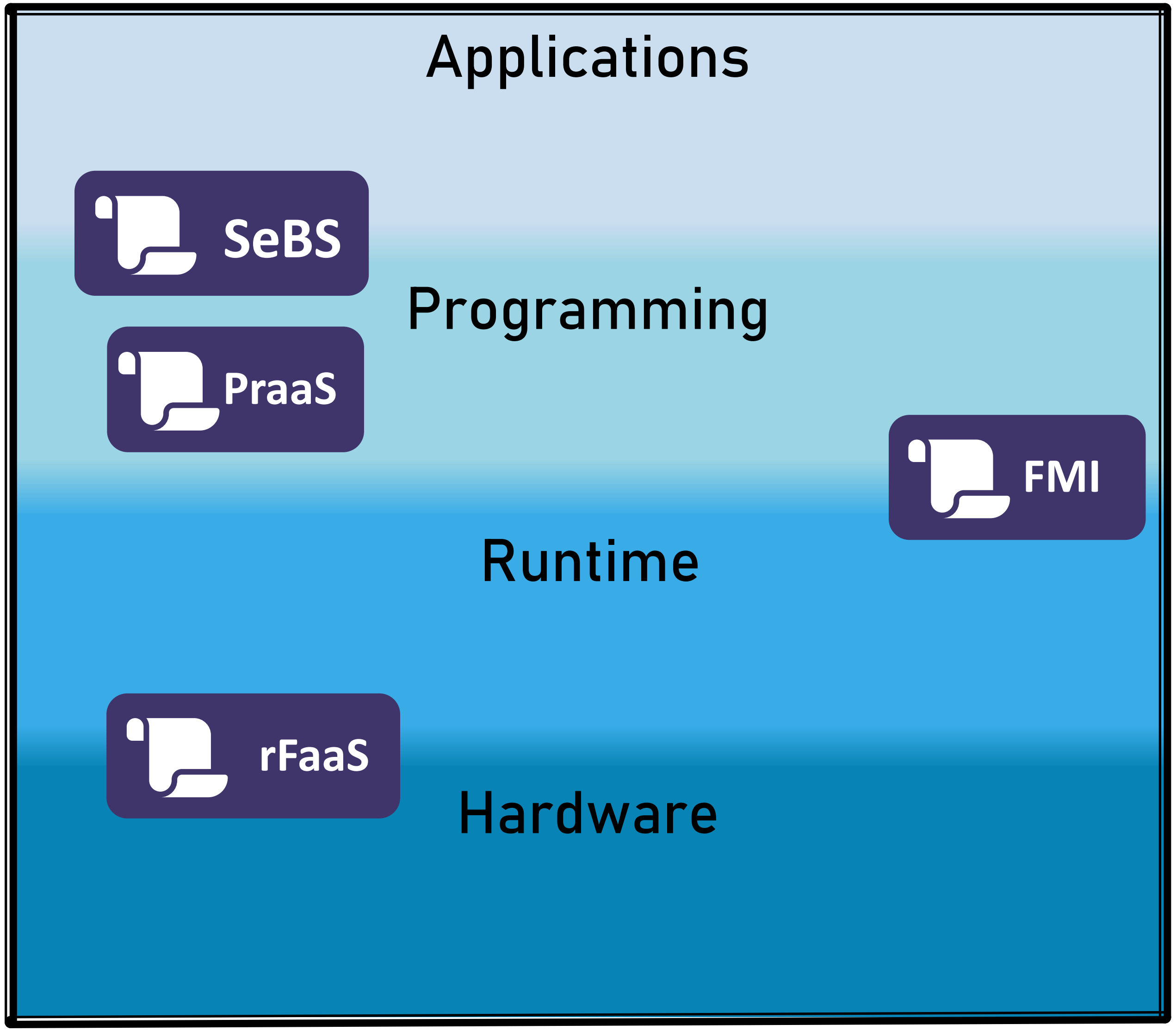
“Process-as-a-Service: Elastic and Stateful Serverless with Cloud Processes”, paper preprint.

Serverless for High-Performance Applications

Functions are expensive to invoke.

Communication is slow and restricted.

Serverless is hard to program.



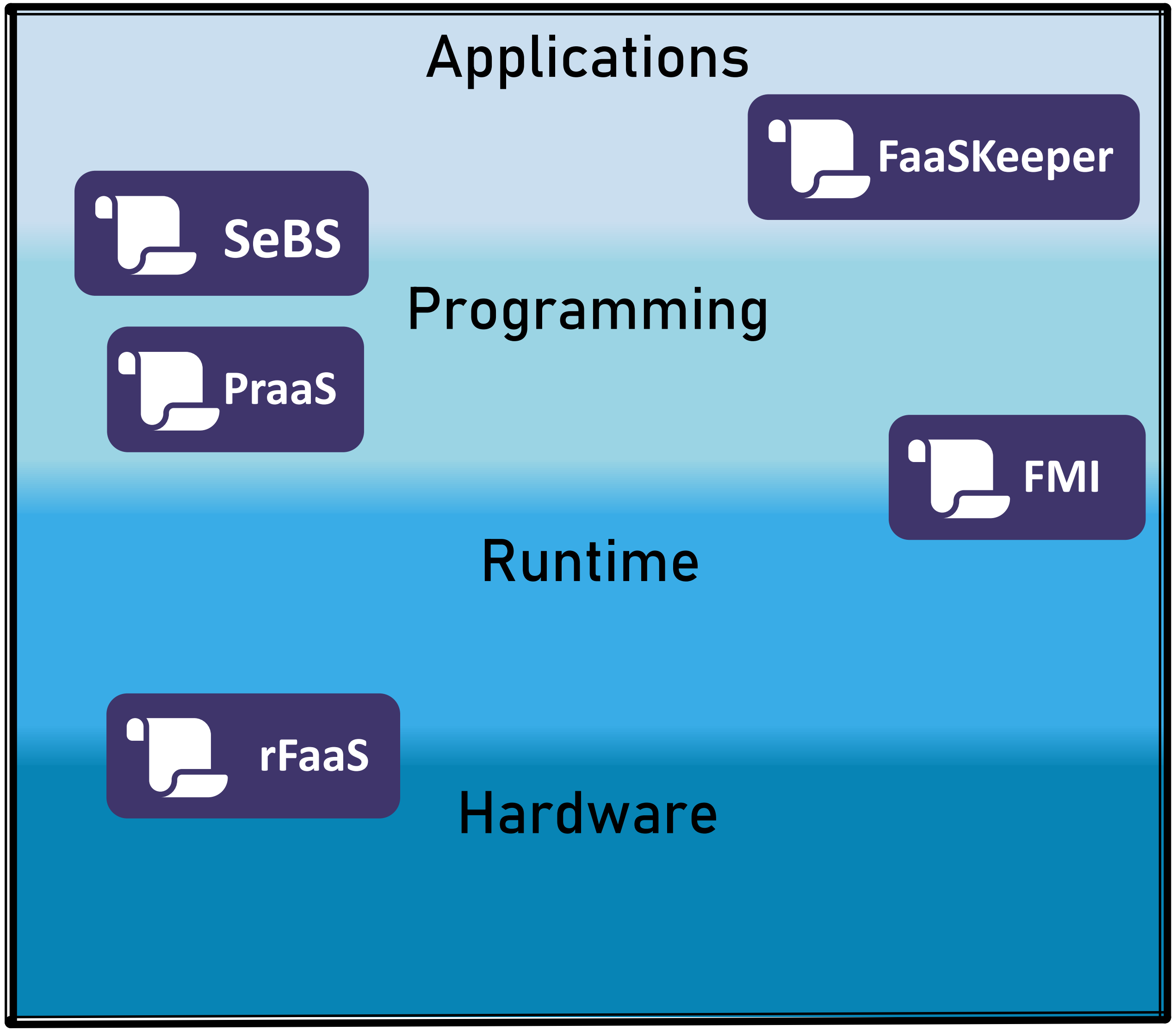
Serverless for High-Performance Applications

Functions are expensive to invoke.

Communication is slow and restricted.

Serverless is hard to program.

How to port existing and complex systems?



