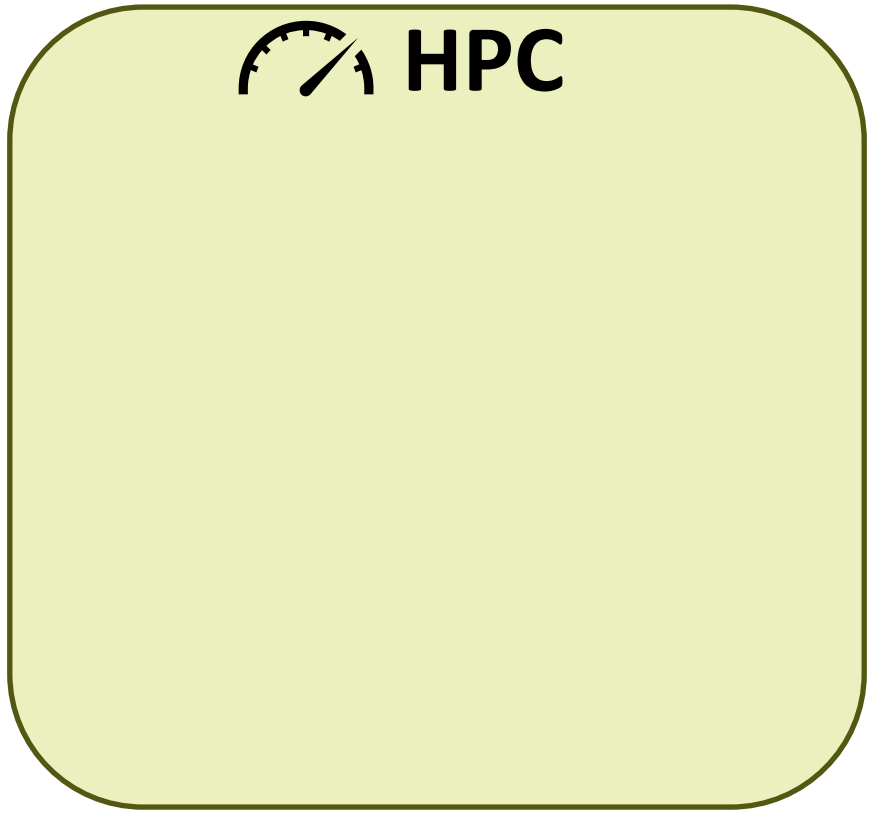
MARCIN COPIK

High Performance Serverless for HPC and Clouds

High-Performance Computing Systems

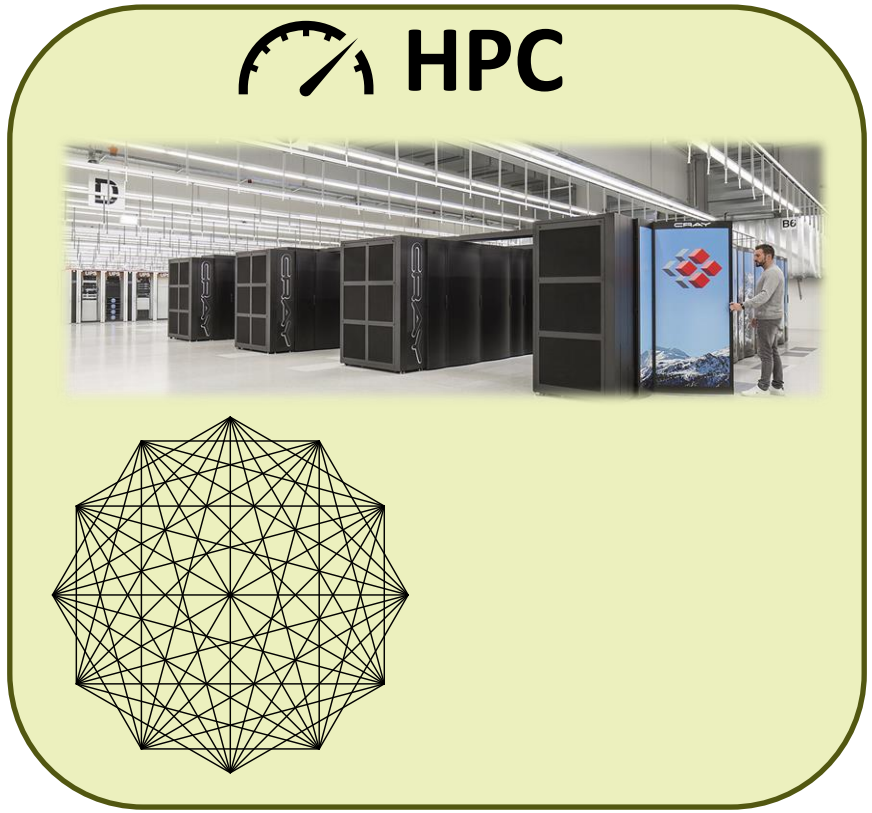
High-Performance Computing Systems



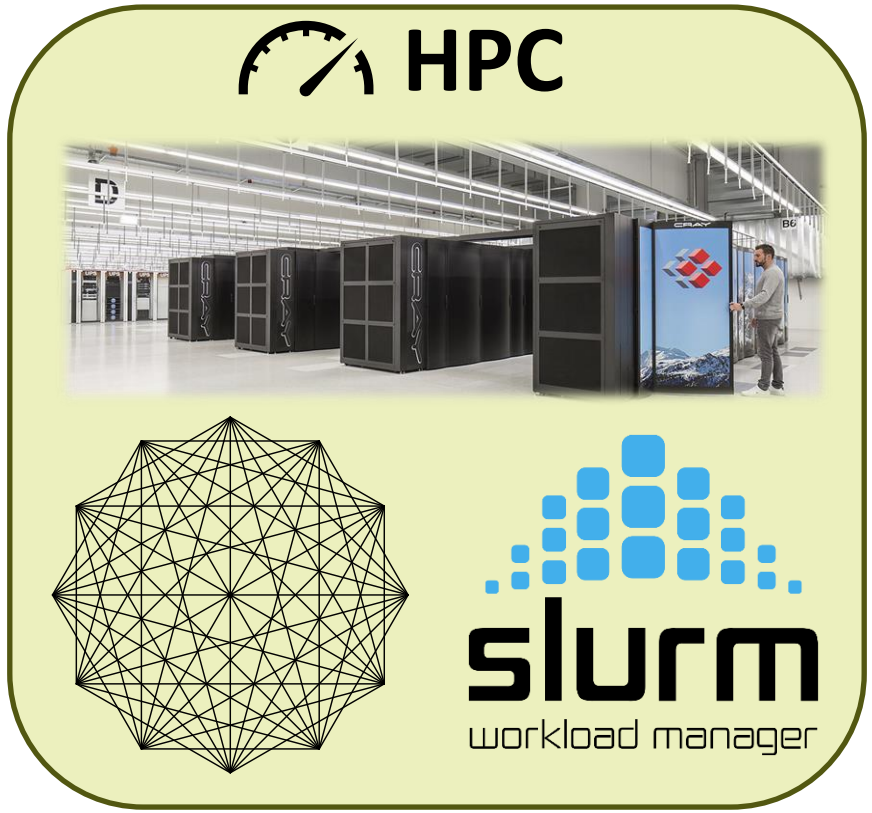
High-Performance Computing Systems

HPC

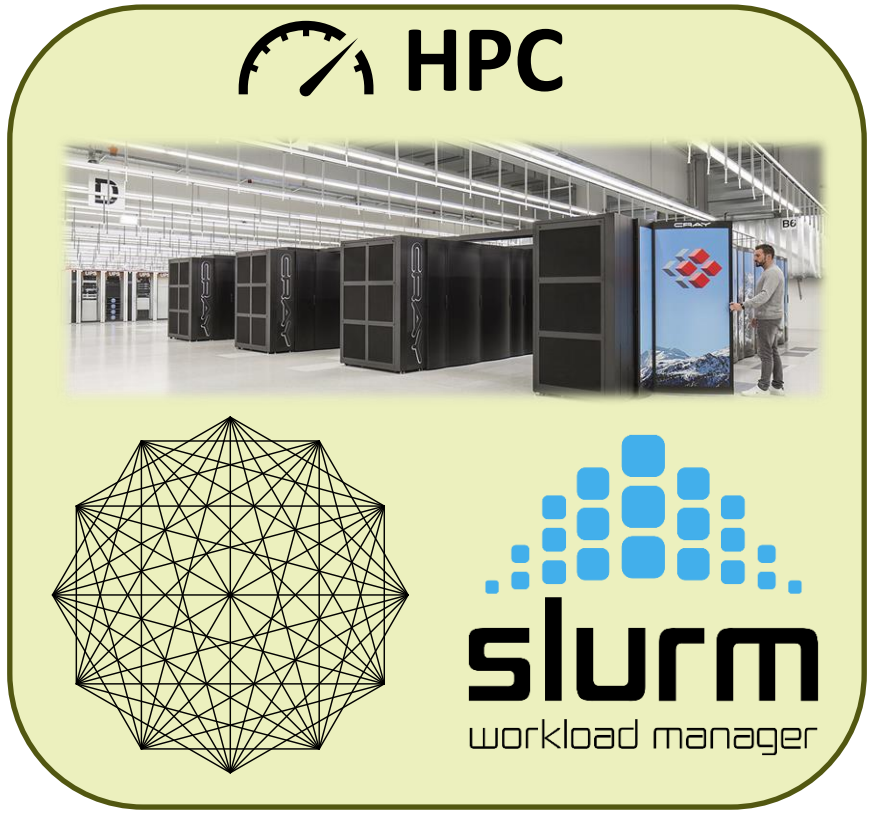
High-Performance Computing Systems



High-Performance Computing Systems



High-Performance Computing Systems



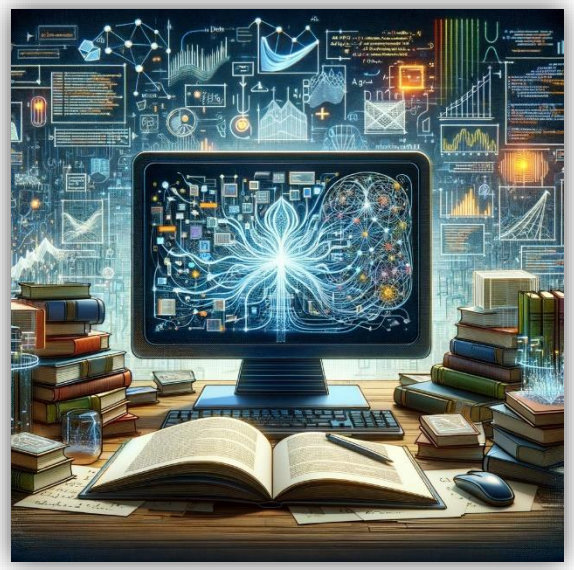
High-Performance Computing Systems

HPC

High-Performance Computing Systems

HPC

slurm workload manager



Scalable Parallel Computing Lab

Scalable Parallel Computing Lab

High-Performance Networking

Scalable Parallel Computing Lab

High-Performance Networking

Programming Models

Scalable Parallel Computing Lab

High-Performance Networking

Programming Models

Machine Learning & LLMs

Scalable Parallel Computing Lab

High-Performance Networking

Programming Models

Machine Learning & LLMs

Cloud & Weather Simulations

Scalable Parallel Computing Lab

High-Performance Networking

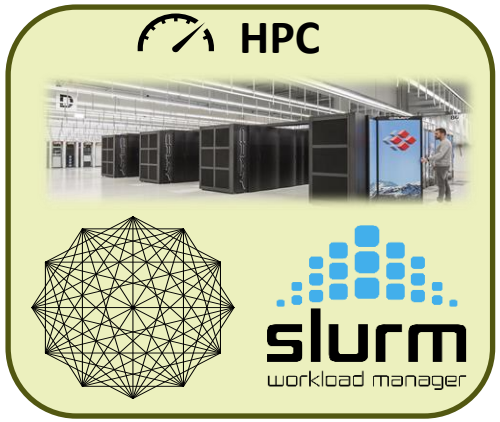
Programming Models

Machine Learning & LLMs

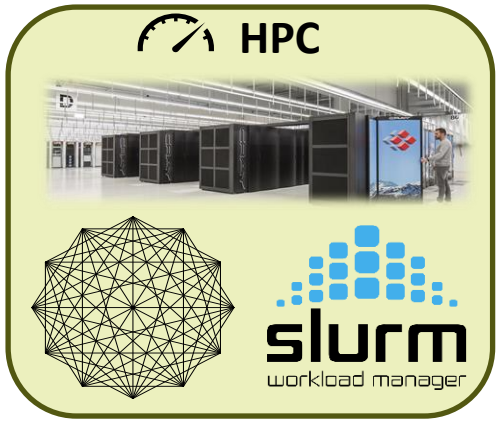
Cloud & Weather Simulations

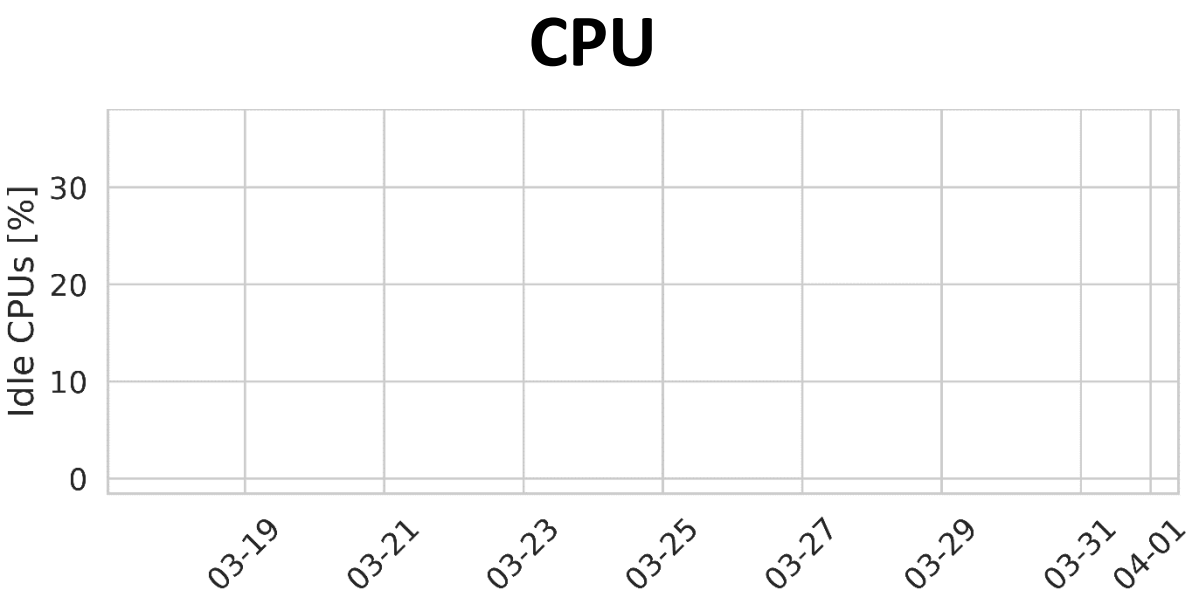
Cloud – HPC Convergence

Tracking Wasted Resources in HPC

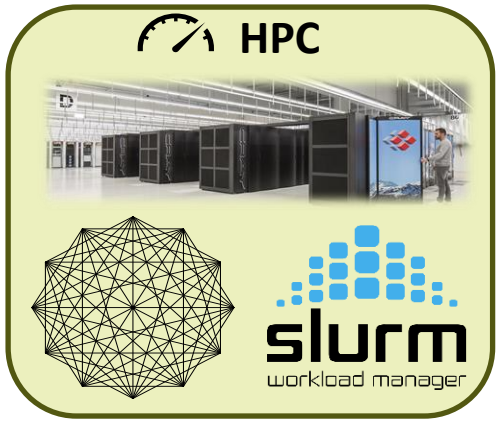


Tracking Wasted Resources in HPC

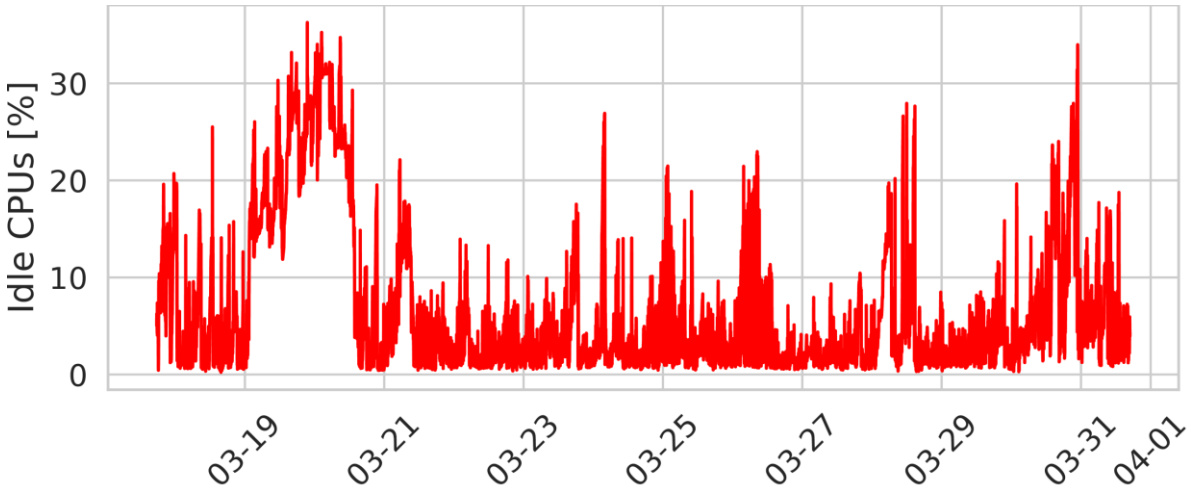




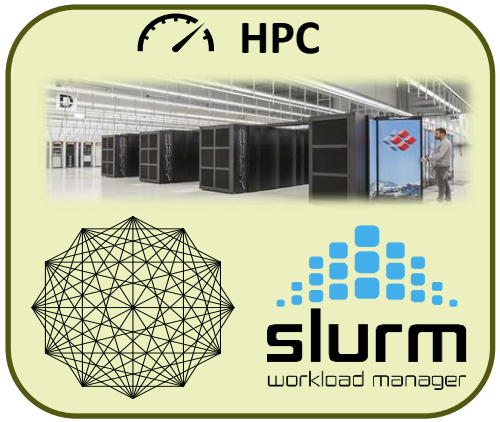
Tracking Wasted Resources in HPC



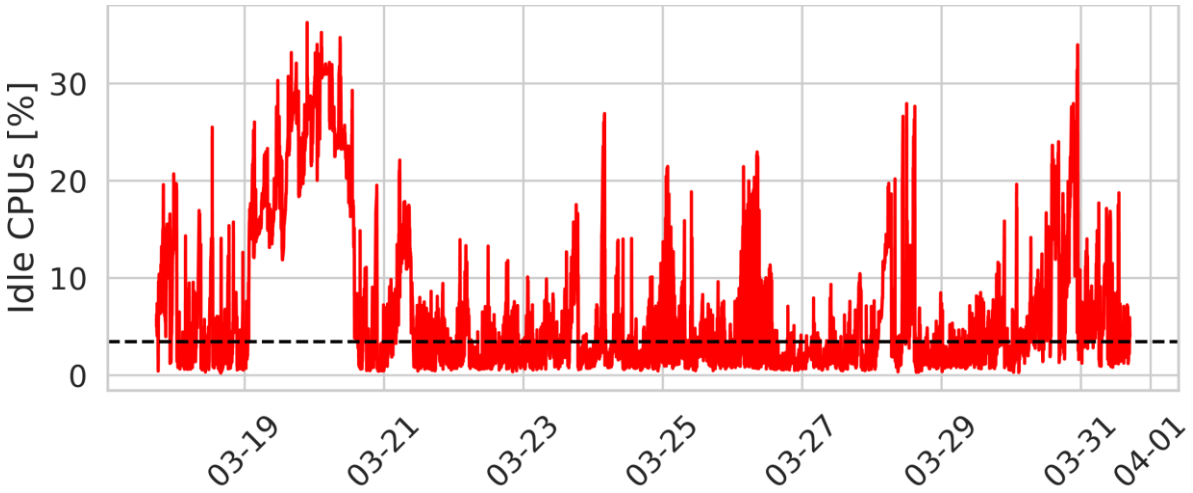
CPU



Tracking Wasted Resources in HPC

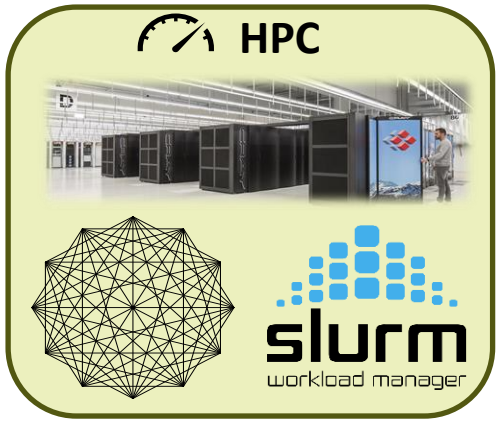


CPU

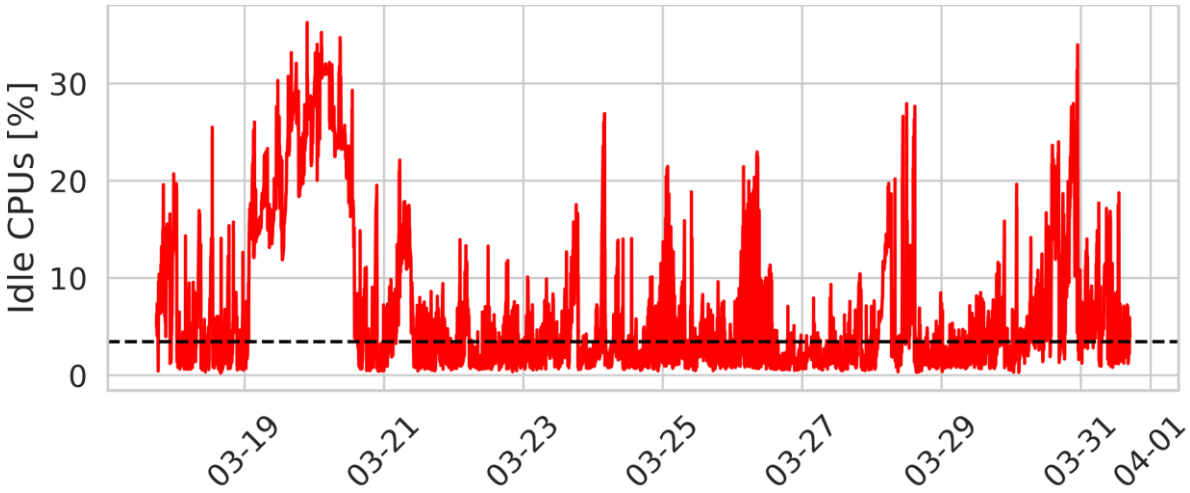


Mean idle CPUs: 6.6%

Tracking Wasted Resources in HPC

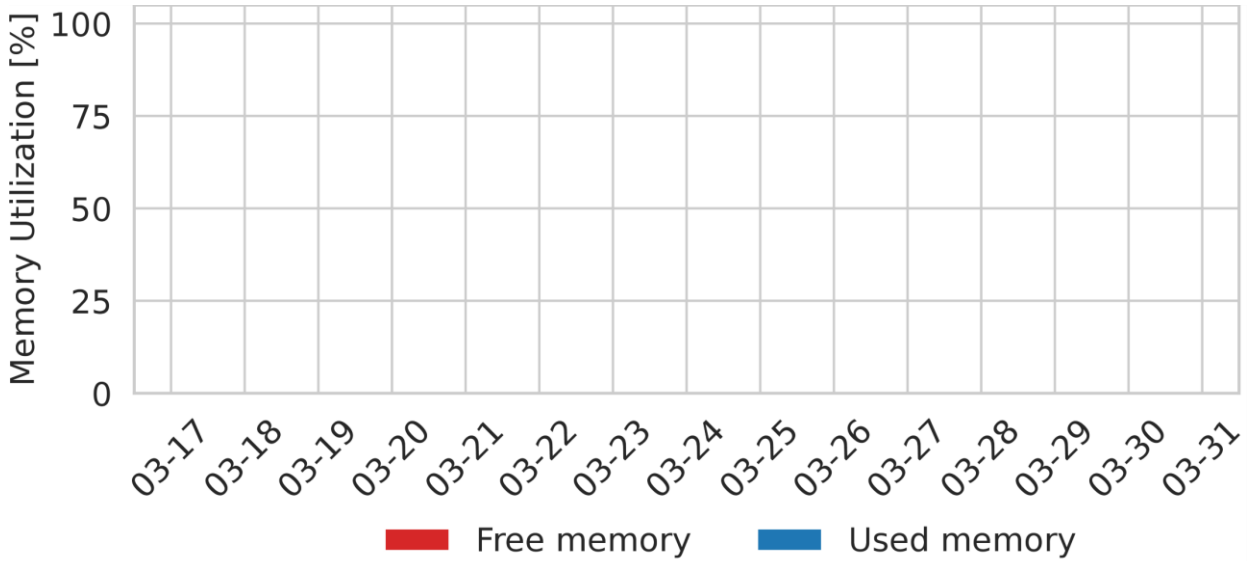


CPU

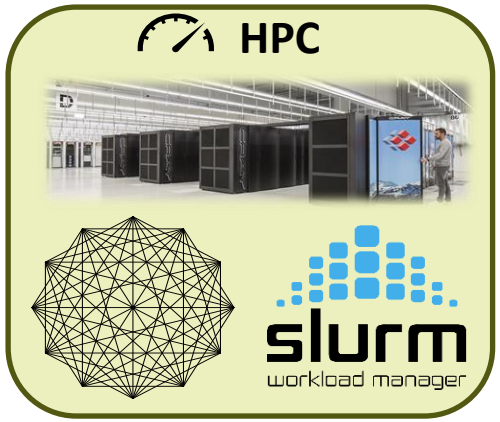


Mean idle CPUs: 6.6%

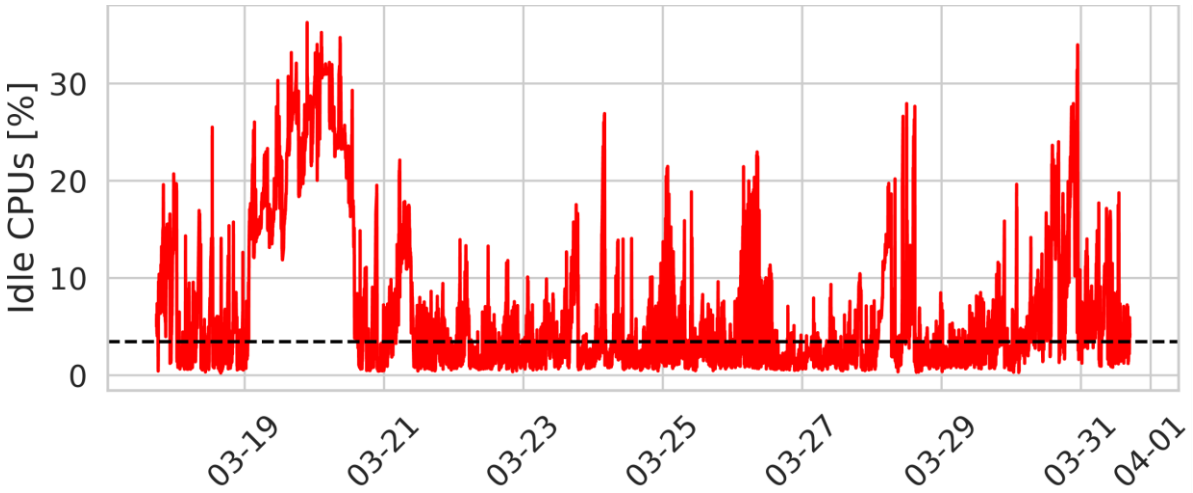
Memory



Tracking Wasted Resources in HPC

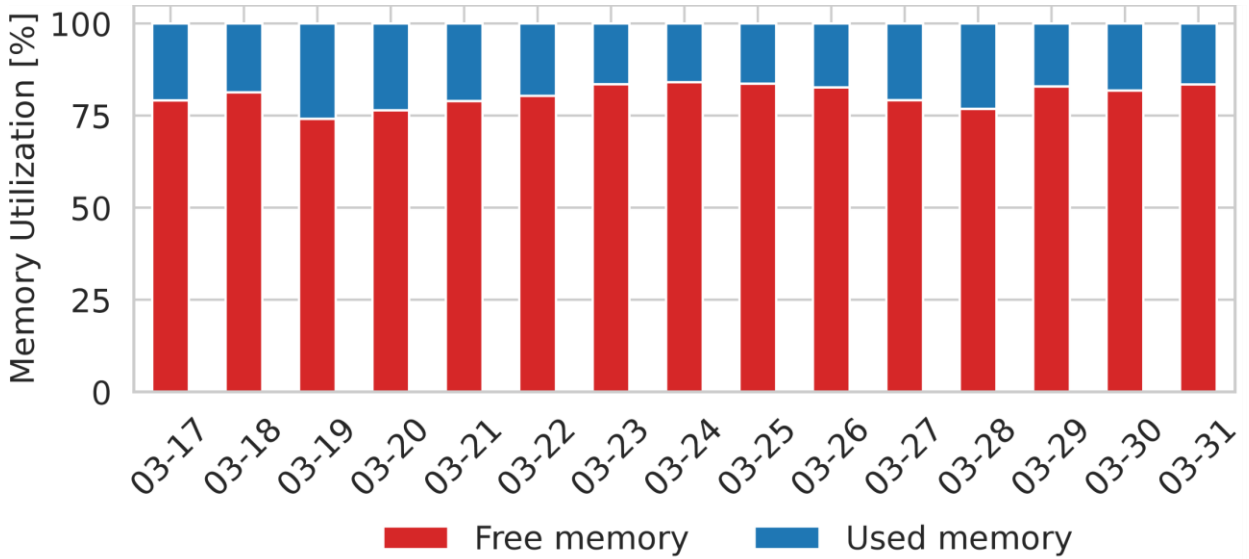


CPU



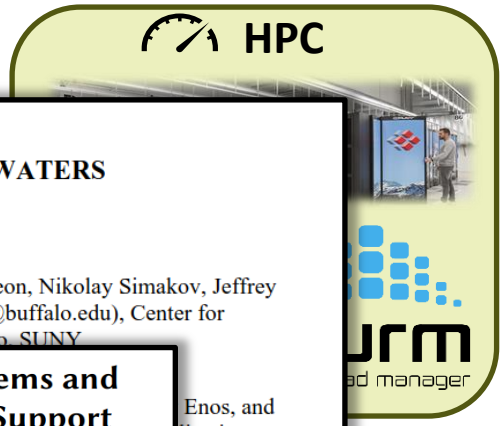
Mean idle CPUs: 6.6%

Memory



Mean free memory: 80.5%

Tracking Wasted Resources in HPC



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, Argonne National Laboratory
Devesh Tiwari, Northeastern University

SC, 2020

FINAL REPORT
WORKLOAD ANALYSIS OF BLUE WATERS
(ACI 1650758)

Matthew D. Jones, Joseph P. White, Martins Innus, Robert L. DeLeon, Nikolay Simakov, Jeffrey T. Palmer, Steven M. Gallo, and Thomas R. Furlani (furlani@buffalo.edu), Center for Computational Research, University at Buffalo, SUNY

Enos, and
lications

Xiv, 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

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 YE
LARRY D
KEREN
JOHN SH

Comprehensive Workload Analysis and Modeling of a Petascale Supercomputer

Haihang You[1] and Hao Zhang[2]

[1] National Institute for Computational Sciences, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA
[2] Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, TN 37996, USA

JSSPP, 2012

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 Liv
Lab
Kathleen Shoga, Shoga1@llnl.gov, Lawrence Livermore National Laboratory, USA
Matthew Legendre, legendre1@llnl.gov, Lawrence Livermore National Laboratory, USA
Todd, gamblin, Lawrence Liv
Lab

ME

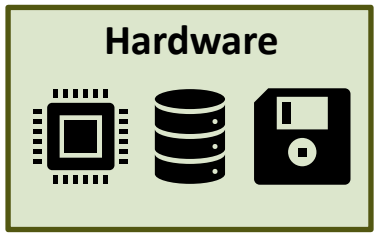
Learning from Five-year Resource-Utilization Data of Titan System

Feiyi Wang*, Sarp Oral†, Satyabrata Sen ‡ and Neena Imam§
Oak Ridge National Laboratory