Serverless for High-Performance Applications

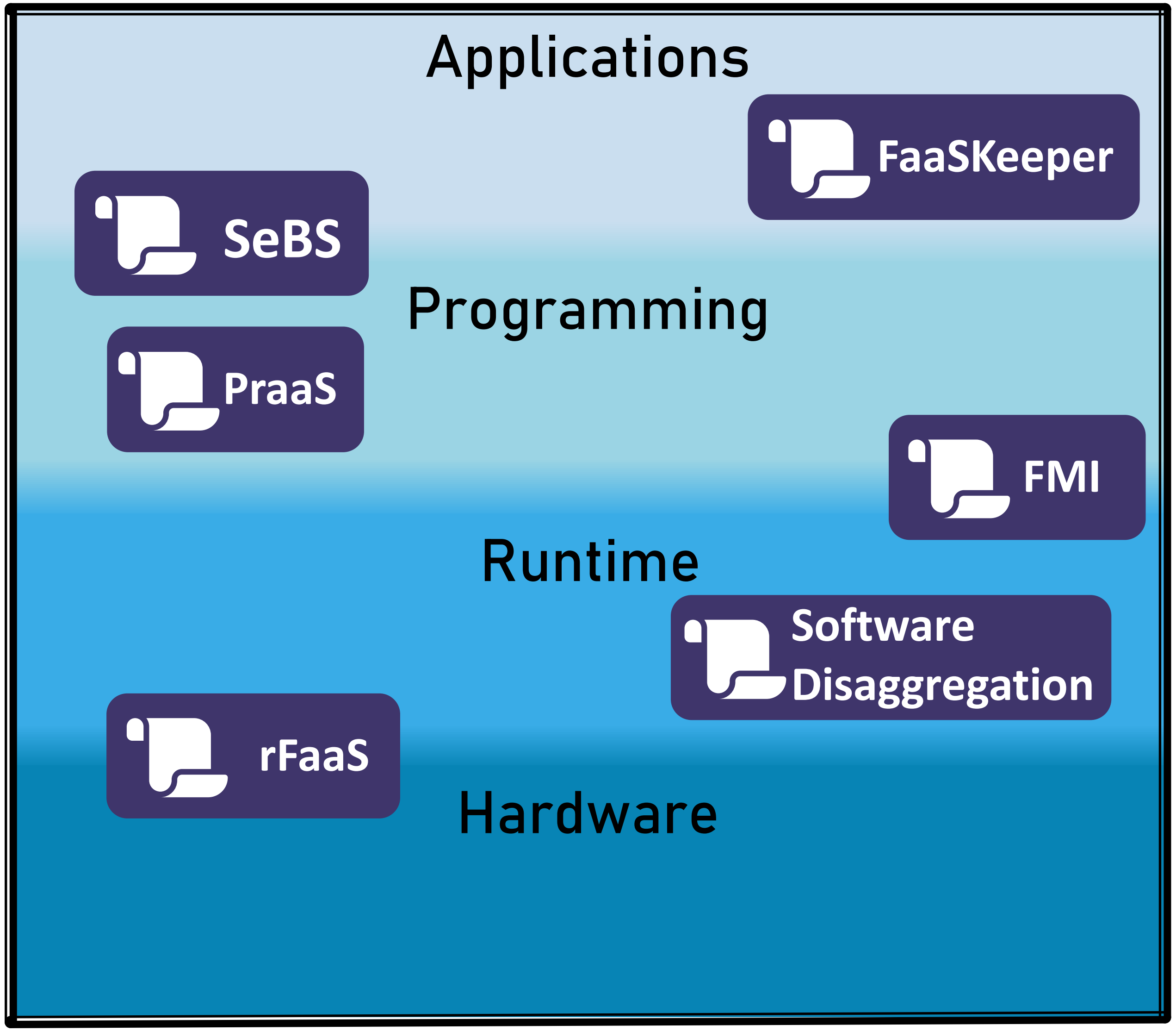
Functions are expensive to invoke.

Communication is slow and restricted.

Serverless is hard to program.

How to port existing and complex systems?

How can serverless improve HPC?



Serverless Solutions for HPC

Serverless Solutions for HPC

spcl/serverless-benchmarks

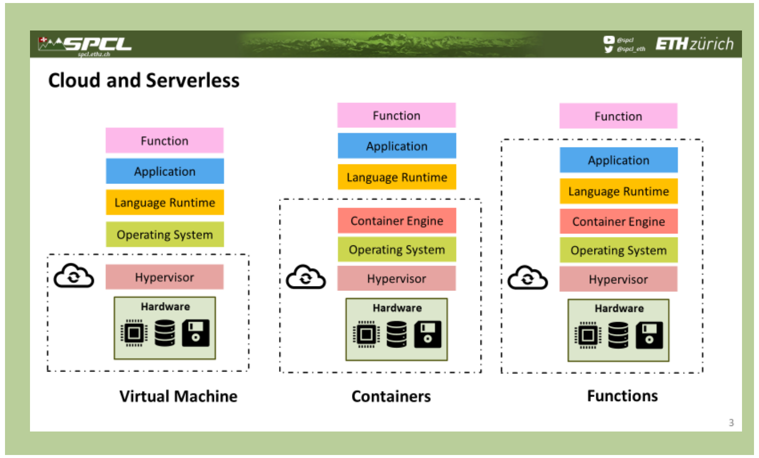
spcl/rFaaS

spcl/fmi

spcl/PraaS

spcl/FaaSKeeper

Conclusions



Cloud and Serverless

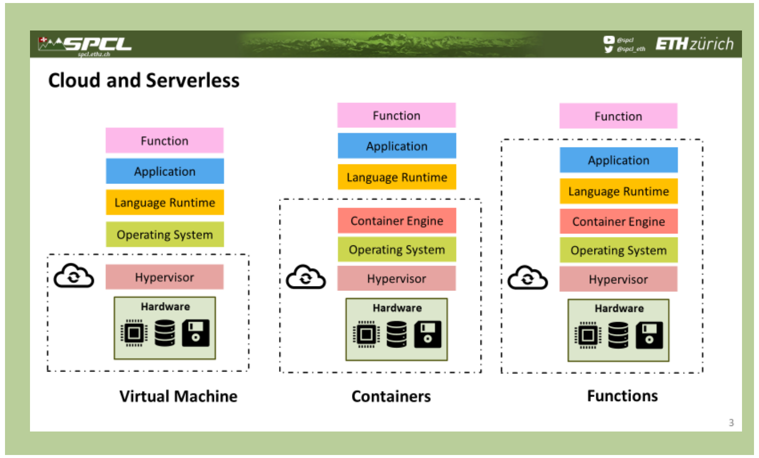
More of SPCL's research:

youtube.com/@spcl	150+ Talks
twitter.com/spcl_eth	1.2K+ Followers
github.com/spcl	2K+ Stars

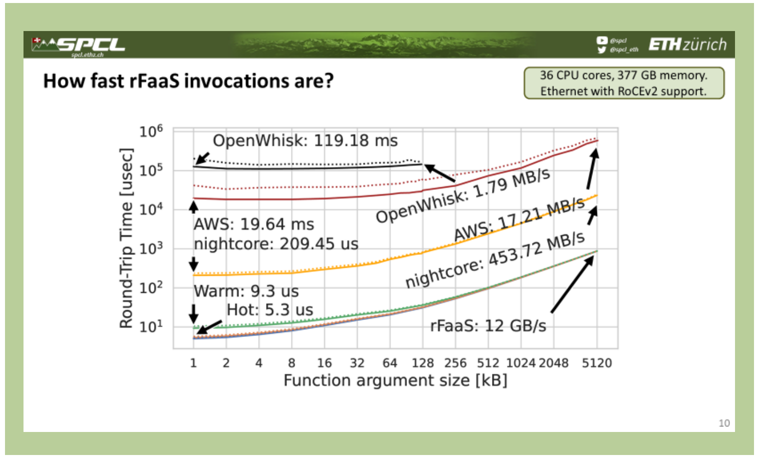
… or spcl.ethz.ch

This work has received funding from the European Research Council (ERC), Swiss National Science Foundation (SNF), and from Amazon Web Services through the AWS Cloud Credits for Research program.

Conclusions



Cloud and Serverless



How fast rFaaS invocations are?

36 CPU cores, 377 GB memory. Ethernet with RoCEv2 support.

More of SPCL's research:

youtube.com/@spcl — 150+ Talks

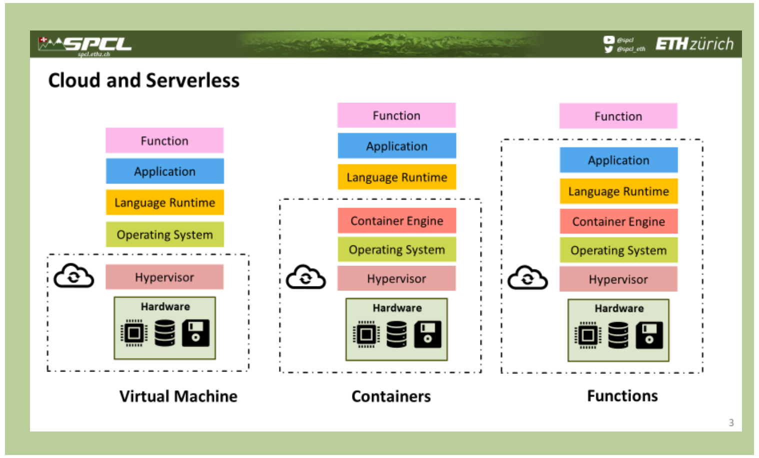
twitter.com/spcl_eth — 1.2K+ Followers

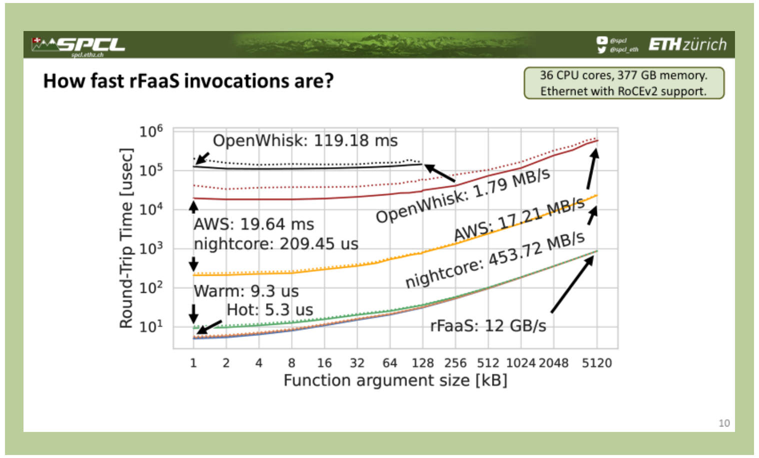
github.com/spcl — 2K+ Stars

… or spcl.ethz.ch

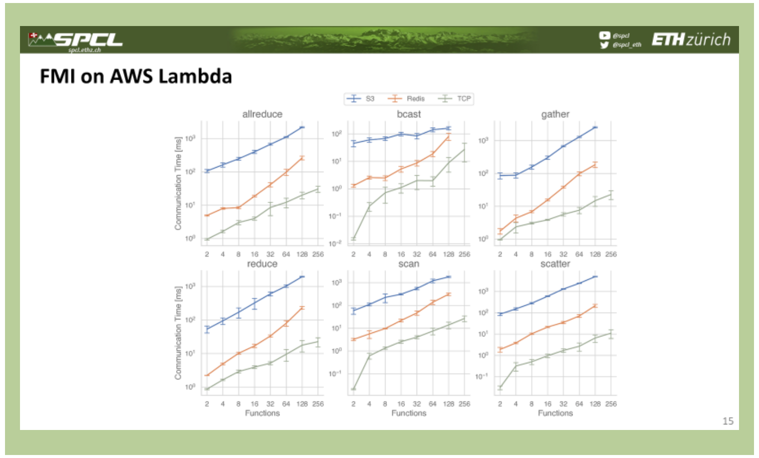
This work has received funding from the European Research Council (ERC), Swiss National Science Foundation (SNF), and from Amazon Web Services through the AWS Cloud Credits for Research program.

Conclusions


Cloud and Serverless


How fast rFaaS invocations are?

36 CPU cores, 377 GB memory. Ethernet with RoCEv2 support.


FMI on AWS Lambda

More of SPCL's research:

youtube.com/@spcl — 150+ Talks

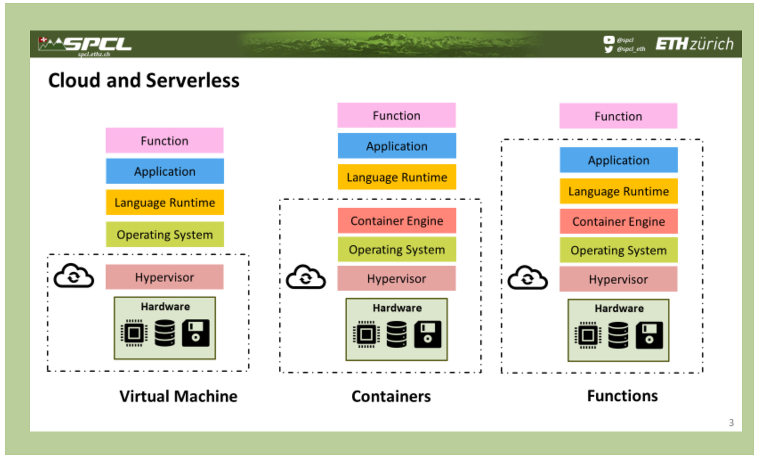
twitter.com/spcl_eth — 1.2K+ Followers

github.com/spcl — 2K+ Stars

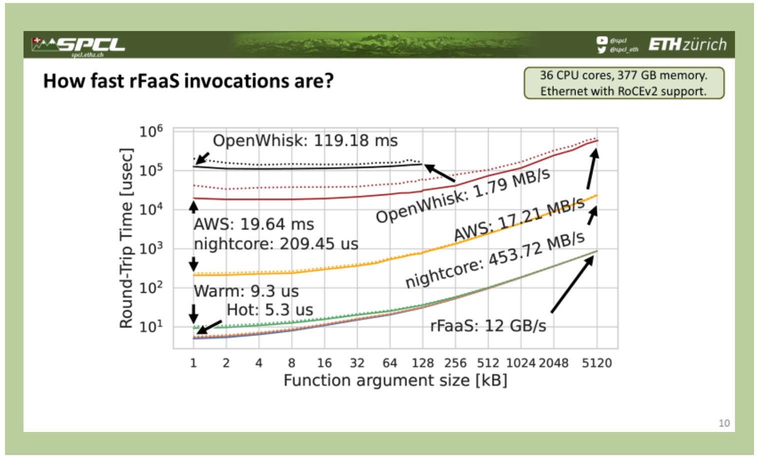
… or spcl.ethz.ch

This work has received funding from the European Research Council (ERC), Swiss National Science Foundation (SNF), and from Amazon Web Services through the AWS Cloud Credits for Research program.

Conclusions

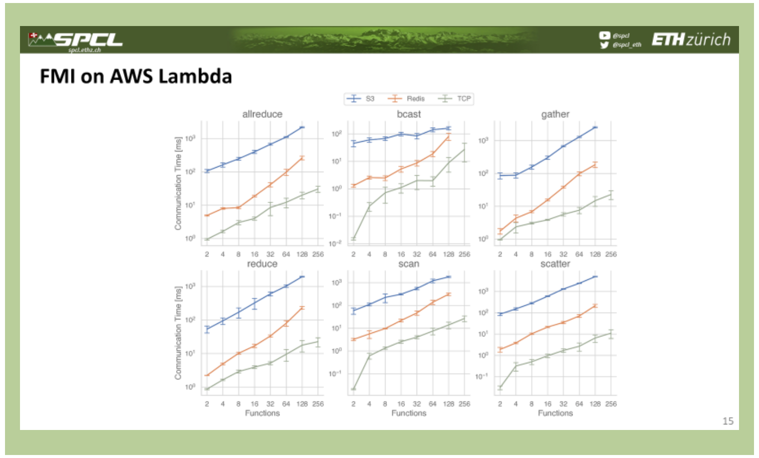


Cloud and Serverless

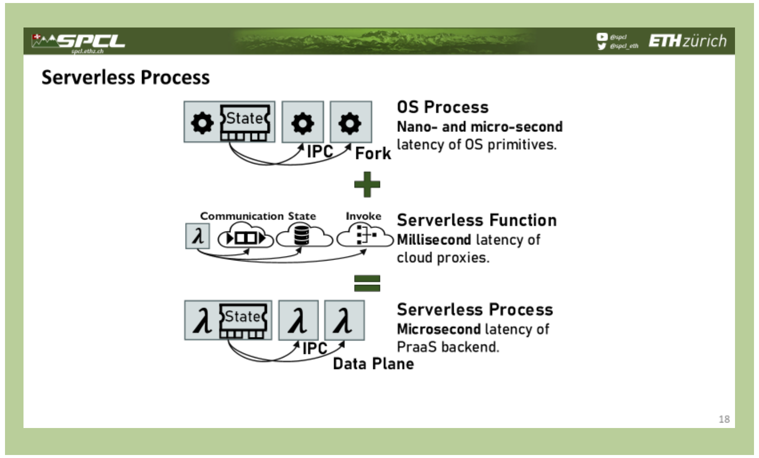


How fast rFaaS invocations are?

36 CPU cores, 377 GB memory. Ethernet with RoCEv2 support.



FMI on AWS Lambda



Serverless Process

OS Process
Nano- and micro-second latency of OS primitives.

Serverless Function
Millisecond latency of cloud proxies.

Serverless Process
Microsecond latency of PraaS backend.

More of SPCL's research:

youtube.com/@spcl — 150+ Talks

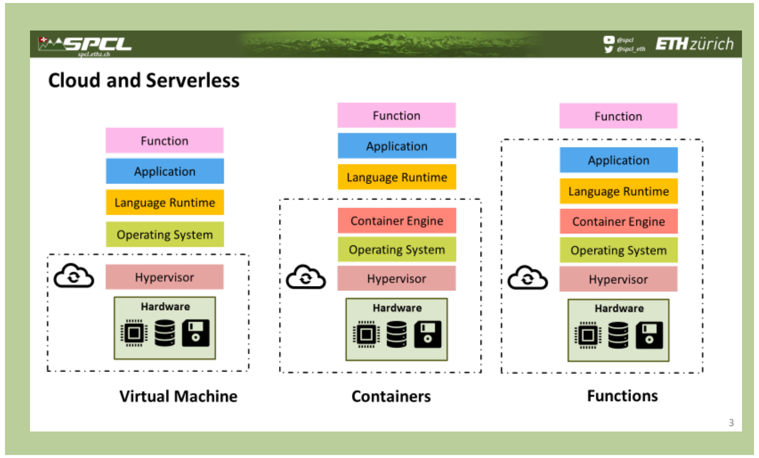
twitter.com/spcl_eth — 1.2K+ Followers

github.com/spcl — 2K+ Stars

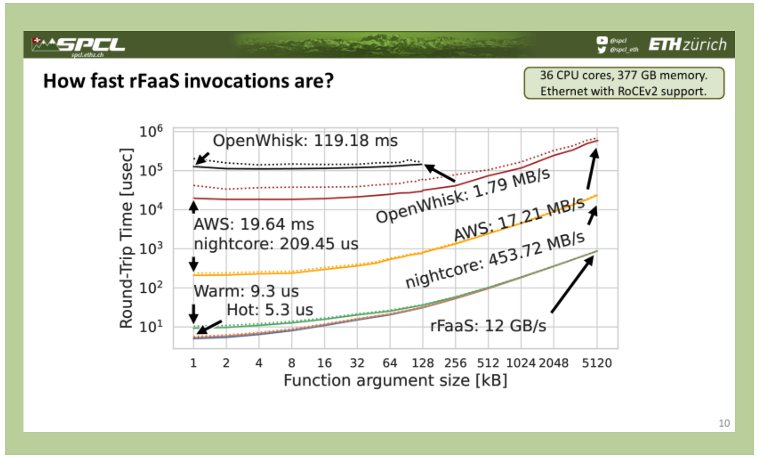
… or spcl.ethz.ch

This work has received funding from the European Research Council (ERC), Swiss National Science Foundation (SNF), and from Amazon Web Services through the AWS Cloud Credits for Research program.