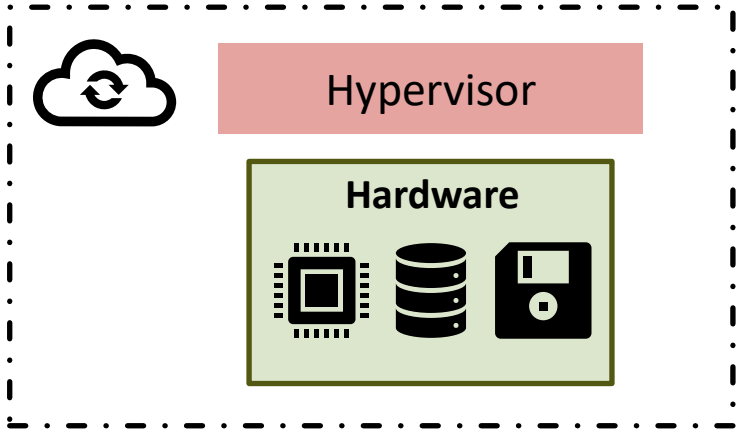
CLUSTER, 2019

Serverless as a Way Forward

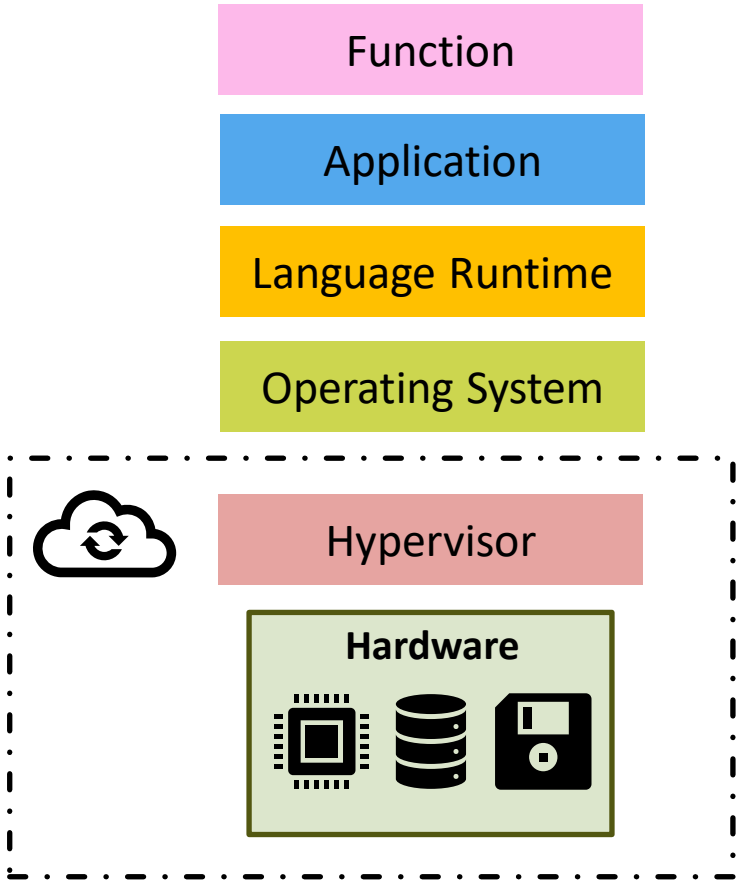
Serverless as a Way Forward



Serverless as a Way Forward

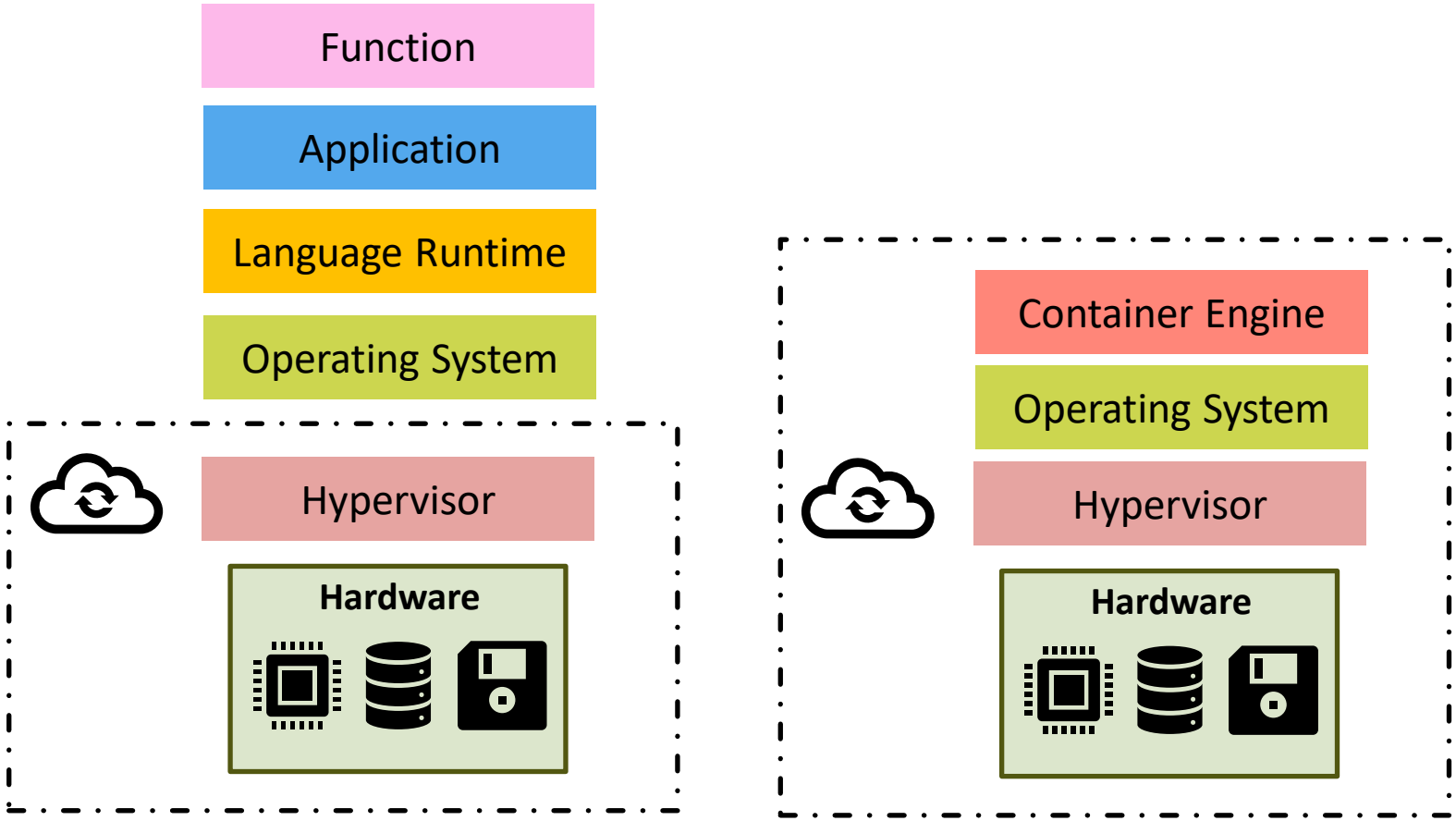


Serverless as a Way Forward

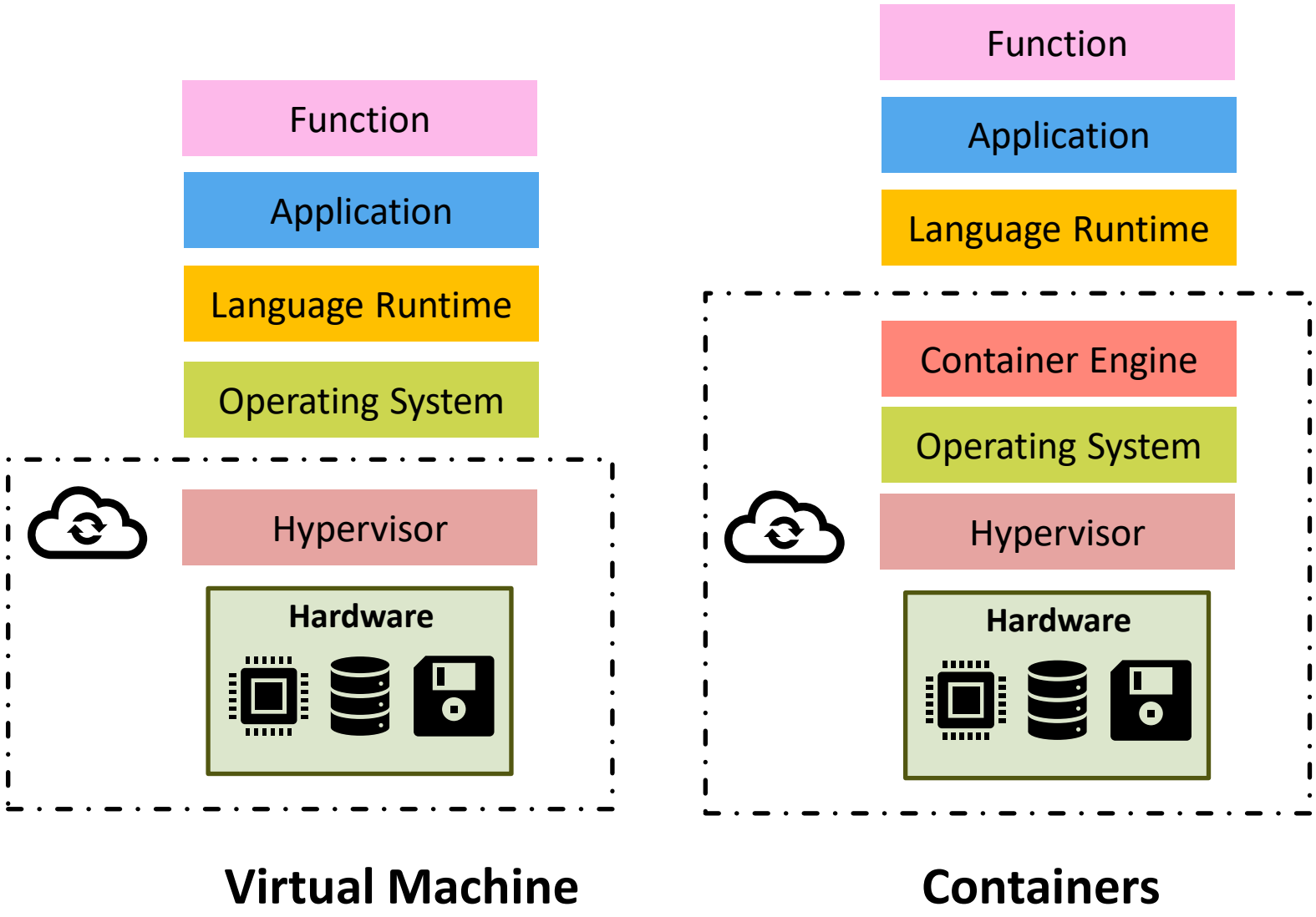


Virtual Machine

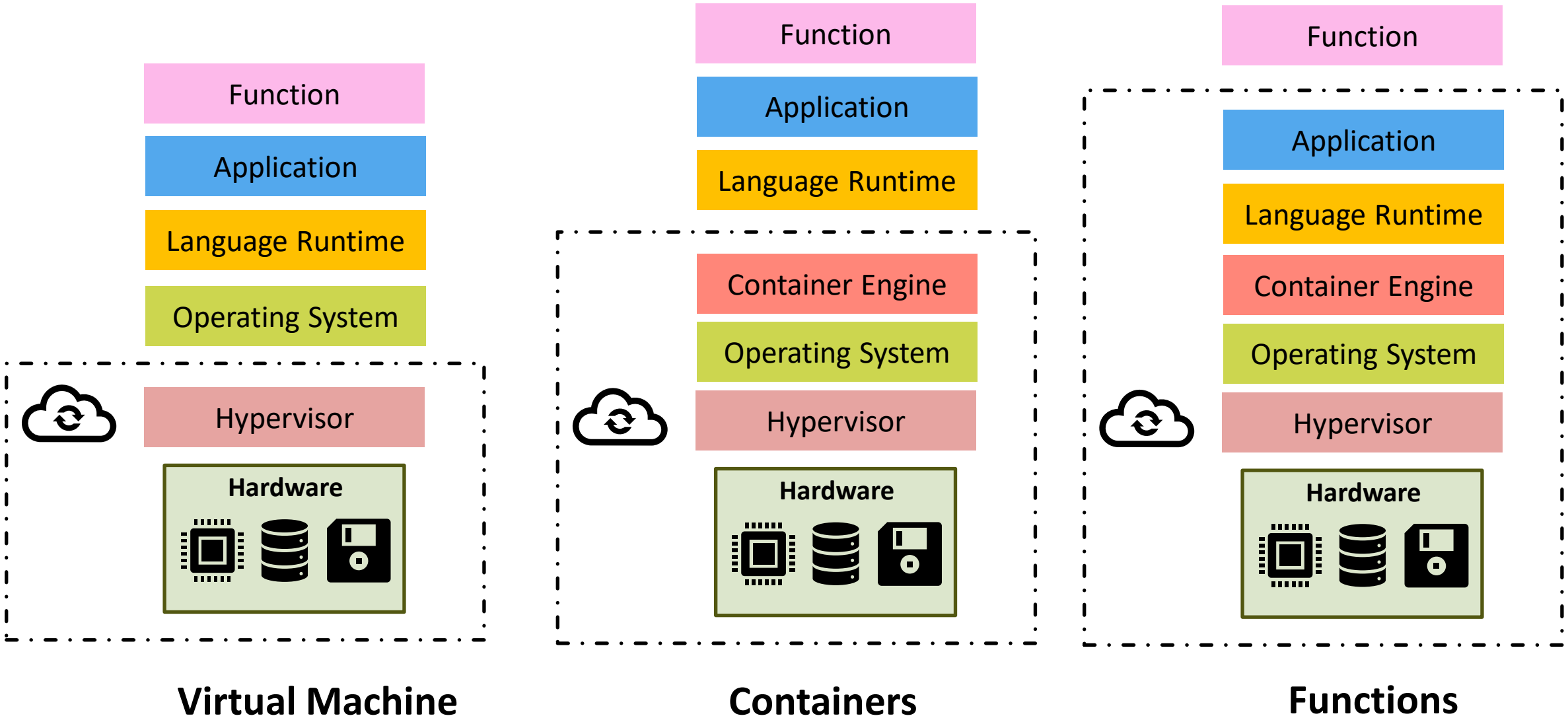
Serverless as a Way Forward



Serverless as a Way Forward



Serverless as a Way Forward



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

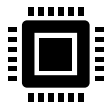
How does Function-as-a-Service (FaaS) work?

```
def handler_function(req: dict, context: dict):

    model = cloud_storage.download_model()

    input = parse_input(req['payload'])

    output = model.inference(input)

    return output
```

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

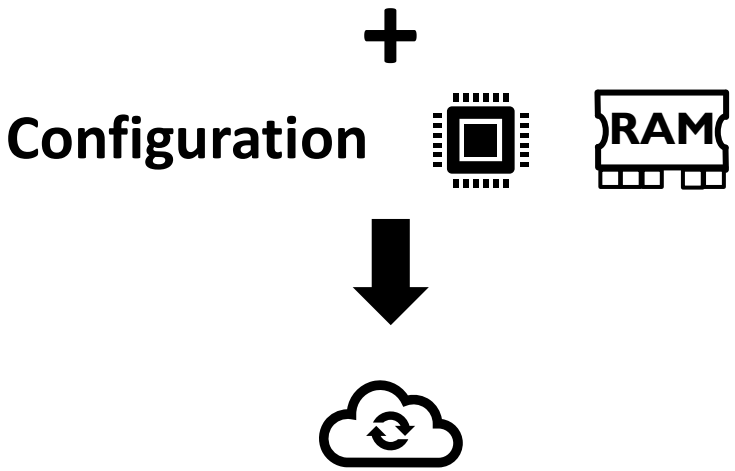
```python
def handler_function(req: dict, context: dict):

    model = cloud_storage.download_model()

    input = parse_input(req['payload'])

    output = model.inference(input)

    return output
```

+

Configuration  

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

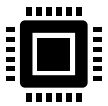
```python
def handler_function(req: dict, context: dict):

    model = cloud_storage.download_model()

    input = parse_input(req['payload'])

    output = model.inference(input)

    return output
```

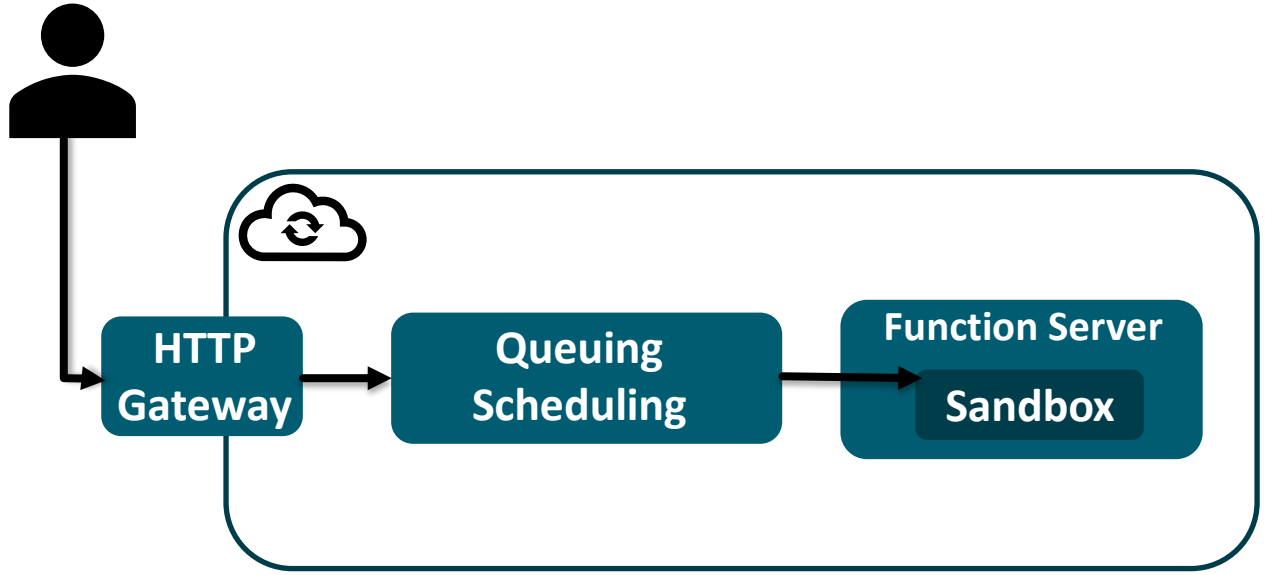


How does Function-as-a-Service (FaaS) work?

```
def handler_function(req: dict, context: dict):

    model = cloud_storage.download_model()

    input = parse_input(req['payload'])

    output = model.inference(input)

    return output
```

+

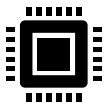
Configuration  

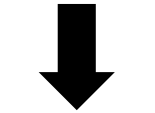


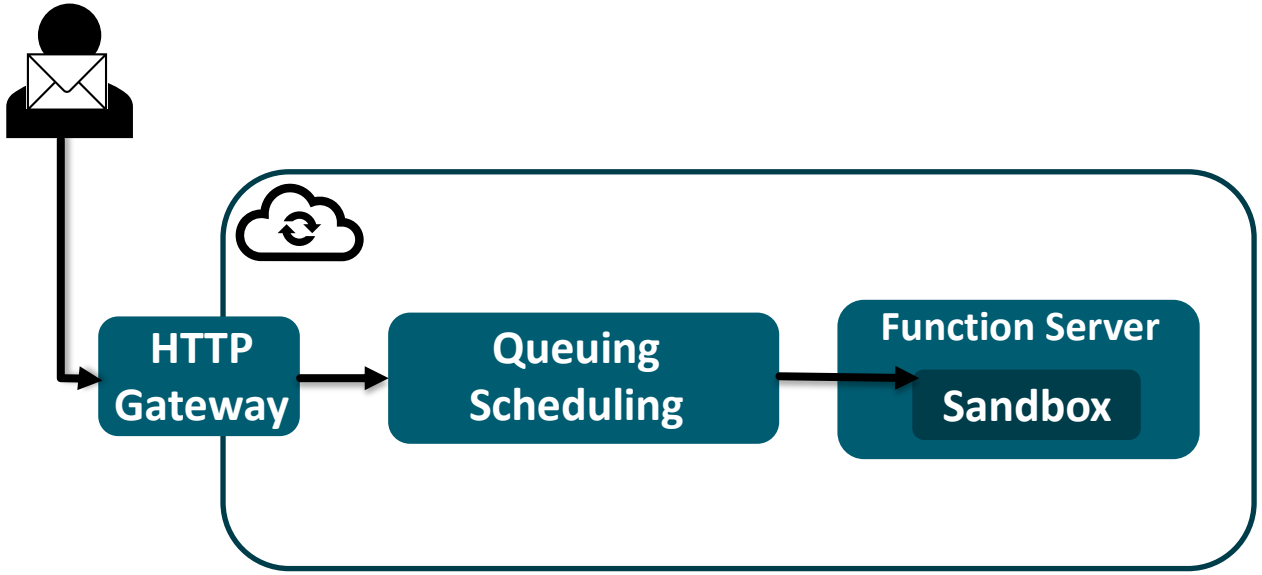
How does Function-as-a-Service (FaaS) work?

```
def handler_function(req: dict, context: dict):

    model = cloud_storage.download_model()

    input = parse_input(req['payload'])

    output = model.inference(input)

    return output
```

+

Configuration  

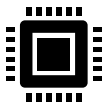


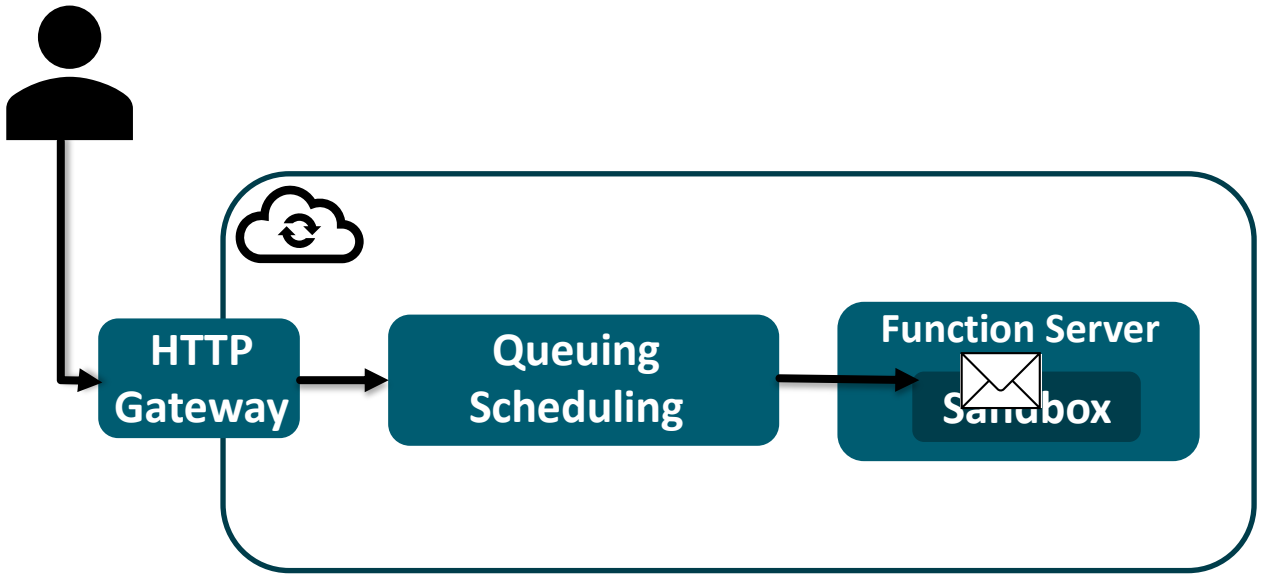


How does Function-as-a-Service (FaaS) work?

```
def handler_function(req: dict, context: dict):

    model = cloud_storage.download_model()

    input = parse_input(req['payload'])

    output = model.inference(input)

    return output
```

+

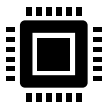
Configuration  

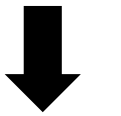


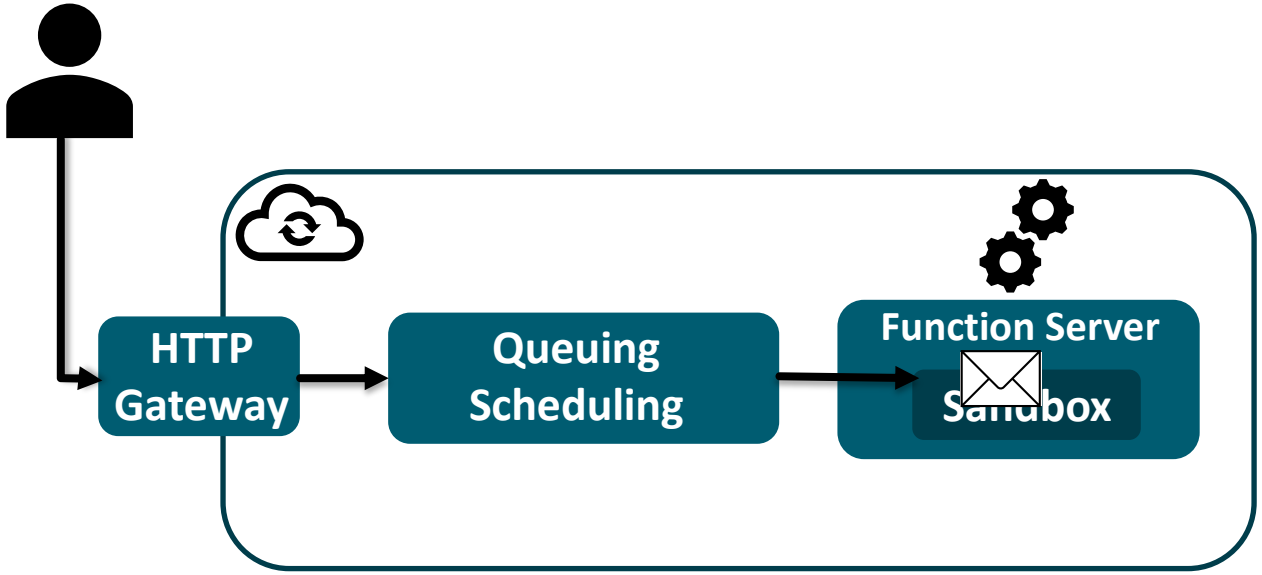
How does Function-as-a-Service (FaaS) work?

```
def handler_function(req: dict, context: dict):

    model = cloud_storage.download_model()

    input = parse_input(req['payload'])

    output = model.inference(input)

    return output
```

+

Configuration  

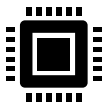


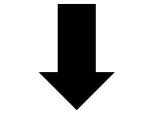


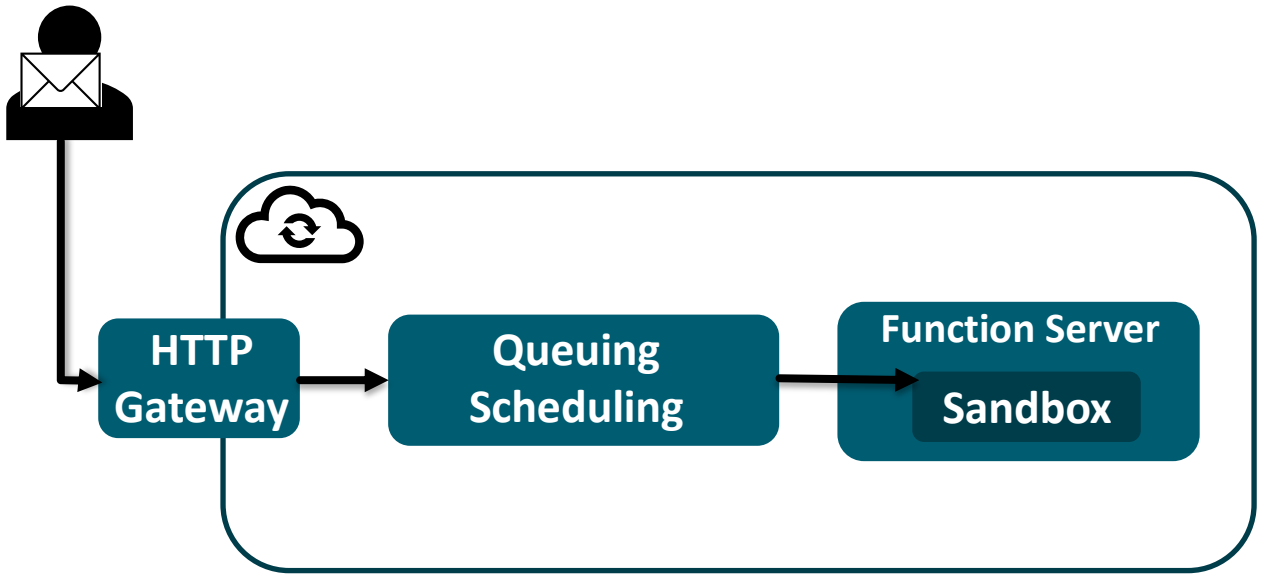
How does Function-as-a-Service (FaaS) work?

```
def handler_function(req: dict, context: dict):

    model = cloud_storage.download_model()

    input = parse_input(req['payload'])

    output = model.inference(input)

    return output
```

+

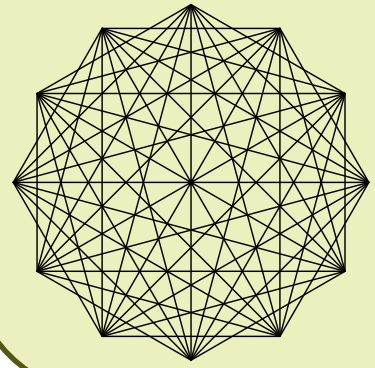
Configuration  





Convergence of HPC and Cloud

HPC

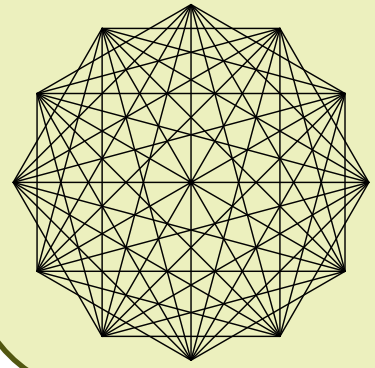


slurm workload manager

Cloud

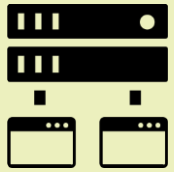
Convergence of HPC and Cloud

HPC



slurm
workload manager

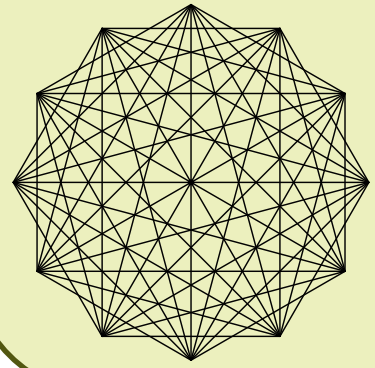
Cloud



Virtualization

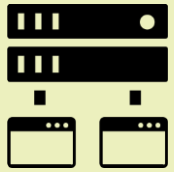
Convergence of HPC and Cloud

HPC



slurm workload manager

Cloud

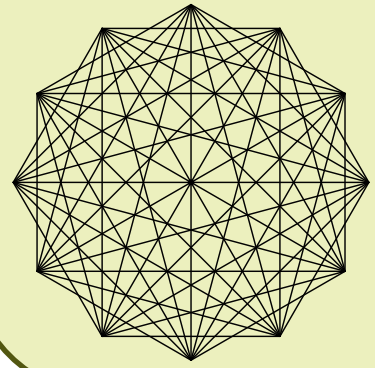


Virtualization

Containers

Convergence of HPC and Cloud

HPC



slurm workload manager

Cloud

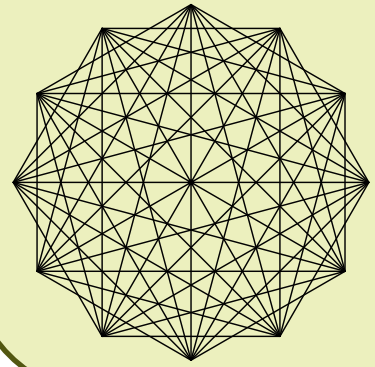
Virtualization

Containers

Pay-as-you-go

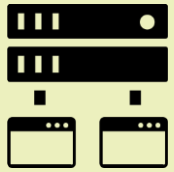
Convergence of HPC and Cloud

HPC



slurm workload manager

Cloud



Virtualization

Containers

Pay-as-you-go

Multi-tenancy

Convergence of HPC and Cloud

HPC

slurm workload manager

Cloud

Virtualization

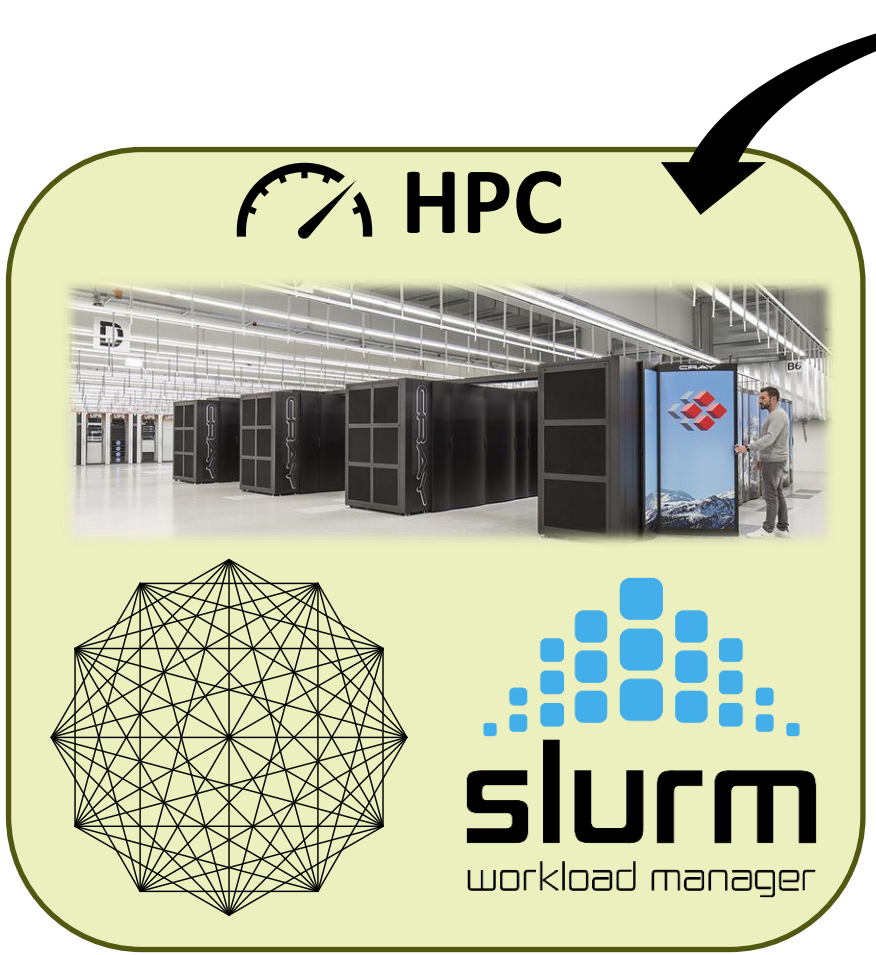
Containers

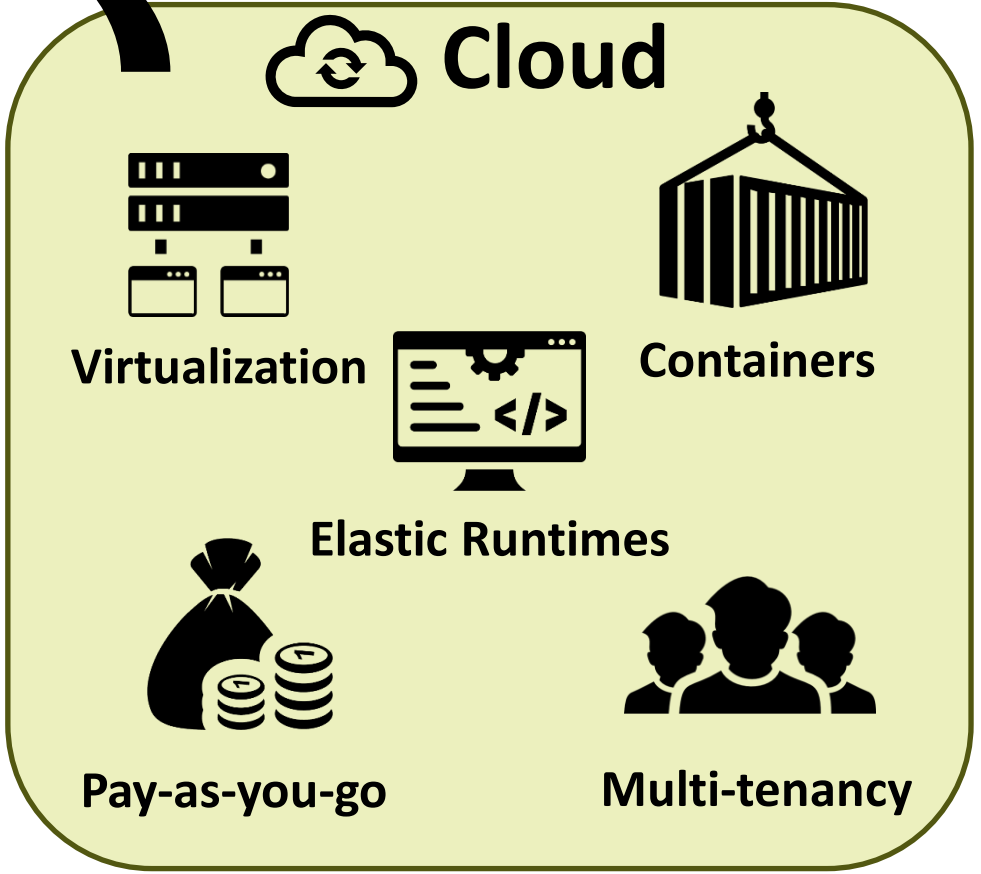
Elastic Runtimes

Pay-as-you-go

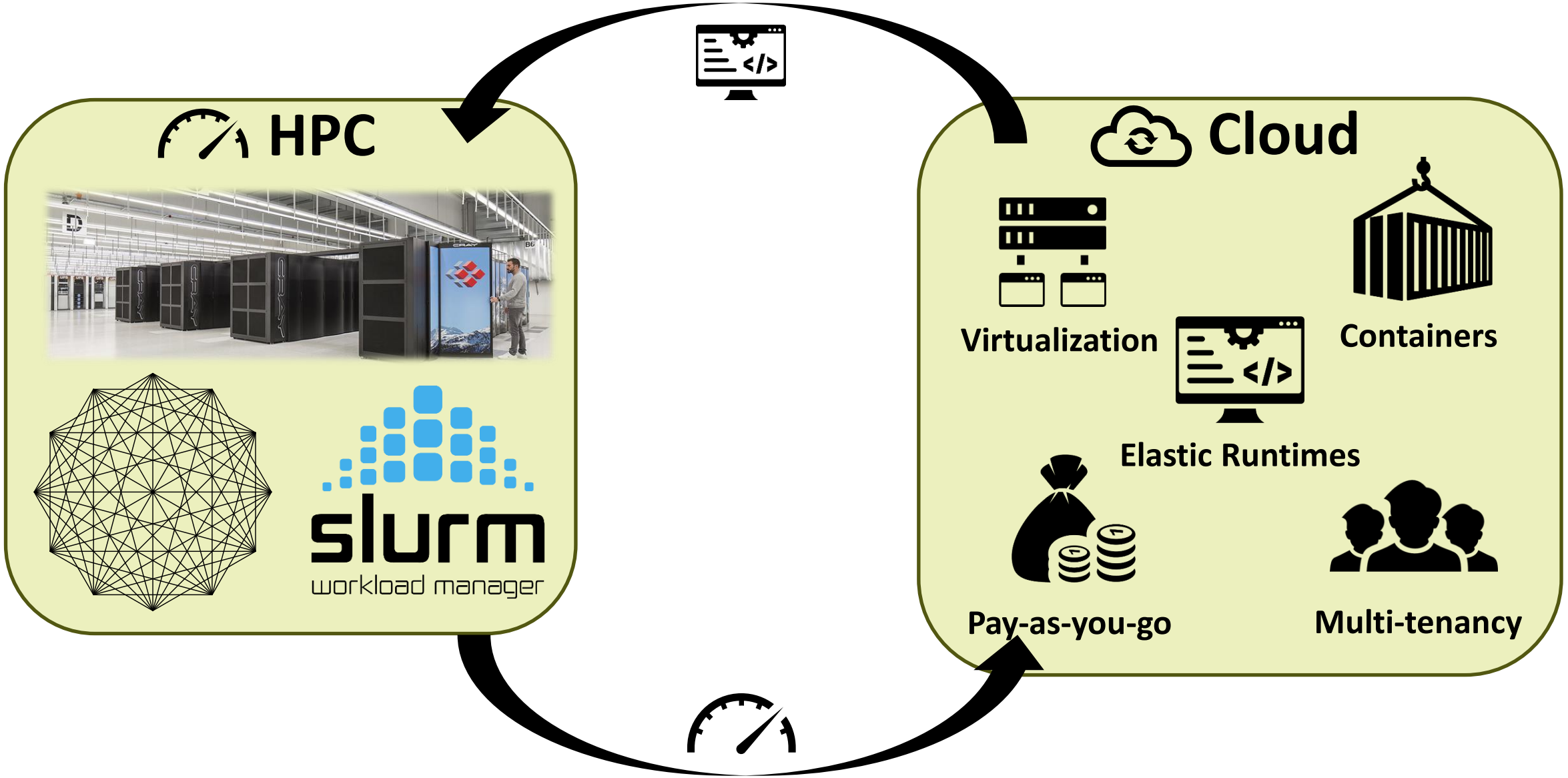
Multi-tenancy

Convergence of HPC and Cloud

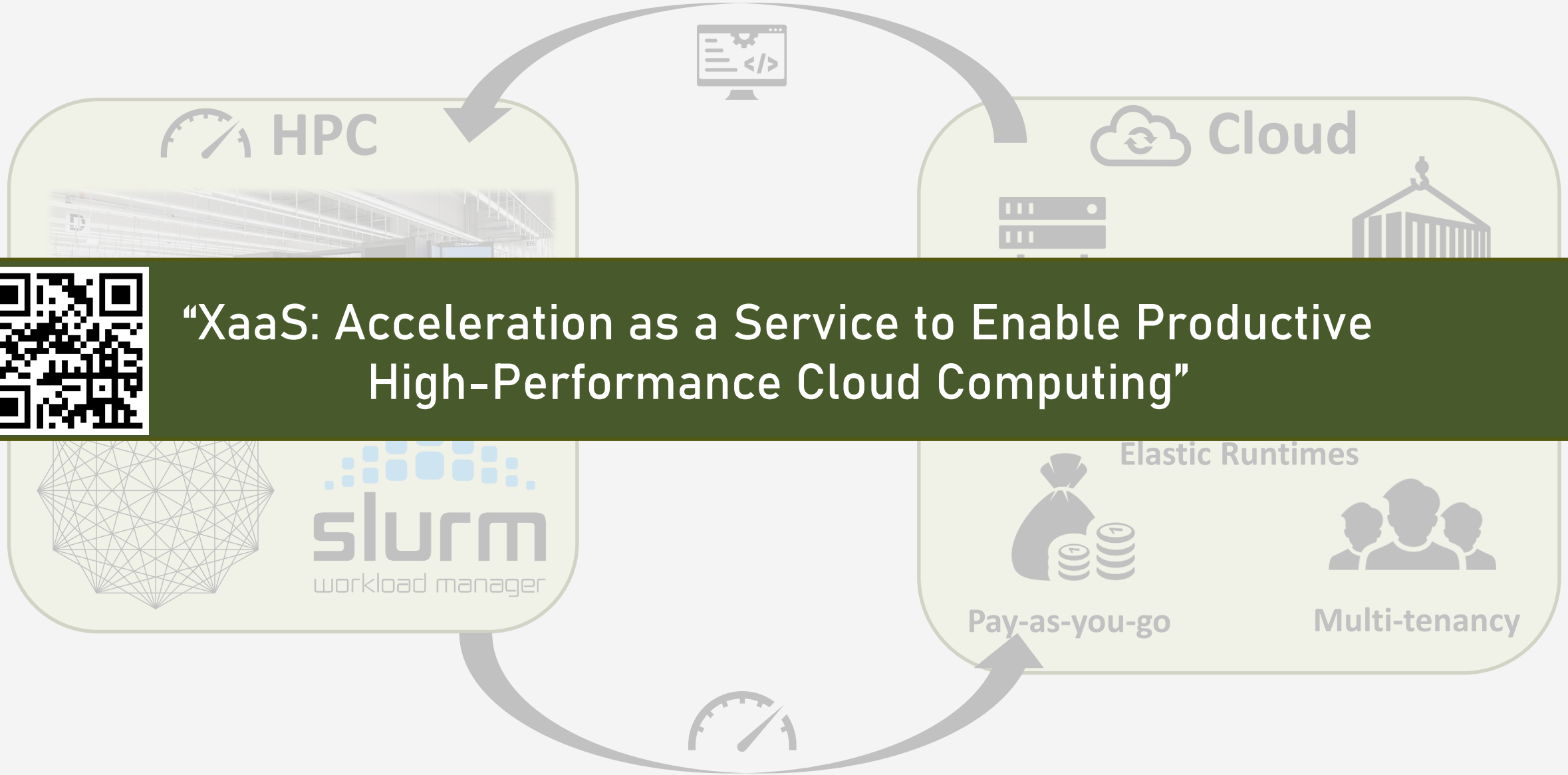




Convergence of HPC and Cloud



Convergence of HPC and Cloud



"XaaS: Acceleration as a Service to Enable Productive High-Performance Cloud Computing"

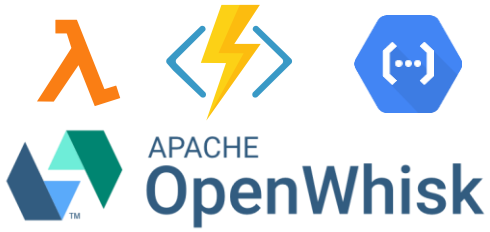
SeBS: The Serverless Benchmark Suite

ACM/IFIP Middleware' 21

SeBS: The Serverless Benchmark Suite

ACM/IFIP Middleware' 21

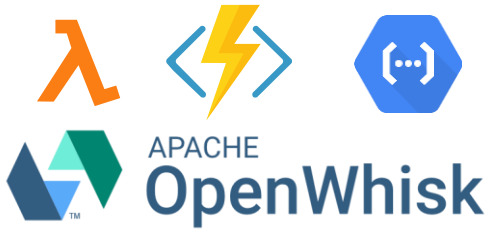
Serverless Platforms



SeBS: The Serverless Benchmark Suite

ACM/IFIP Middleware' 21

Serverless Platforms

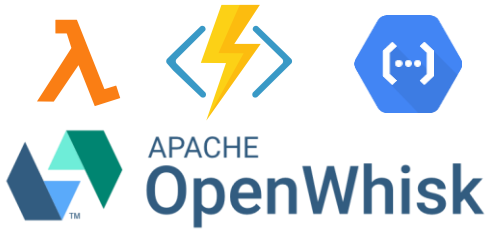


Benchmarks

SeBS: The Serverless Benchmark Suite

ACM/IFIP Middleware' 21

Serverless Platforms



Benchmarks

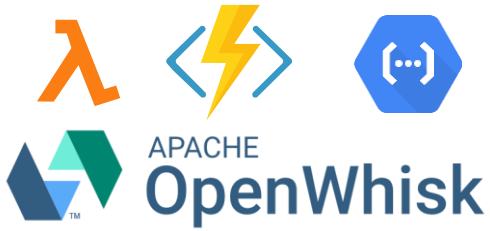
Experiments

Performance & Cost
Invocation Overhead
Container Eviction
Serverless Communication
Serverless Workflows