Conclusions

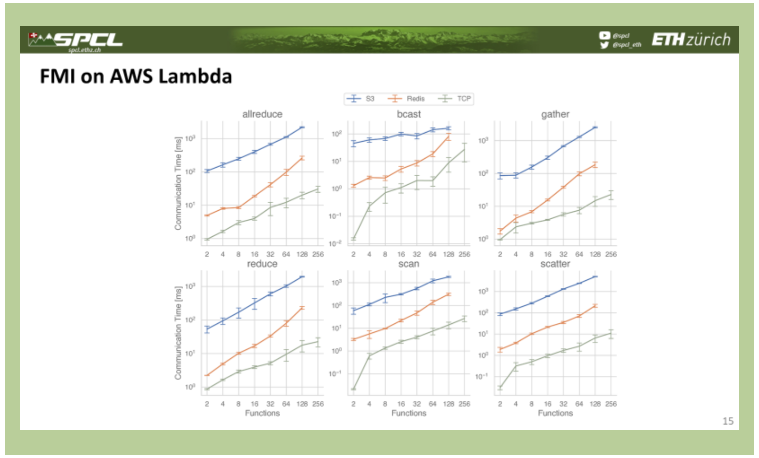


Cloud and Serverless

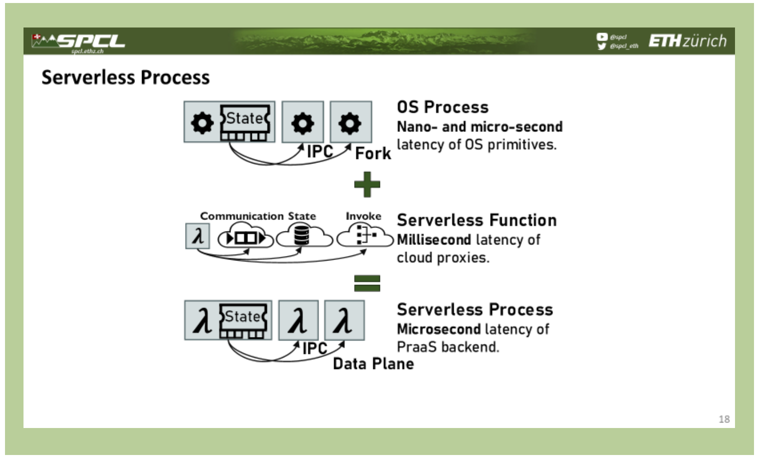


How fast rFaaS invocations are?

36 CPU cores, 377 GB memory. Ethernet with RoCEv2 support.



FMI on AWS Lambda



Serverless Process

OS Process
Nano- and micro-second latency of OS primitives.

Serverless Function
Millisecond latency of cloud proxies.

Serverless Process
Microsecond latency of PraaS backend.

More of SPCL's research:

youtube.com/@spcl	150+ Talks
twitter.com/spcl_eth	1.2K+ Followers
github.com/spcl	2K+ Stars

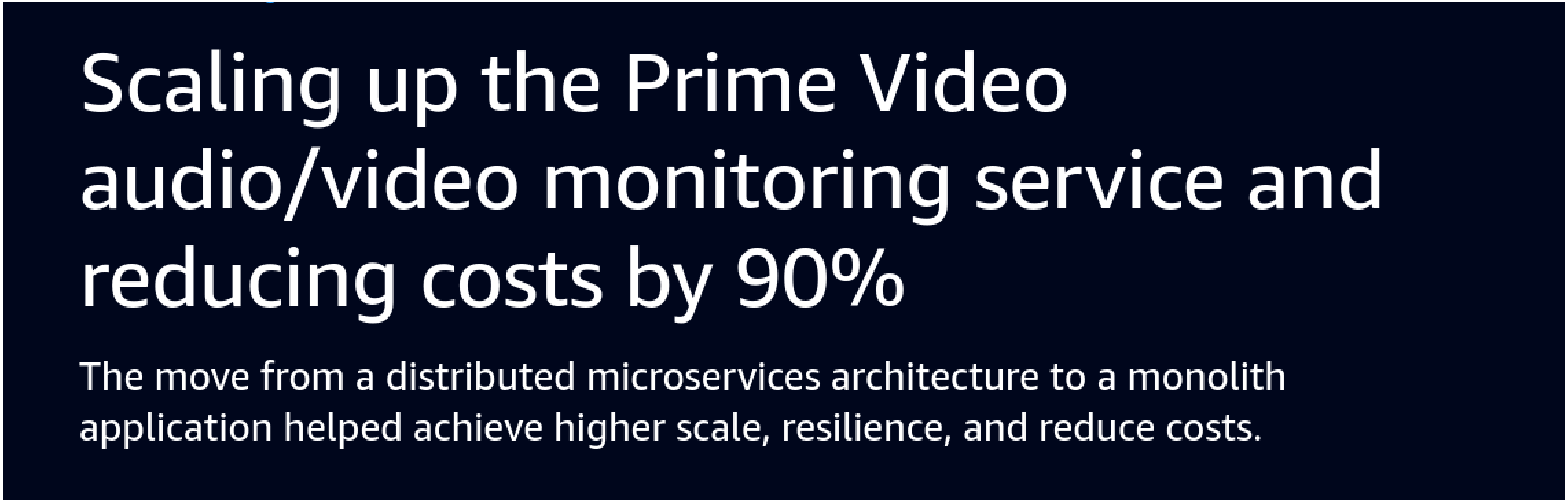
… or spcl.ethz.ch

Poster **All projects.**

This work has received funding from the European Research Council (ERC), Swiss National Science Foundation (SNF), and from Amazon Web Services through the AWS Cloud Credits for Research program.

“But serverless is slow and expensive”

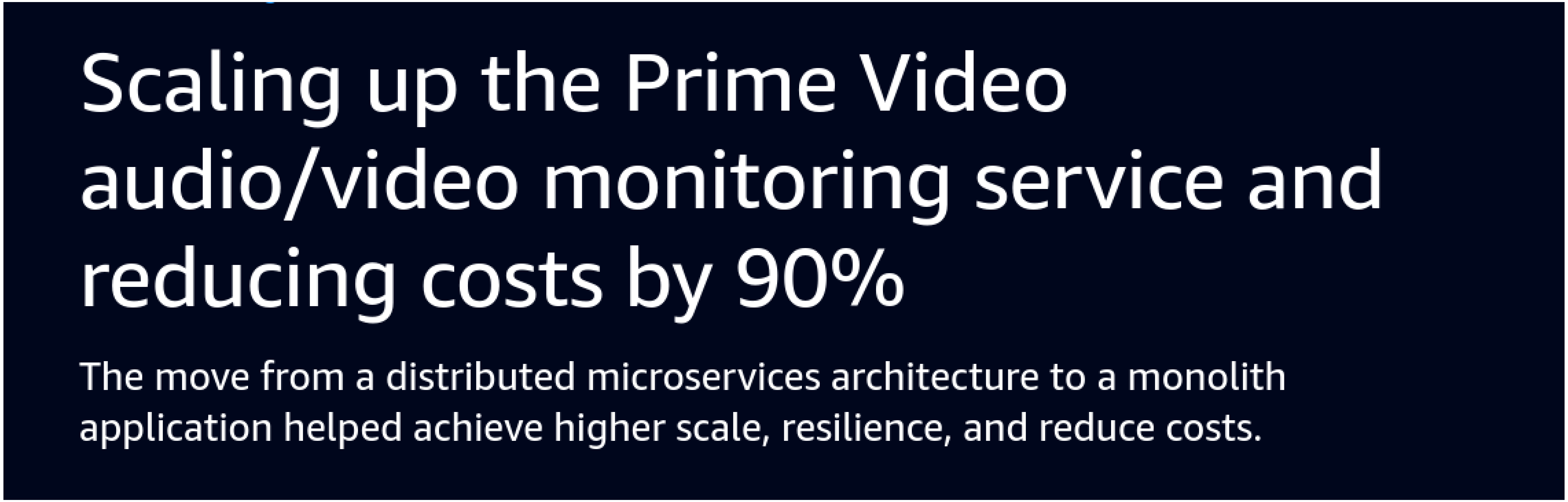
“But serverless is slow and expensive”



Scaling up the Prime Video audio/video monitoring service and reducing costs by 90%

The move from a distributed microservices architecture to a monolith application helped achieve higher scale, resilience, and reduce costs.

“But serverless is slow and expensive”



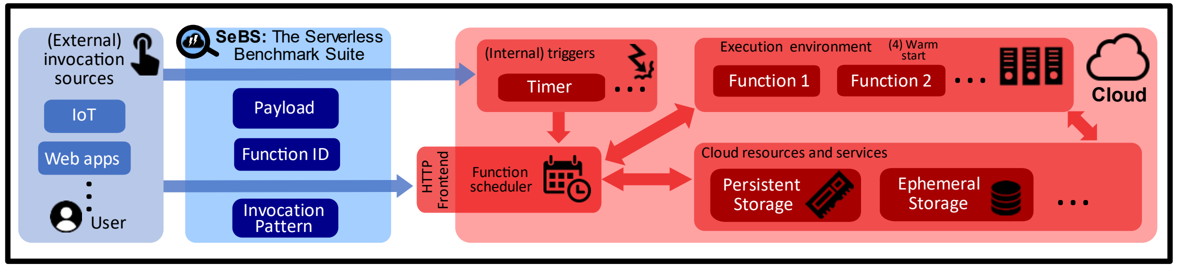
Scaling up the Prime Video audio/video monitoring service and reducing costs by 90%

The move from a distributed microservices architecture to a monolith application helped achieve higher scale, resilience, and reduce costs.

Benchmarking Serverless

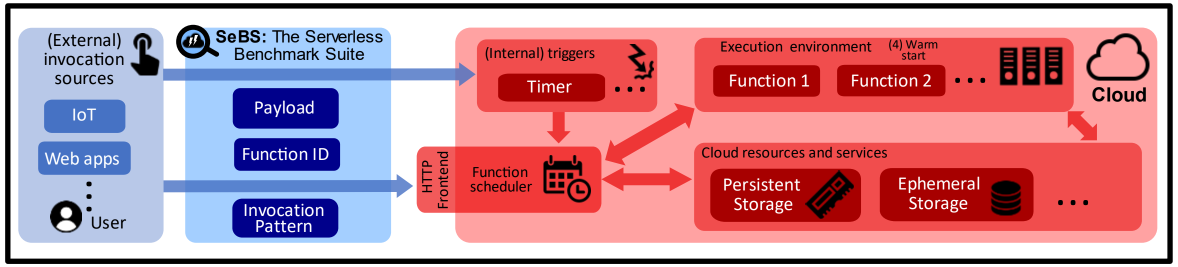
“SeBS: a Serverless Benchmark Suite for Function-as-a-Service Computing”, ACM/IFIP Middleware 2021

Benchmarking Serverless



“SeBS: a Serverless Benchmark Suite for Function-as-a-Service Computing”, ACM/IFIP Middleware 2021

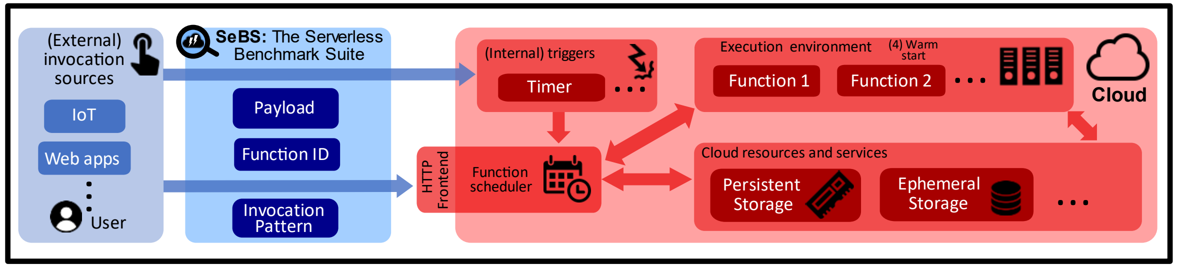
Benchmarking Serverless

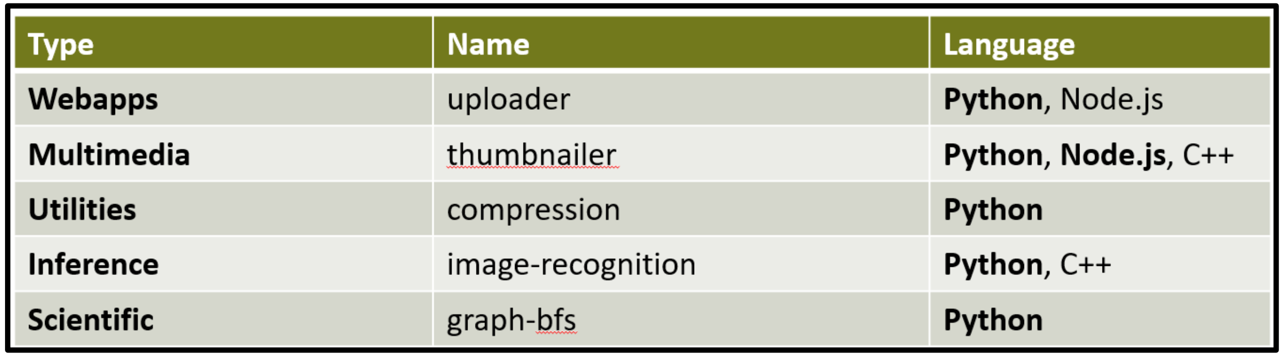


Type	Name	Language
Webapps	uploader	**Python**, Node.js
Multimedia	thumbnailer	**Python**, **Node.js**, C++
Utilities	compression	**Python**
Inference	image-recognition	**Python**, C++
Scientific	graph-bfs	**Python**

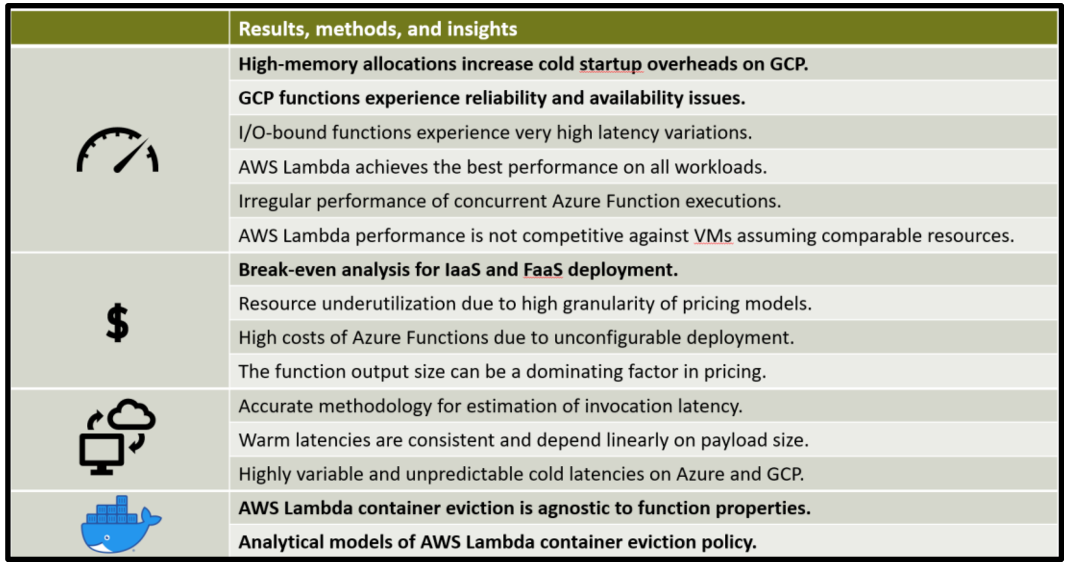
“SeBS: a Serverless Benchmark Suite for Function-as-a-Service Computing”, ACM/IFIP Middleware 2021

Benchmarking Serverless





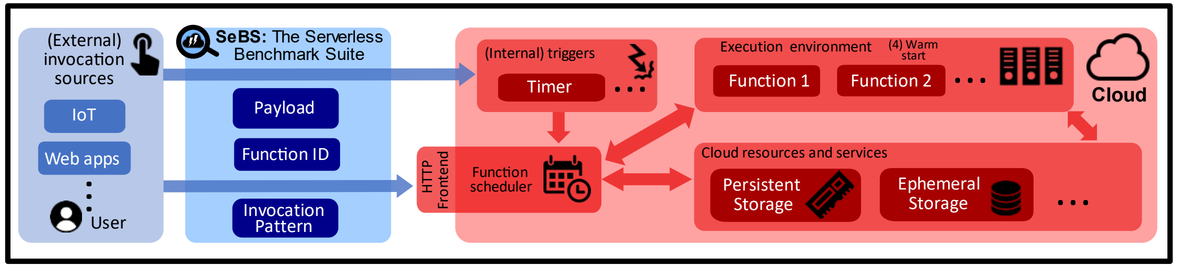
Type	Name	Language
Webapps	uploader	**Python**, Node.js
Multimedia	thumbnailer	**Python**, **Node.js**, C++
Utilities	compression	**Python**
Inference	image-recognition	**Python**, C++
Scientific	graph-bfs	**Python**



	Results, methods, and insights
	High-memory allocations increase cold startup overheads on GCP.
	GCP functions experience reliability and availability issues.
	I/O-bound functions experience very high latency variations.
	AWS Lambda achieves the best performance on all workloads.
	Irregular performance of concurrent Azure Function executions.
	AWS Lambda performance is not competitive against VMs assuming comparable resources.
$	**Break-even analysis for IaaS and FaaS deployment.**
	Resource underutilization due to high granularity of pricing models.
	High costs of Azure Functions due to unconfigurable deployment.
	The function output size can be a dominating factor in pricing.
	Accurate methodology for estimation of invocation latency.
	Warm latencies are consistent and depend linearly on payload size.
	Highly variable and unpredictable cold latencies on Azure and GCP.
	AWS Lambda container eviction is agnostic to function properties.
	Analytical models of AWS Lambda container eviction policy.