SeBS: The Serverless Benchmark Suite

ACM/IFIP Middleware' 21

Serverless Platforms

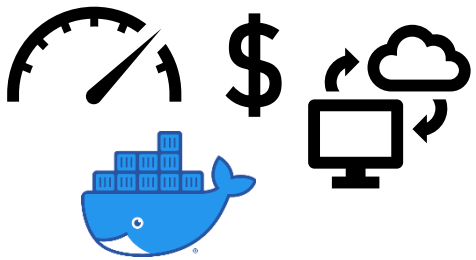


Benchmarks

Experiments

Performance & Cost
Invocation Overhead
Container Eviction
Serverless Communication
Serverless Workflows

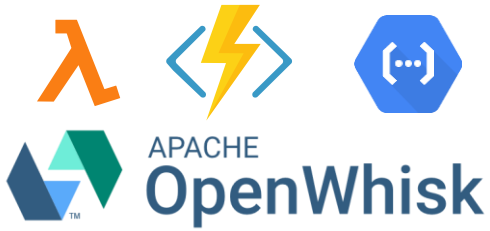
Insights



SeBS: The Serverless Benchmark Suite

ACM/IFIP Middleware' 21

Serverless Platforms

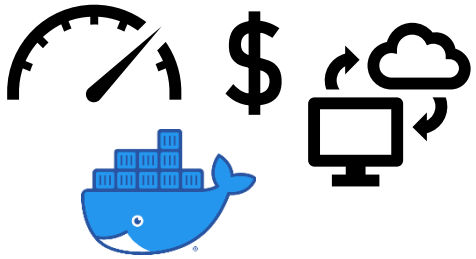


Benchmarks

Experiments

Performance & Cost
Invocation Overhead
Container Eviction
Serverless Communication
Serverless Workflows

Insights

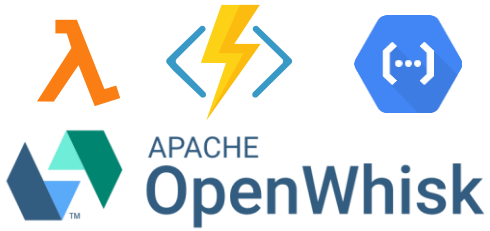


Adoption

SeBS: The Serverless Benchmark Suite

ACM/IFIP Middleware' 21

Serverless Platforms

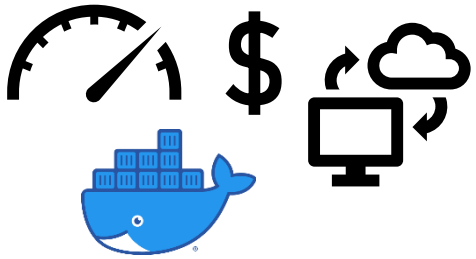


Benchmarks

Experiments

Performance & Cost
Invocation Overhead
Container Eviction
Serverless Communication
Serverless Workflows

Insights



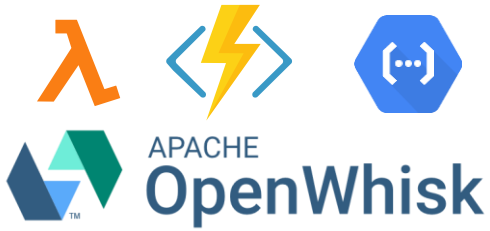
Adoption

118 stars
61 forks
18 contributors

SeBS: The Serverless Benchmark Suite

ACM/IFIP Middleware' 21

Serverless Platforms

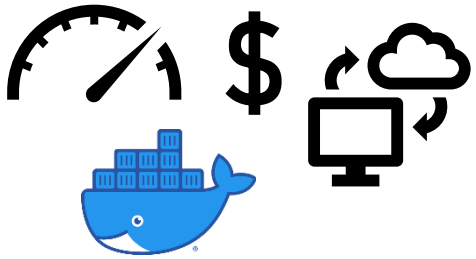


Benchmarks

Experiments

Performance & Cost
Invocation Overhead
Container Eviction
Serverless Communication
Serverless Workflows

Insights



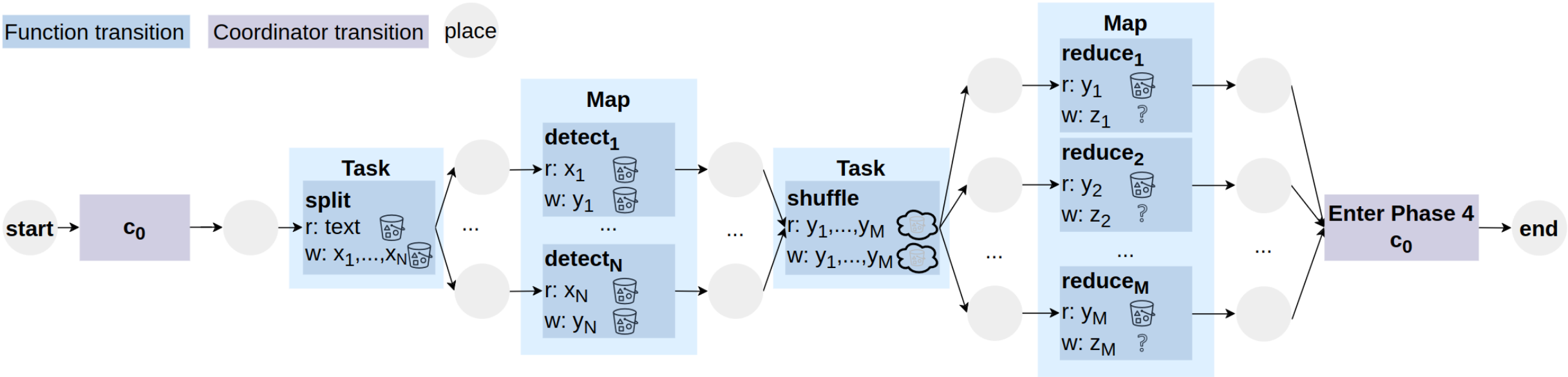
Adoption

118 stars
61 forks
18 contributors

114 citations

SeBS-Flow: The Serverless Benchmark Suite

MSc Thesis
In submission



Idea: Pushing the Serverless Envelope in SeBS

Idea: Pushing the Serverless Envelope in SeBS

Stateful Applications

Idea: Pushing the Serverless Envelope in SeBS

Stateful Applications

I/O-Heavy Workloads

Idea: Pushing the Serverless Envelope in SeBS

Stateful Applications

I/O-Heavy Workloads

Complex Applications

Idea: Pushing the Serverless Envelope in SeBS

Stateful Applications

I/O-Heavy Workloads

Complex Applications

Large Microservices

Idea: Pushing the Serverless Envelope in SeBS

Stateful Applications

I/O-Heavy Workloads

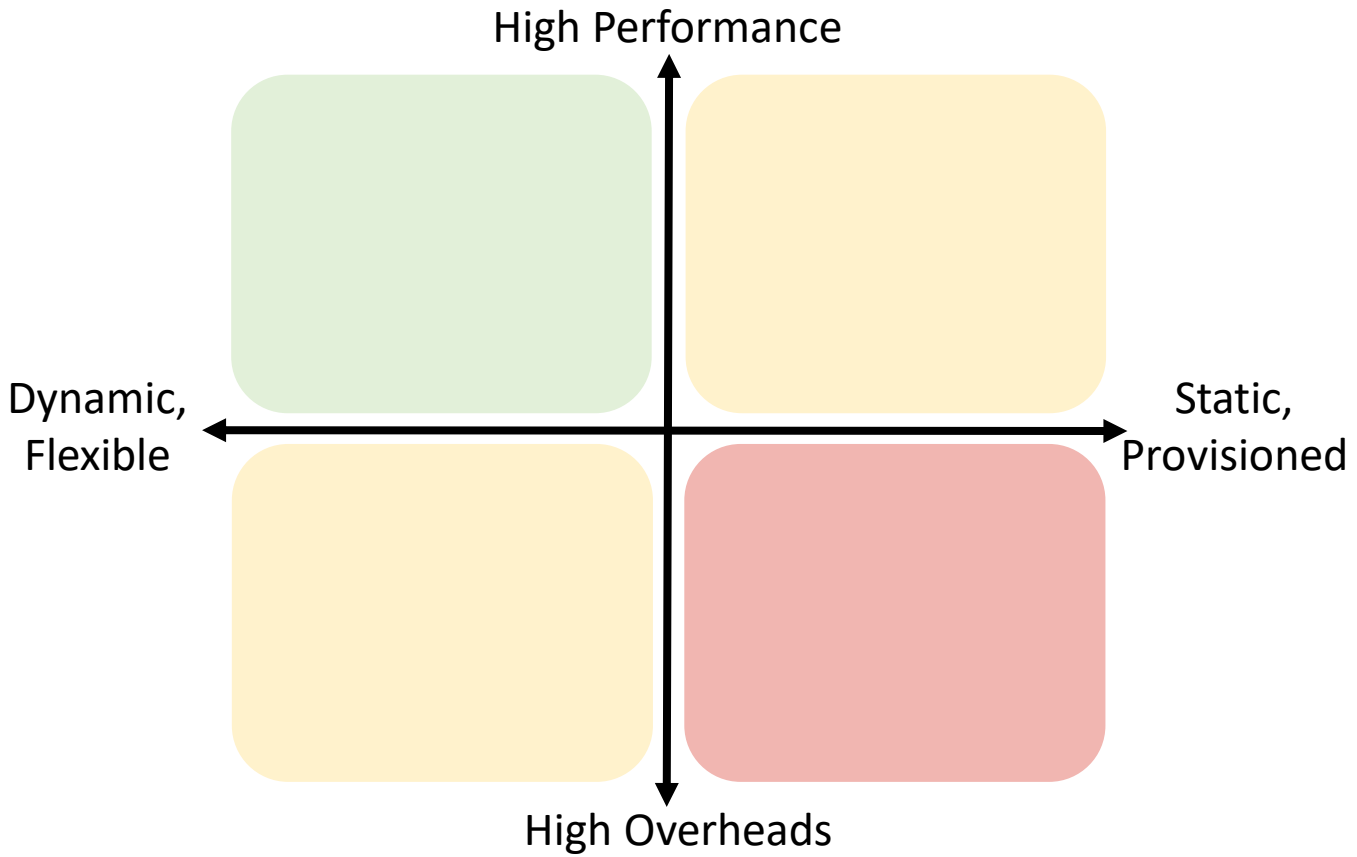
Complex Applications

Large Microservices

Hybrid Applications

Function-as-a-Service for HPC

IEEE IPDPS 2023

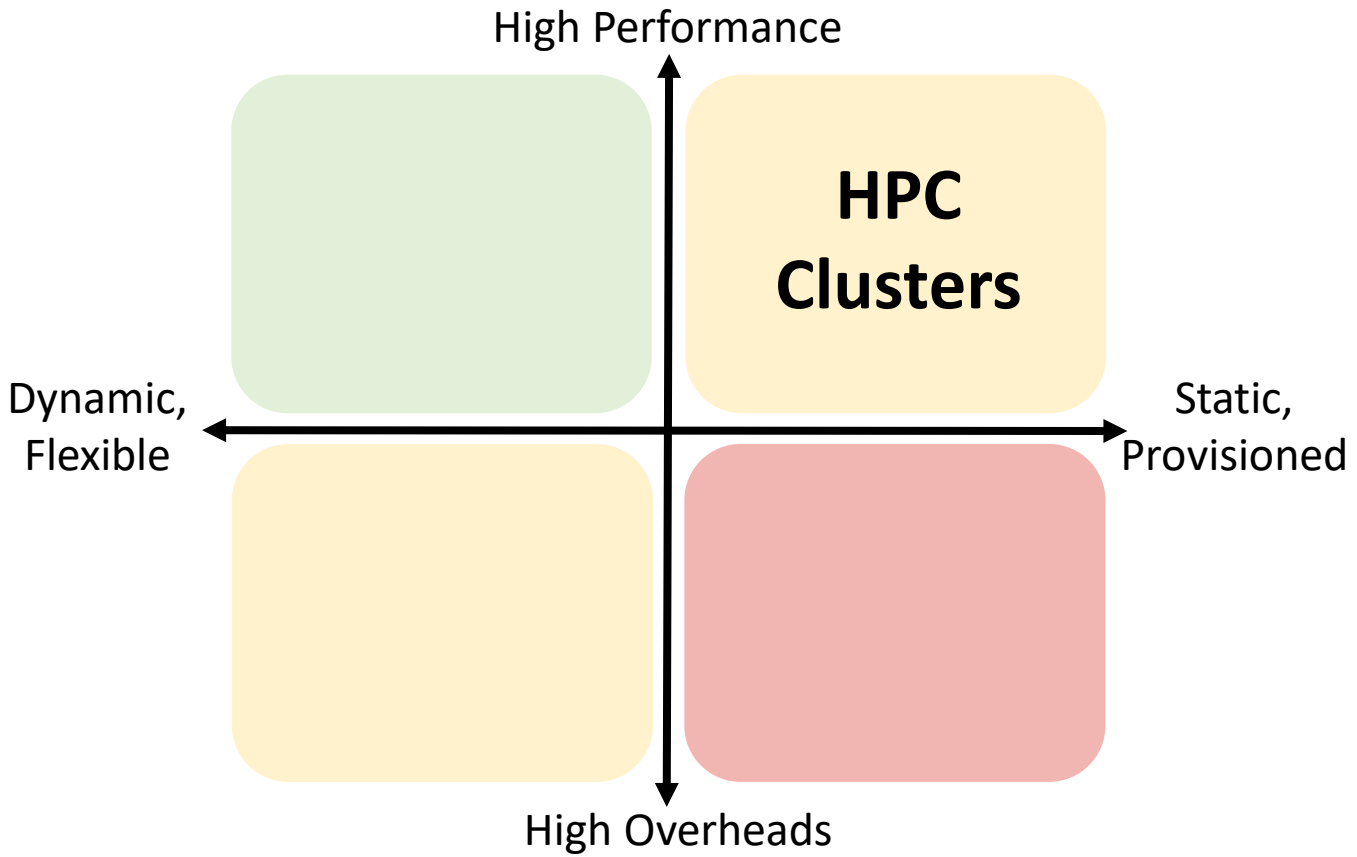


Function-as-a-Service for HPC

IEEE IPDPS 2023

Long-running jobs

Static parallelism

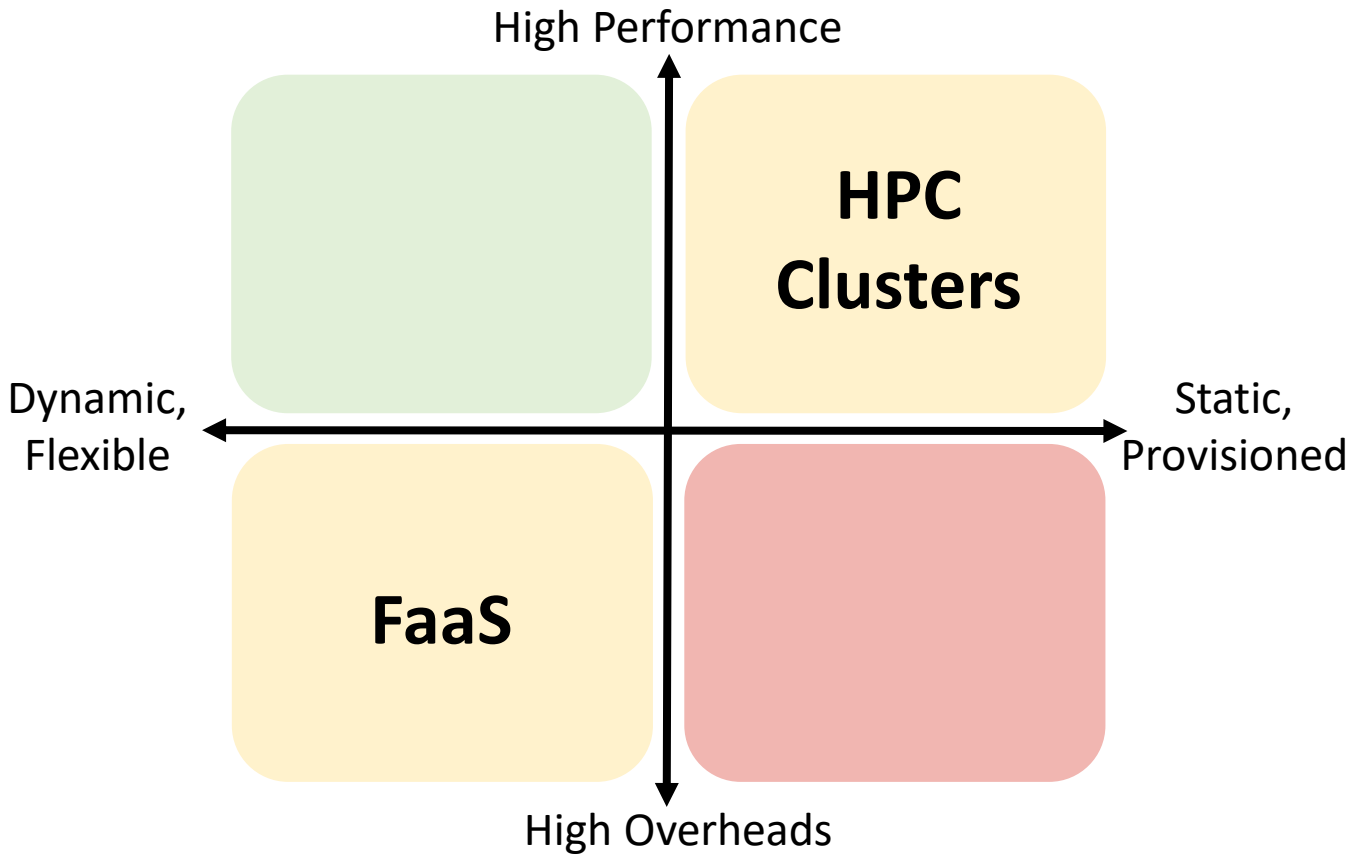


Function-as-a-Service for HPC

IEEE IPDPS 2023

Long-running jobs

Static parallelism



Malleable jobs

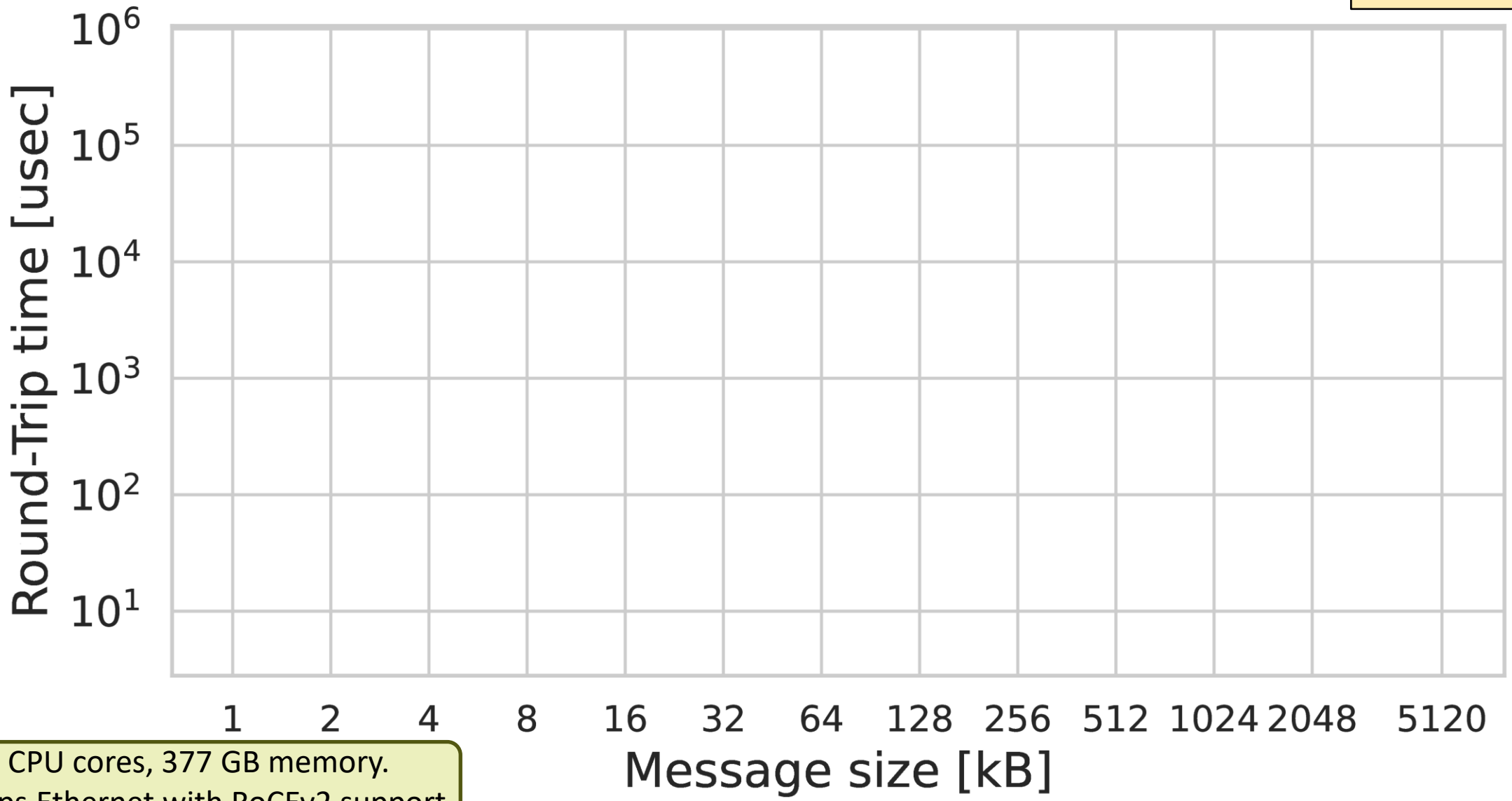
Latency-sensitive jobs

Interactive computations

Dynamic parallelism

How fast are invocations in FaaS?

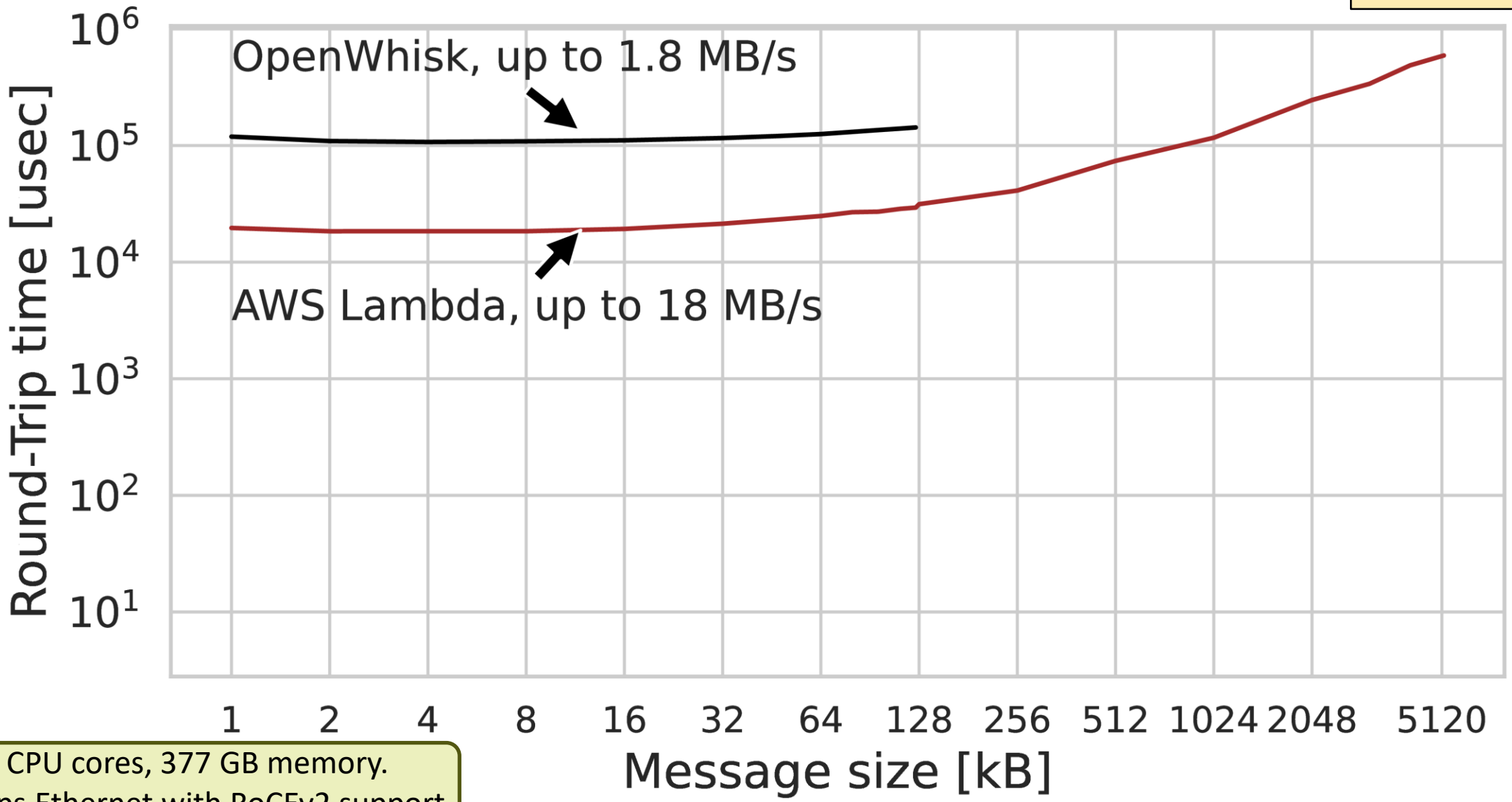
IEEE IPDPS 2023



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

How fast are invocations in FaaS?

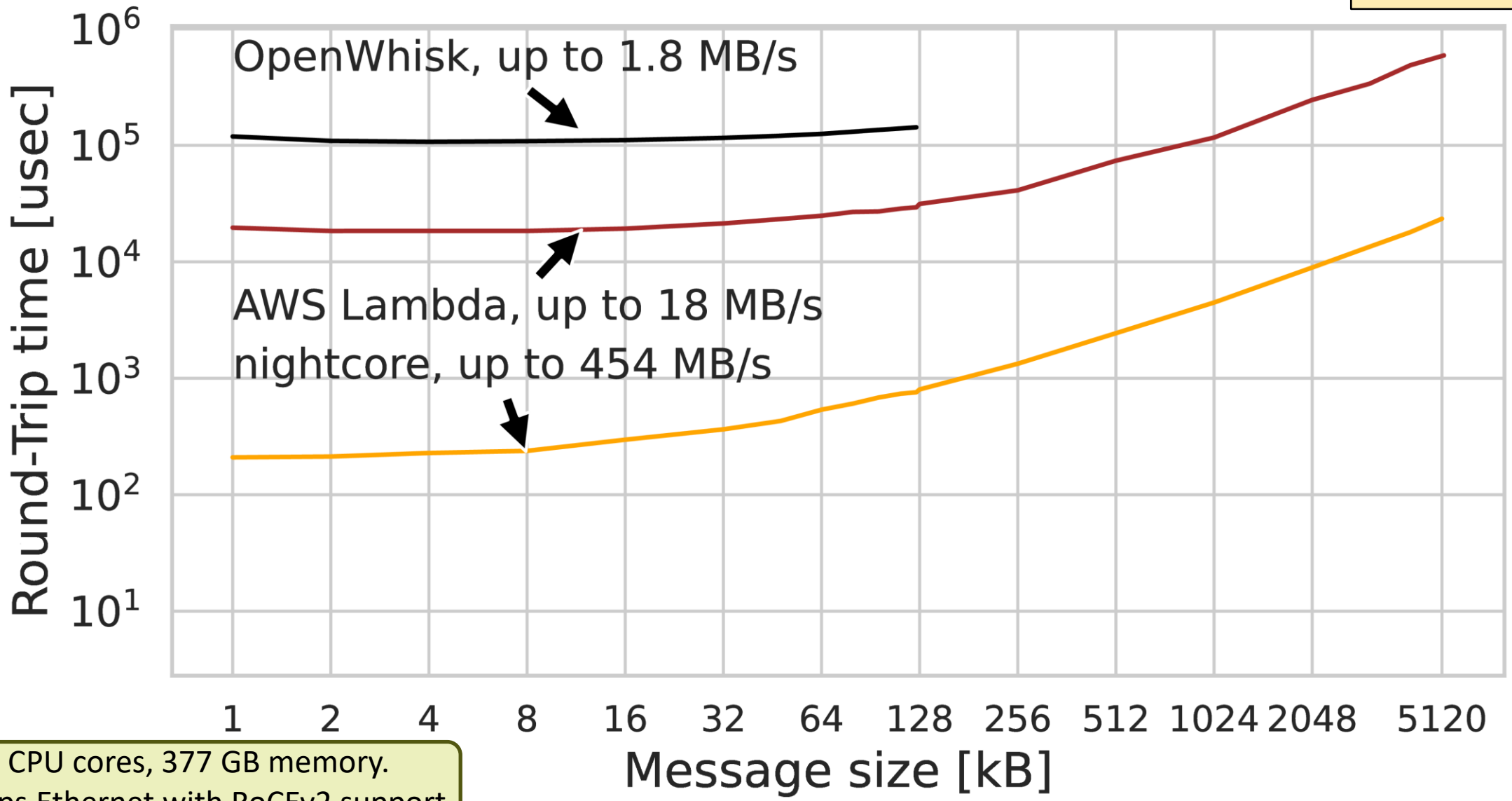
IEEE IPDPS 2023



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

How fast are invocations in FaaS?

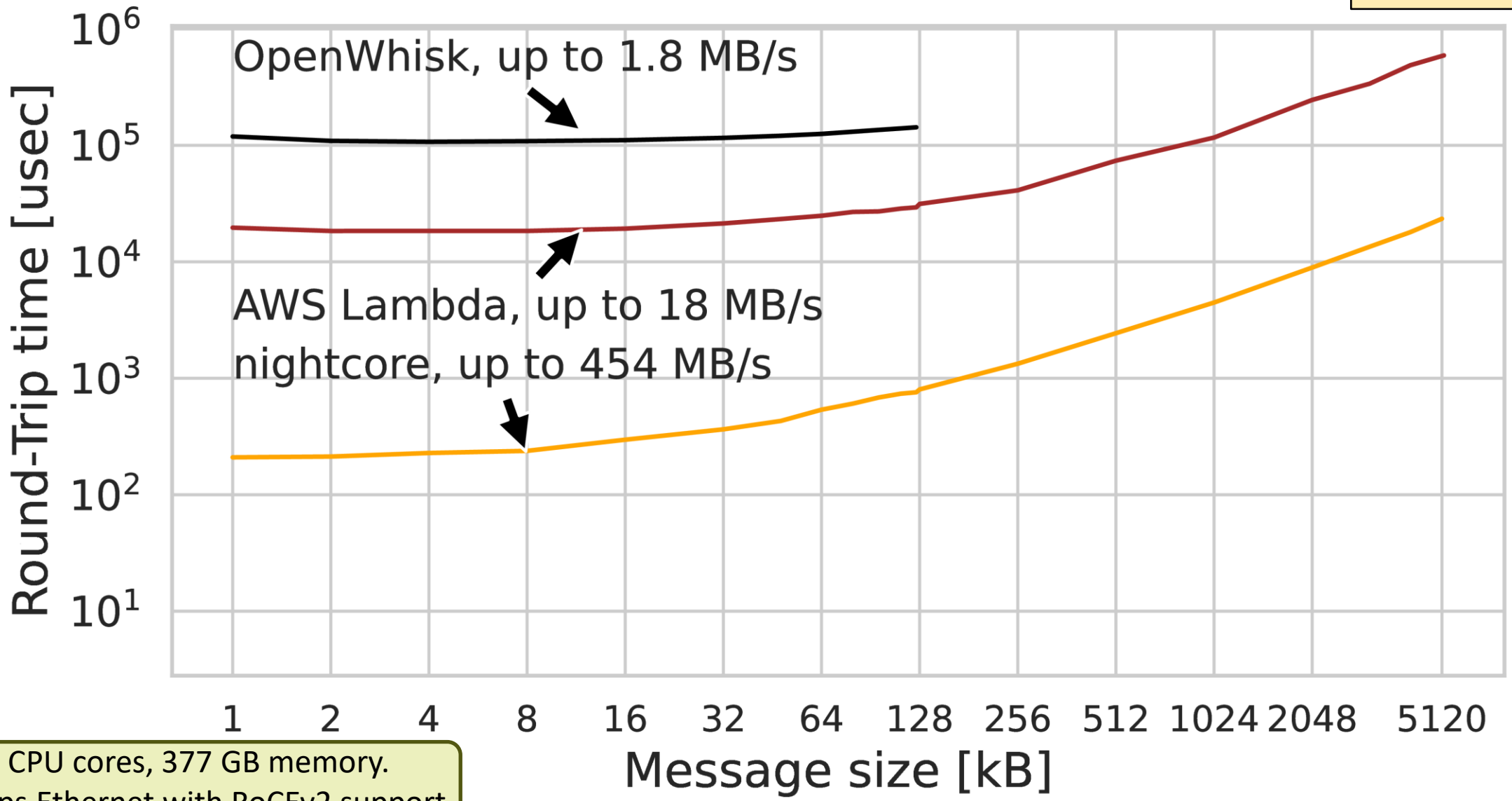
IEEE IPDPS 2023



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

How fast are invocations in FaaS?

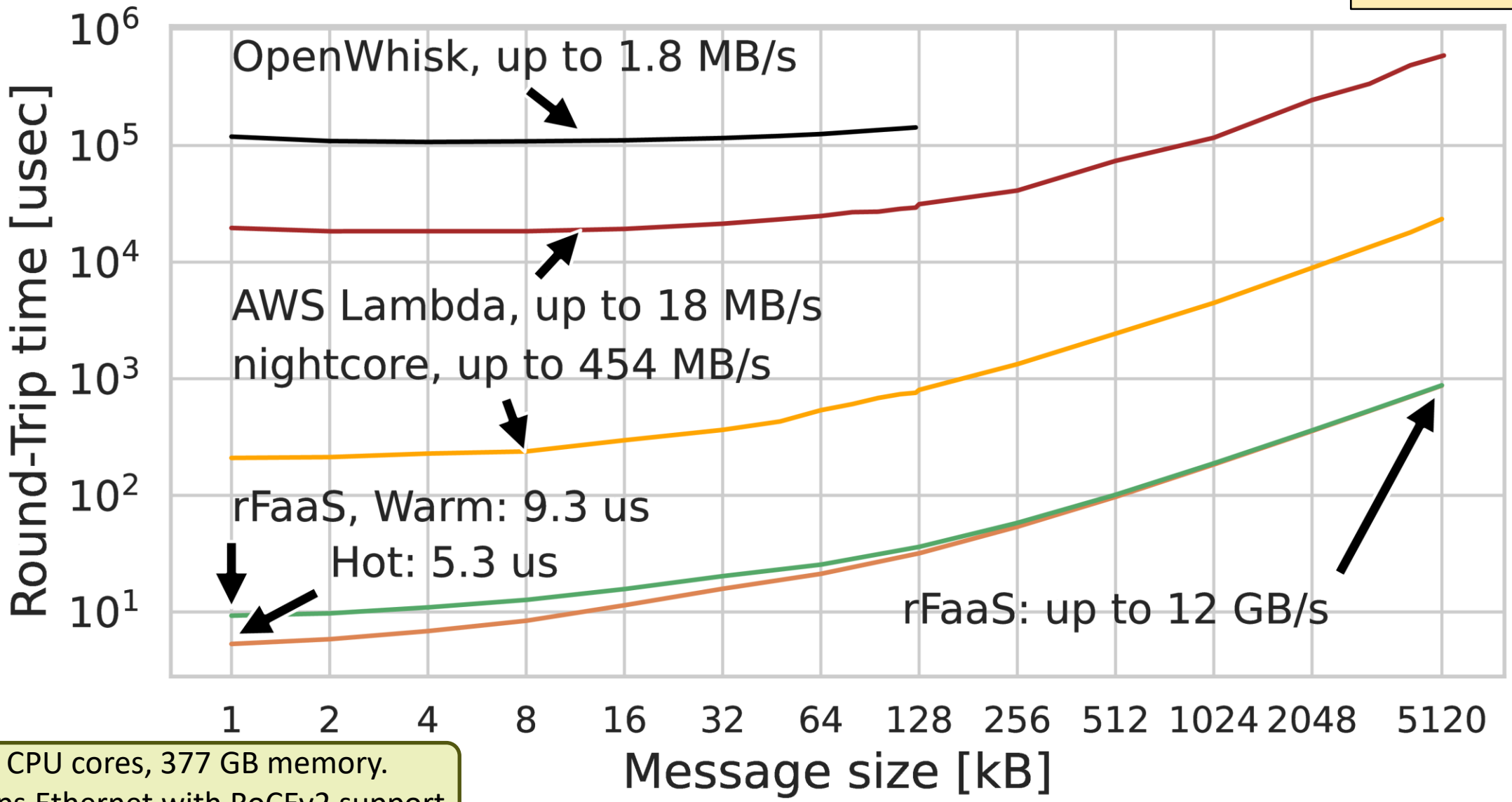
IEEE IPDPS 2023



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

How fast are invocations in FaaS?

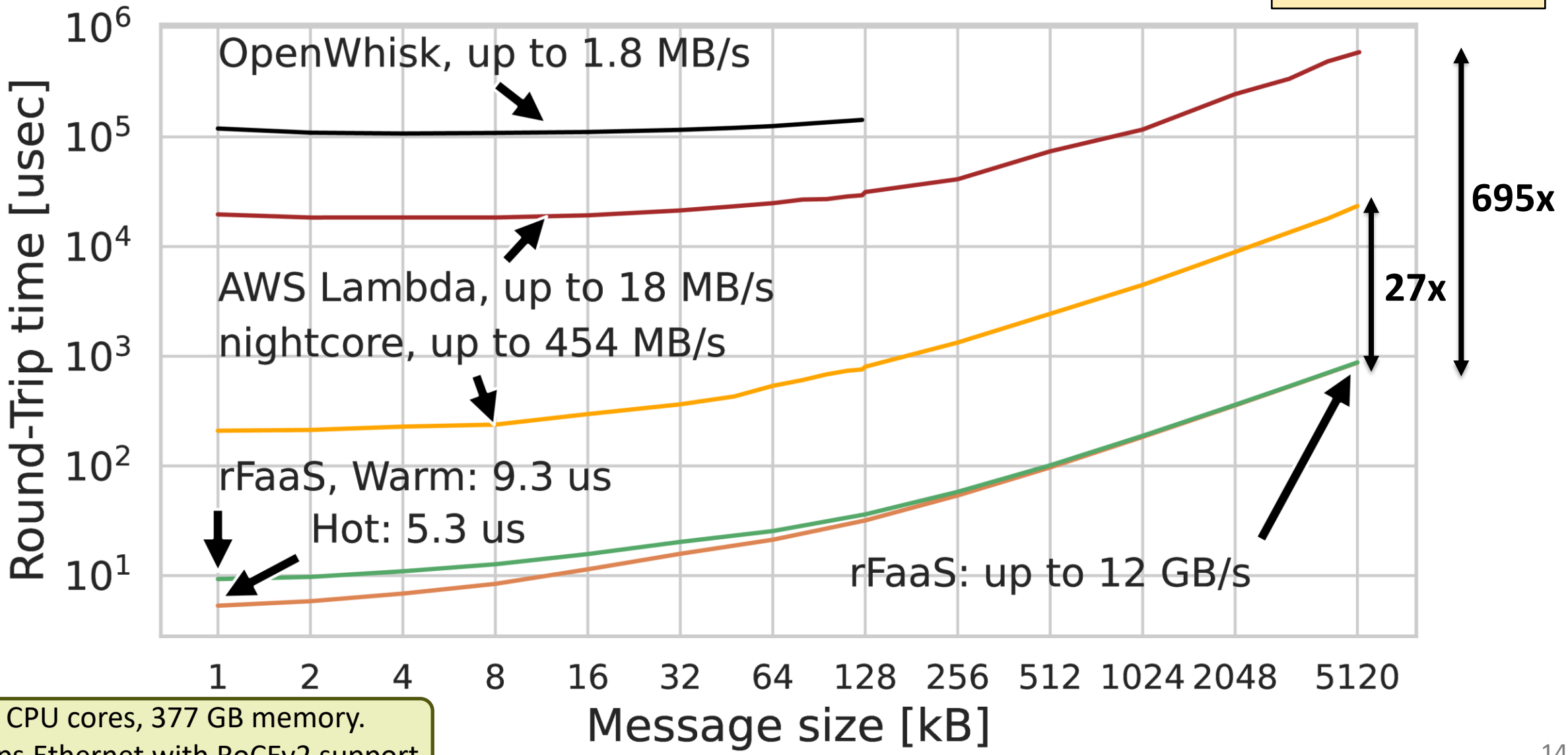
IEEE IPDPS 2023



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

How fast are invocations in FaaS?