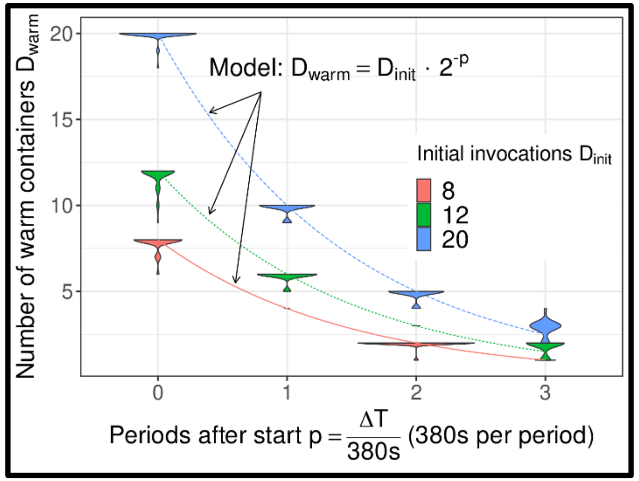
“SeBS: a Serverless Benchmark Suite for Function-as-a-Service Computing”, ACM/IFIP Middleware 2021

Benchmarking Serverless



Type	Name	Language
Webapps	uploader	**Python**, Node.js
Multimedia	thumbnailer	**Python**, **Node.js**, C++
Utilities	compression	**Python**
Inference	image-recognition	**Python**, C++
Scientific	graph-bfs	**Python**

	Results, methods, and insights
	High-memory allocations increase cold startup overheads on GCP.
	GCP functions experience reliability and availability issues.
	I/O-bound functions experience very high latency variations.
	AWS Lambda achieves the best performance on all workloads.
	Irregular performance of concurrent Azure Function executions.
	AWS Lambda performance is not competitive against VMs assuming comparable resources.
$	**Break-even analysis for IaaS and FaaS deployment.**
	Resource underutilization due to high granularity of pricing models.
	High costs of Azure Functions due to unconfigurable deployment.
	The function output size can be a dominating factor in pricing.
	Accurate methodology for estimation of invocation latency.
	Warm latencies are consistent and depend linearly on payload size.
	Highly variable and unpredictable cold latencies on Azure and GCP.
	AWS Lambda container eviction is agnostic to function properties.
	Analytical models of AWS Lambda container eviction policy.



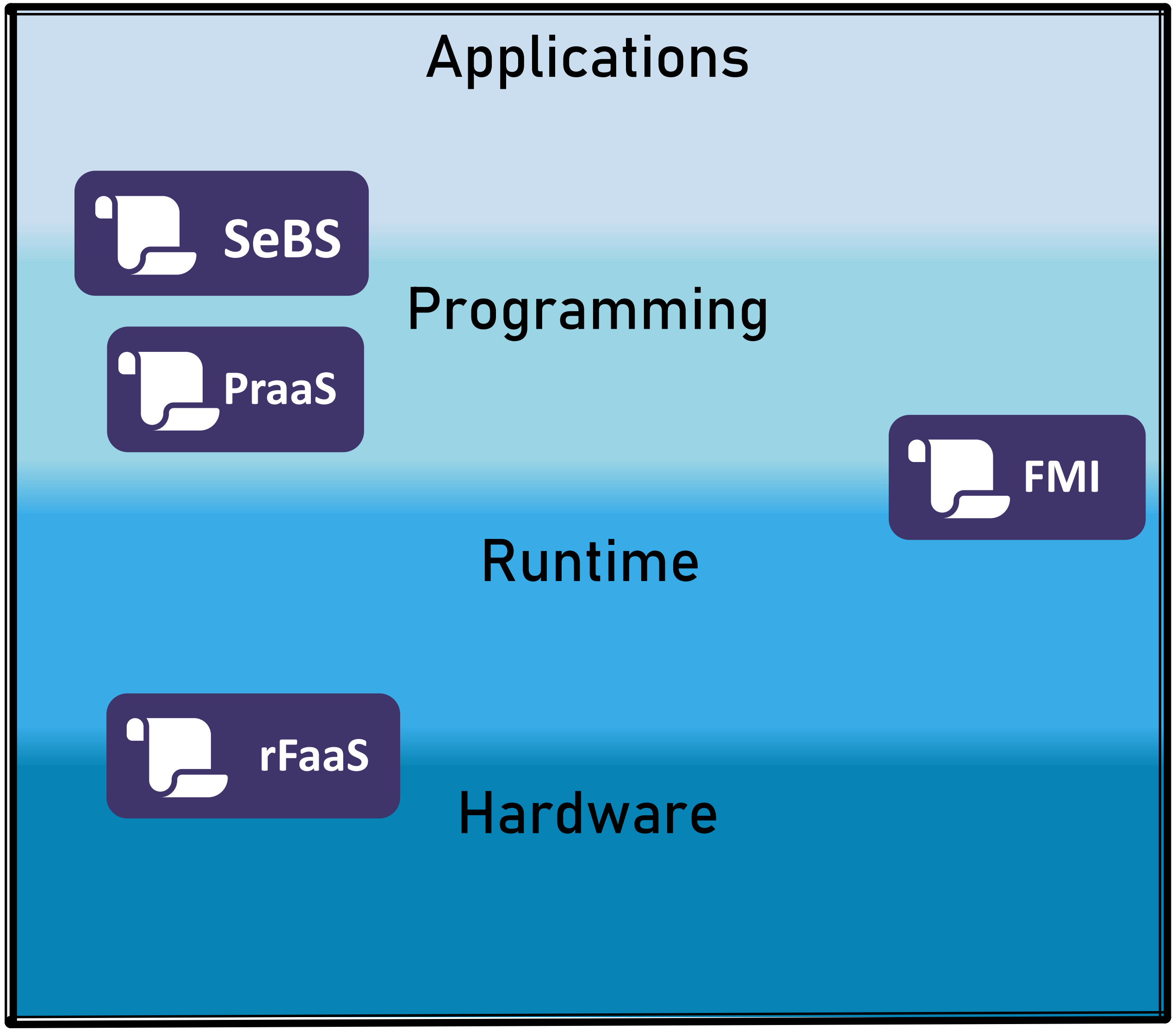
“SeBS: a Serverless Benchmark Suite for Function-as-a-Service Computing”, ACM/IFIP Middleware 2021

Serverless for High-Performance Applications

Functions are expensive to invoke.

Communication is slow and restricted.

Serverless is hard to program.



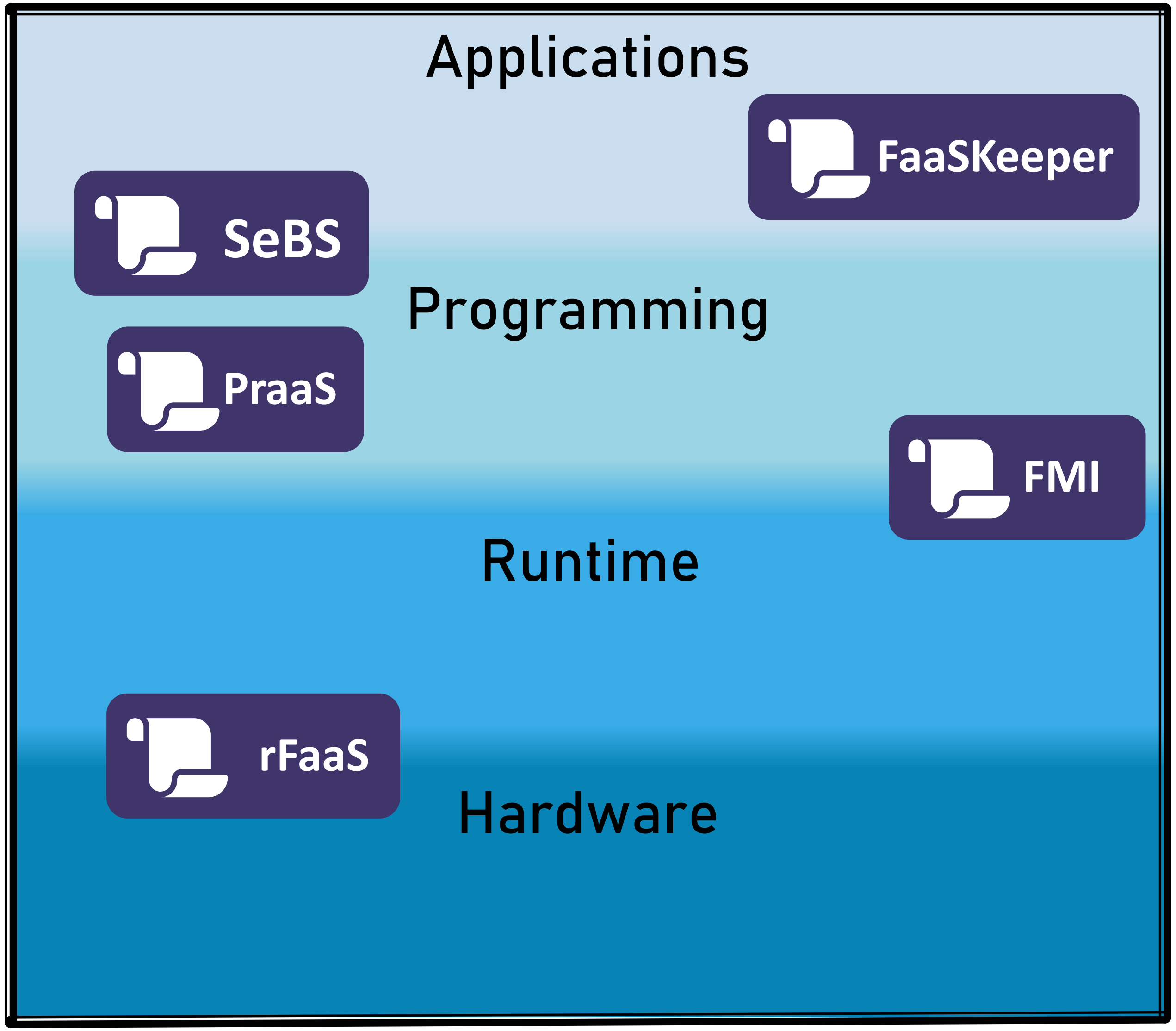
Serverless for High-Performance Applications

Functions are expensive to invoke.