IEEE IPDPS 2023



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

Communication in serverless

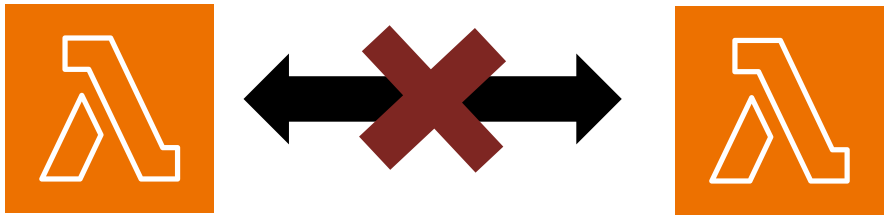
ACM ICS 2023
MSc Thesis

Communication in serverless

ACM ICS 2023
MSc Thesis

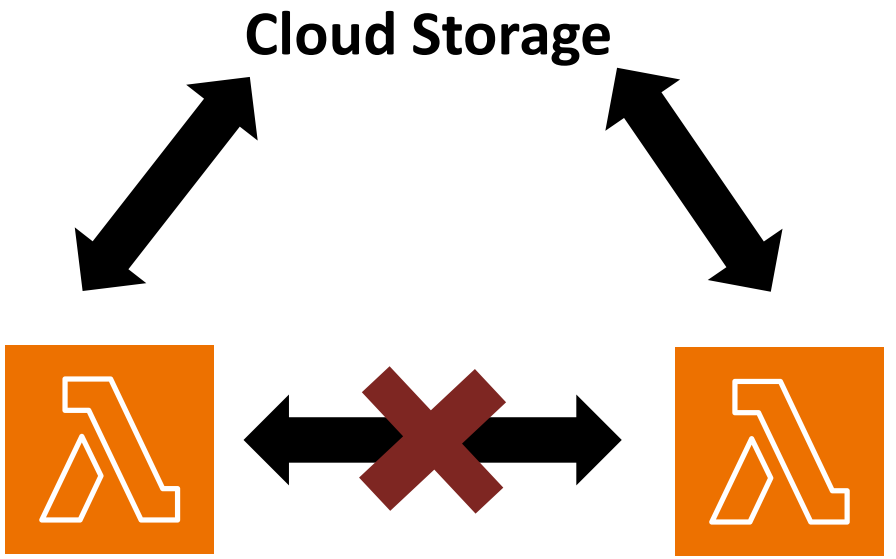
Communication in serverless

ACM ICS 2023
MSc Thesis



Communication in serverless

ACM ICS 2023
MSc Thesis

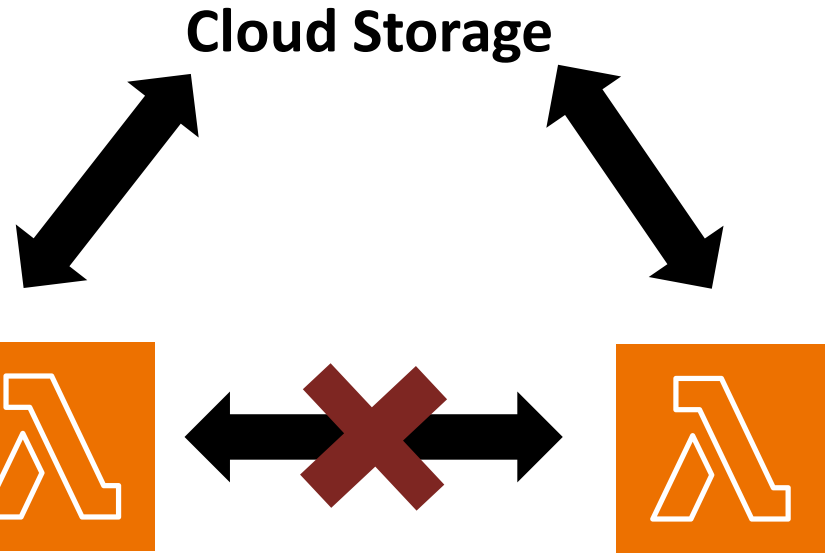


Communication in serverless

ACM ICS 2023
MSc Thesis

High Latency
For Small Messages

S3



Communication in serverless

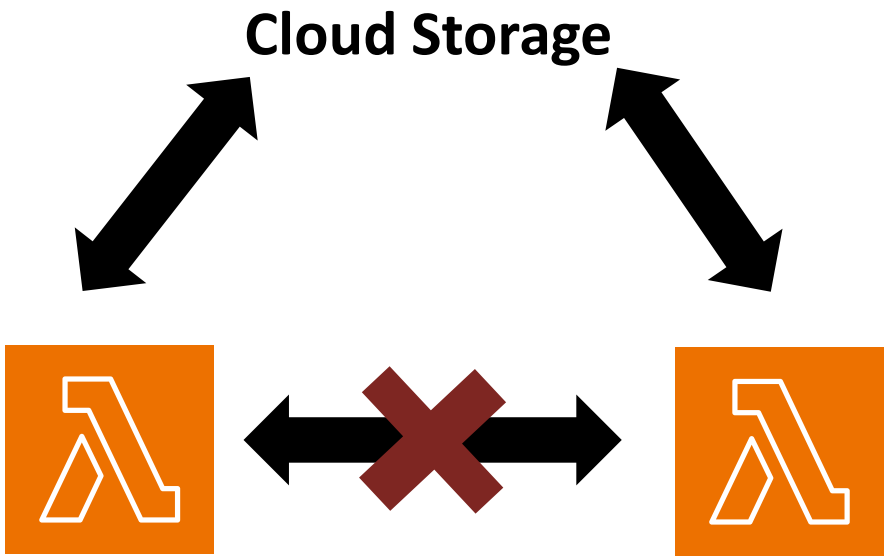
ACM ICS 2023
MSc Thesis

High Latency
For Small Messages

S3

Expensive for
Large Messages

DynamoDB



Communication in serverless

ACM ICS 2023
MSc Thesis

High Latency
For Small Messages

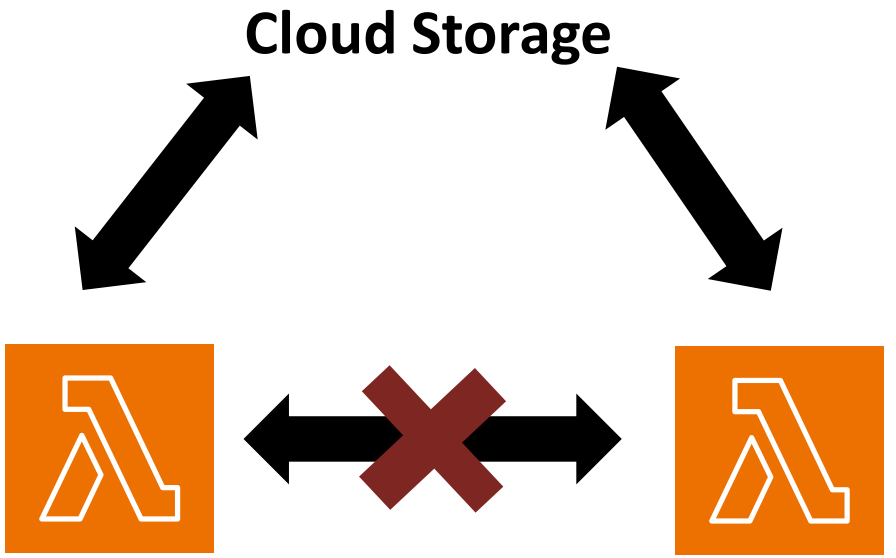
S3

Expensive for
Large Messages

DynamoDB

Not Serverless

Redis

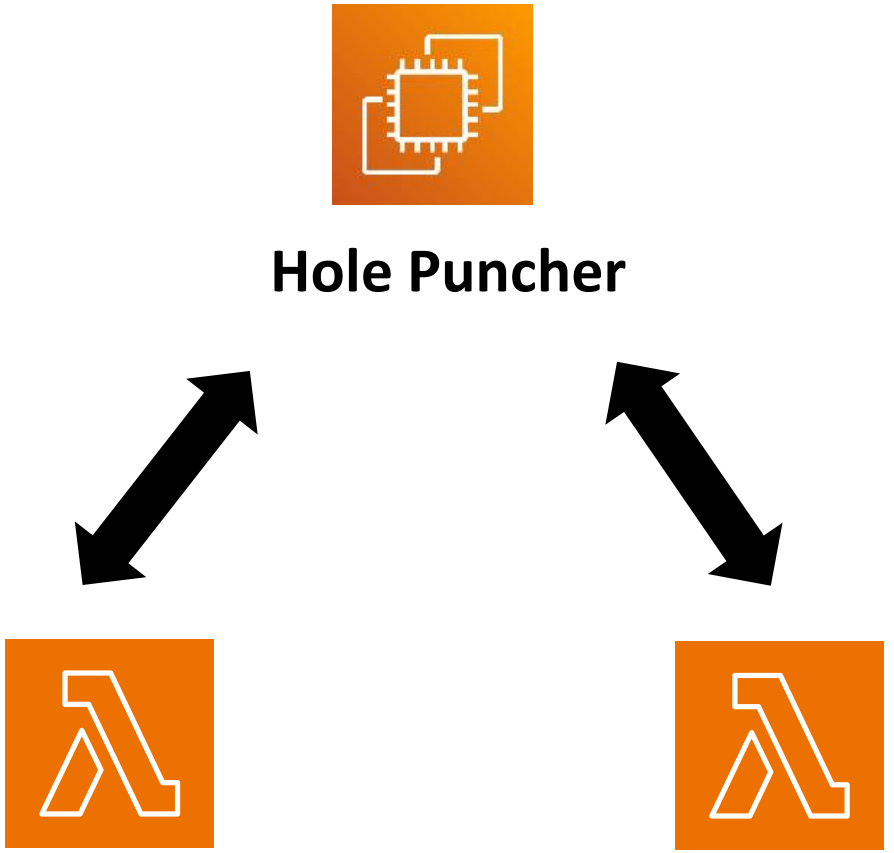


Communication in serverless

ACM ICS 2023
MSc Thesis

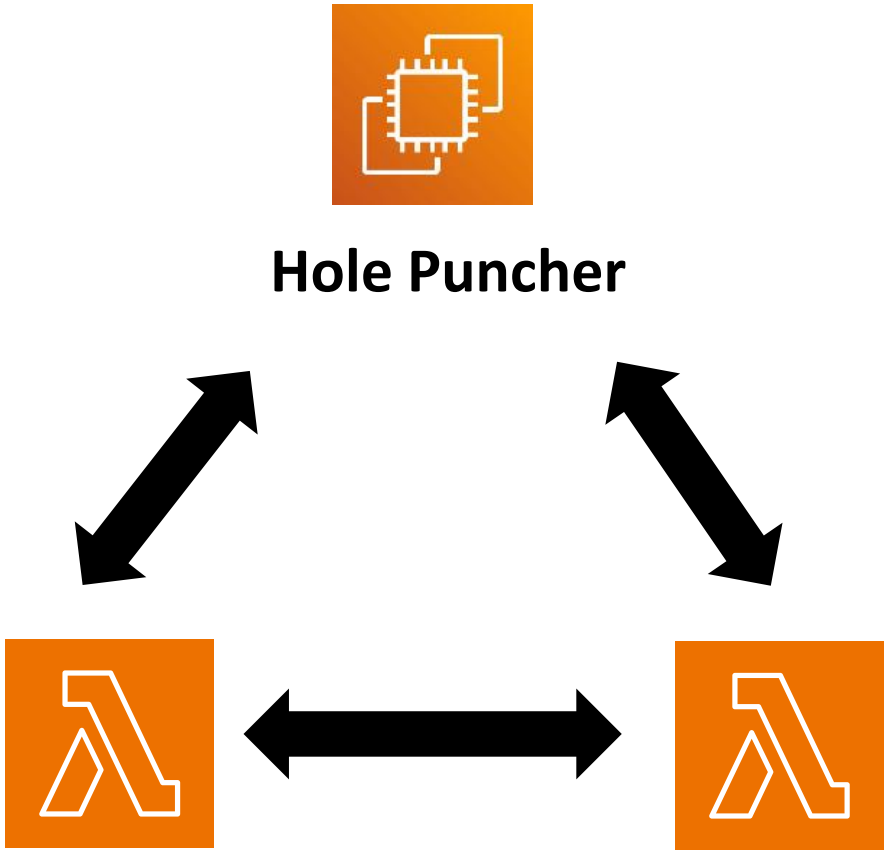
Communication in serverless

ACM ICS 2023
MSc Thesis



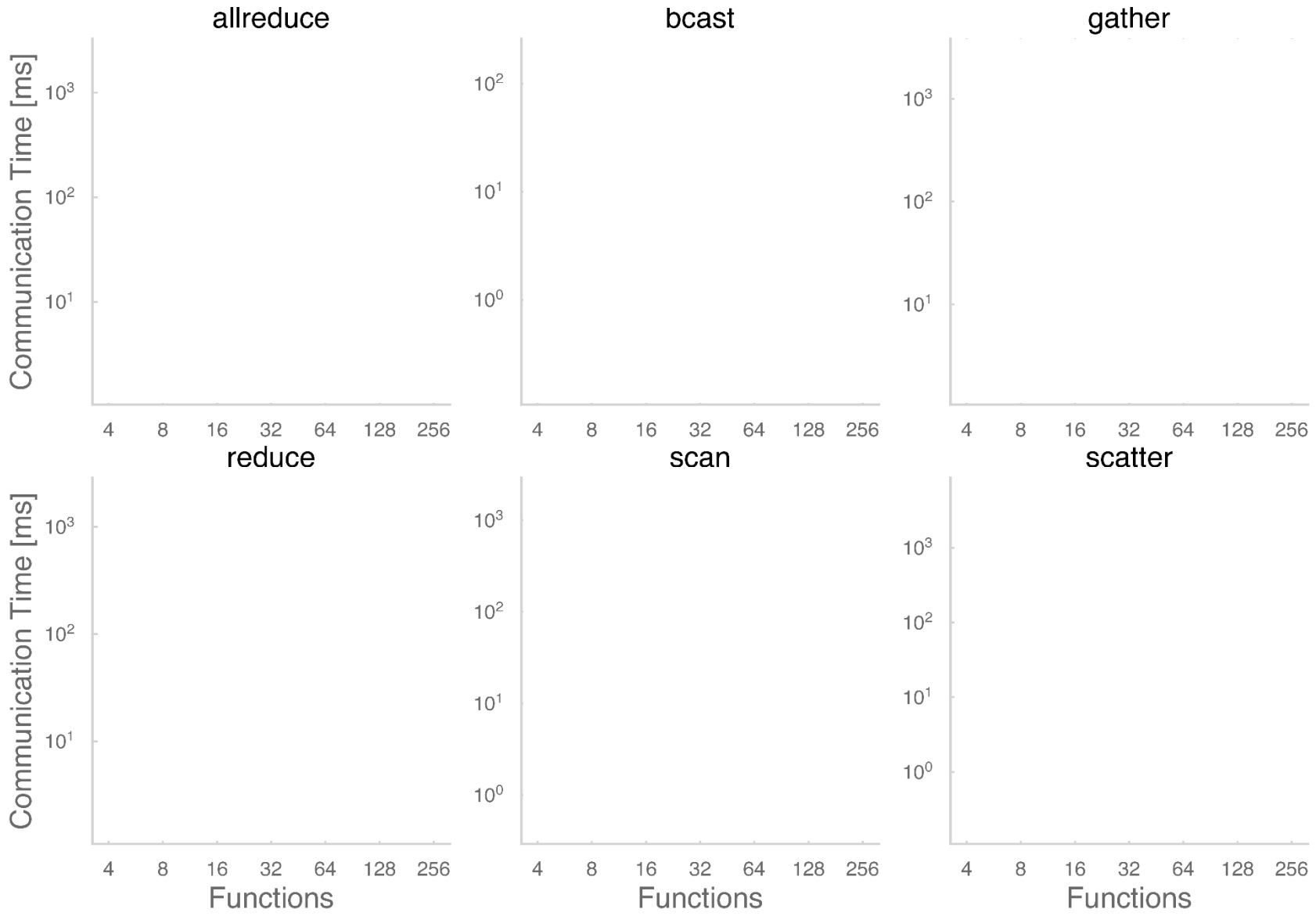
Communication in serverless

ACM ICS 2023
MSc Thesis



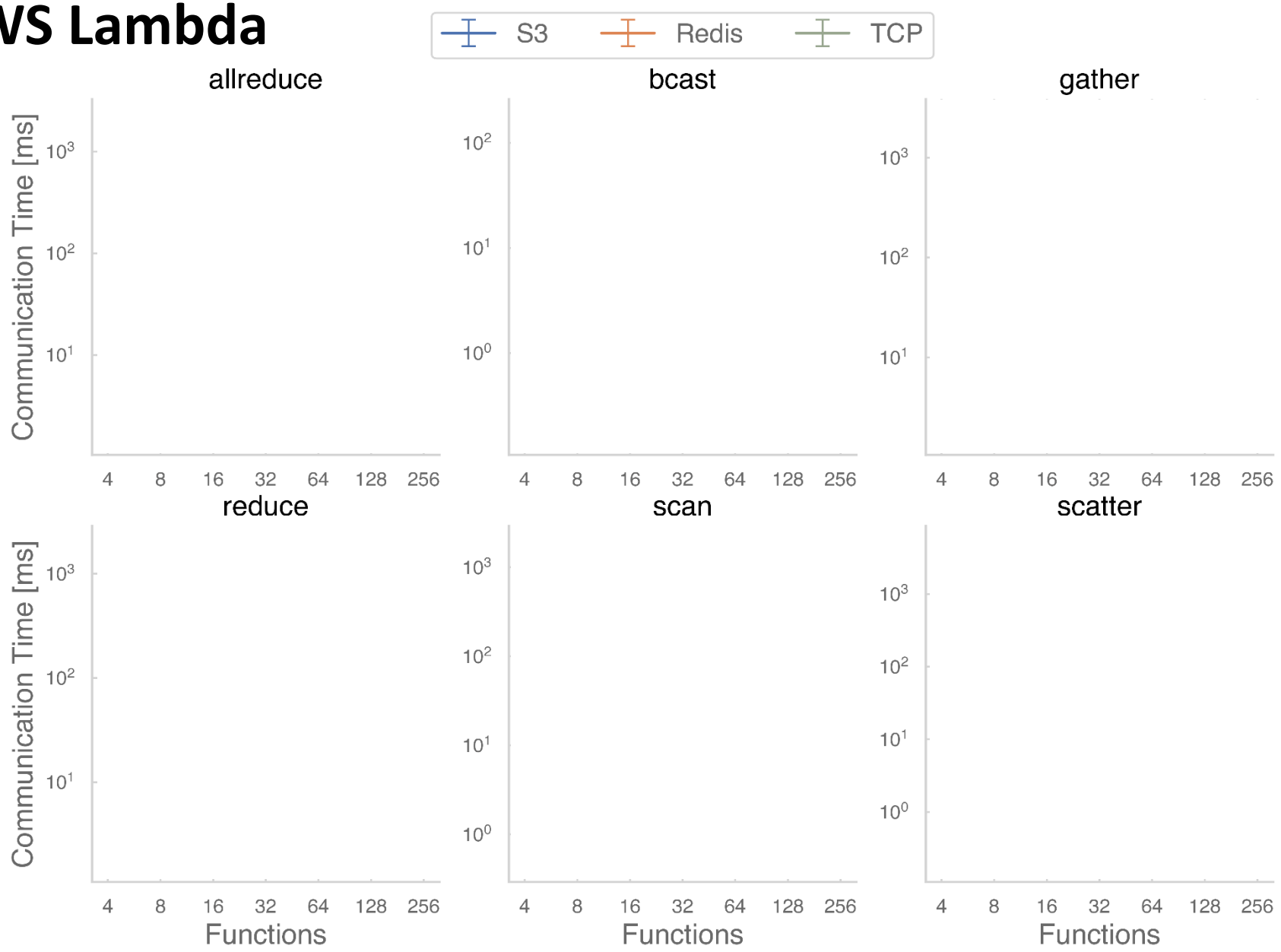
FMI on AWS Lambda

ACM ICS 2023
MSc Thesis



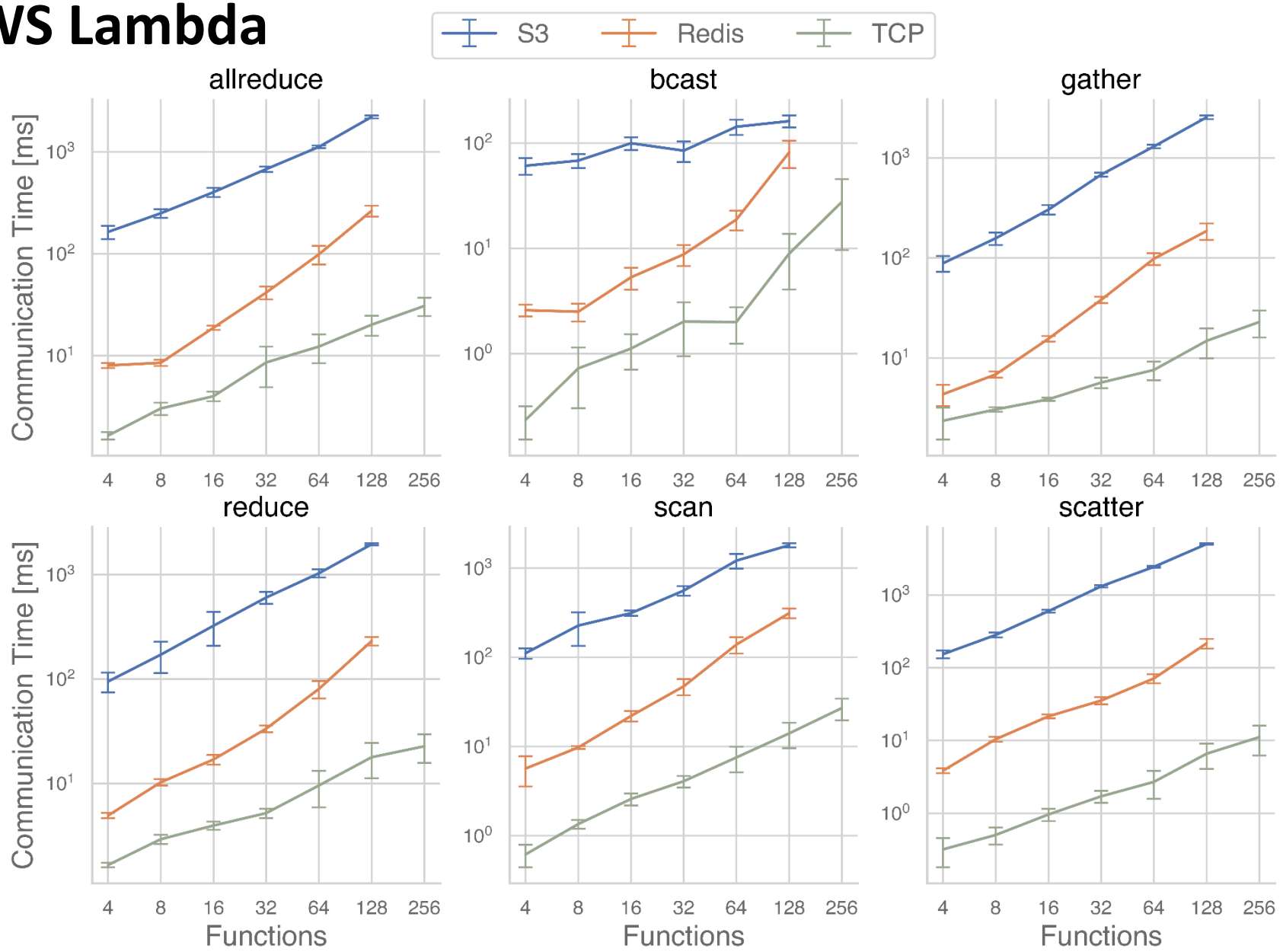
FMI on AWS Lambda

ACM ICS 2023
MSc Thesis

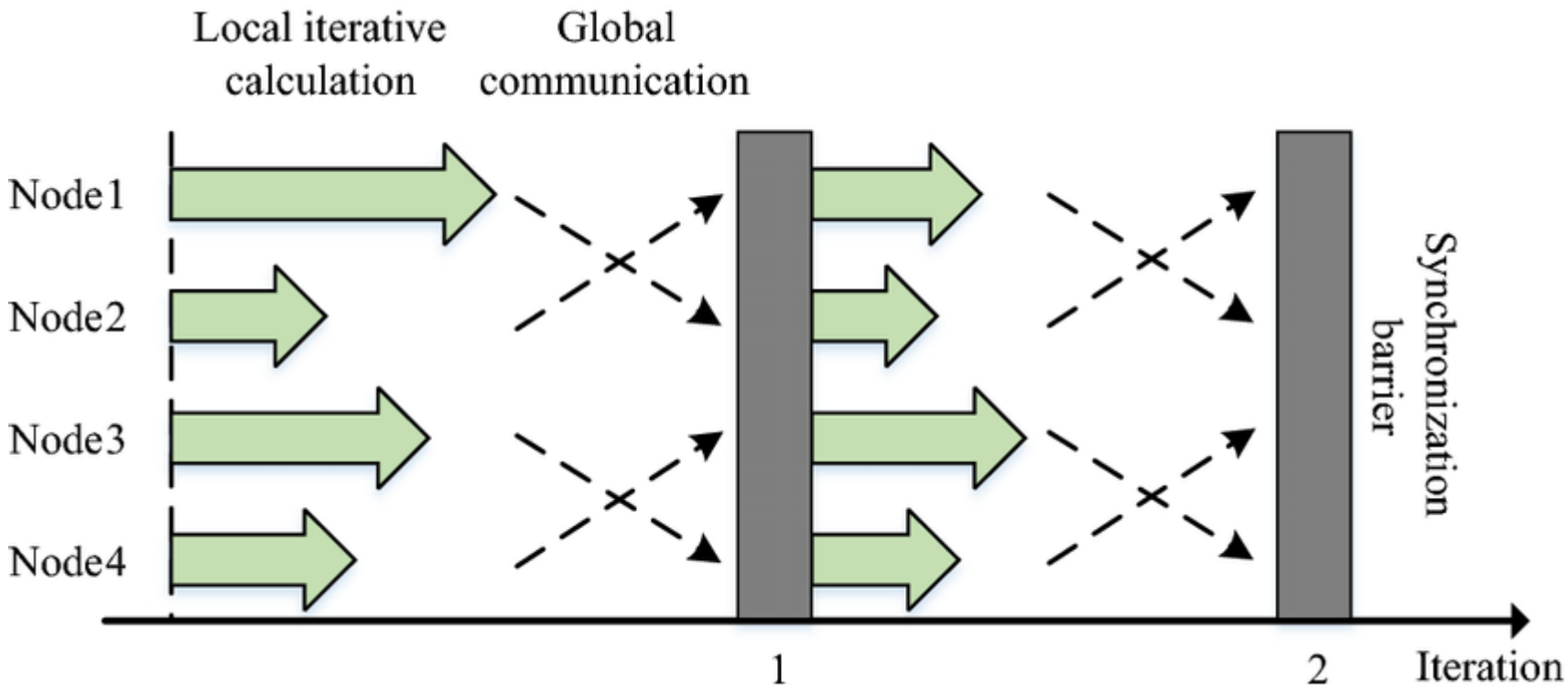


FMI on AWS Lambda

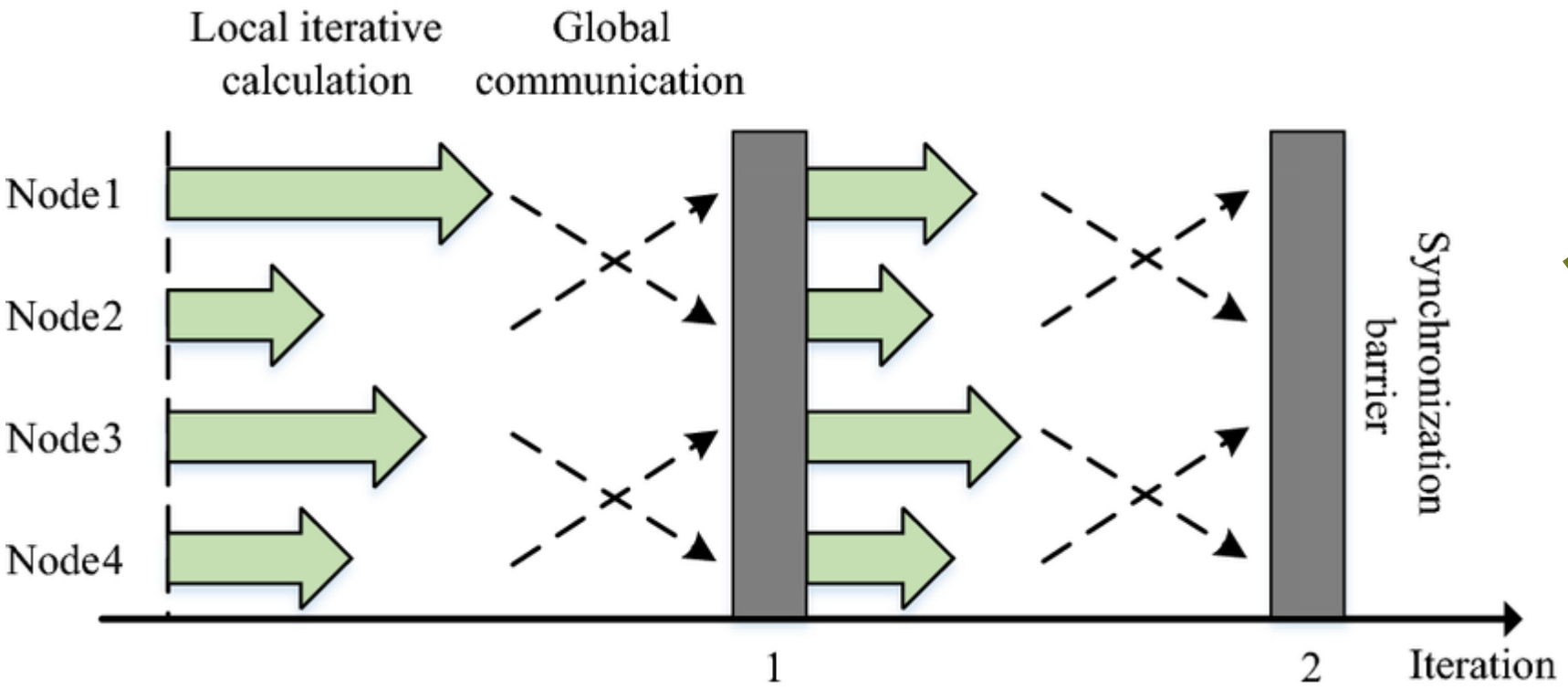
ACM ICS 2023
MSc Thesis



Idea: Elastic Rigidness of MPI

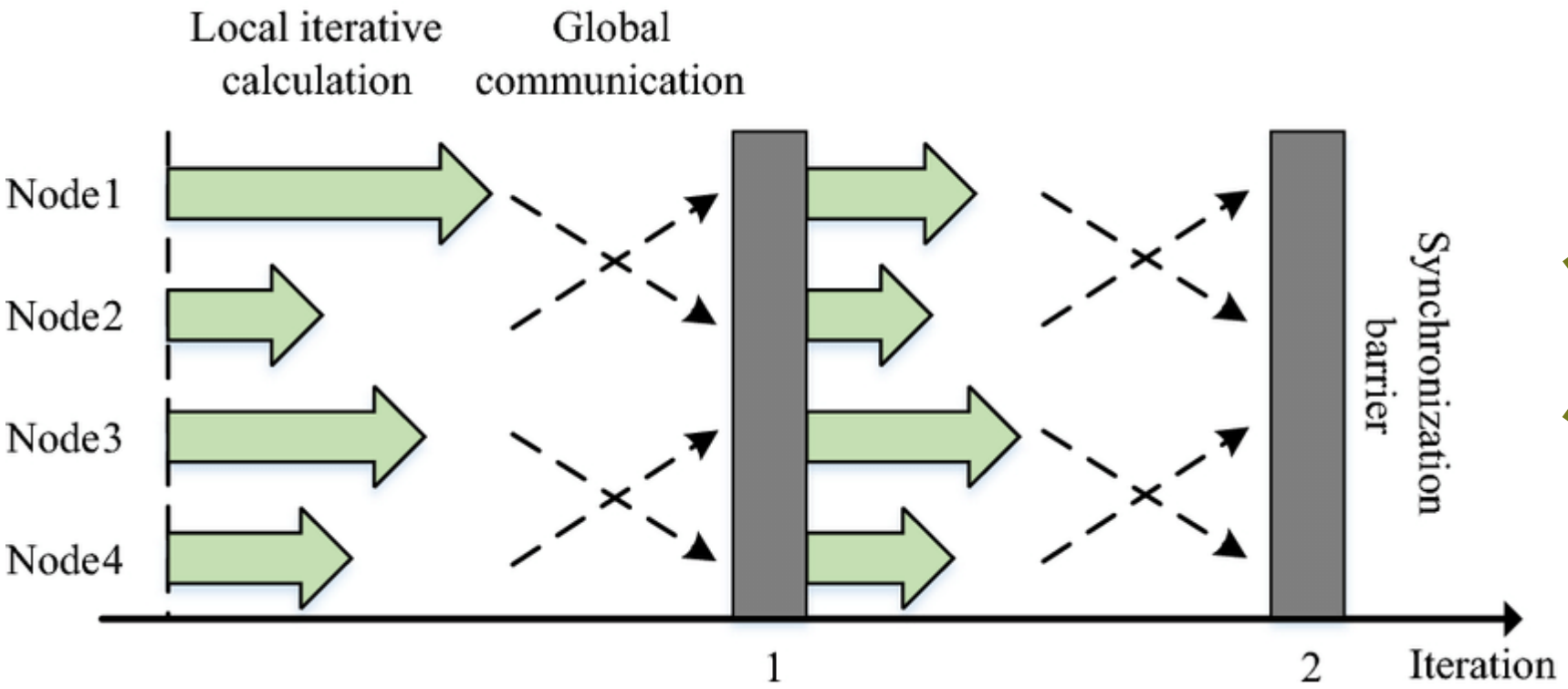


Idea: Elastic Rigidness of MPI

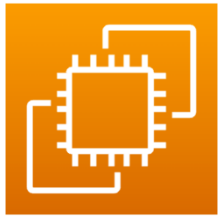


Serverless functions

Idea: Elastic Rigidness of MPI

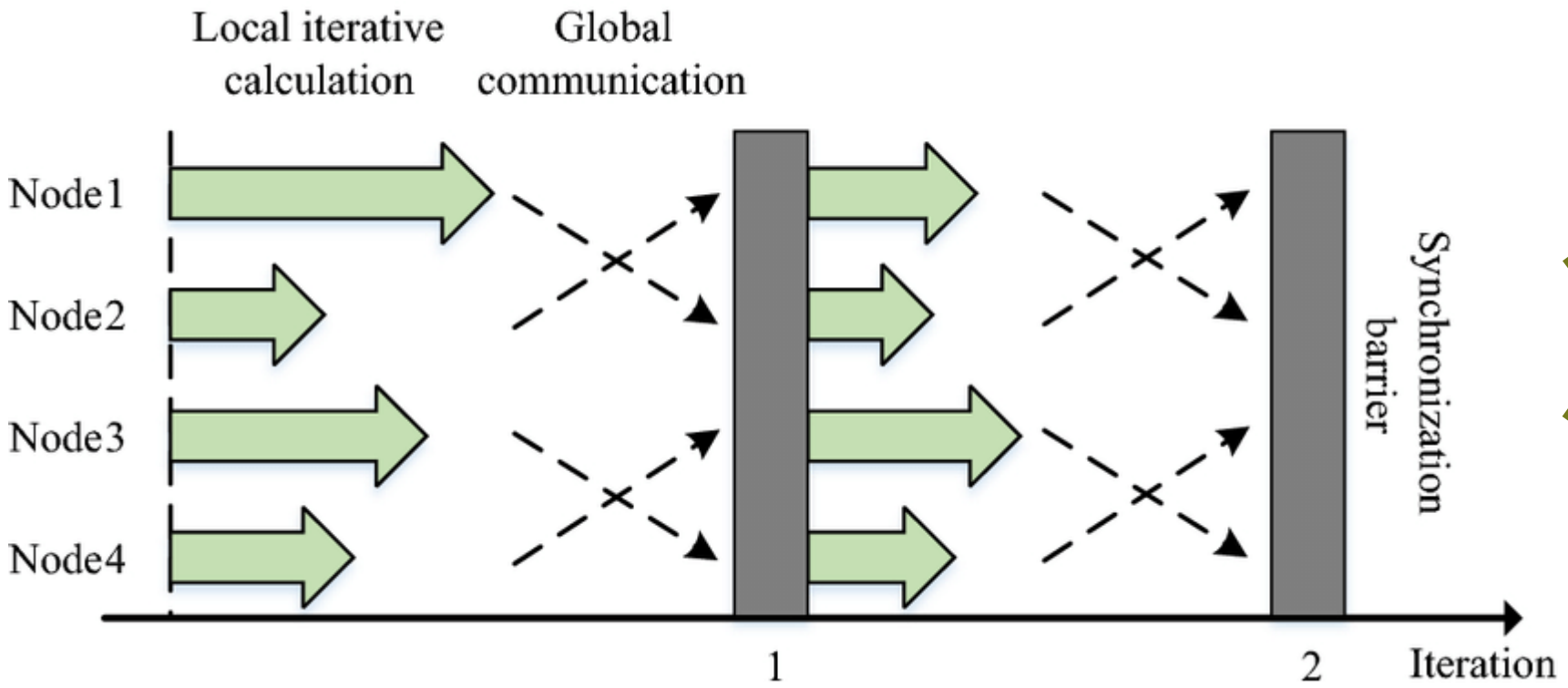


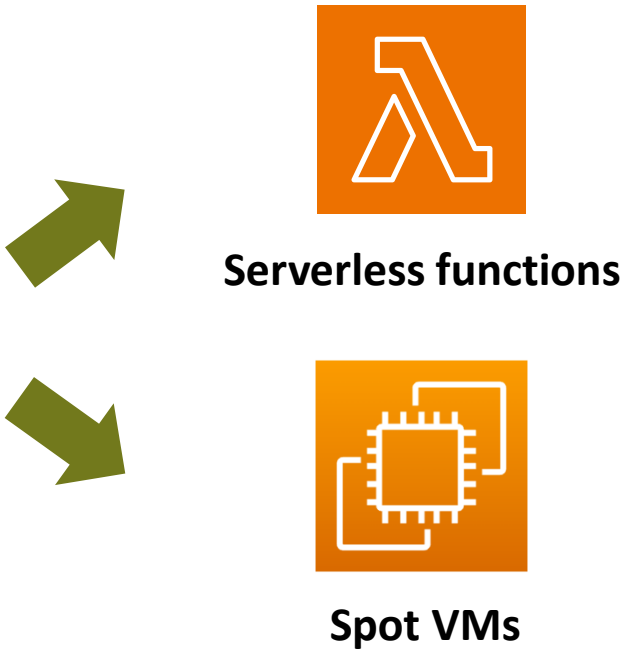
Serverless functions



Spot VMs

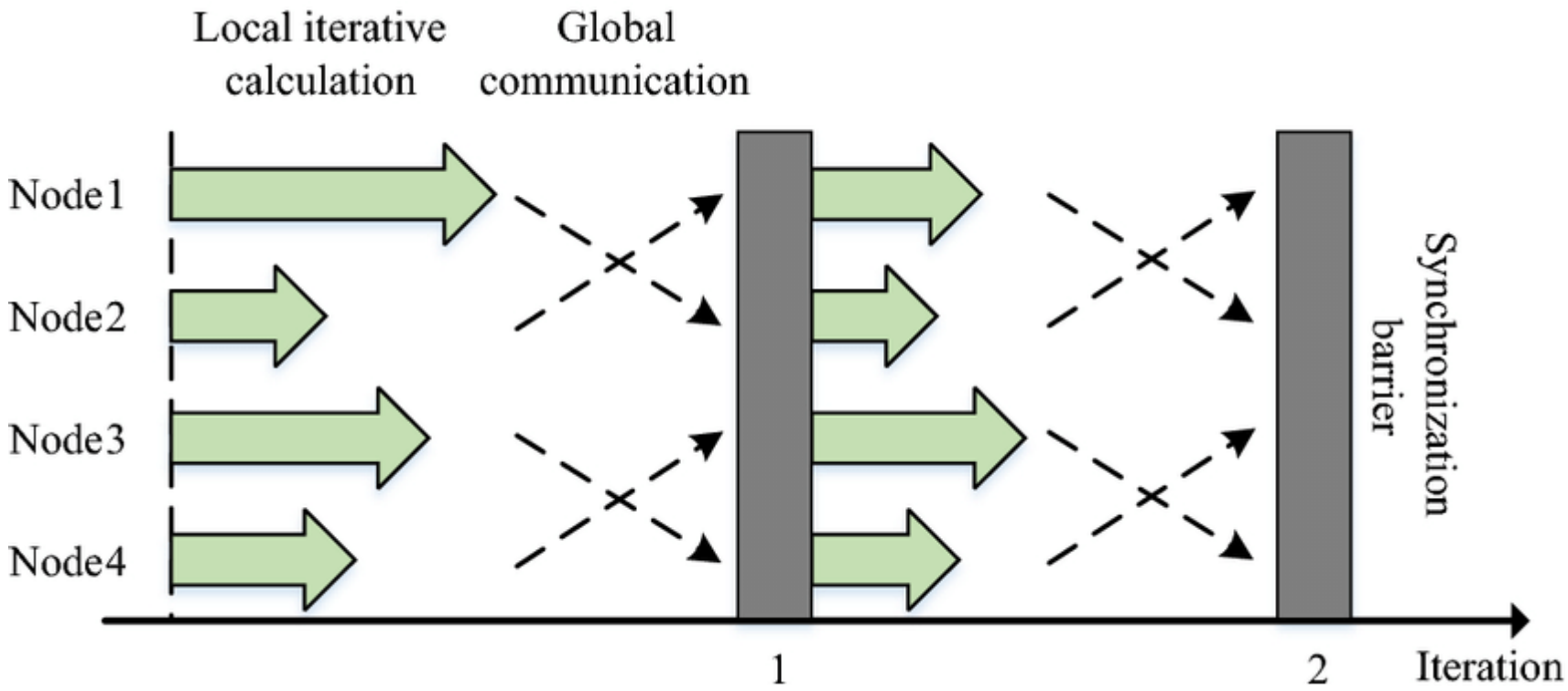
Idea: Elastic Rigidness of MPI

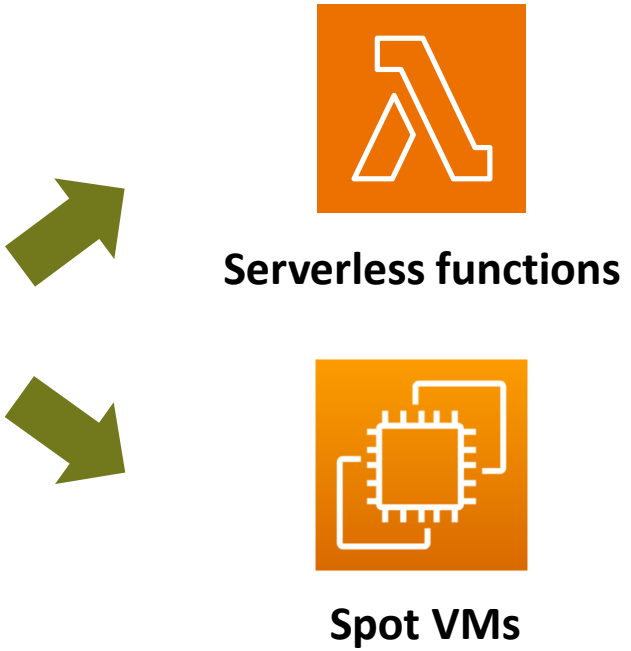




How elastic is Checkpoint/Restore?

Idea: Elastic Rigidness of MPI

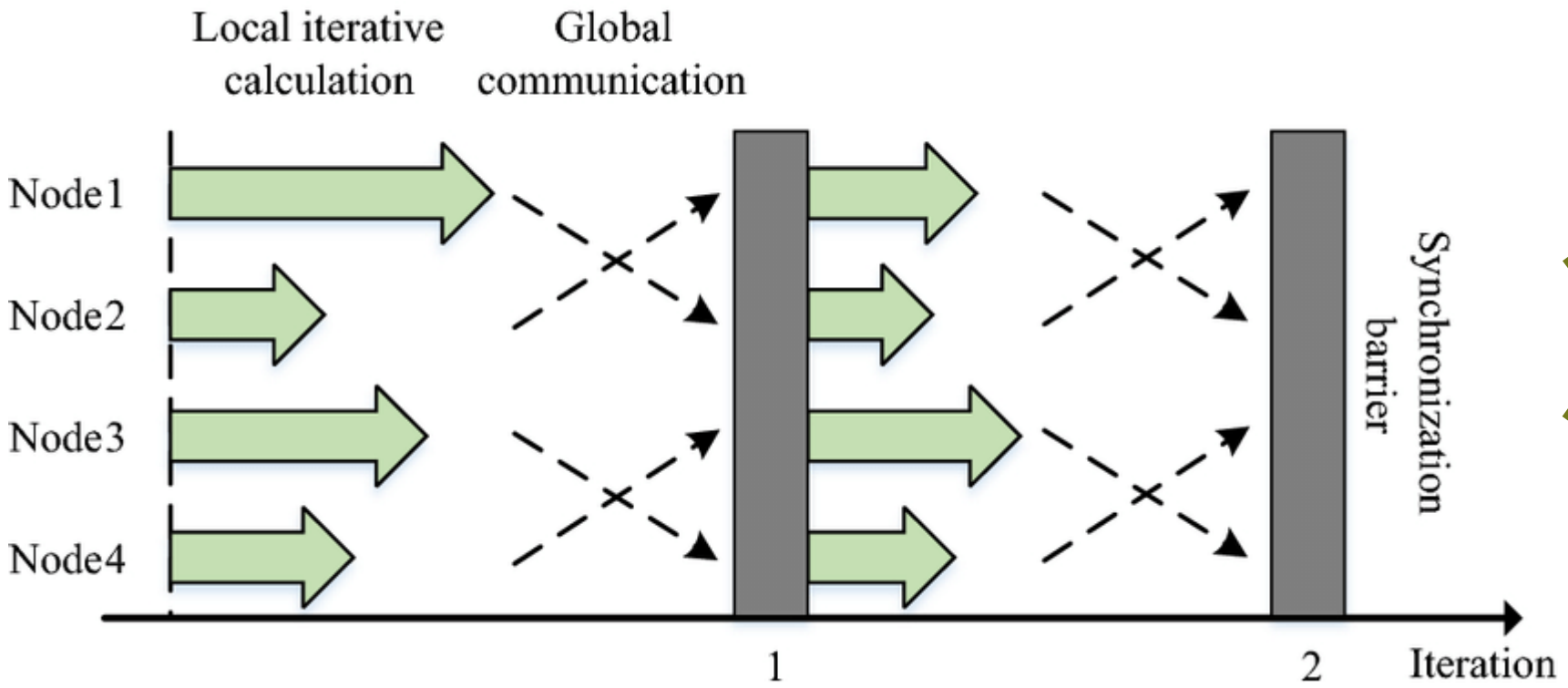


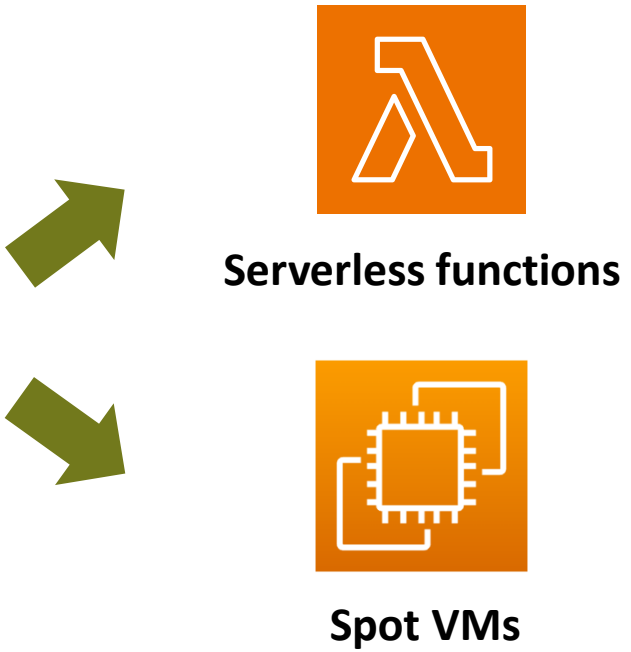


How elastic is Checkpoint/Restore?

Does MPI reconfiguration method work in practice?

Idea: Elastic Rigidness of MPI





How elastic is Checkpoint/Restore?

Does MPI reconfiguration method work in practice?

Can we transparently migrate HW context?

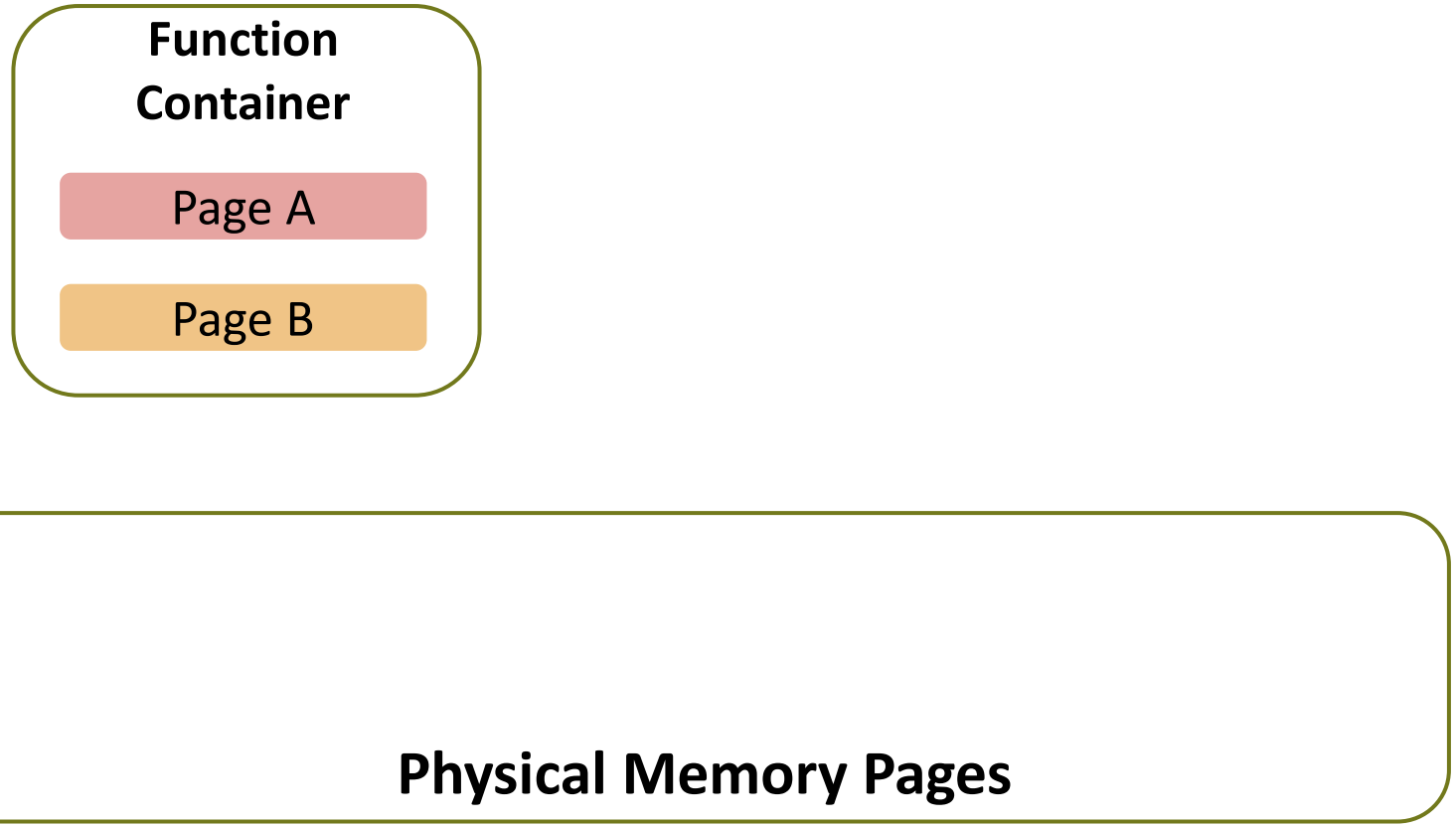
Memory Duplication in Serverless

IEEE BigData 2023
MSc Thesis

Physical Memory Pages

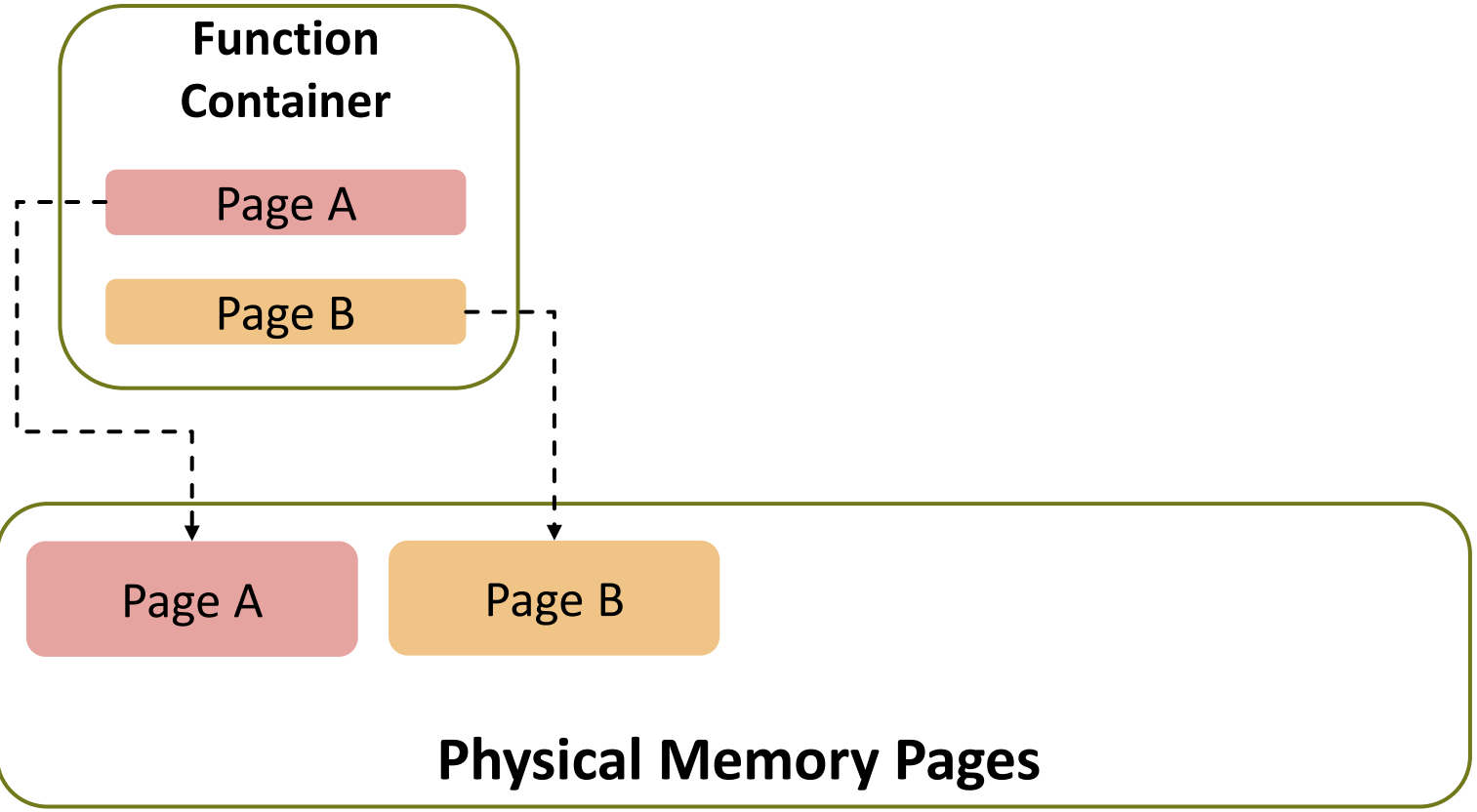
Memory Duplication in Serverless

IEEE BigData 2023
MSc Thesis



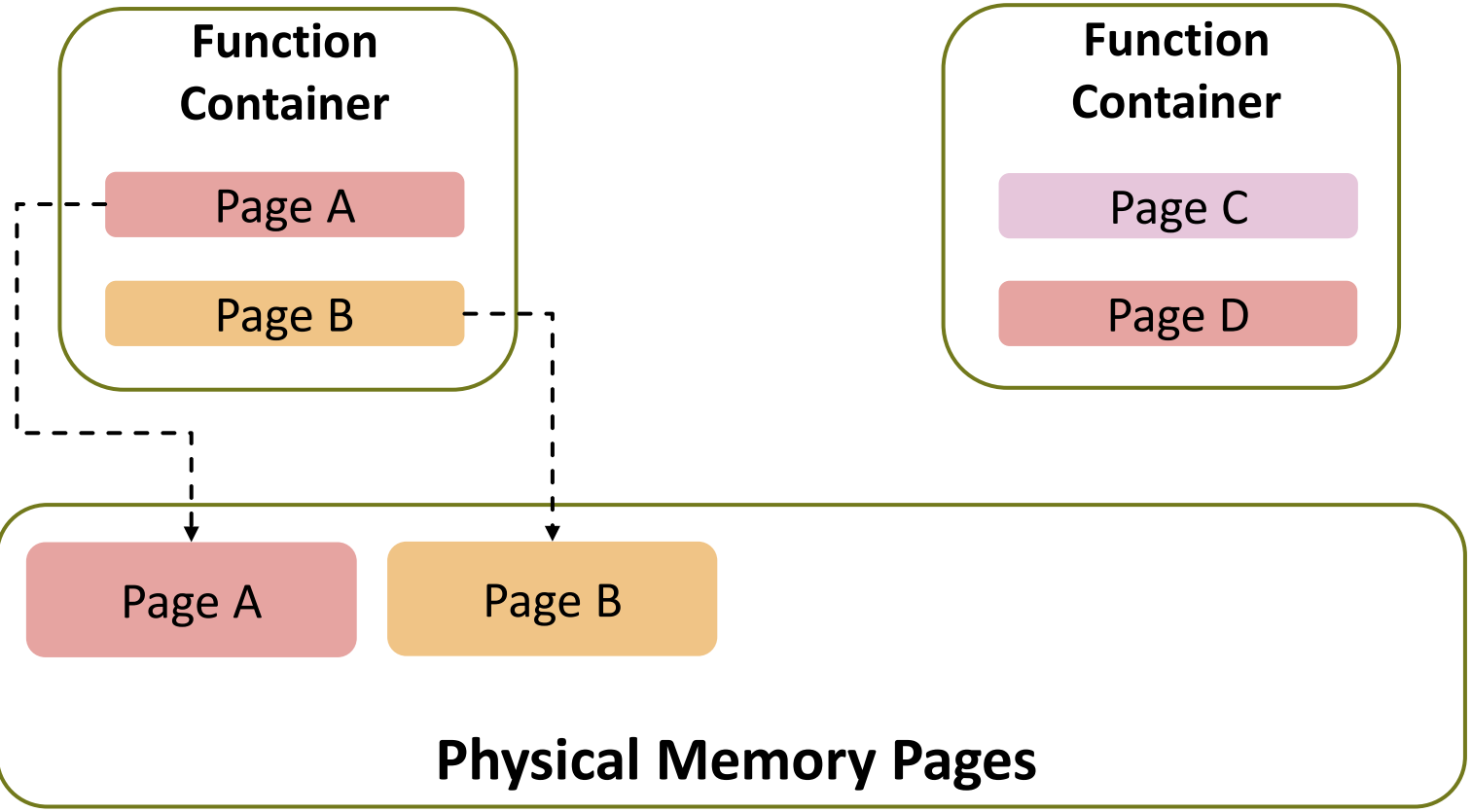
Memory Duplication in Serverless

IEEE BigData 2023
MSc Thesis



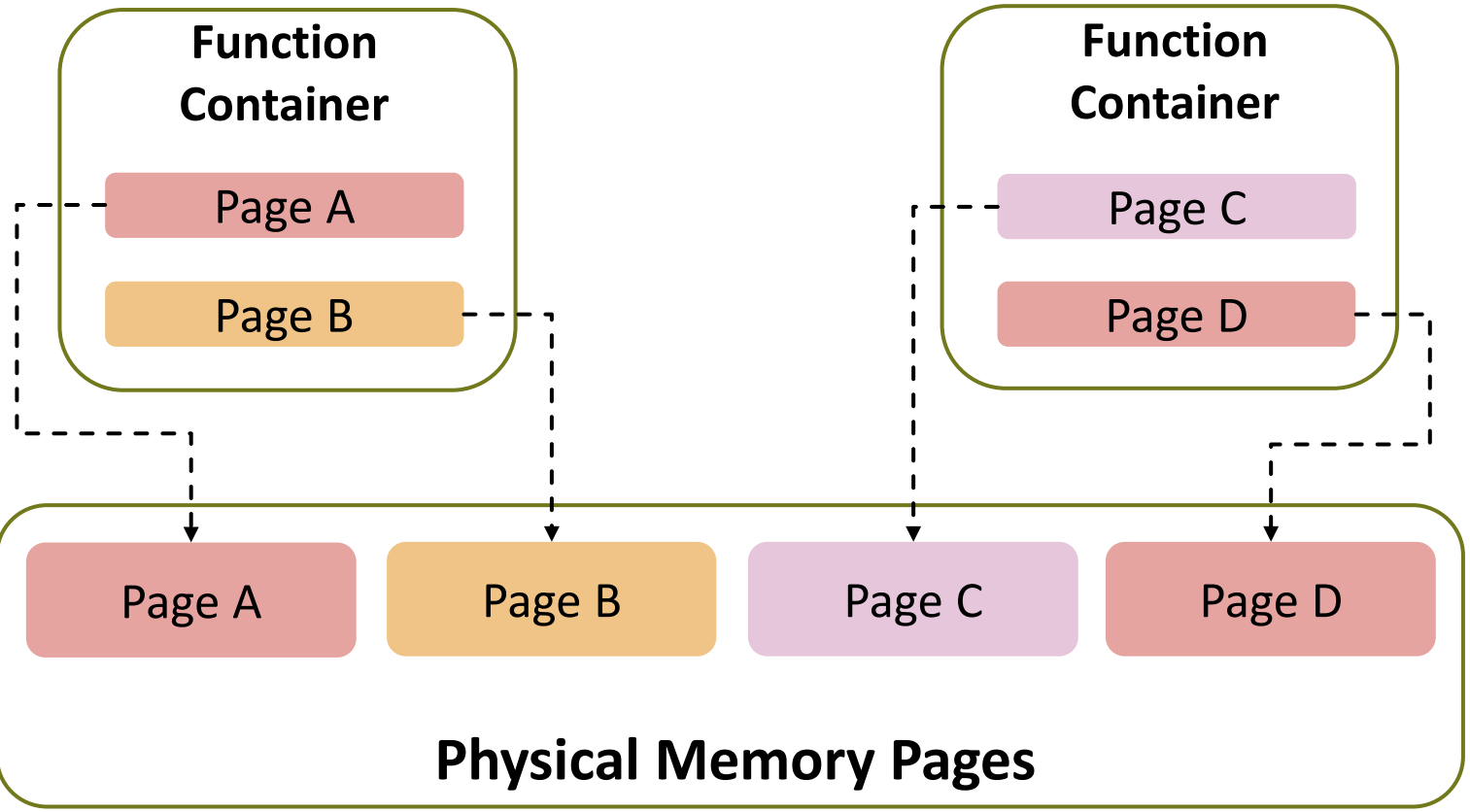
Memory Duplication in Serverless

IEEE BigData 2023
MSc Thesis



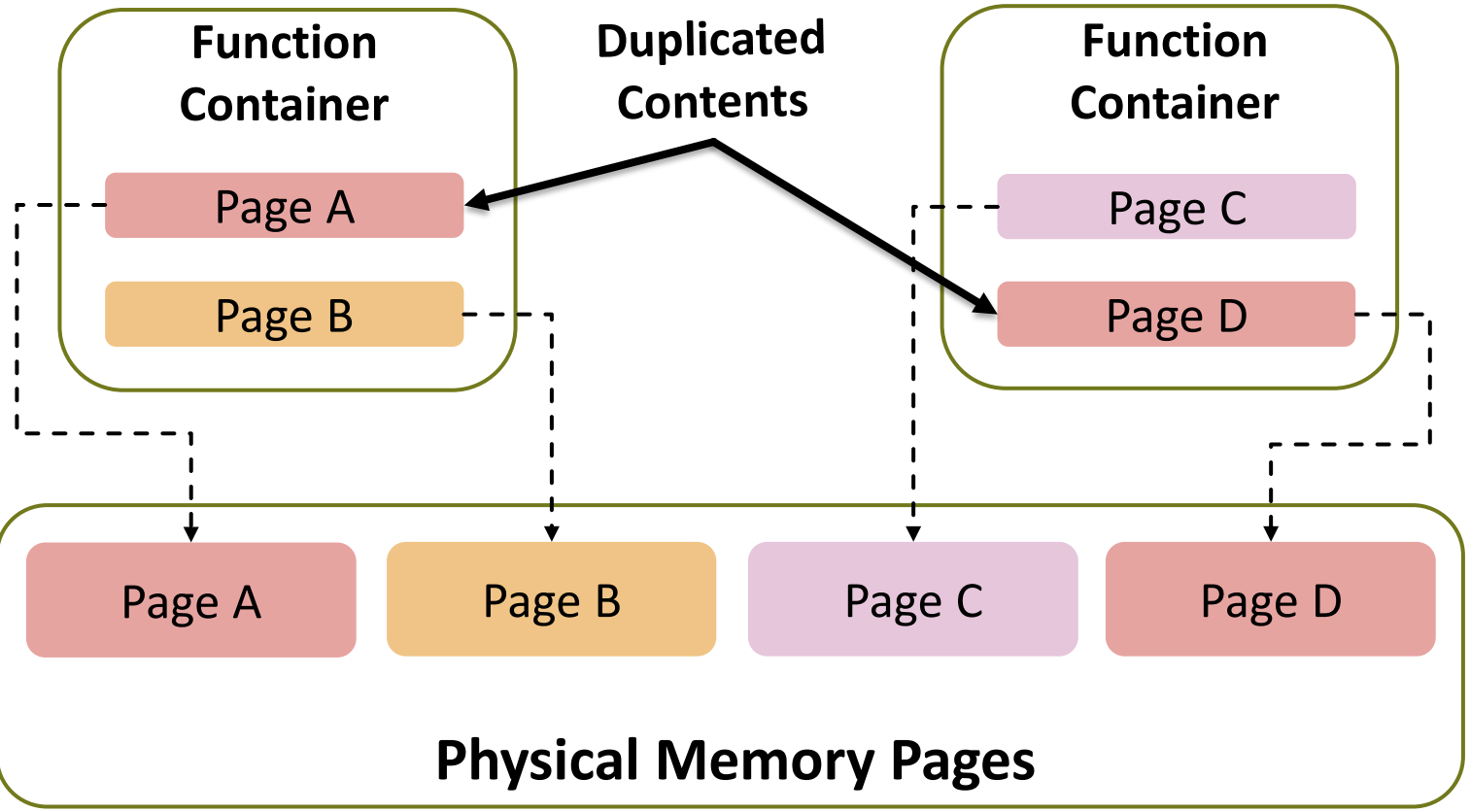
Memory Duplication in Serverless

IEEE BigData 2023
MSc Thesis



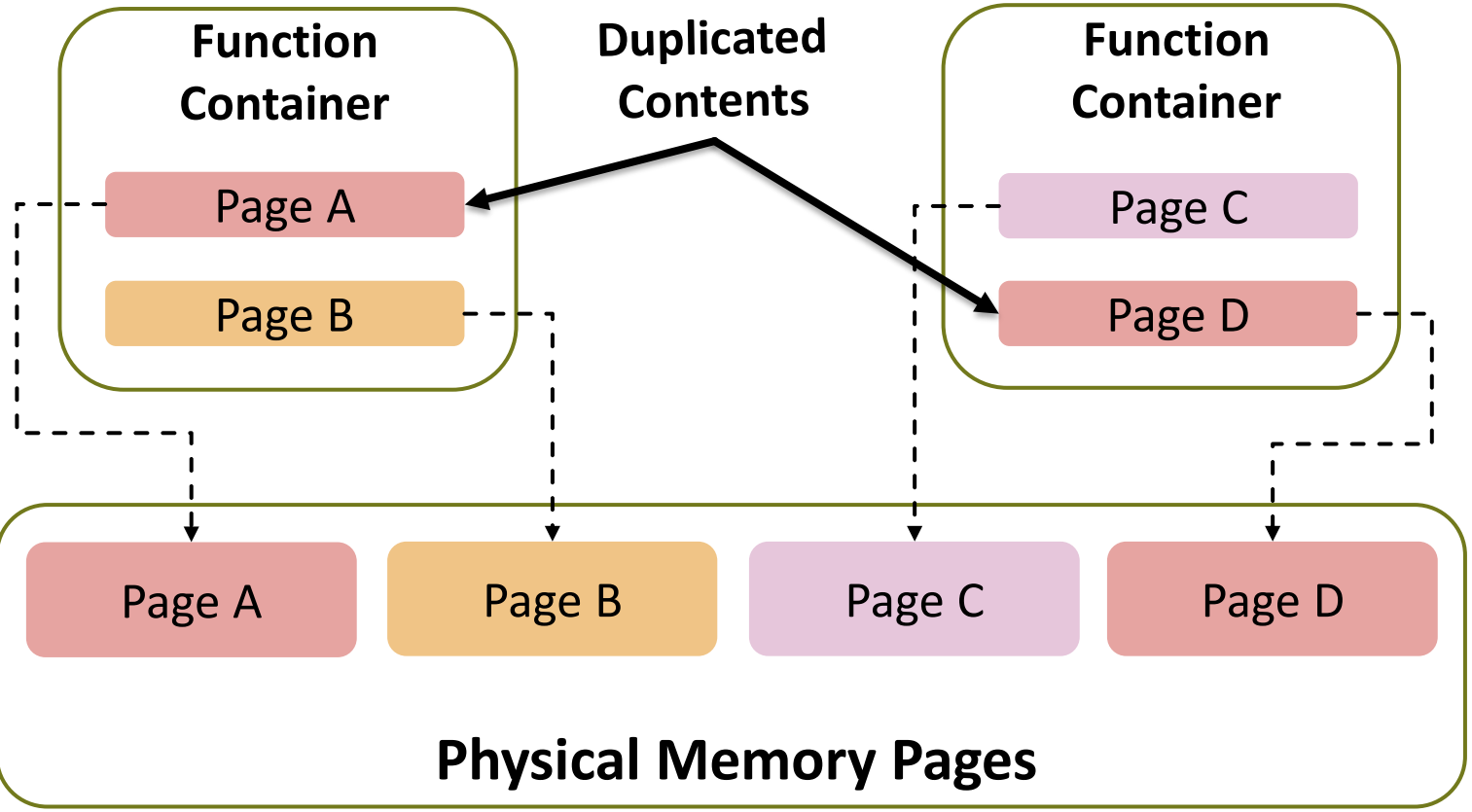
Memory Duplication in Serverless

IEEE BigData 2023
MSc Thesis

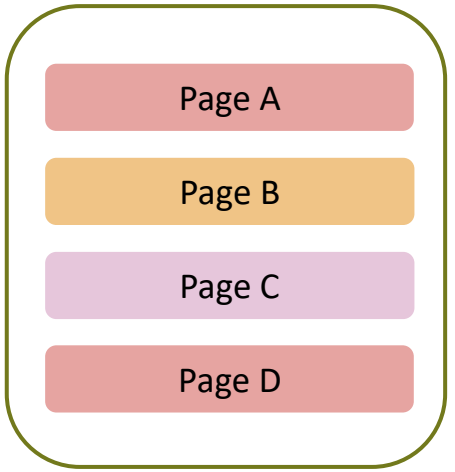


Memory Duplication in Serverless

IEEE BigData 2023
MSc Thesis

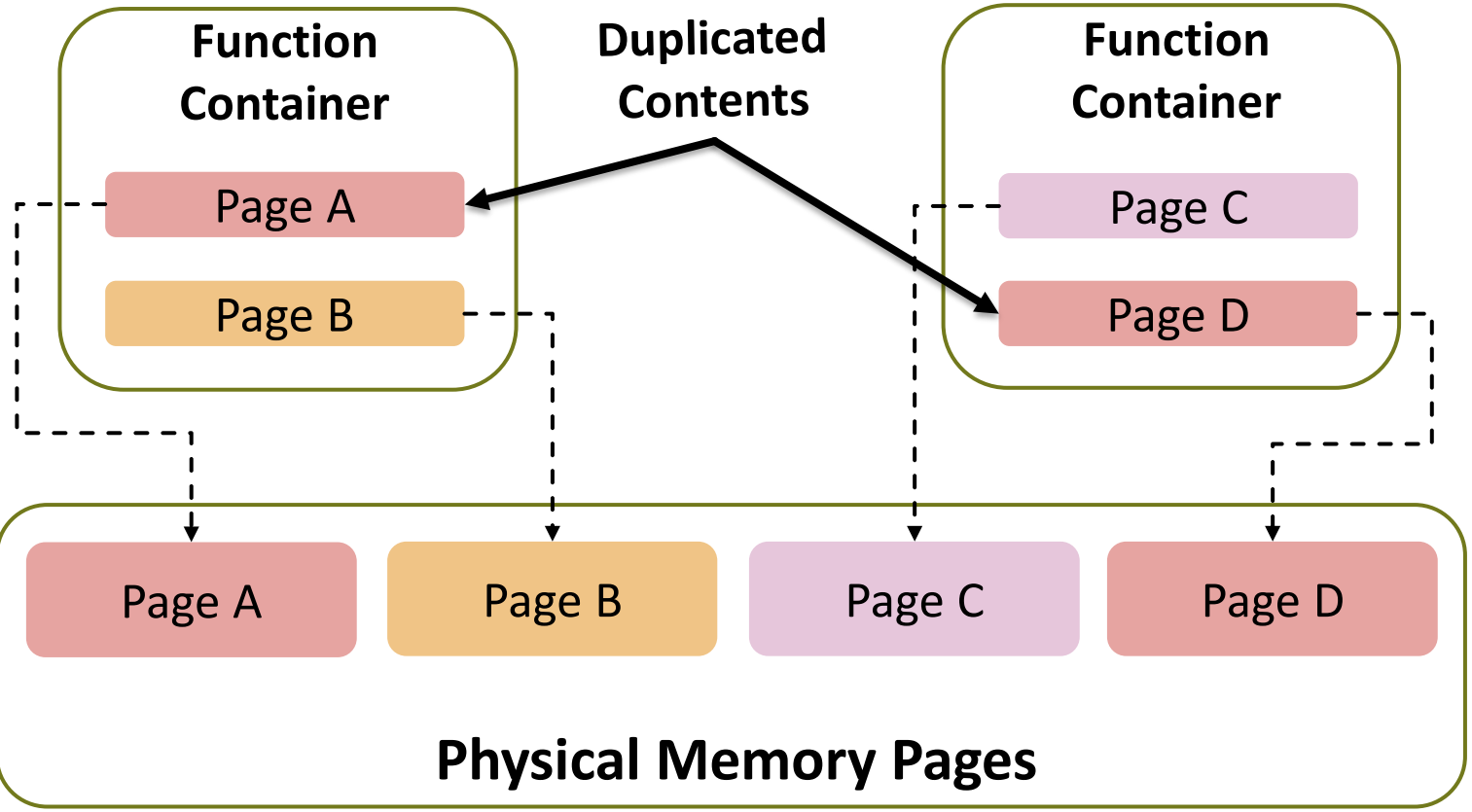


Example: Kernel Samepage Merging (KSM)

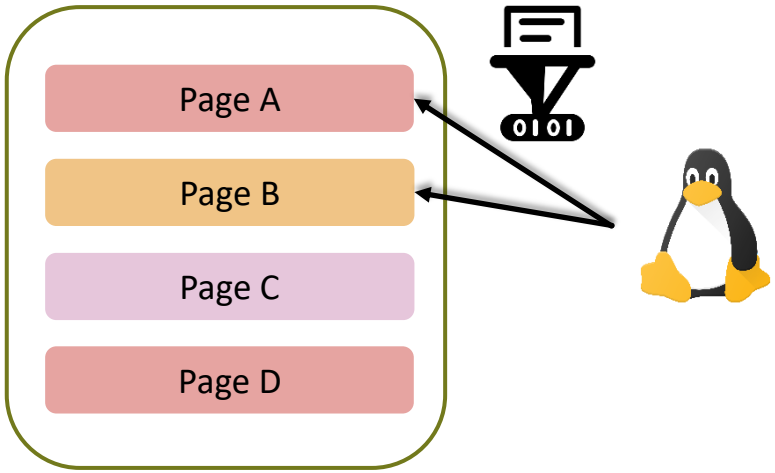


Memory Duplication in Serverless

IEEE BigData 2023
MSc Thesis

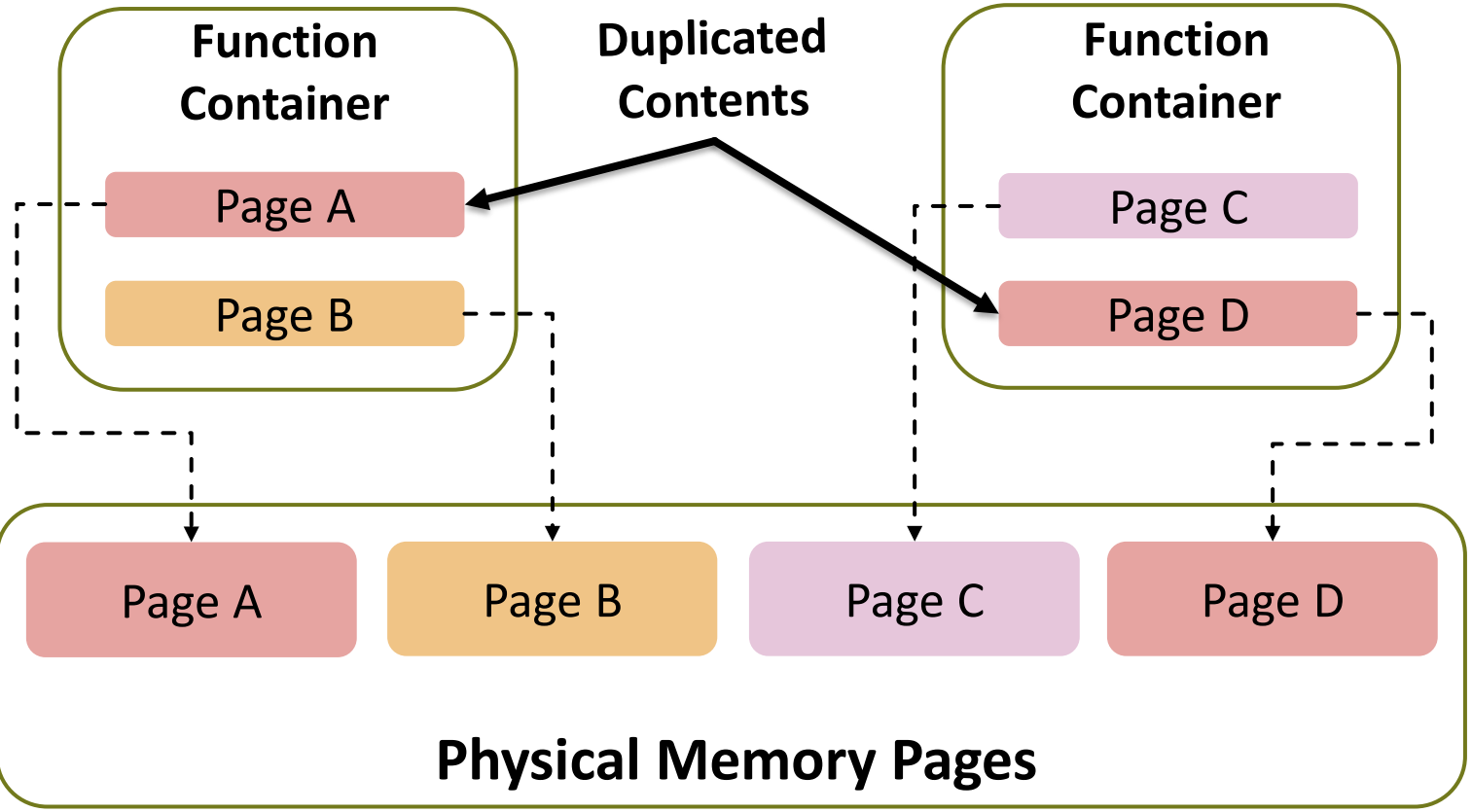


Example: Kernel Samepage Merging (KSM)