Communication is slow and restricted.

Serverless is hard to program.

How to port existing and complex systems?



Serverless for High-Performance Applications

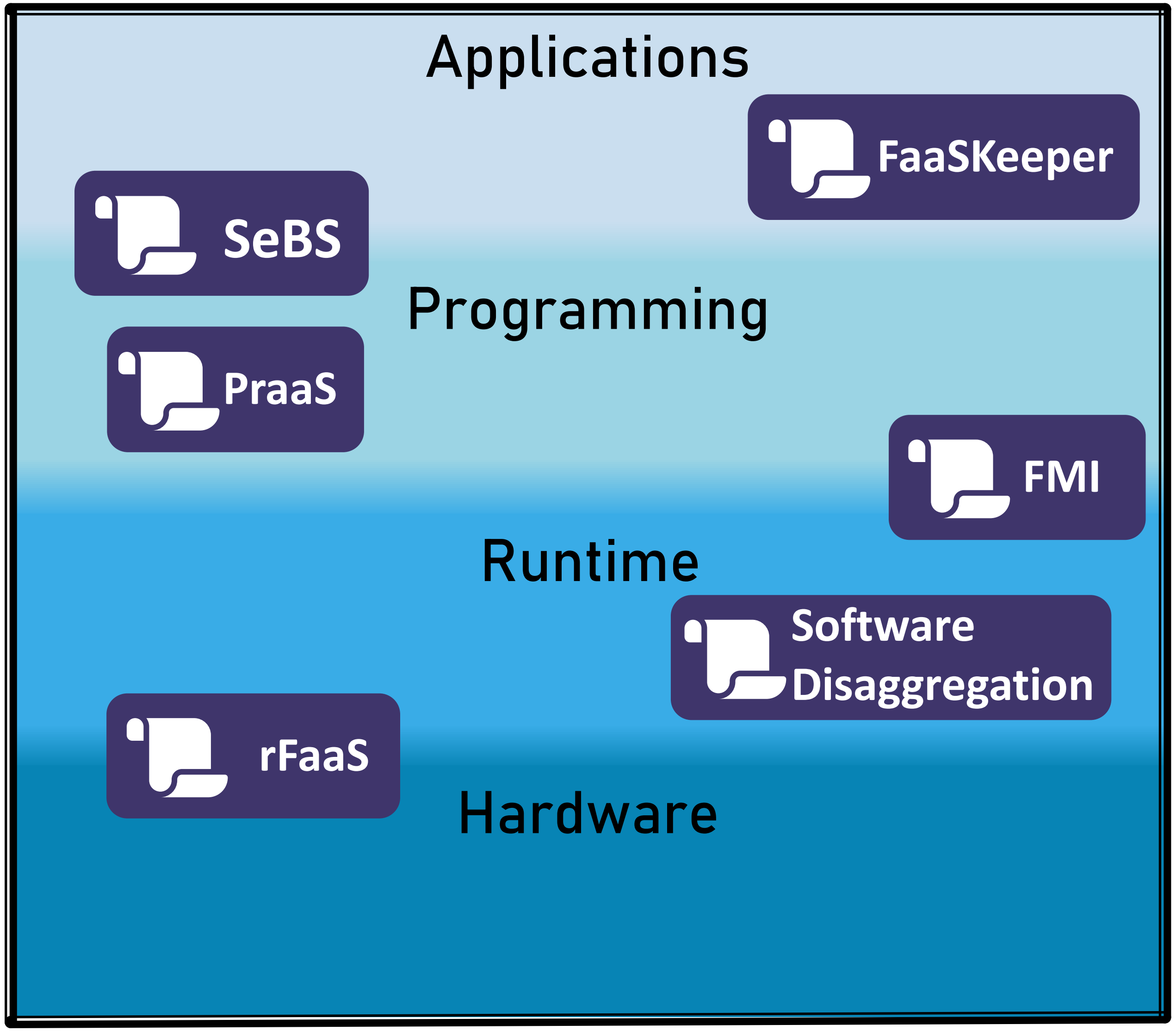
Functions are expensive to invoke.

Communication is slow and restricted.

Serverless is hard to program.

How to port existing and complex systems?

How can serverless improve HPC?



Serverless for High-Performance Applications

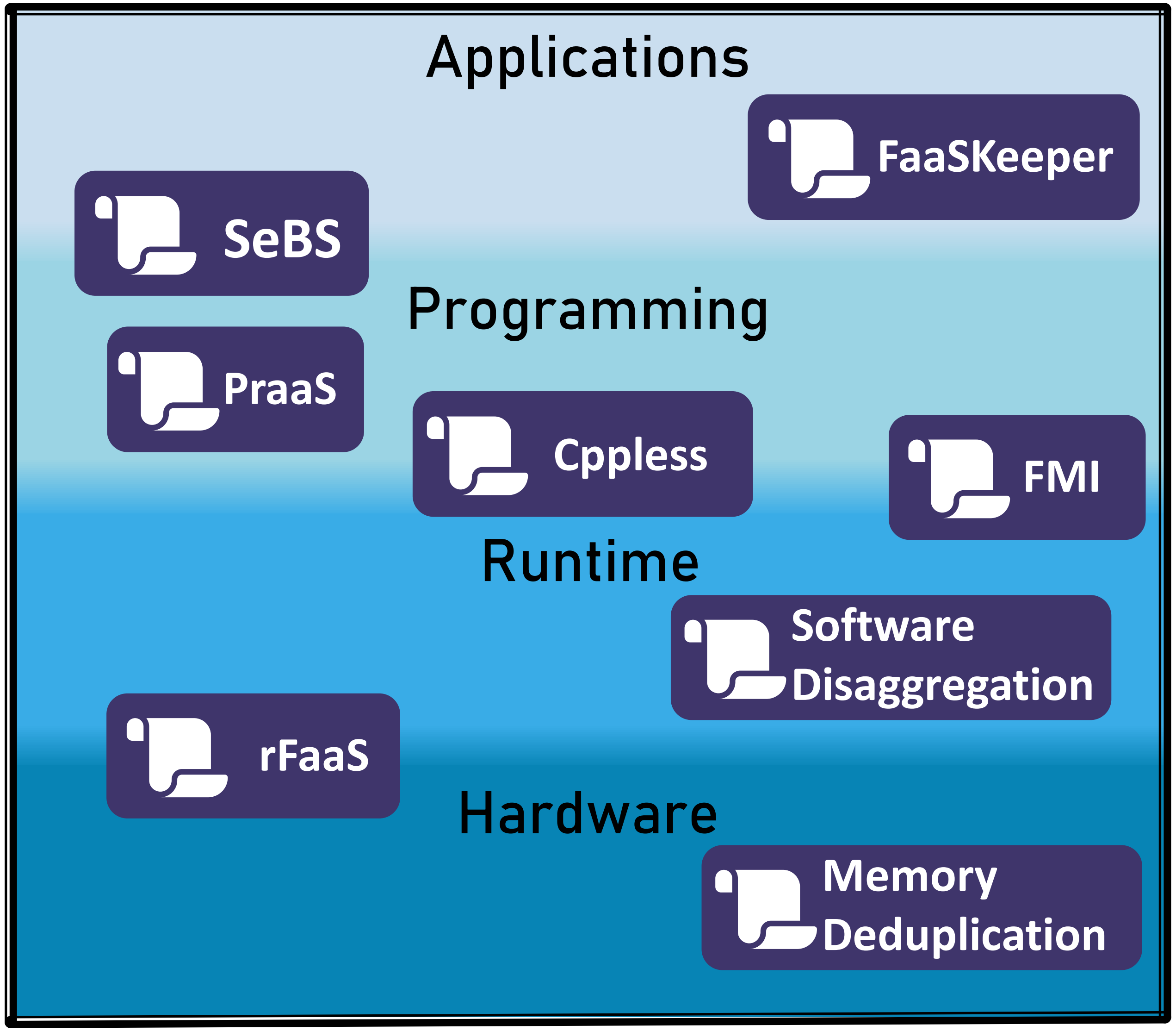
Functions are expensive to invoke.

Communication is slow and restricted.

Serverless is hard to program.

How to port existing and complex systems?

How can serverless improve HPC?