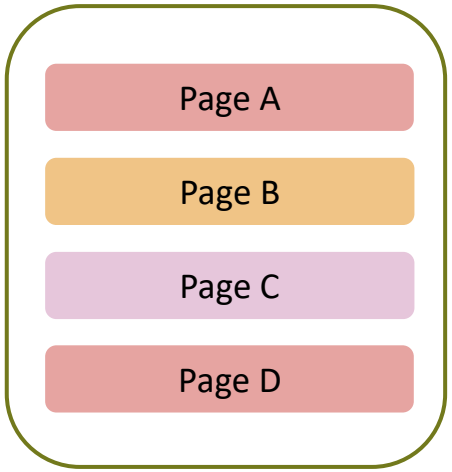


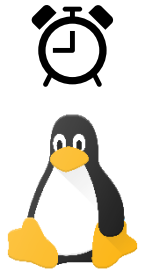
Memory Duplication in Serverless

IEEE BigData 2023
MSc Thesis



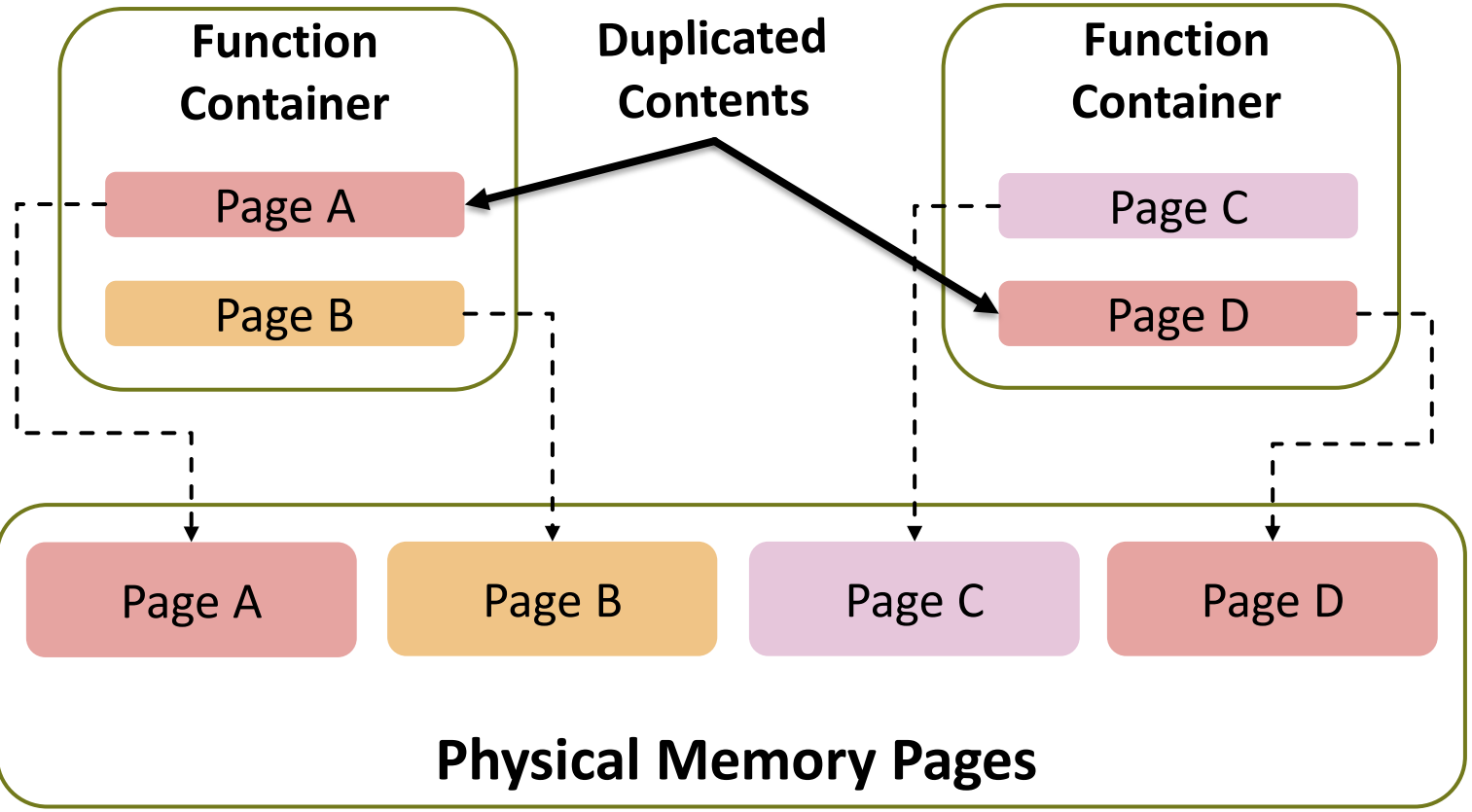
Example: Kernel Samepage Merging (KSM)



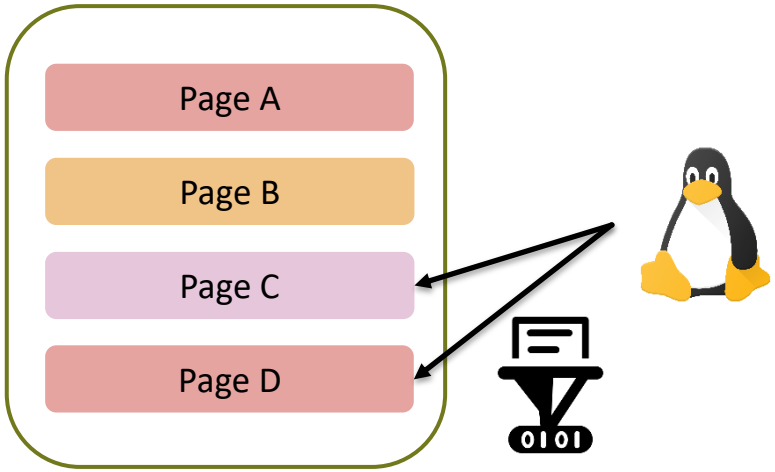


Memory Duplication in Serverless

IEEE BigData 2023
MSc Thesis

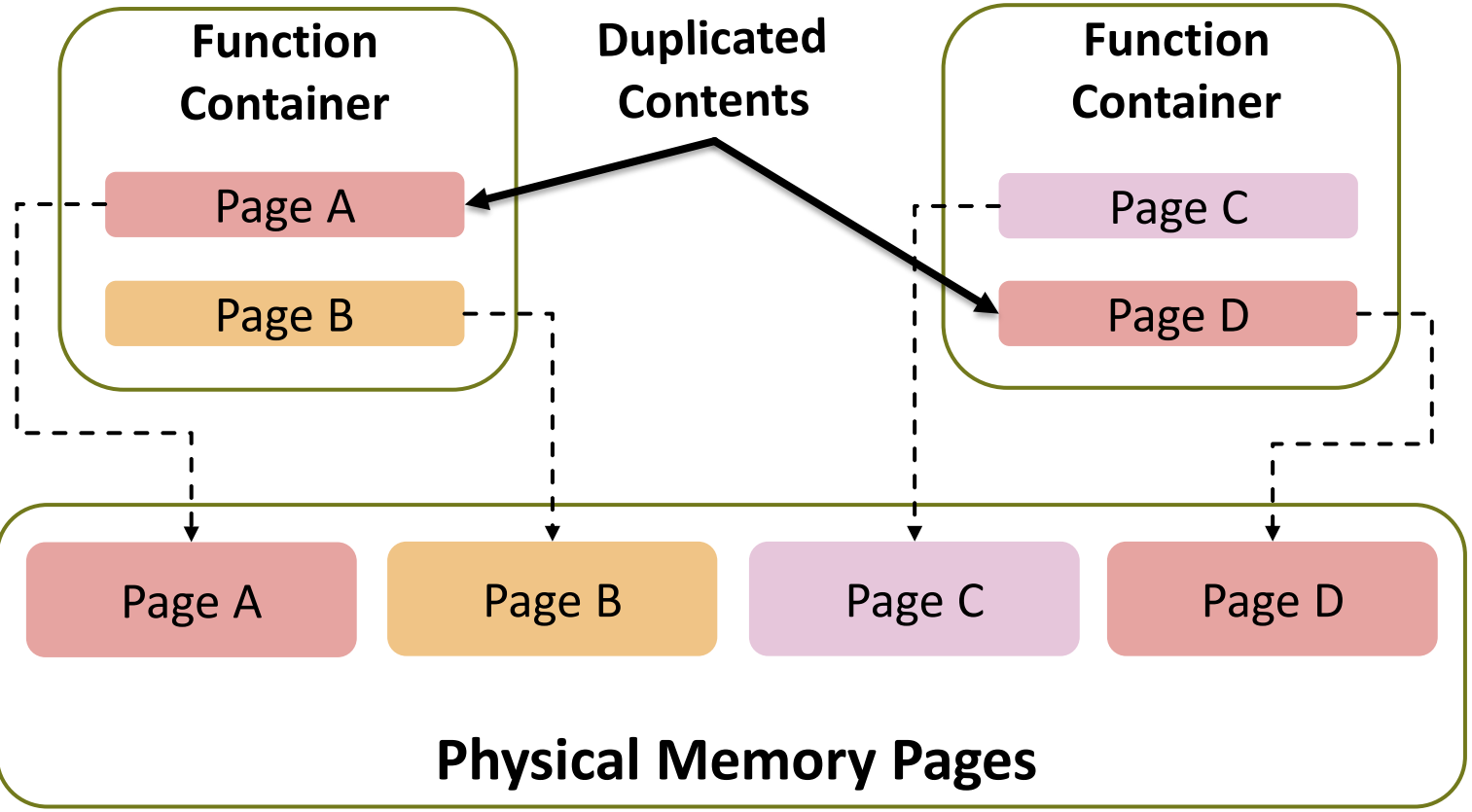


Example: Kernel Samepage Merging (KSM)

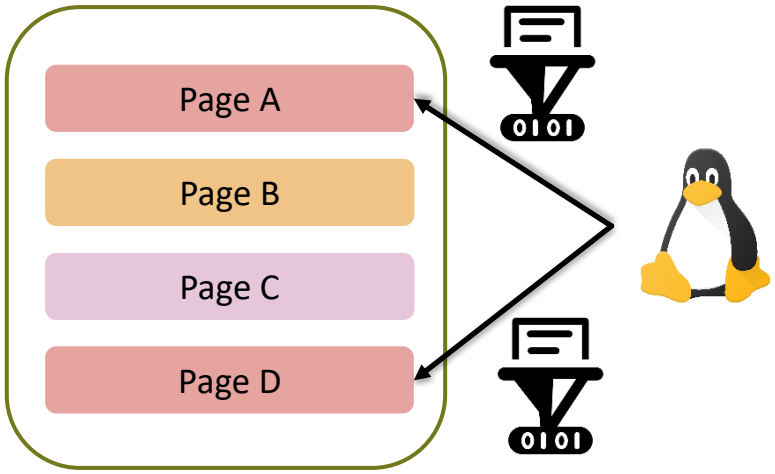


Memory Duplication in Serverless

IEEE BigData 2023
MSc Thesis

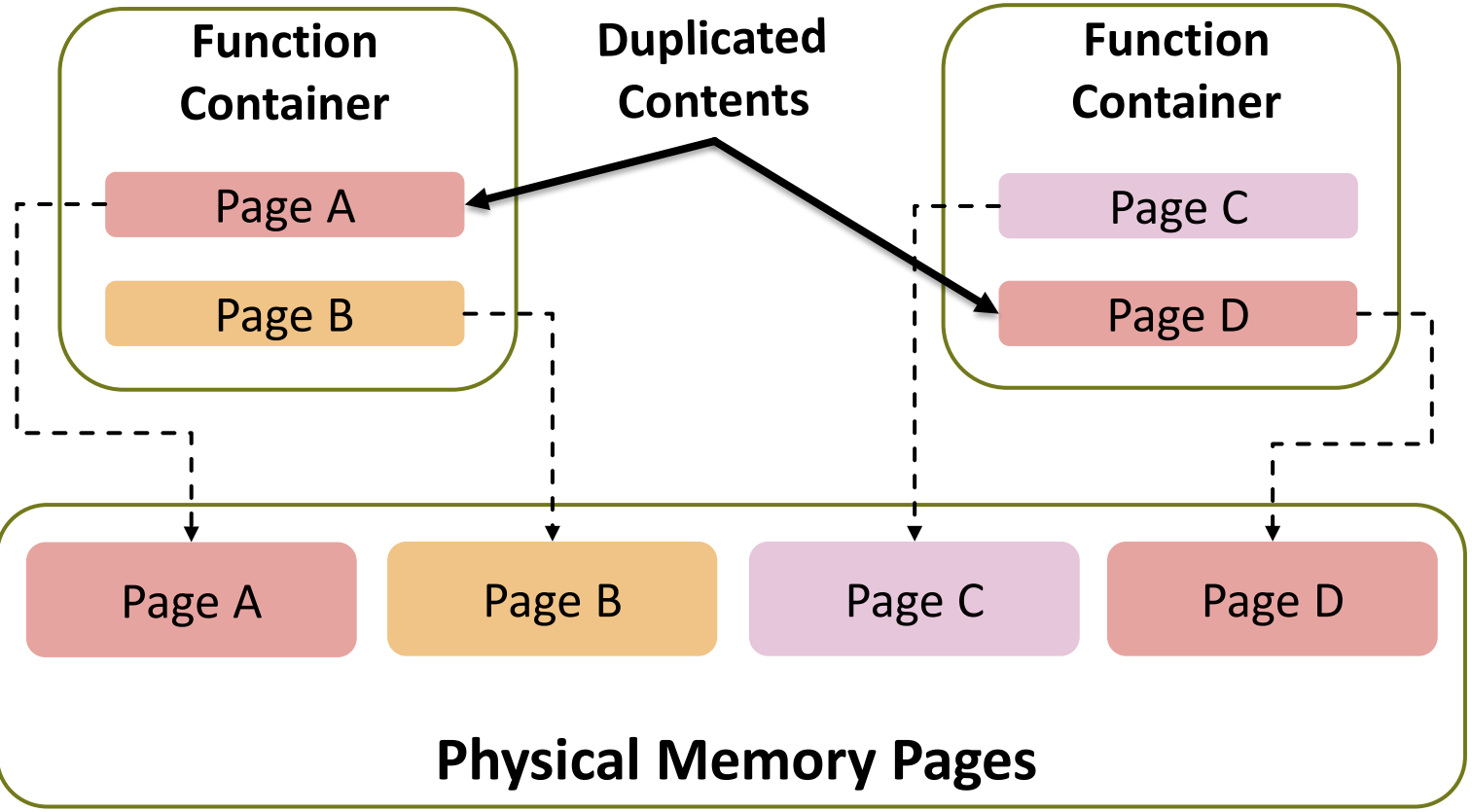


Example: Kernel Samepage Merging (KSM)

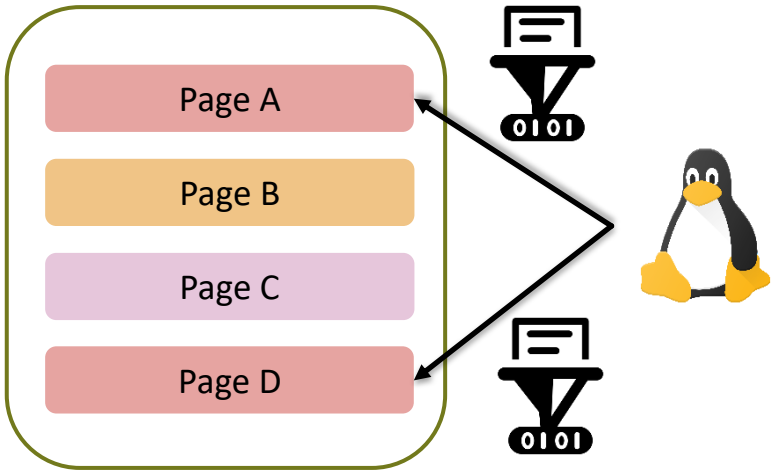


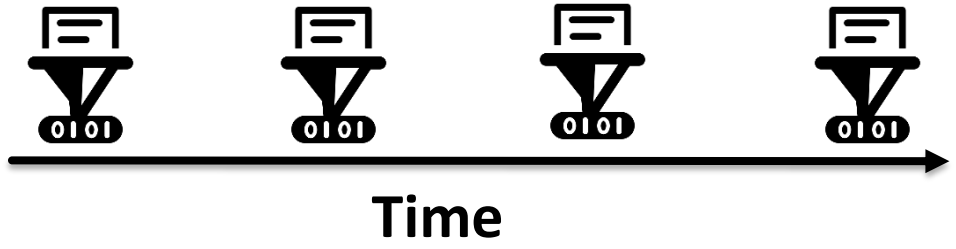
Memory Duplication in Serverless

IEEE BigData 2023
MSc Thesis



Example: Kernel Samepage Merging (KSM)





UPM: User-Guided Page Merging

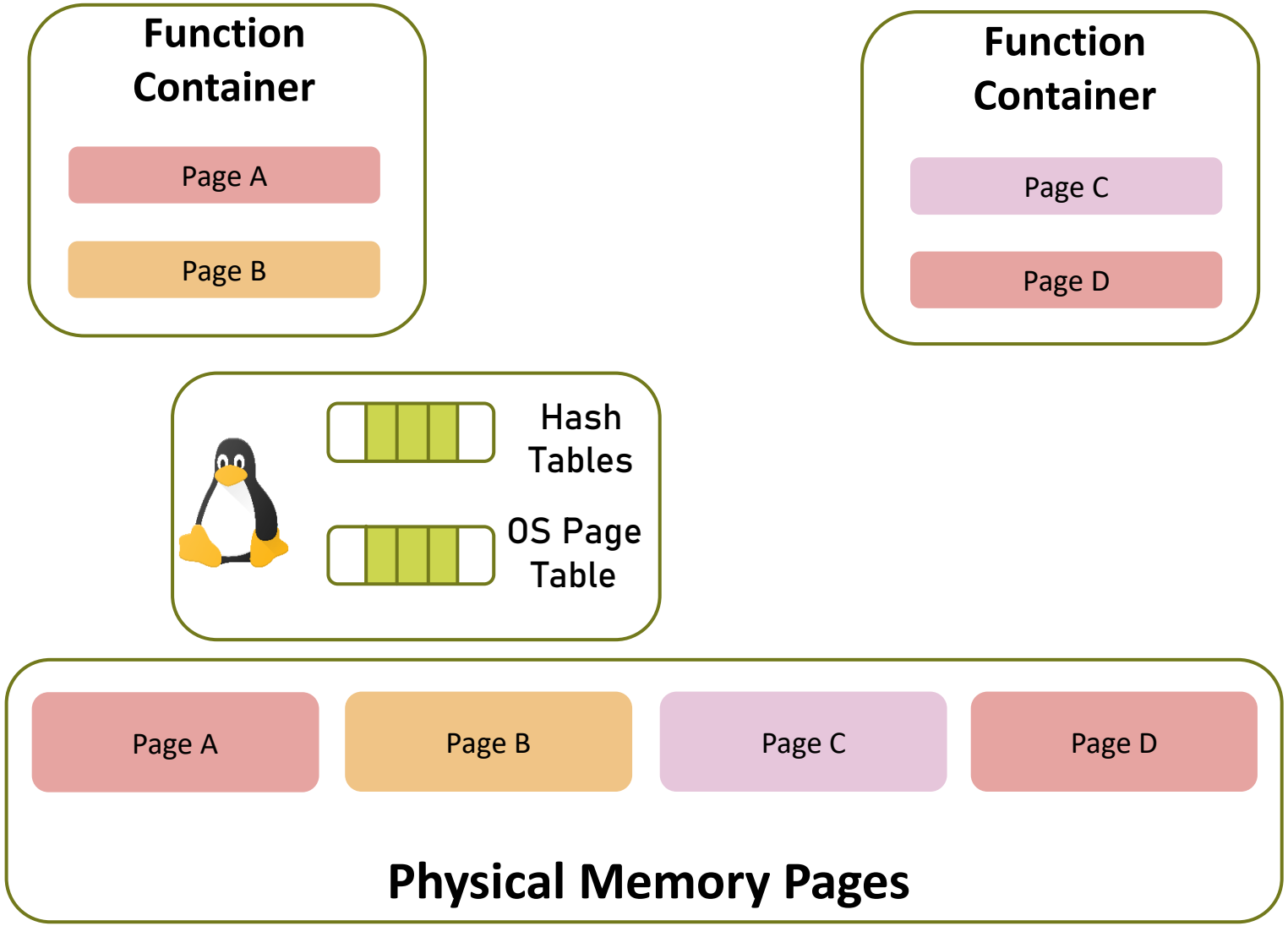
IEEE BigData 2023
MSc Thesis

UPM: User-Guided Page Merging

IEEE BigData 2023
MSc Thesis

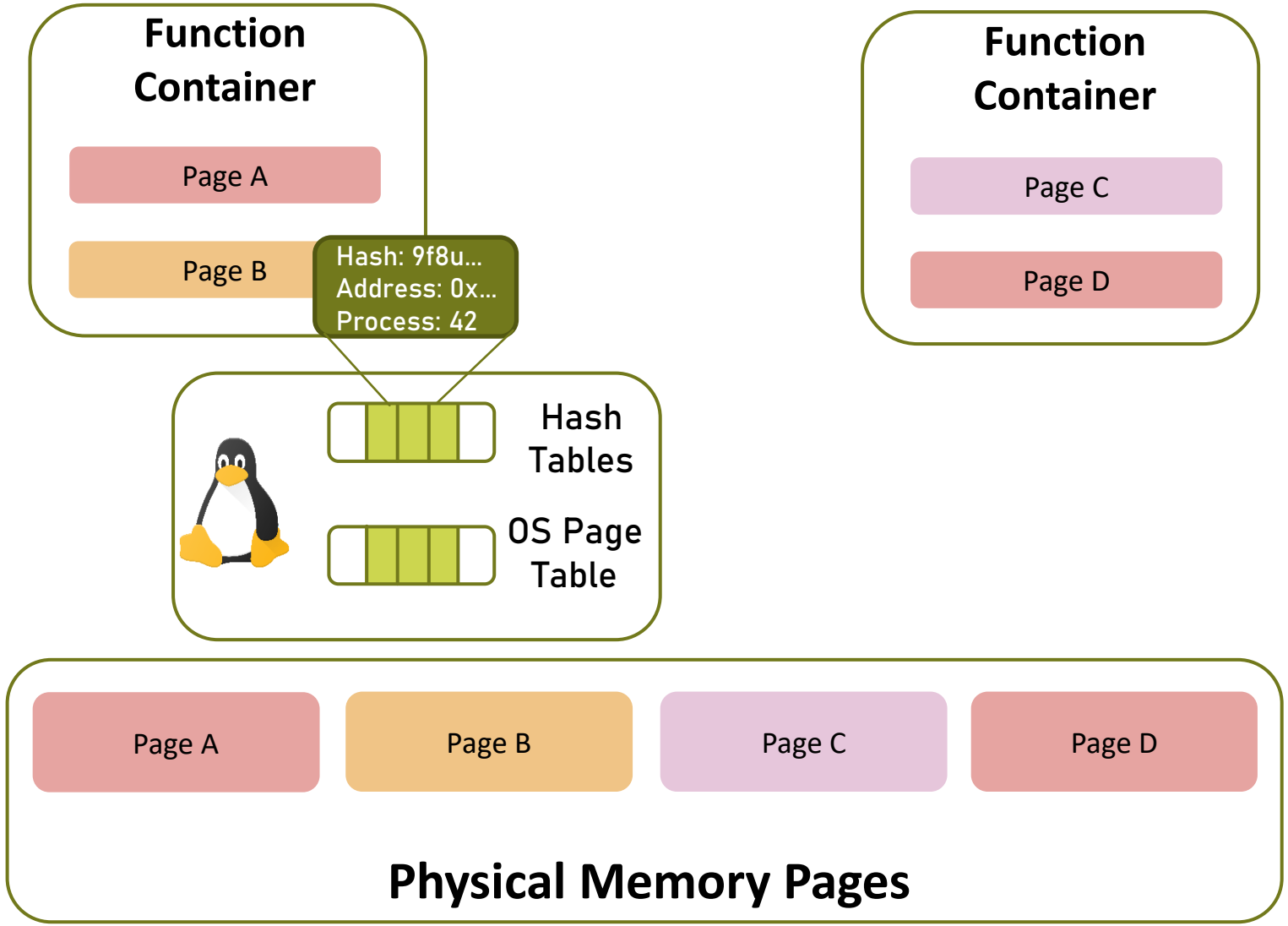
UPM: User-Guided Page Merging

IEEE BigData 2023
MSc Thesis



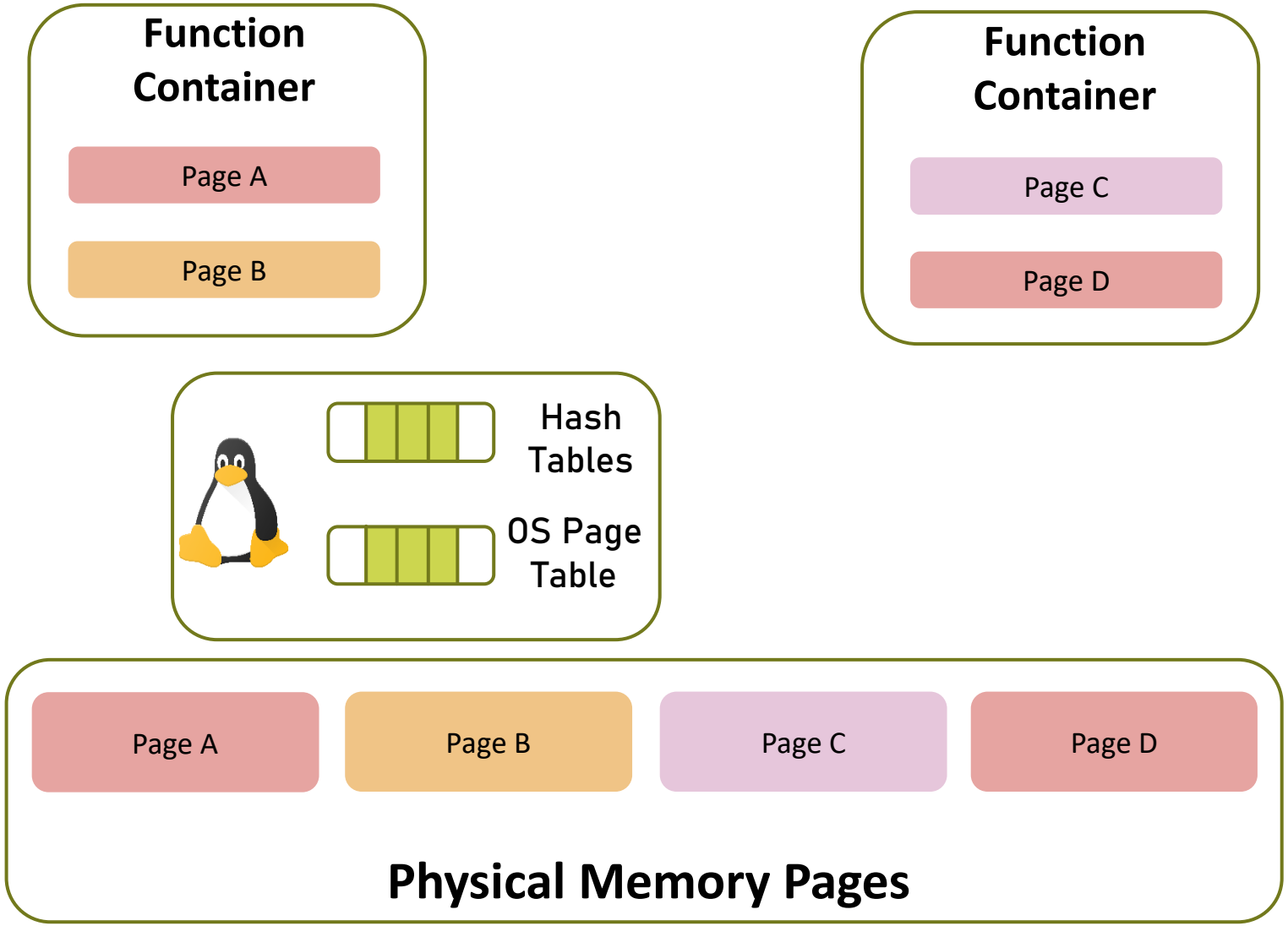
UPM: User-Guided Page Merging

IEEE BigData 2023
MSc Thesis



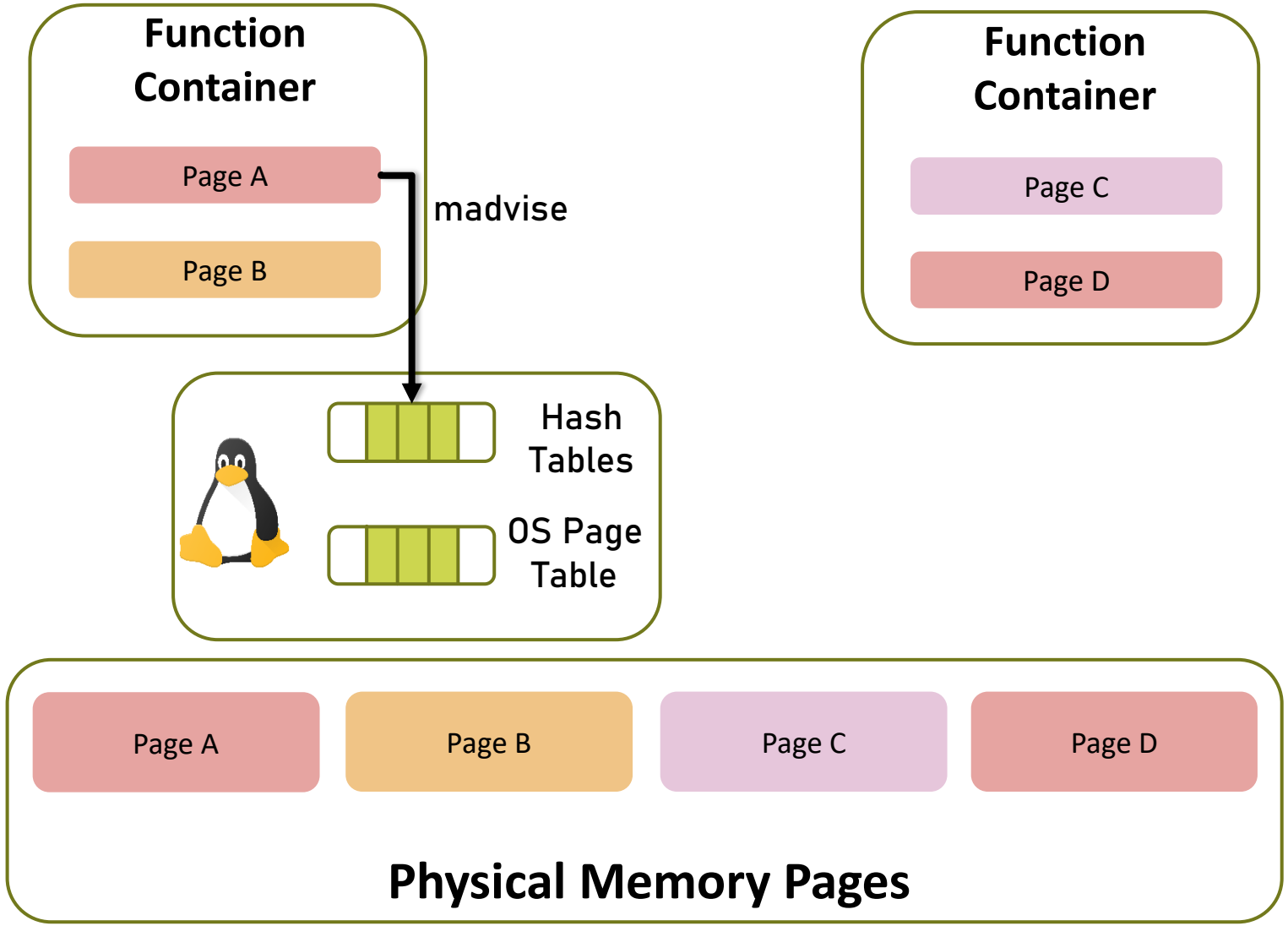
UPM: User-Guided Page Merging

IEEE BigData 2023
MSc Thesis



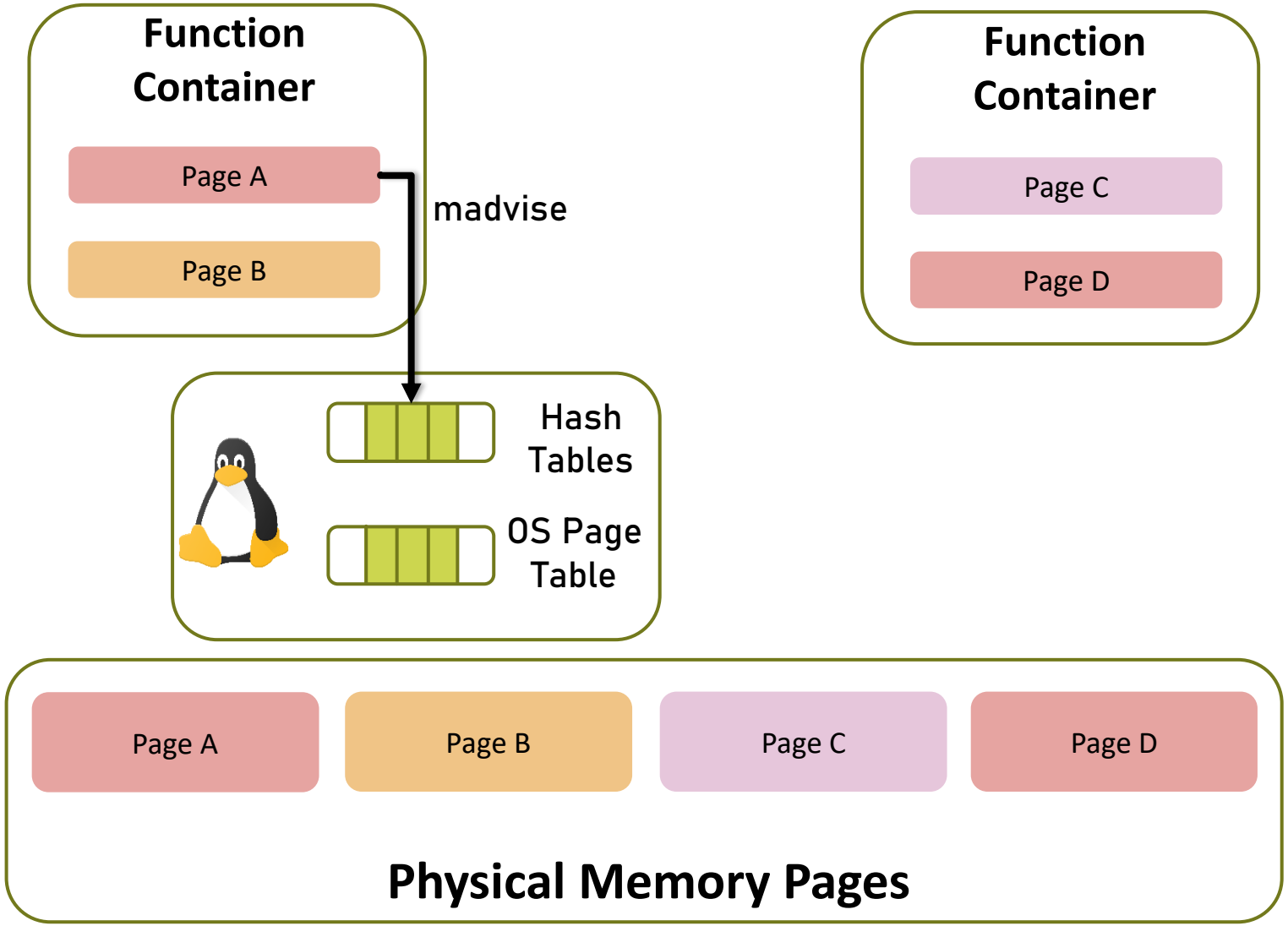
UPM: User-Guided Page Merging

IEEE BigData 2023
MSc Thesis



UPM: User-Guided Page Merging

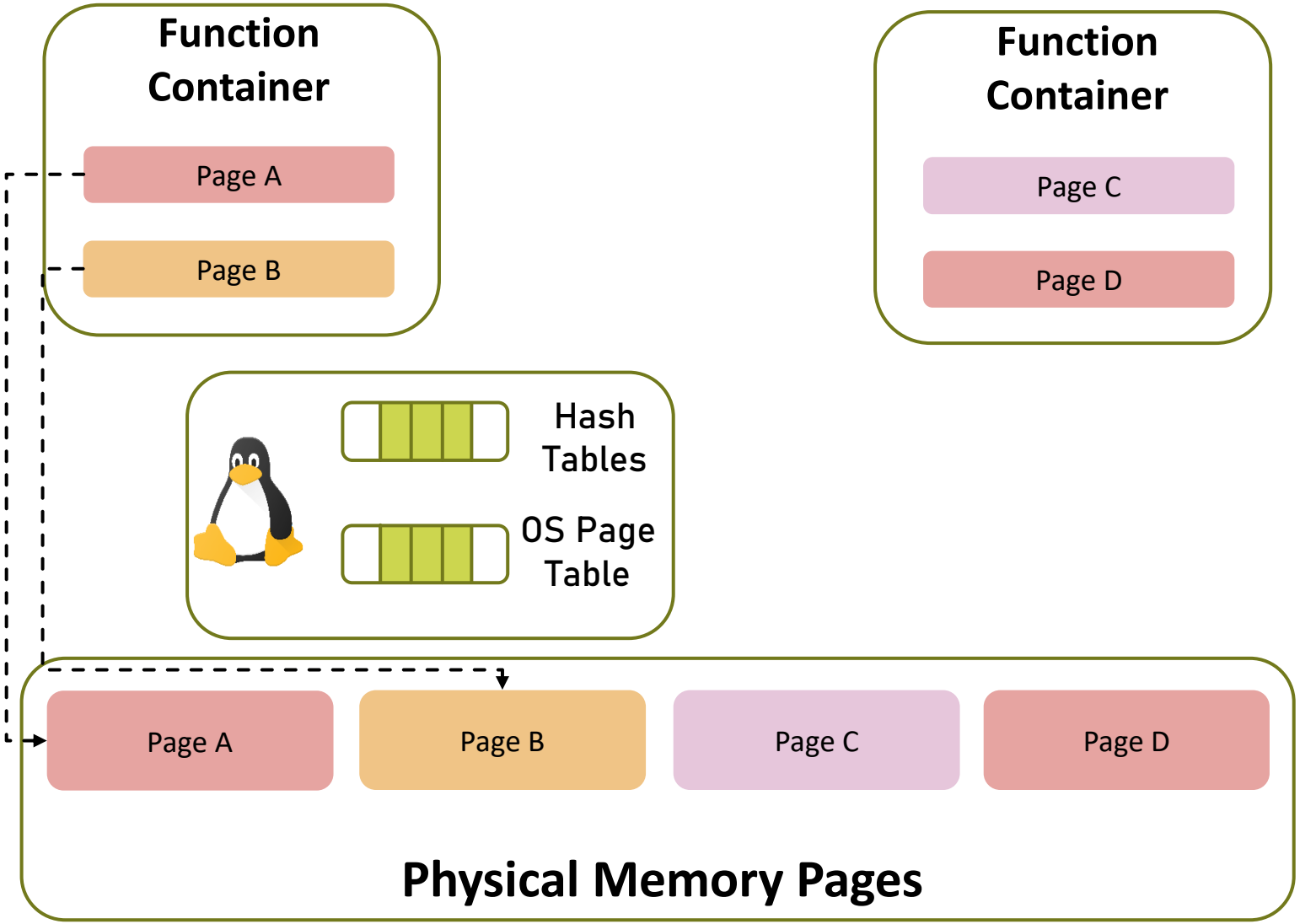
IEEE BigData 2023
MSc Thesis




```
int madvise(
    void *addr, size_t length, int advice
);
```

UPM: User-Guided Page Merging

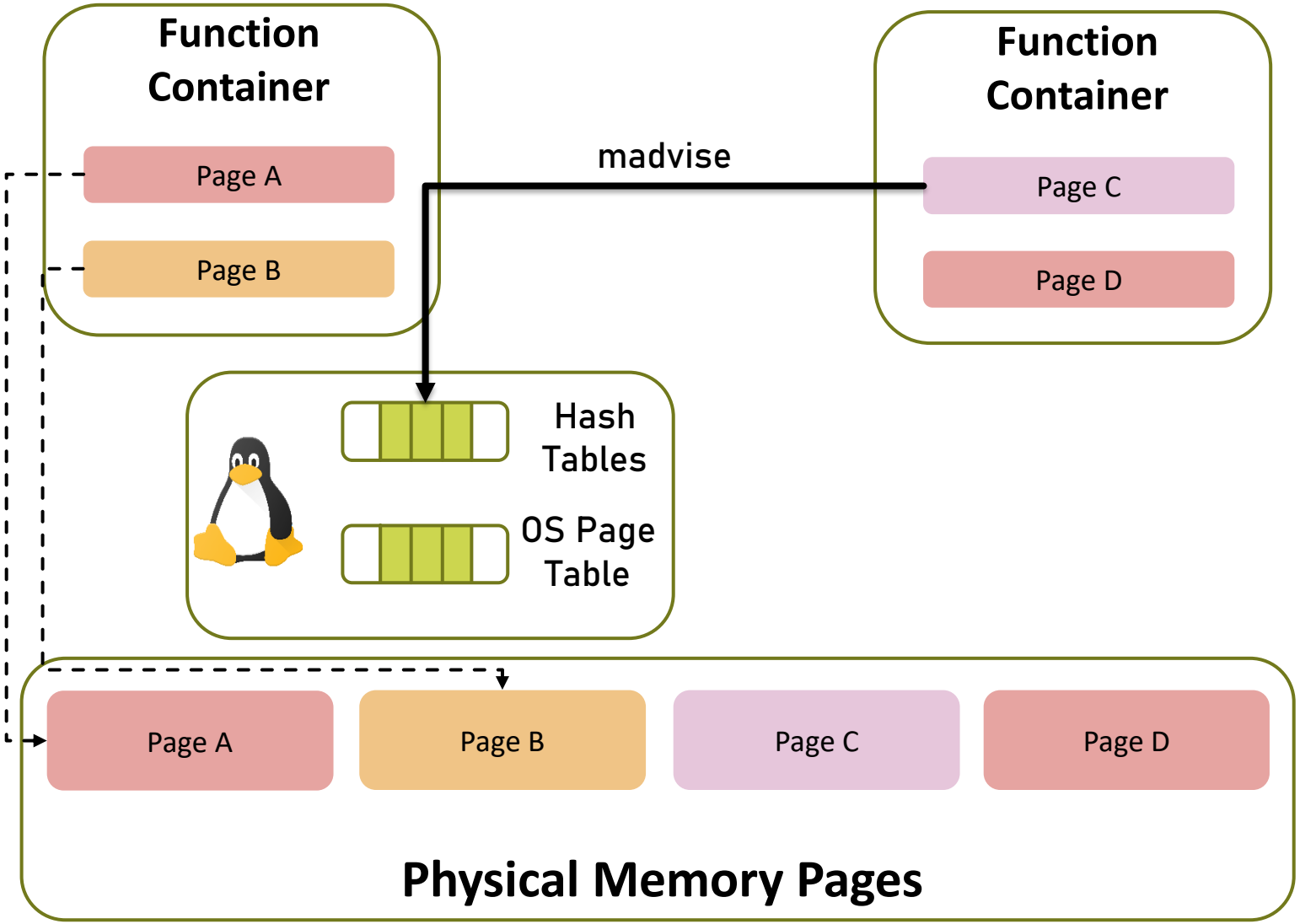
IEEE BigData 2023
MSc Thesis




```
int madvise(
    void *addr, size_t length, int advice
);
```

UPM: User-Guided Page Merging

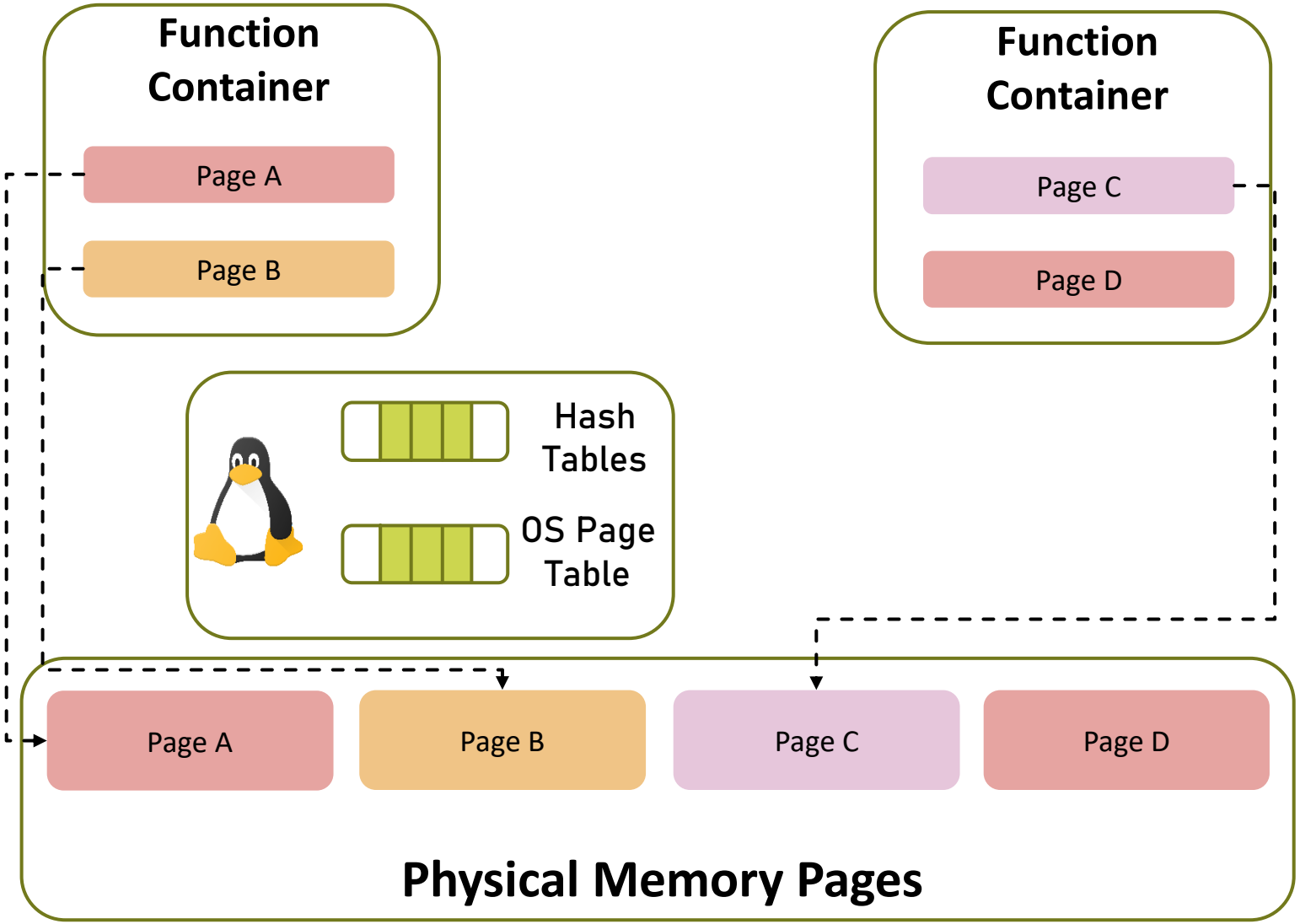
IEEE BigData 2023
MSc Thesis




```
int madvise(
    void *addr, size_t length, int advice
);
```

UPM: User-Guided Page Merging

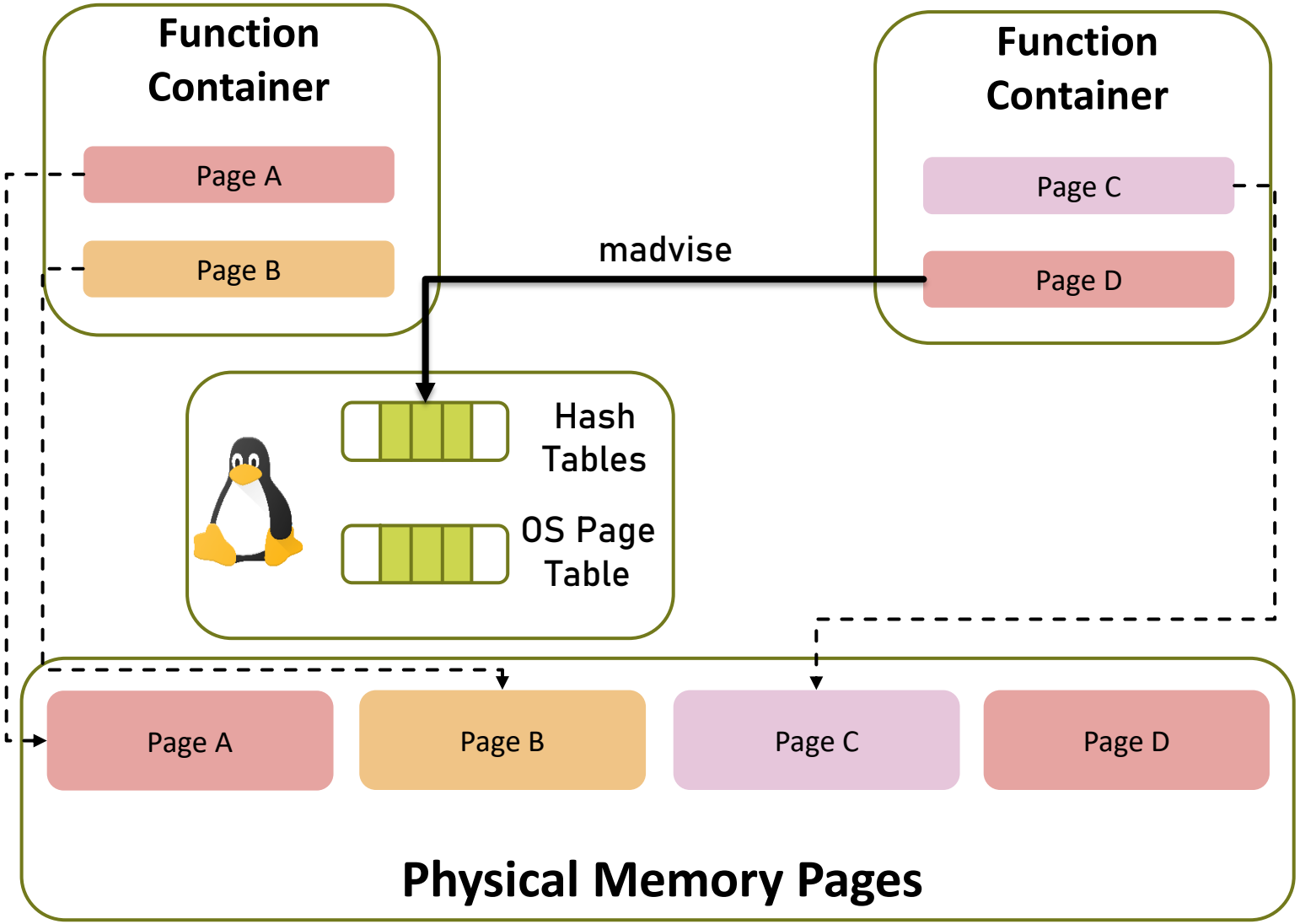
IEEE BigData 2023
MSc Thesis




```
int madvise(
    void *addr, size_t length, int advice
);
```

UPM: User-Guided Page Merging

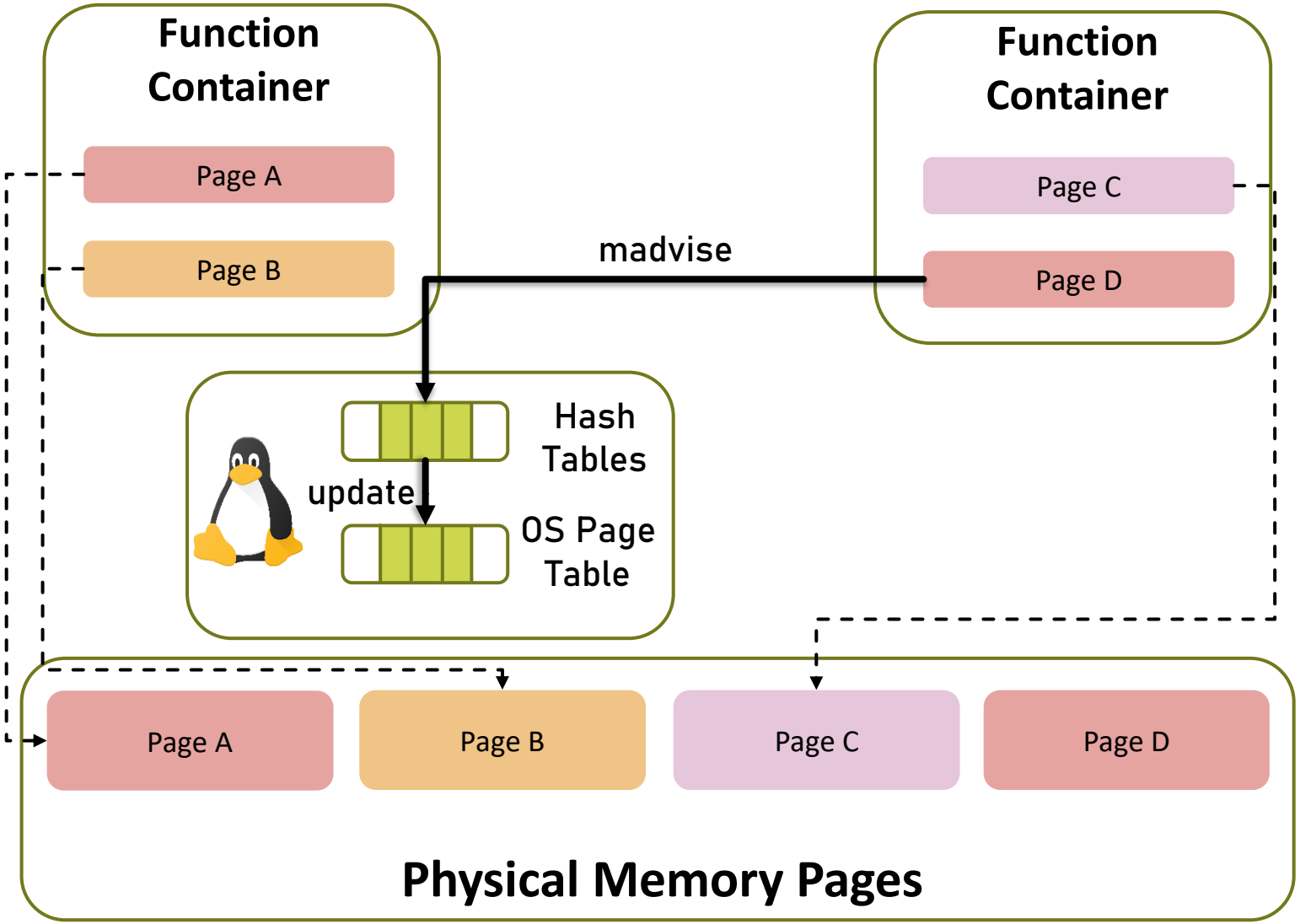
IEEE BigData 2023
MSc Thesis




```
int madvise(
    void *addr, size_t length, int advice
);
```

UPM: User-Guided Page Merging

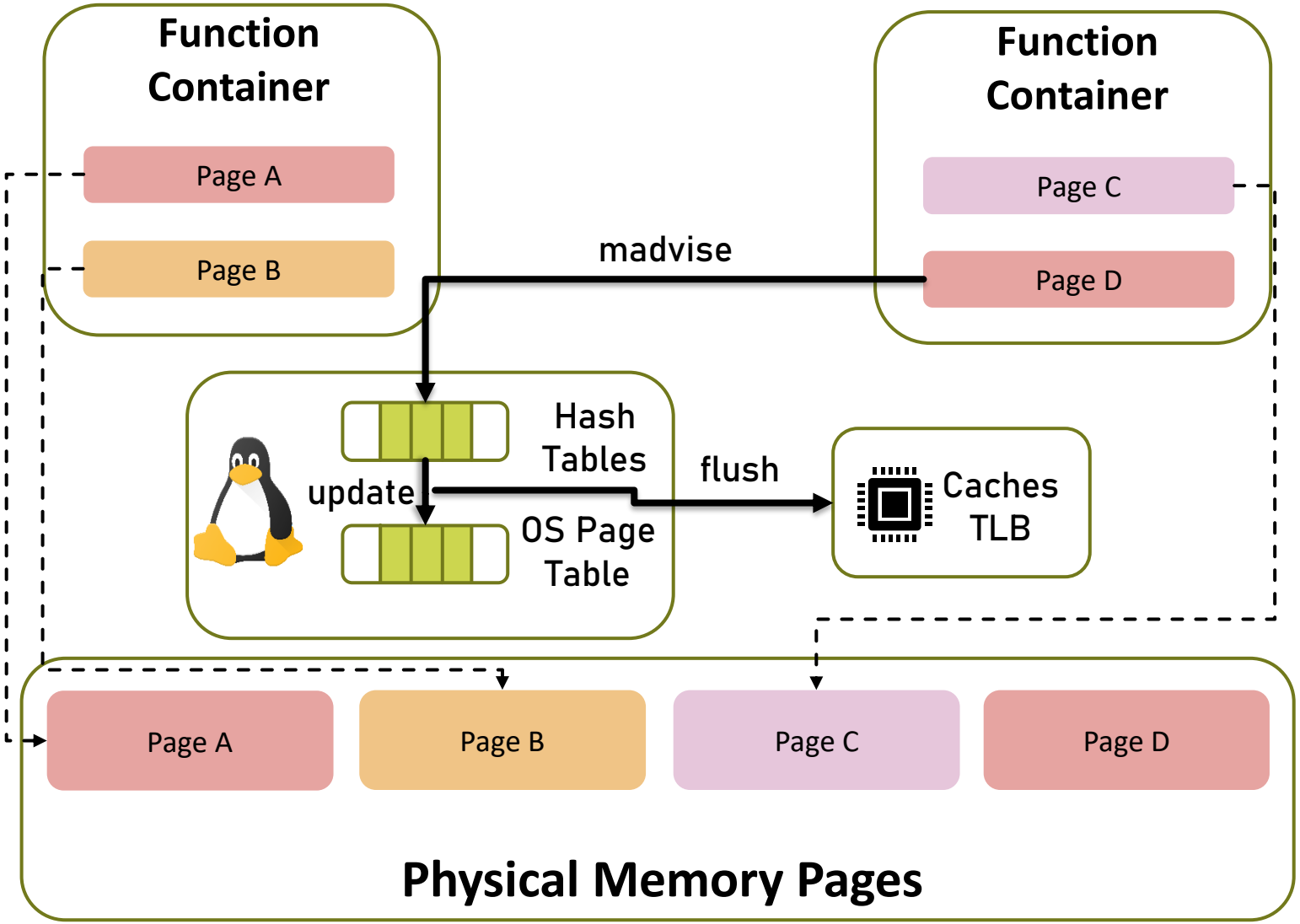
IEEE BigData 2023
MSc Thesis




```
int madvise(
    void *addr, size_t length, int advice
);
```

UPM: User-Guided Page Merging

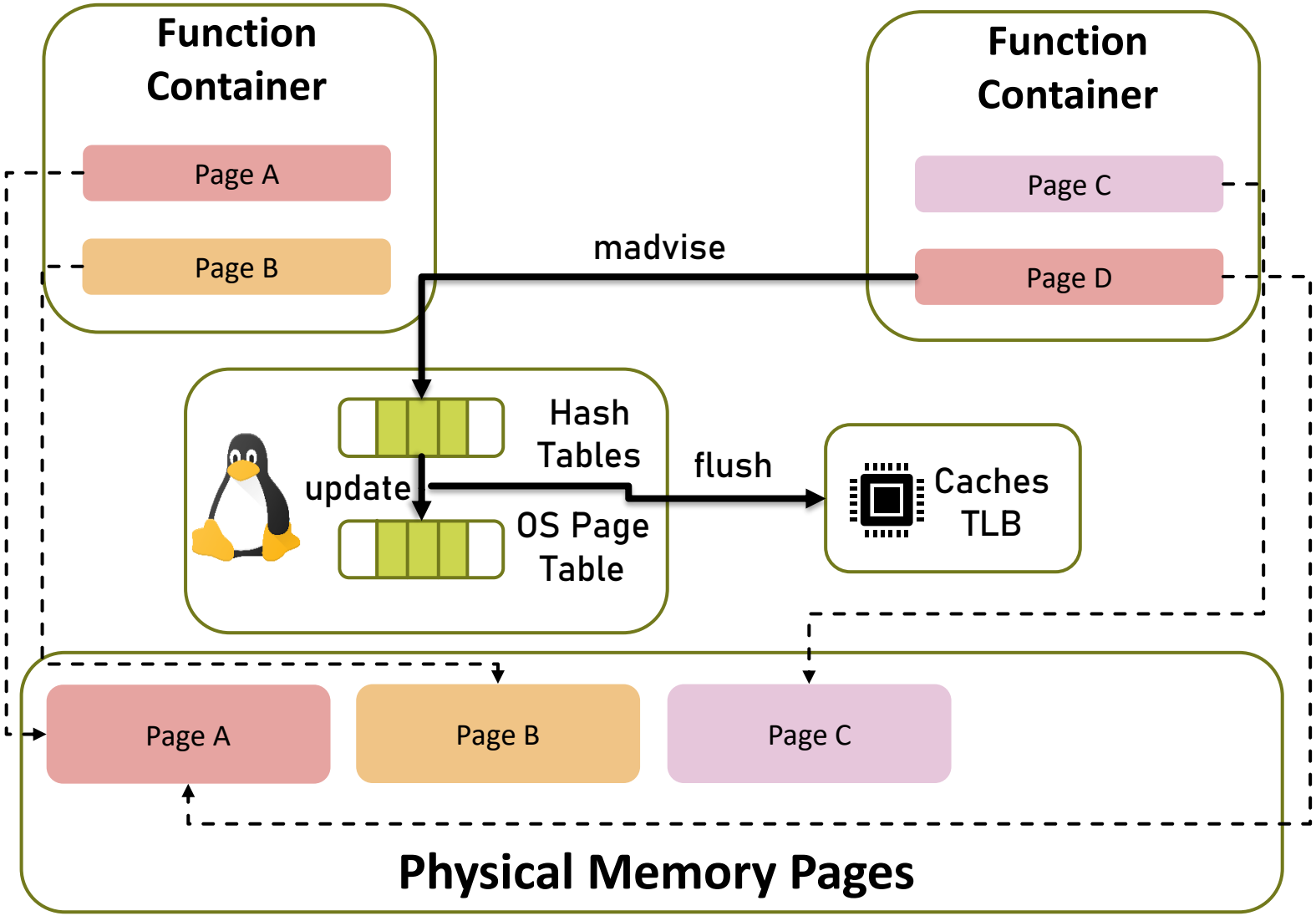
IEEE BigData 2023
MSc Thesis




```
int madvise(
    void *addr, size_t length, int advice
);
```

UPM: User-Guided Page Merging

IEEE BigData 2023
MSc Thesis

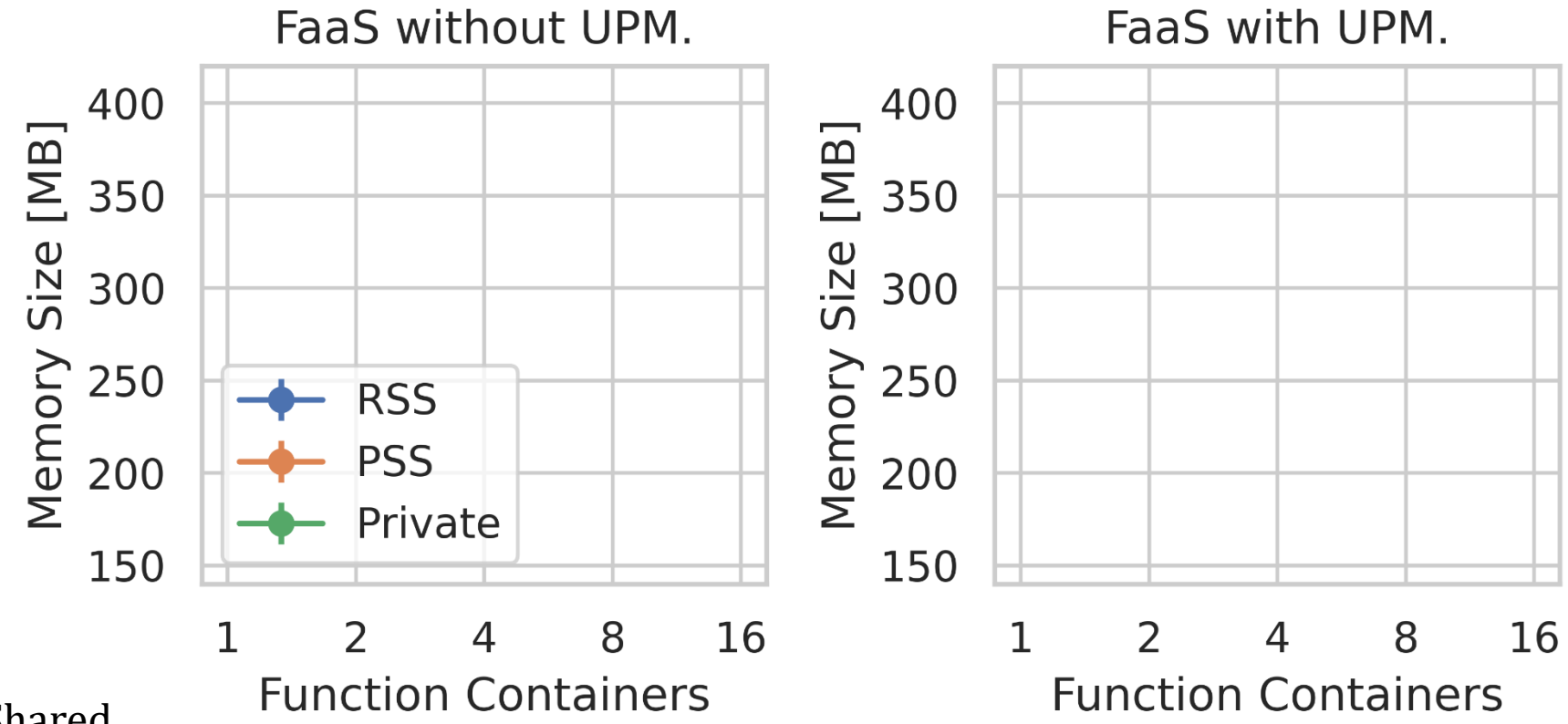



```
int madvise(
    void *addr, size_t length, int advice
);
```

Function Memory Footprint – AlexNet

IEEE BigData 2023
MSc Thesis

4x Intel Xeon X7550 @ 2.00GHz,
64 cores total. 1 TB memory.



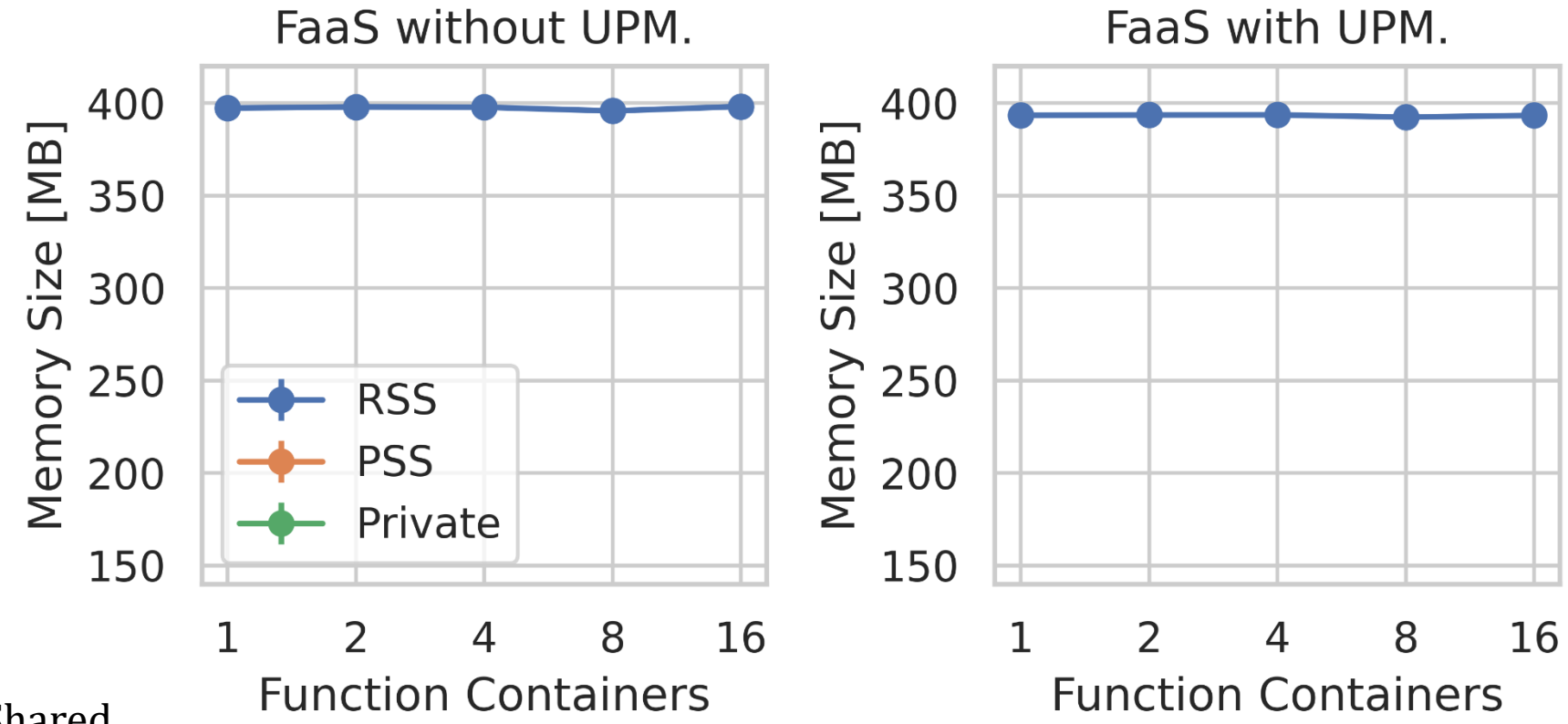
$$\text{RSS} = \text{Private} + \text{Shared}$$

$$\text{PSS} = \text{Private} + \frac{\text{Shared}}{\#\text{Processes}}$$

Function Memory Footprint – AlexNet

IEEE BigData 2023
MSc Thesis

4x Intel Xeon X7550 @ 2.00GHz,
64 cores total. 1 TB memory.



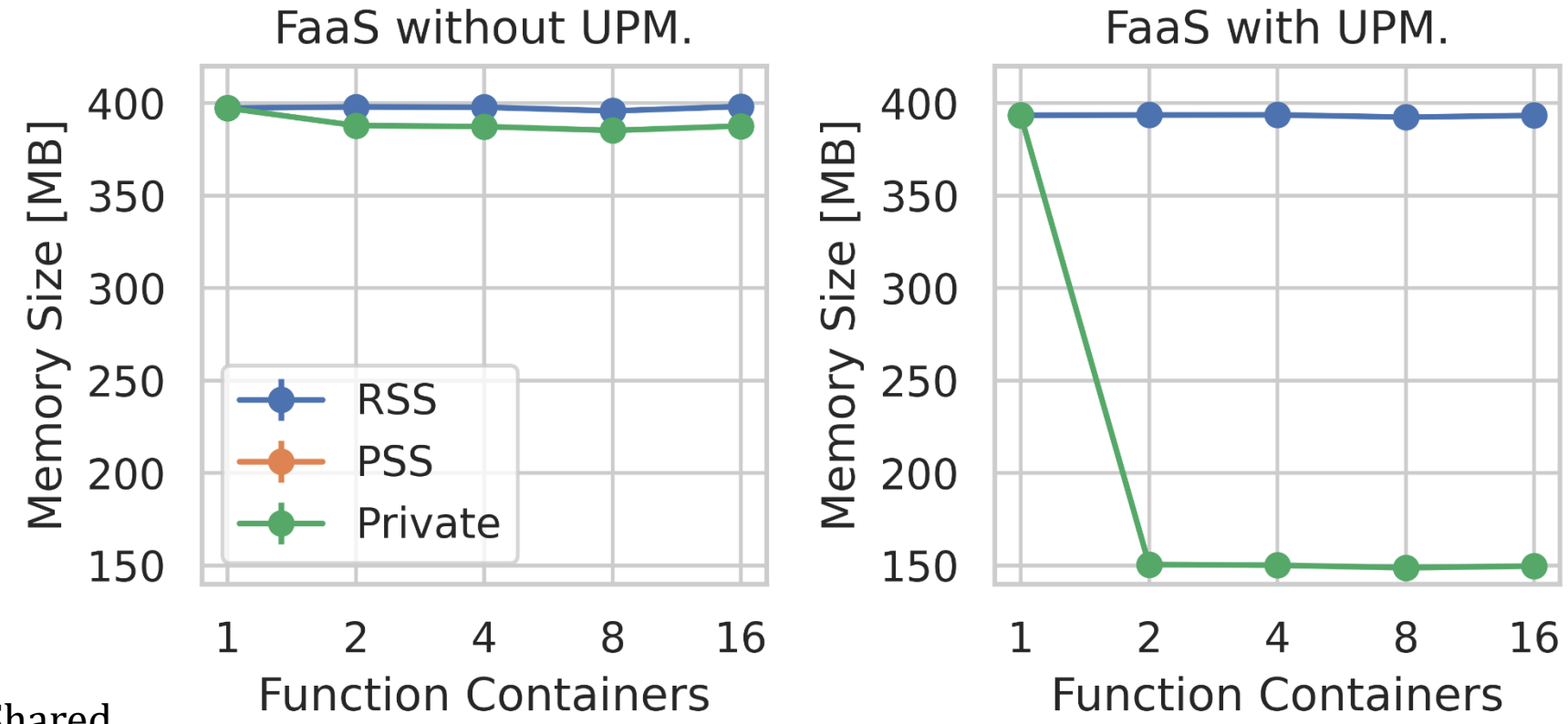
$$\text{RSS} = \text{Private} + \text{Shared}$$

$$\text{PSS} = \text{Private} + \frac{\text{Shared}}{\#\text{Processes}}$$

Function Memory Footprint – AlexNet

IEEE BigData 2023
MSc Thesis

4x Intel Xeon X7550 @ 2.00GHz,
64 cores total. 1 TB memory.



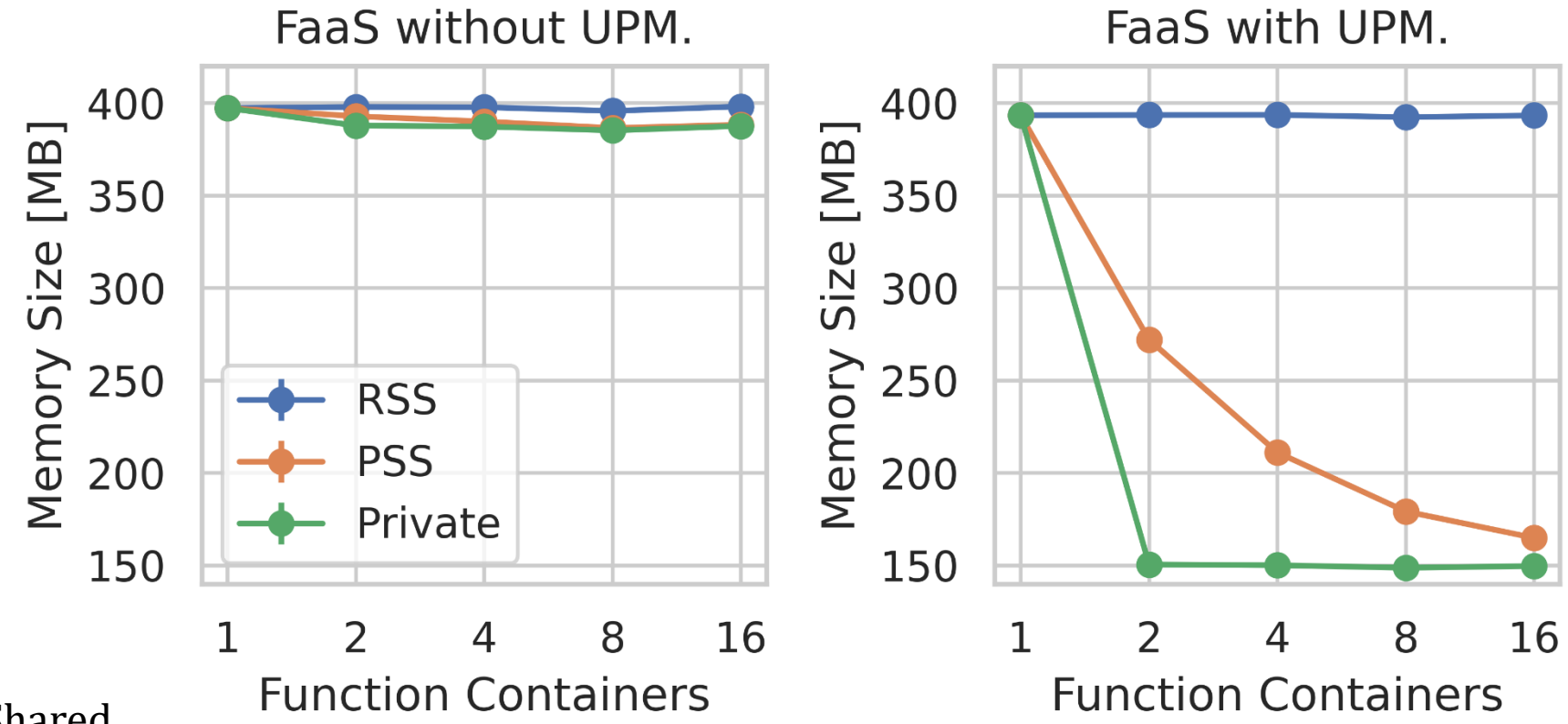
$$\text{RSS} = \text{Private} + \text{Shared}$$

$$\text{PSS} = \text{Private} + \frac{\text{Shared}}{\#\text{Processes}}$$

Function Memory Footprint – AlexNet

IEEE BigData 2023
MSc Thesis

4x Intel Xeon X7550 @ 2.00GHz,
64 cores total. 1 TB memory.