Applications

FaaSKeeper

SeBS

Programming

PraaS

Cppless

FMI

Runtime

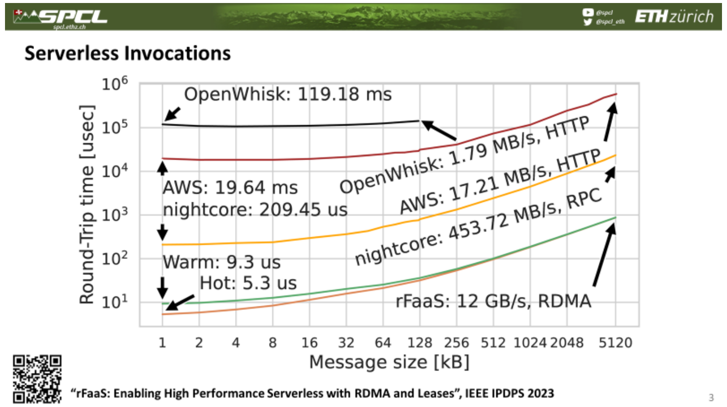
Software Disaggregation

rFaaS

Hardware

Memory Deduplication

Conclusions



Serverless Invocations

“rFaaS: Enabling High Performance Serverless with RDMA and Leases”, IEEE IPDPS 2023

3

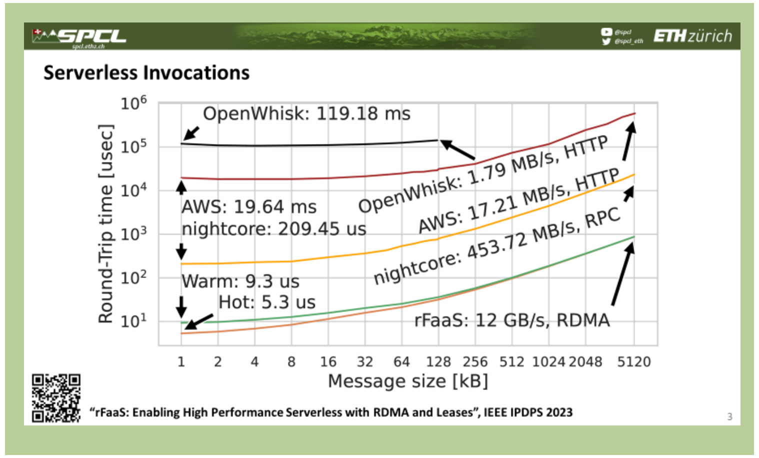
More of SPCL’s research:

youtube.com/@spcl	150+ Talks
twitter.com/spcl_eth	1.2K+ Followers
github.com/spcl	2K+ Stars

… or spcl.ethz.ch

mcopik.github.io/projects/praas

Conclusions



Serverless Invocations

"rFaaS: Enabling High Performance Serverless with RDMA and Leases", IEEE IPDPS 2023

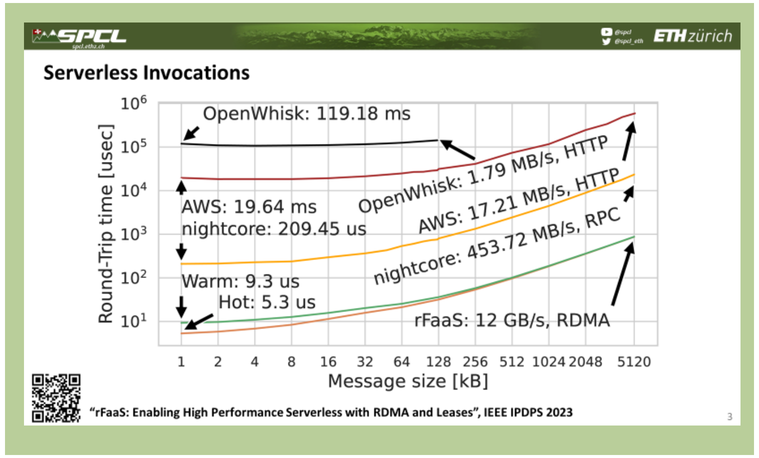
More of SPCL's research:

▶	youtube.com/@spcl	150+ Talks
🐦	twitter.com/spcl_eth	1.2K+ Followers
	github.com/spcl	2K+ Stars

… or spcl.ethz.ch

mcopik.github.io/projects/praas

Conclusions



Serverless Invocations

"rFaaS: Enabling High Performance Serverless with RDMA and Leases", IEEE IPDPS 2023

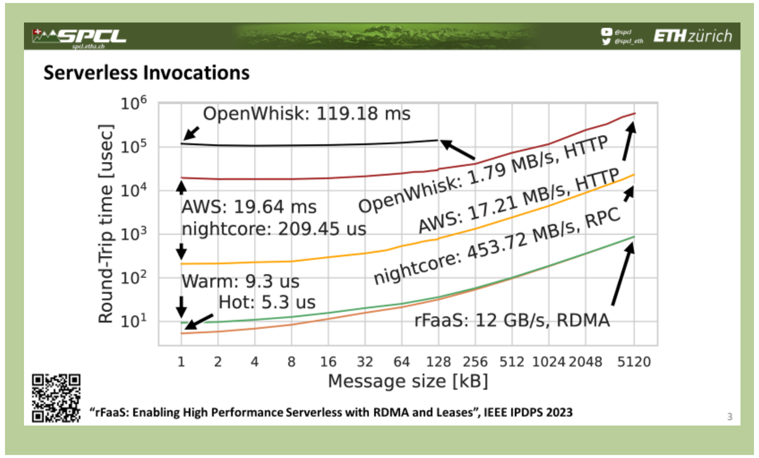
More of SPCL's research:

youtube.com/@spcl	150+ Talks
twitter.com/spcl_eth	1.2K+ Followers
github.com/spcl	2K+ Stars

… or spcl.ethz.ch

mcopik.github.io/projects/praas

Conclusions



Serverless Invocations

"rFaaS: Enabling High Performance Serverless with RDMA and Leases", IEEE IPDPS 2023

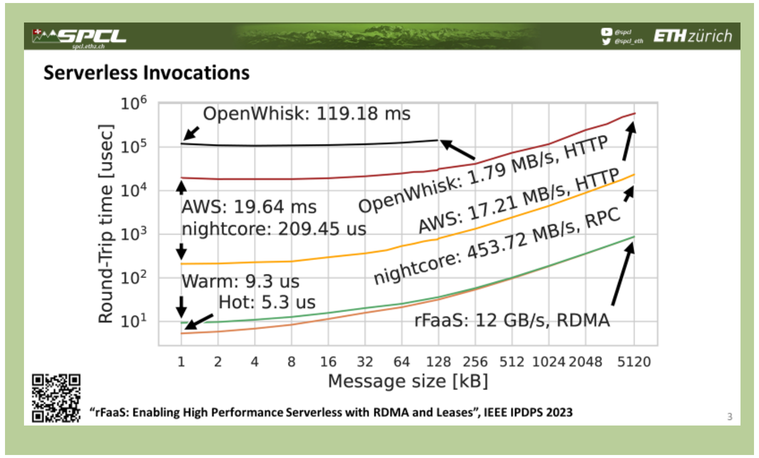
More of SPCL's research:

youtube.com/@spcl	150+ Talks
twitter.com/spcl_eth	1.2K+ Followers
github.com/spcl	2K+ Stars

… or spcl.ethz.ch

mcopik.github.io/projects/praas

Conclusions



Serverless Invocations

“rFaaS: Enabling High Performance Serverless with RDMA and Leases”, IEEE IPDPS 2023

3

More of SPCL’s research:

youtube.com/@spcl 150+ Talks

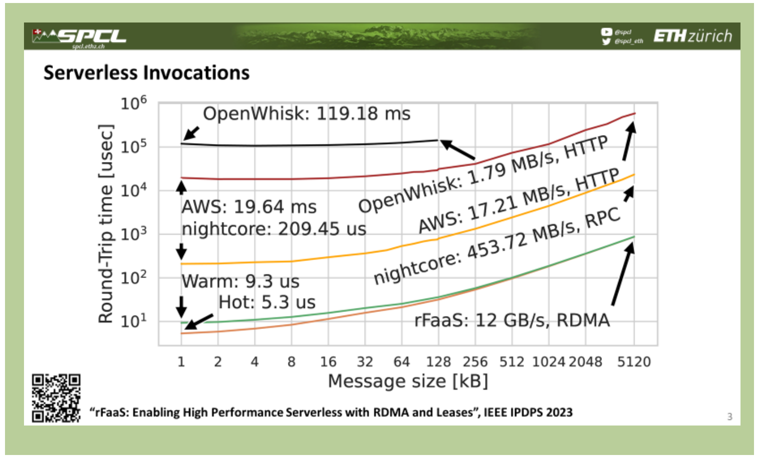
twitter.com/spcl_eth 1.2K+ Followers

github.com/spcl 2K+ Stars

… or spcl.ethz.ch

mcopik.github.io/projects/praas

Conclusions



Serverless Invocations

"rFaaS: Enabling High Performance Serverless with RDMA and Leases", IEEE IPDPS 2023

More of SPCL's research:

youtube.com/@spcl	150+ Talks
twitter.com/spcl_eth	1.2K+ Followers
github.com/spcl	2K+ Stars

… or spcl.ethz.ch

GitHub spcl/praas

Paper preprint

mcopik.github.io/projects/praas