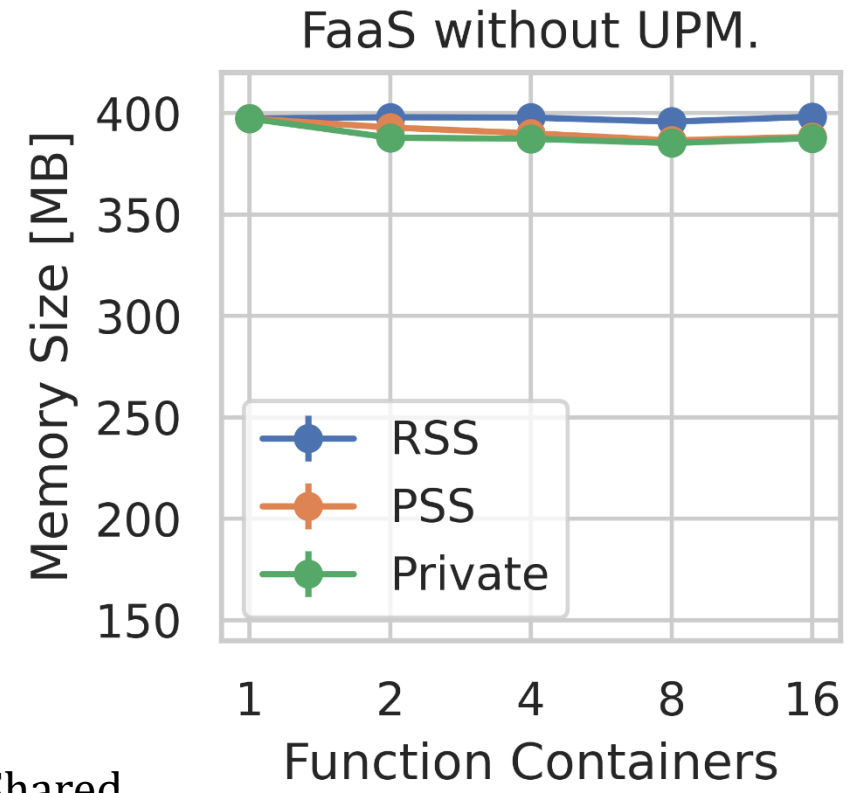
$$\text{RSS} = \text{Private} + \text{Shared}$$

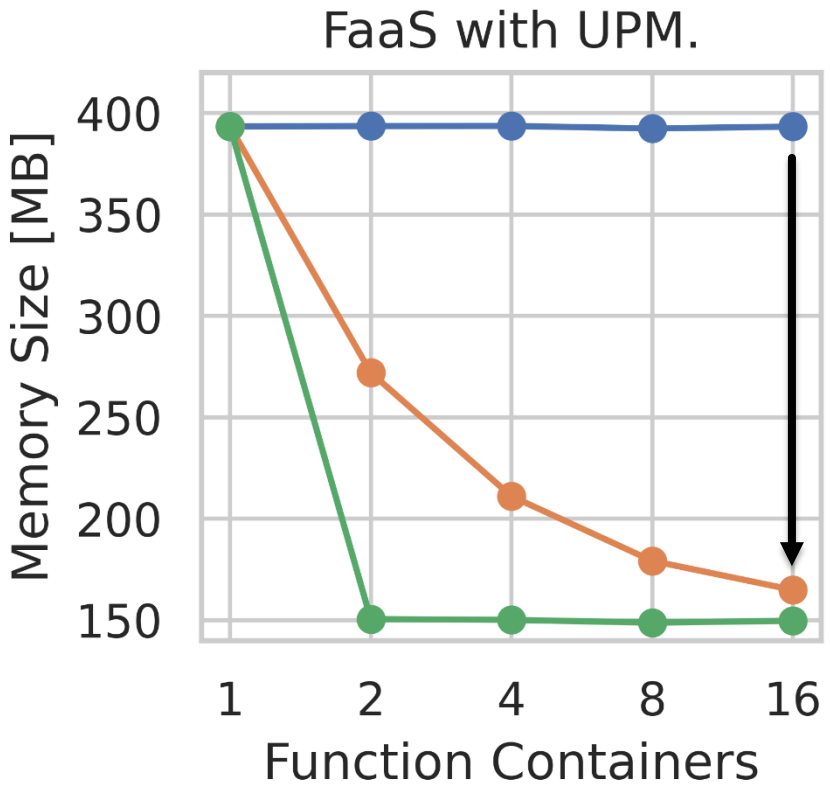
$$\text{PSS} = \text{Private} + \frac{\text{Shared}}{\#\text{Processes}}$$

Function Memory Footprint – AlexNet

IEEE BigData 2023
MSc Thesis

4x Intel Xeon X7550 @ 2.00GHz,
64 cores total. 1 TB memory.



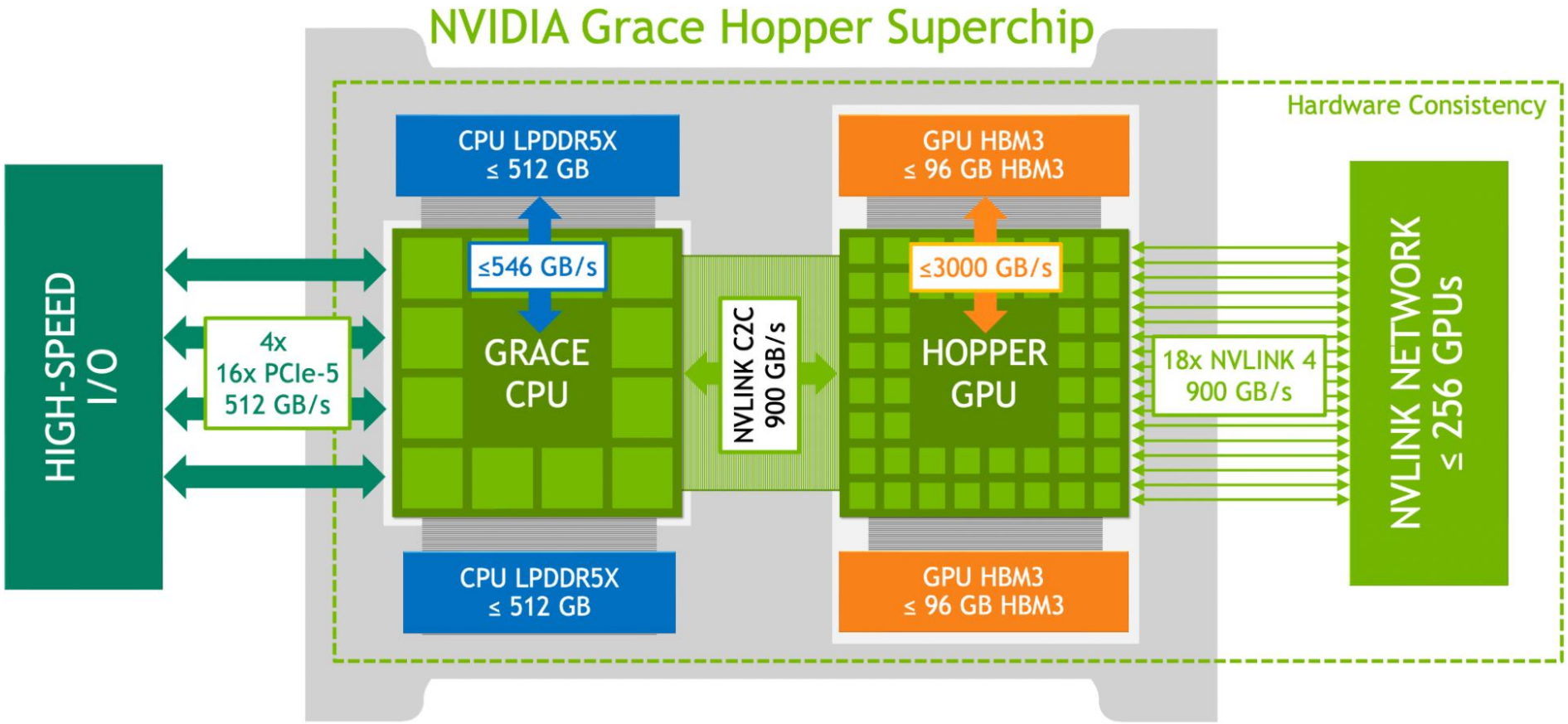


57.5% reduction

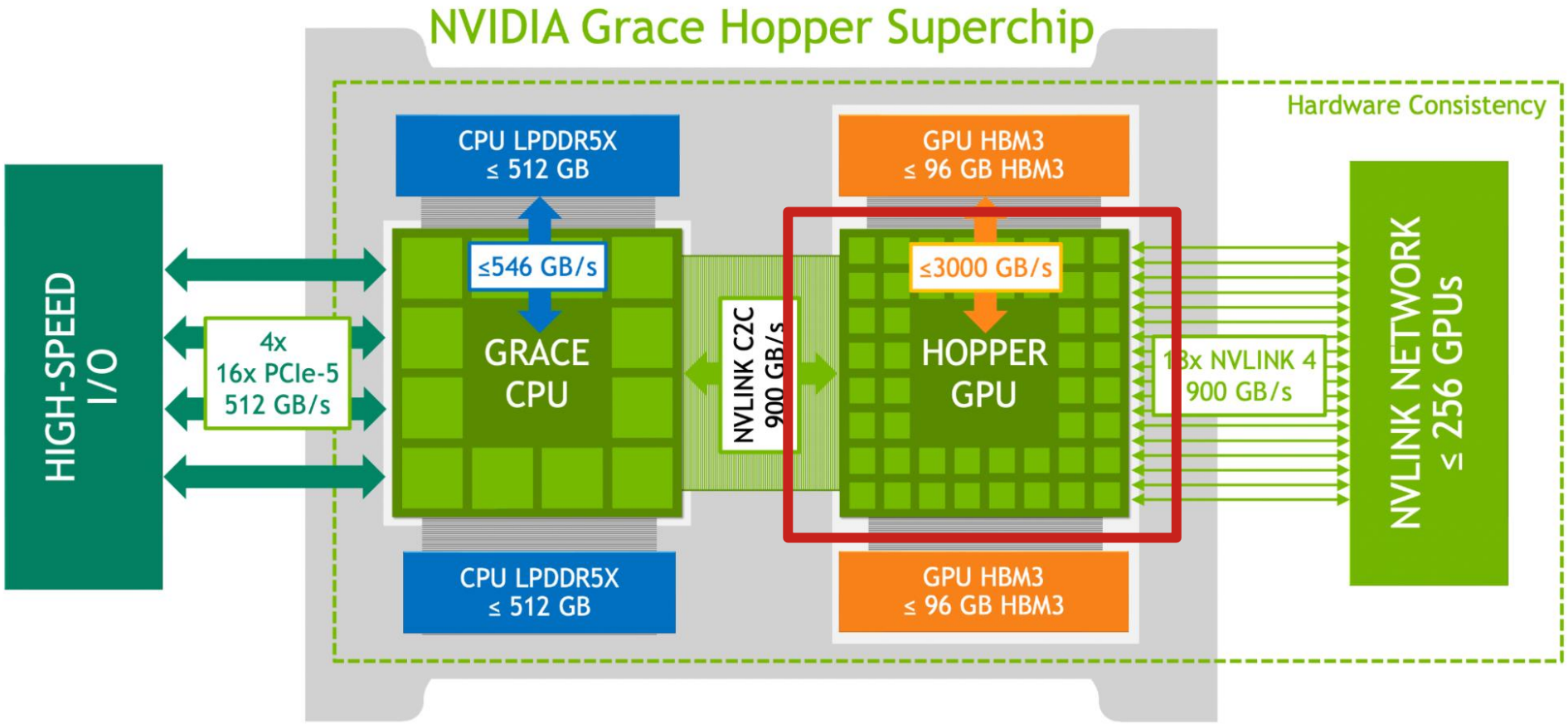
$$\text{RSS} = \text{Private} + \text{Shared}$$

$$\text{PSS} = \text{Private} + \frac{\text{Shared}}{\#\text{Processes}}$$

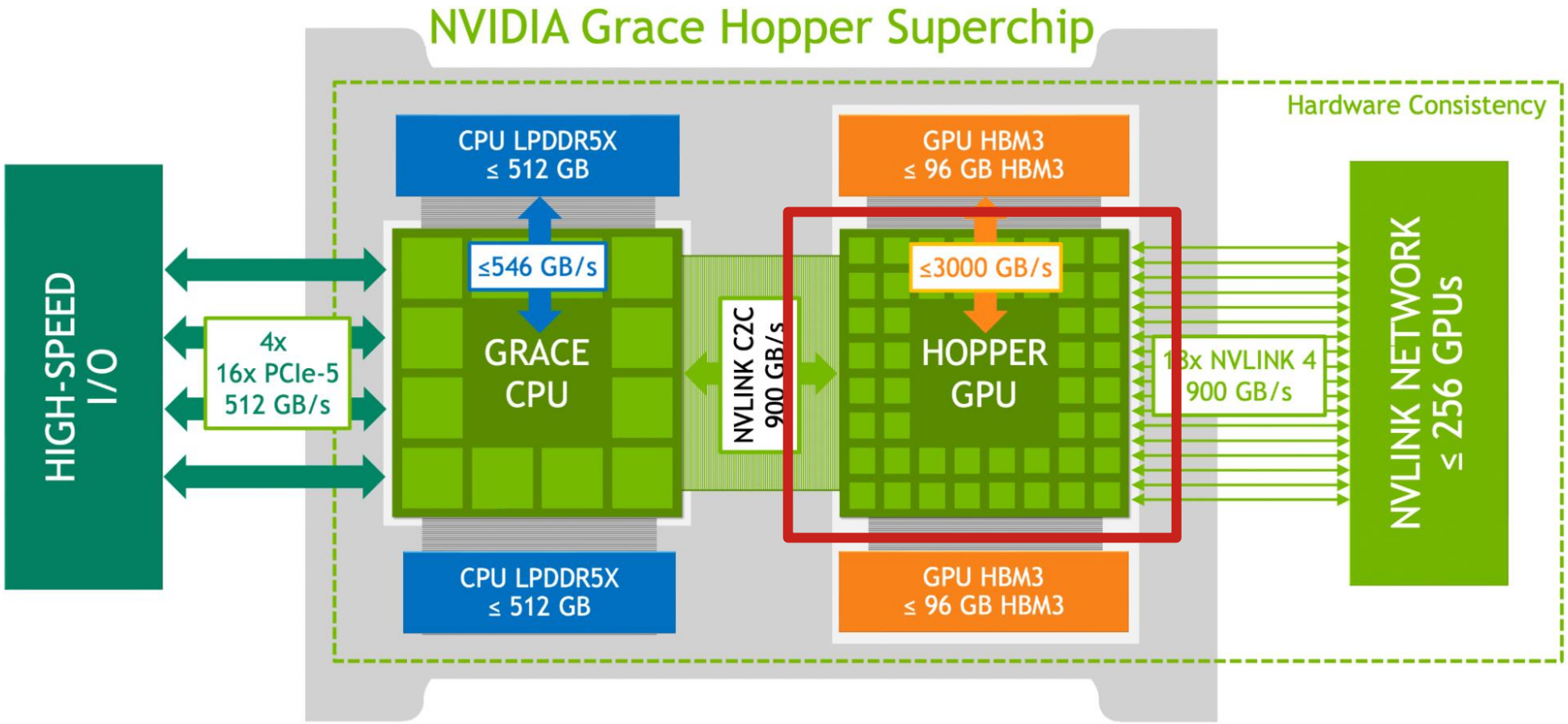
Function Memory Footprint – AlexNet



Function Memory Footprint – AlexNet

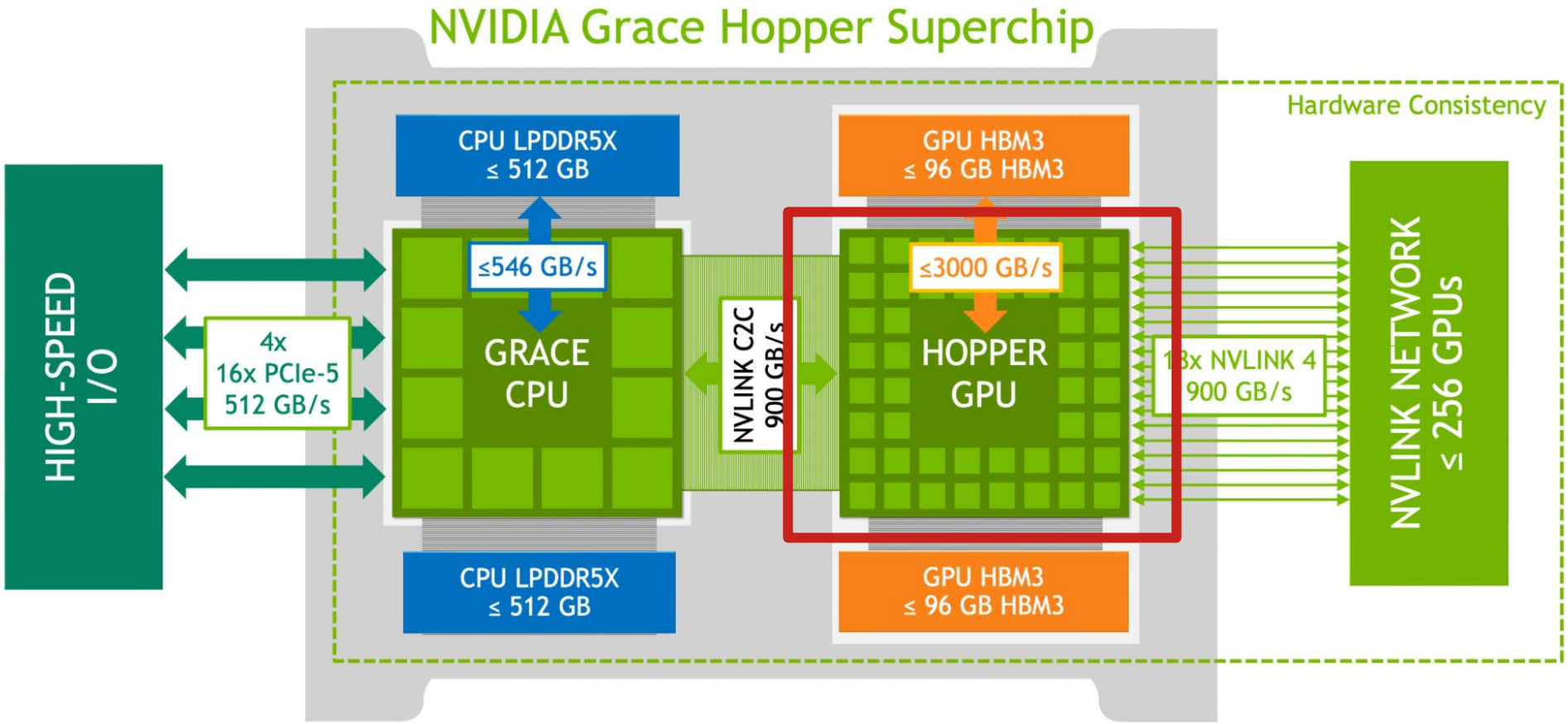


Function Memory Footprint – AlexNet



What is the deduplication potential?

Function Memory Footprint – AlexNet



What is the deduplication potential?

How can we implement copy-on-write on GPU?

High-Performance Serverless Solutions

High-Performance Serverless Solutions


spcl/serverless-benchmarks


spcl/rFaaS


spcl/FMI


spcl/PraaS


spcl/FaaSKeeper

Conclusions

More of SPCL's research:

youtube.com/@spcl — 180+ Talks

twitter.com/spcl_eth — 1.4K+ Followers

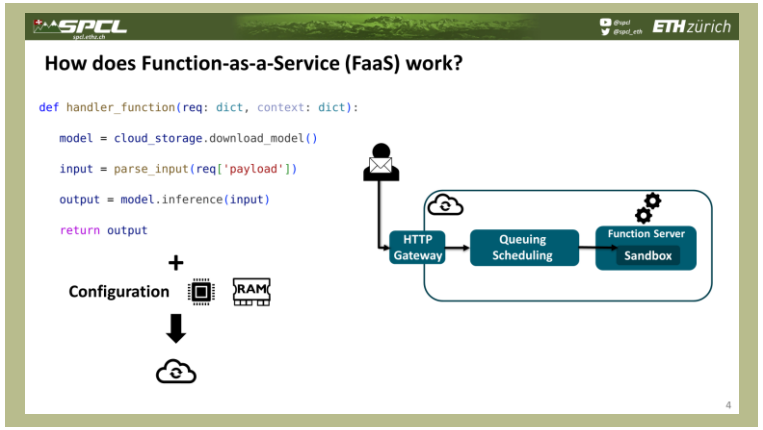
github.com/spcl — 3.8K+ Stars

… or spcl.ethz.ch

This work has received funding from the European Research Council (ERC).
We acknowledge support from the Swiss National Supercomputing Centre (CSCS).

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

Conclusions



More of SPCL's research:

youtube.com/@spcl — 180+ Talks

twitter.com/spcl_eth — 1.4K+ Followers

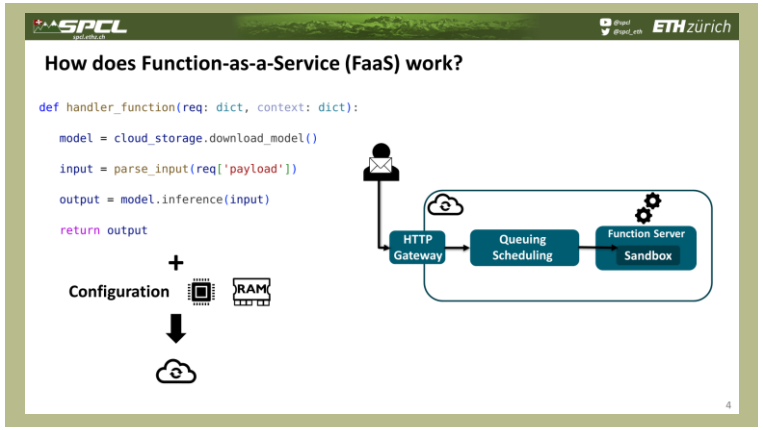
github.com/spcl — 3.8K+ Stars

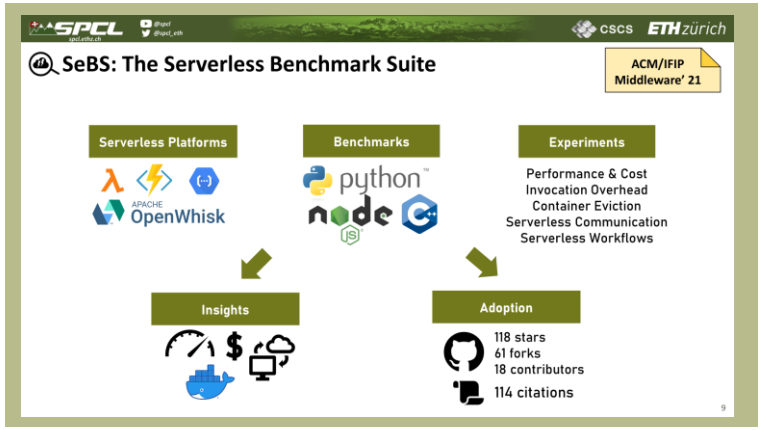
… or spcl.ethz.ch

This work has received funding from the European Research Council (ERC).
We acknowledge support from the Swiss National Supercomputing Centre (CSCS).

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

Conclusions





More of SPCL's research:

youtube.com/@spcl — 180+ Talks

twitter.com/spcl_eth — 1.4K+ Followers

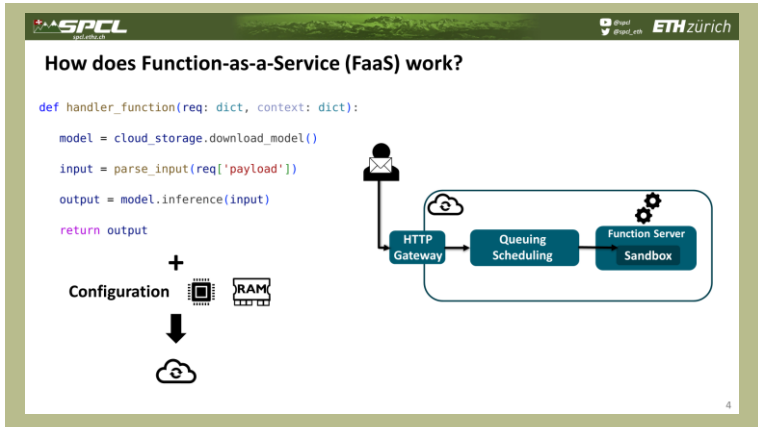
github.com/spcl — 3.8K+ Stars

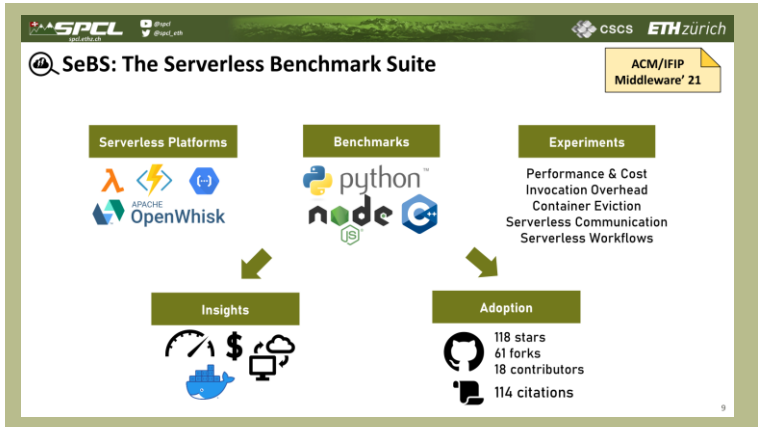
… or spcl.ethz.ch

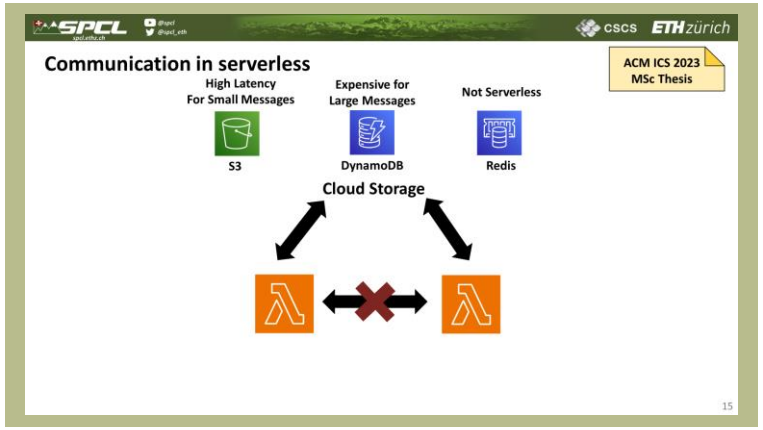
This work has received funding from the European Research Council (ERC).
We acknowledge support from the Swiss National Supercomputing Centre (CSCS).

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

Conclusions







More of SPCL's research:

youtube.com/@spcl — 180+ Talks

twitter.com/spcl_eth — 1.4K+ Followers

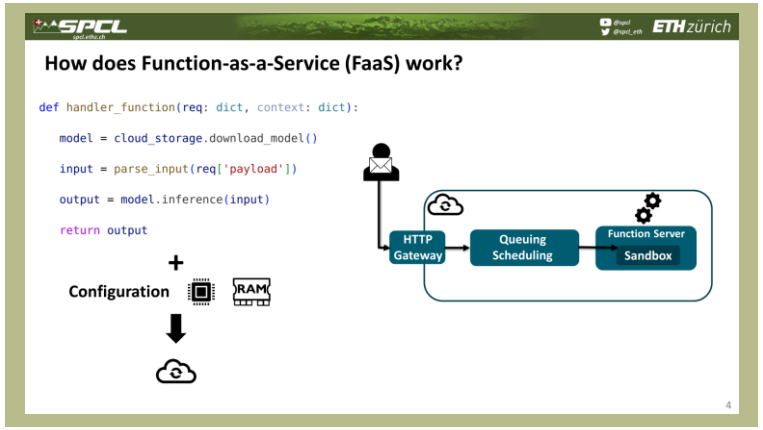
github.com/spcl — 3.8K+ Stars

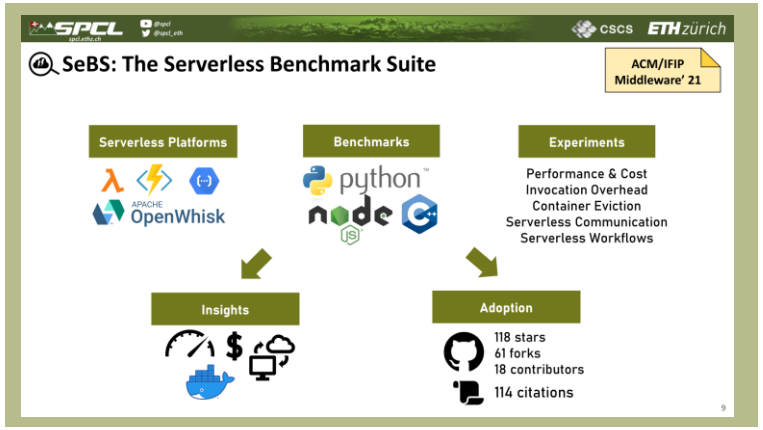
… or spcl.ethz.ch

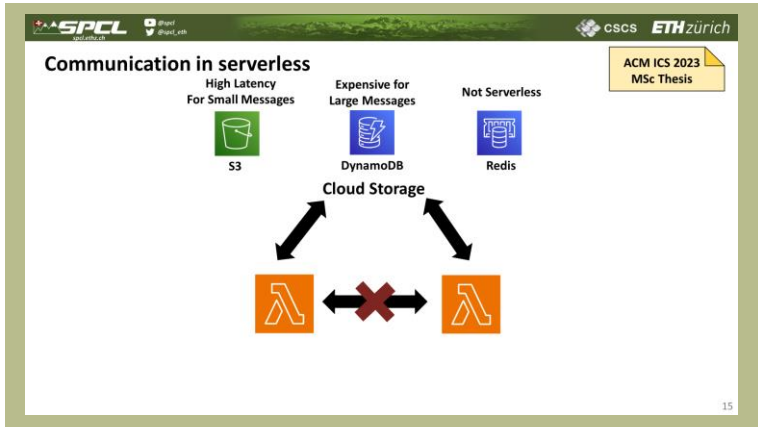
This work has received funding from the European Research Council (ERC).
We acknowledge support from the Swiss National Supercomputing Centre (CSCS).

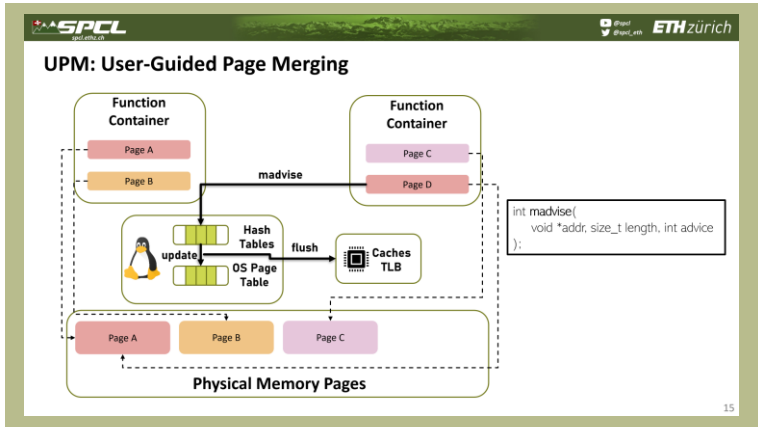
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

Conclusions









More of SPCL's research:

youtube.com/@spcl — 180+ Talks

twitter.com/spcl_eth — 1.4K+ Followers

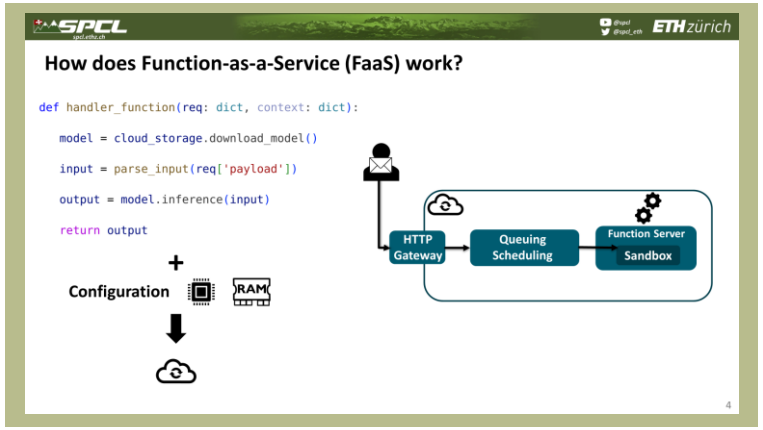
github.com/spcl — 3.8K+ Stars

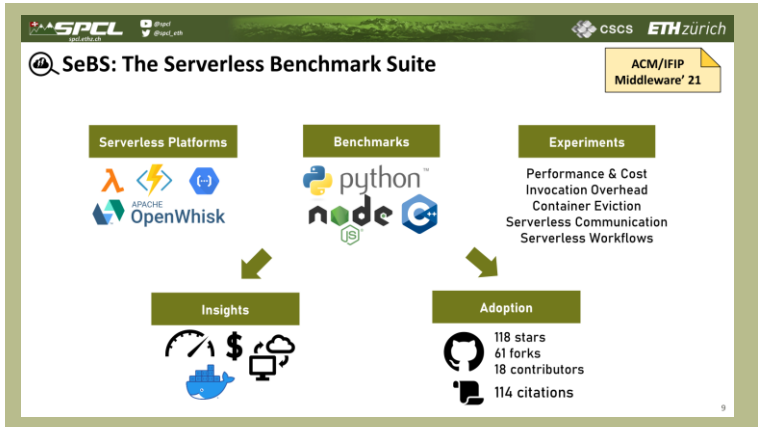
… or spcl.ethz.ch

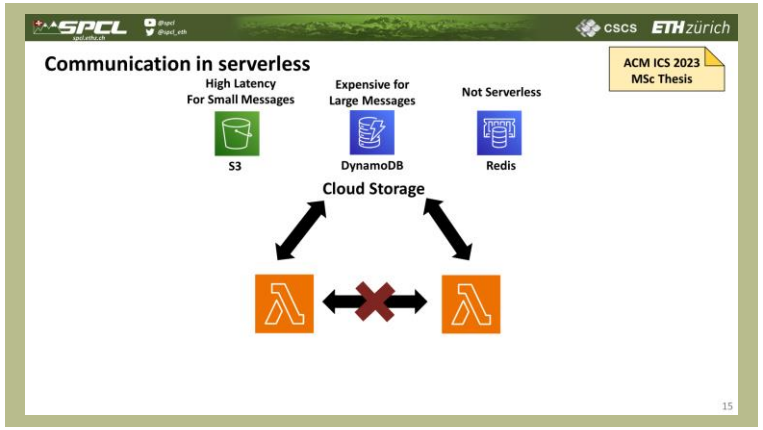
This work has received funding from the European Research Council (ERC).
We acknowledge support from the Swiss National Supercomputing Centre (CSCS).

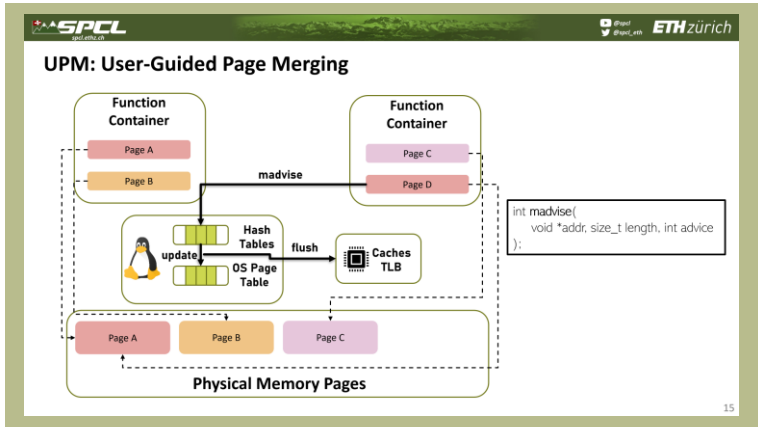
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

Conclusions









More of SPCL's research:

youtube.com/@spcl — 180+ Talks

twitter.com/spcl_eth — 1.4K+ Followers

github.com/spcl — 3.8K+ Stars

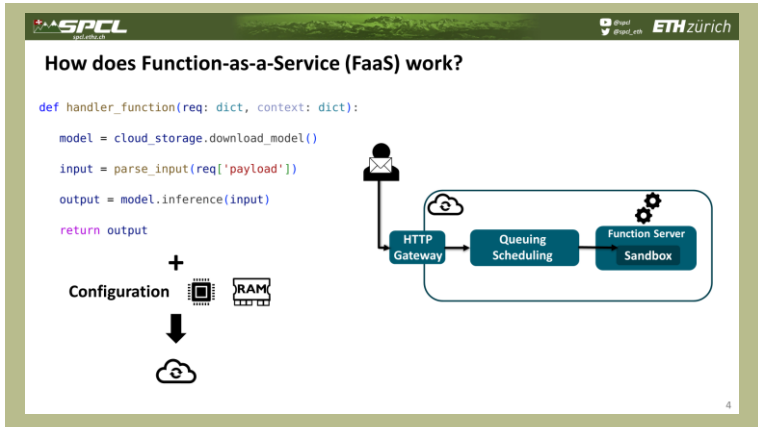
… or spcl.ethz.ch

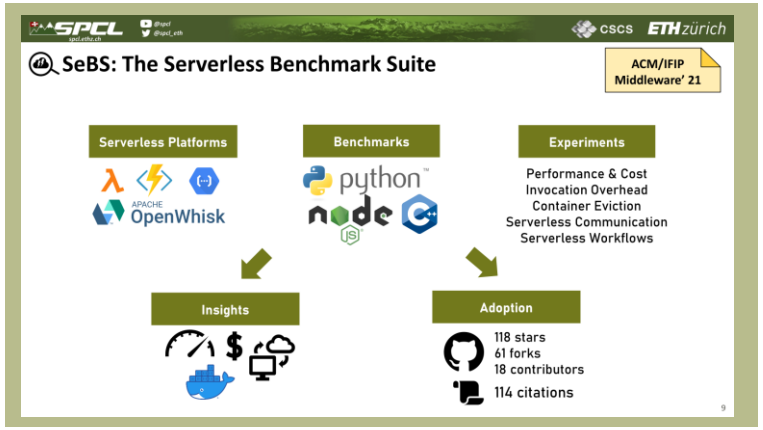
Other Projects

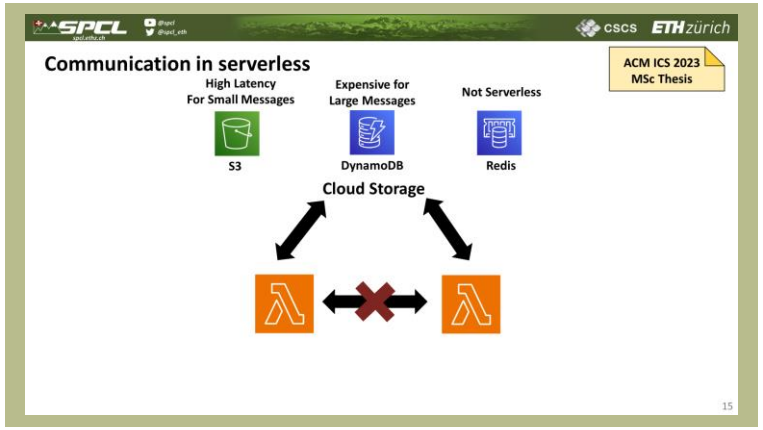
This work has received funding from the European Research Council (ERC).
We acknowledge support from the Swiss National Supercomputing Centre (CSCS).

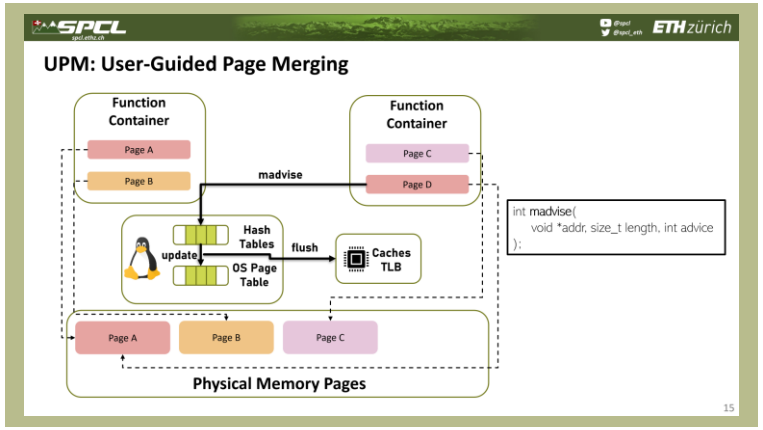
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

Conclusions









More of SPCL's research:

youtube.com/@spcl — 180+ Talks

twitter.com/spcl_eth — 1.4K+ Followers

github.com/spcl — 3.8K+ Stars

… or spcl.ethz.ch

Other Projects

Questions?

marcin.copik@inf.ethz.ch

This work has received funding from the European Research Council (ERC).
We acknowledge support from the Swiss National Supercomputing Centre (CSCS).

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre