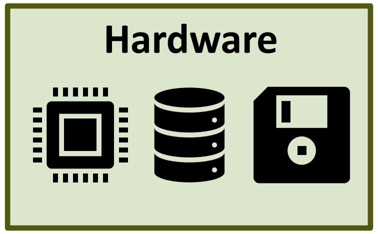
FMI: Fast and Cheap Message Passing for Serverless Functions

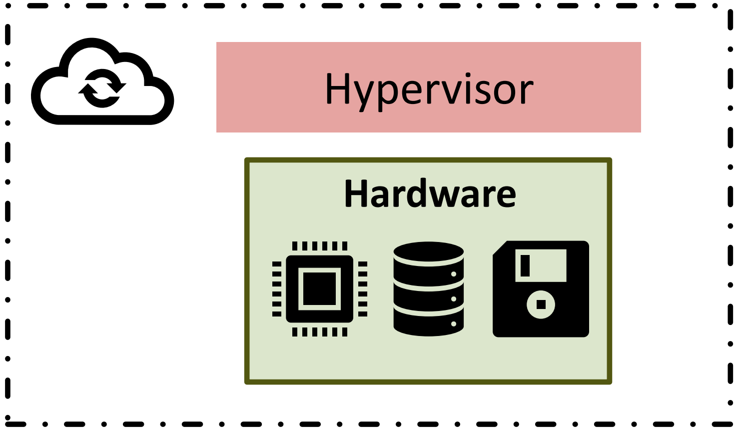
Marcin Copik, **Roman Böhringer,** Alexandru Calotoiu, Torsten Hoefler

Cloud and Serverless

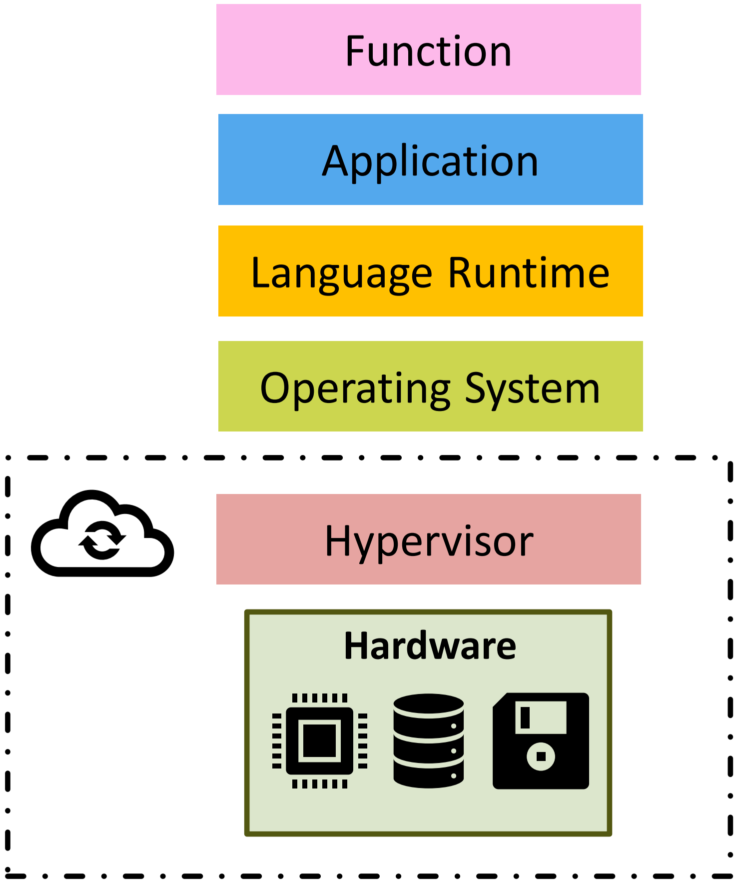
Cloud and Serverless



Cloud and Serverless

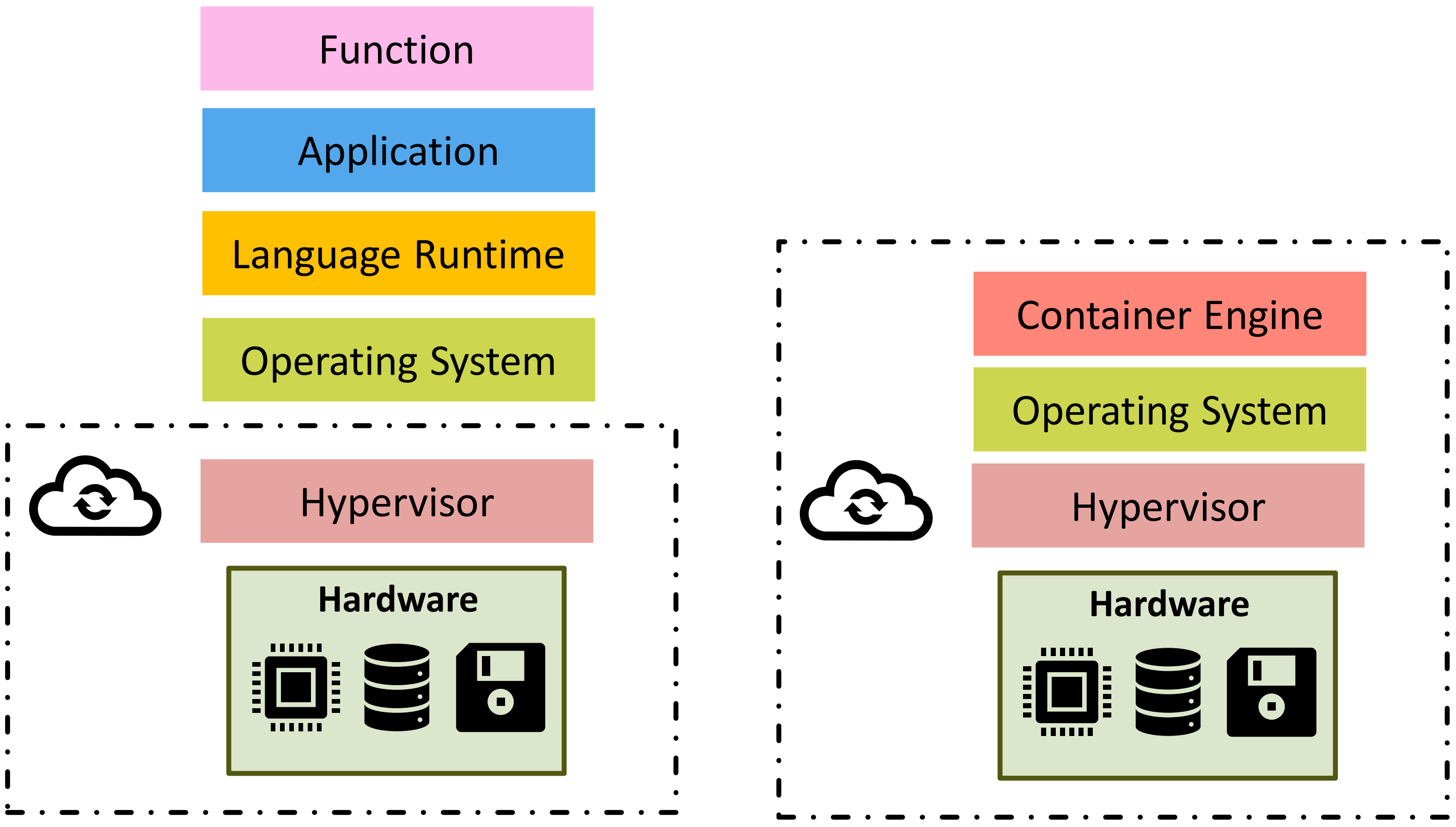


Cloud and Serverless



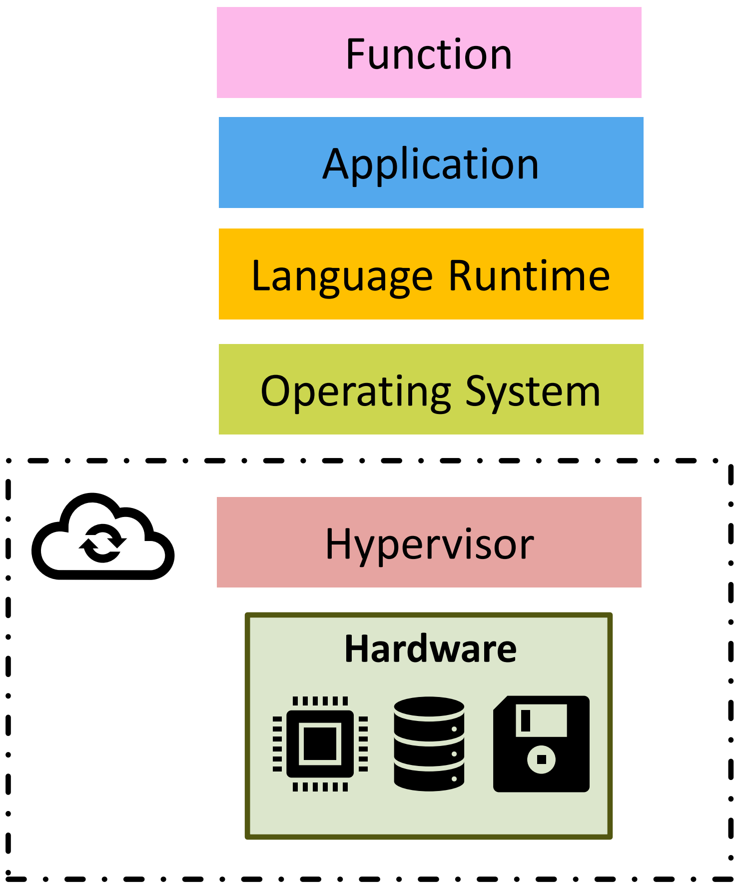
Virtual Machine

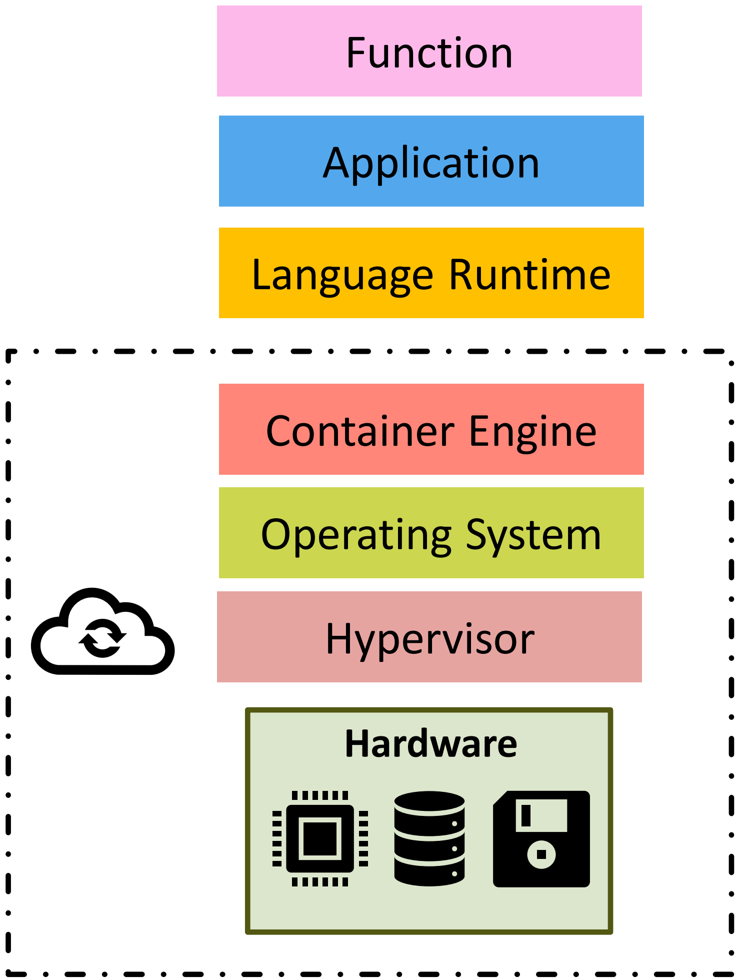
Cloud and Serverless



Virtual Machine **Containers**

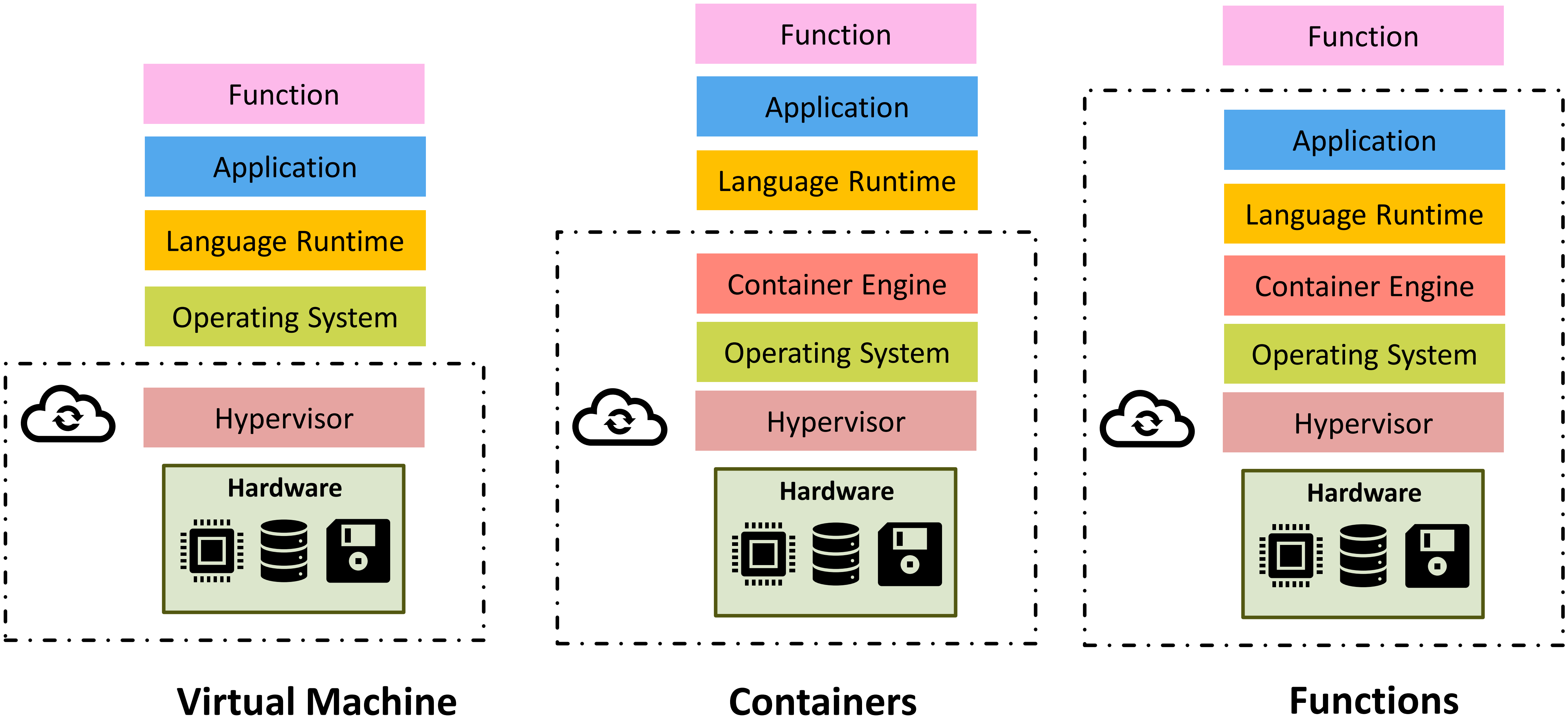
Cloud and Serverless





Virtual Machine **Containers**

Cloud and Serverless



Why is serverless interesting for HPC?

Why is serverless interesting for HPC?

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

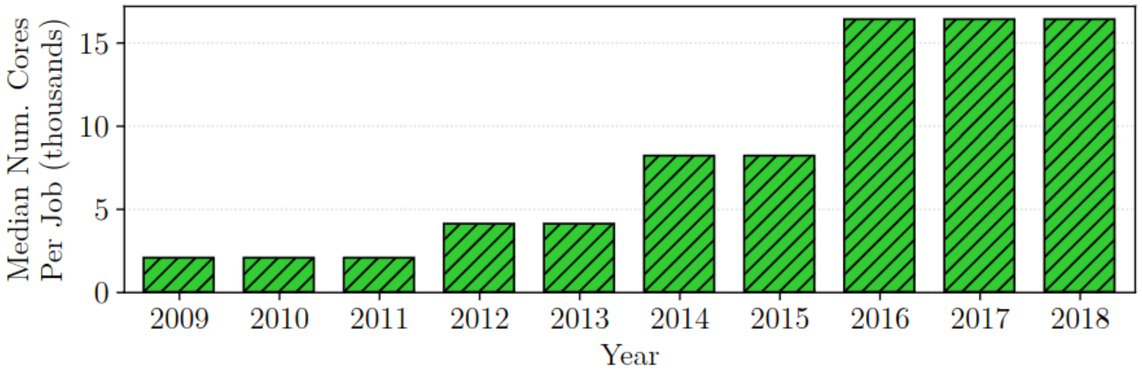
Tirthak Patel
Northeastern University

Zhengchun Liu, Raj Kettimuthu
Argonne National Laboratory

Paul Rich, William Allcock
Argonne National Laboratory

Devesh Tiwari
Northeastern University

SC, 2020



Why is serverless interesting for 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

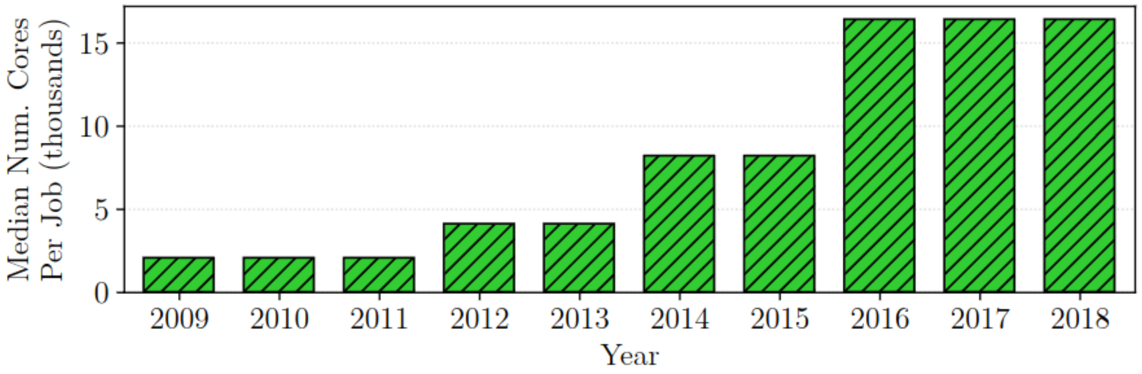
SC, 2020

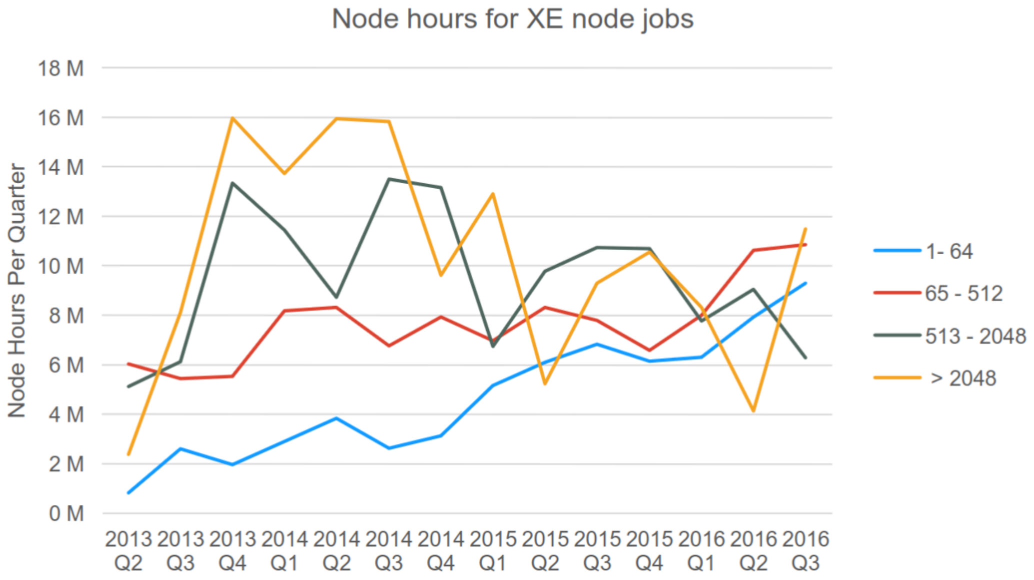
FINAL REPORT
WORKLOAD ANALYSIS OF BLUE WATERS
(ACI 1650758)

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

Michael Showerman, Robert Brunner, Andry Kot, Gregory Bauer, Brett Bode, Jeremy Enos, and William Kramer (wtkramer@illinois.edu), National Center for Supercomputing Applications (NCSA), University of Illinois at Urbana Champaign

arXiv, 2017





Why is serverless interesting for HPC?

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

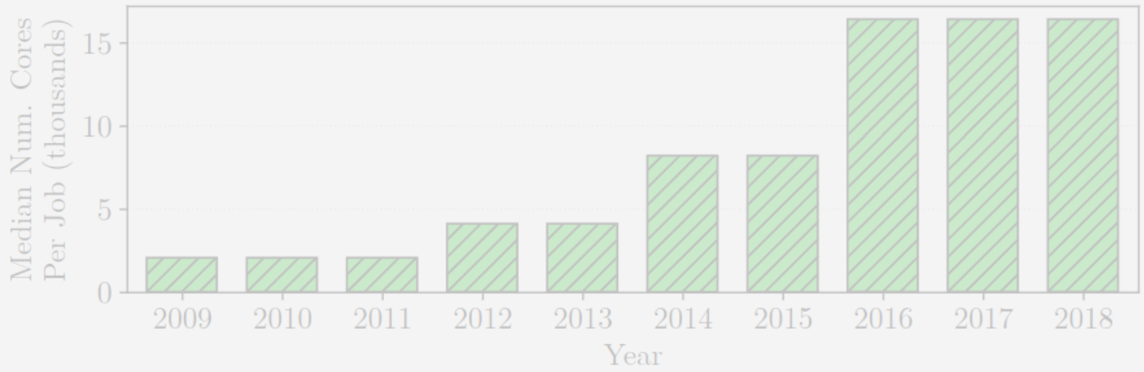
Tirthak Patel, Northeastern University
Zhengchun Liu, Raj Kettimuthu, Argonne National Laboratory
Paul Rich, William Allcock, Argonne National Laboratory
Devesh Tiwari, Northeastern University

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

Michael Showerman, Robert Brunner, Andry Kot, Gregory Bauer, Brett Bode, Jeremy Enos, and William Kramer (wtkramer@illinois.edu), National Center for Supercomputing Applications (NCSA), University of Illinois at Urbana Champaign

On AWS Lambda, we can run by default up to 1000 concurrent functions, each with up to 6 vCPUs.



Communication in serverless: we need to do better!

Communication in serverless: we need to do better!

- We need **high performance.**

Communication in serverless: we need to do better!

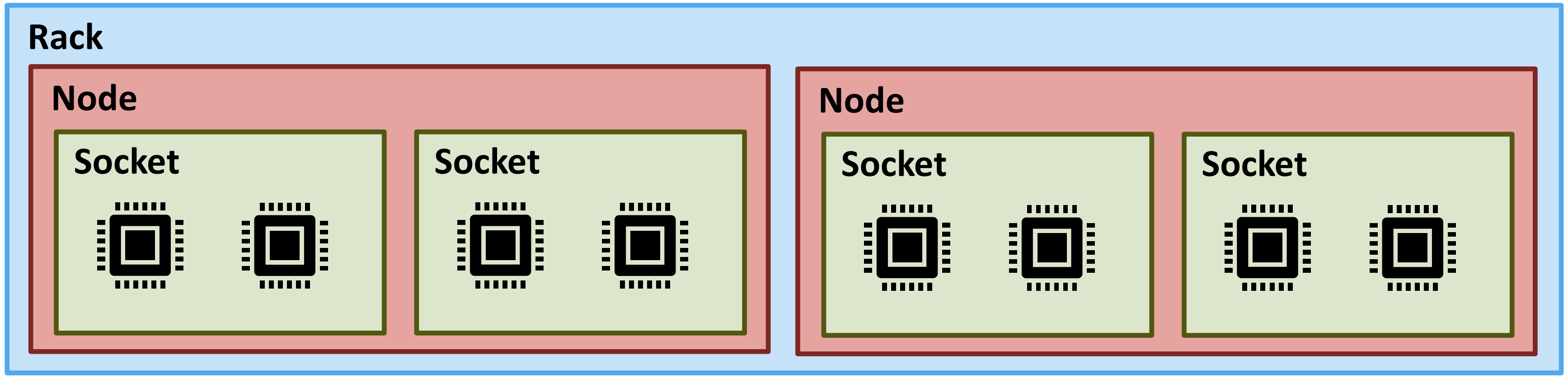
- We need **high performance.**
- We need **division of labor.**

Collectives provide *"division of labor: the programmer thinks in terms of these primitives and the library is responsible for implementing them efficiently".*

Sanders et al., "Two-Tree Algorithms for Full Bandwidth Broadcast, Reduction and Scan", Parallel Computing 2009

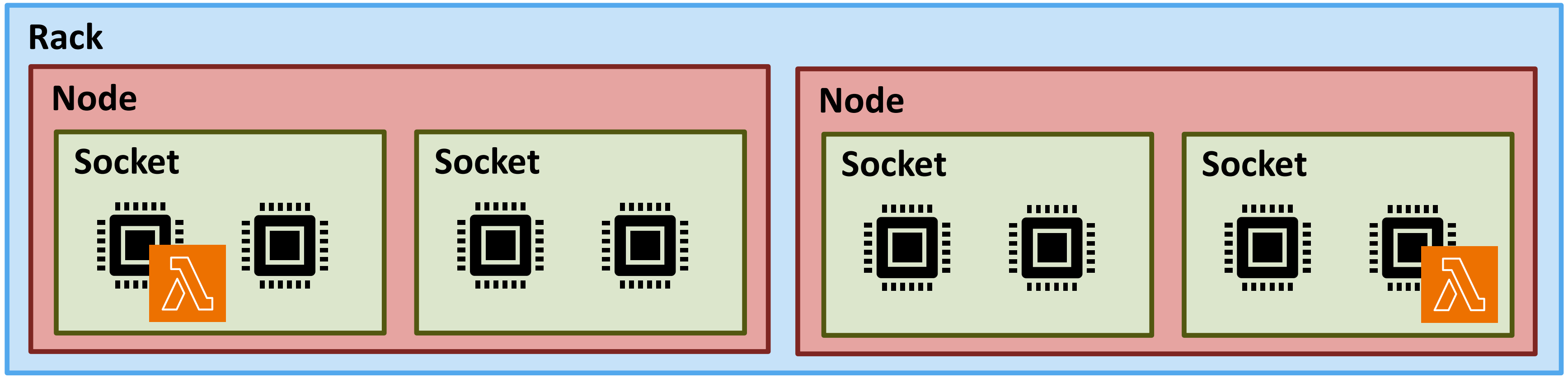
Communication in serverless: we need to do better!

- We need **high performance.**
- We need **division of labor.**
- We need **portable performance.**



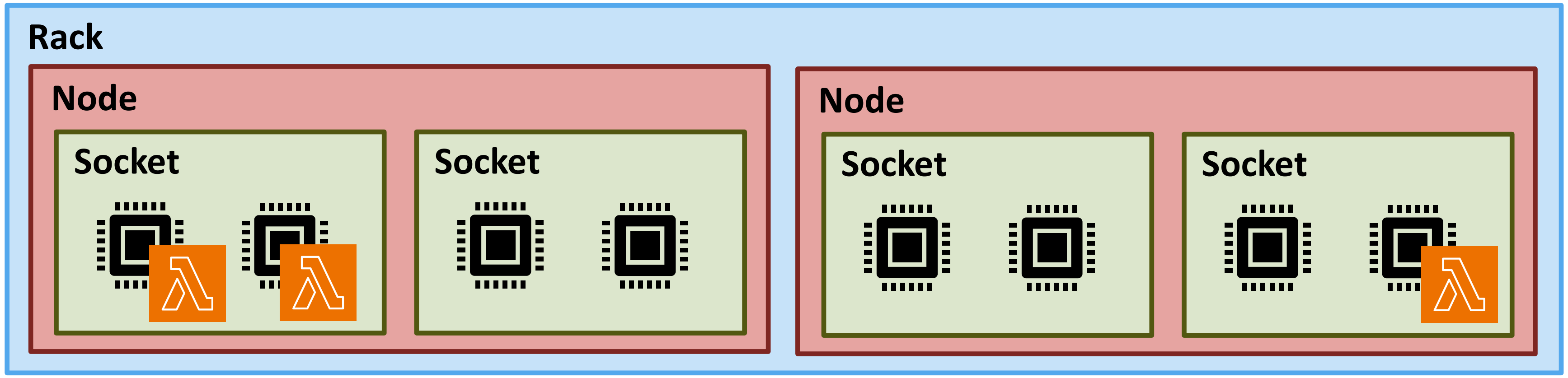
Communication in serverless: we need to do better!

- We need **high performance.**
- We need **division of labor.**
- We need **portable performance.**



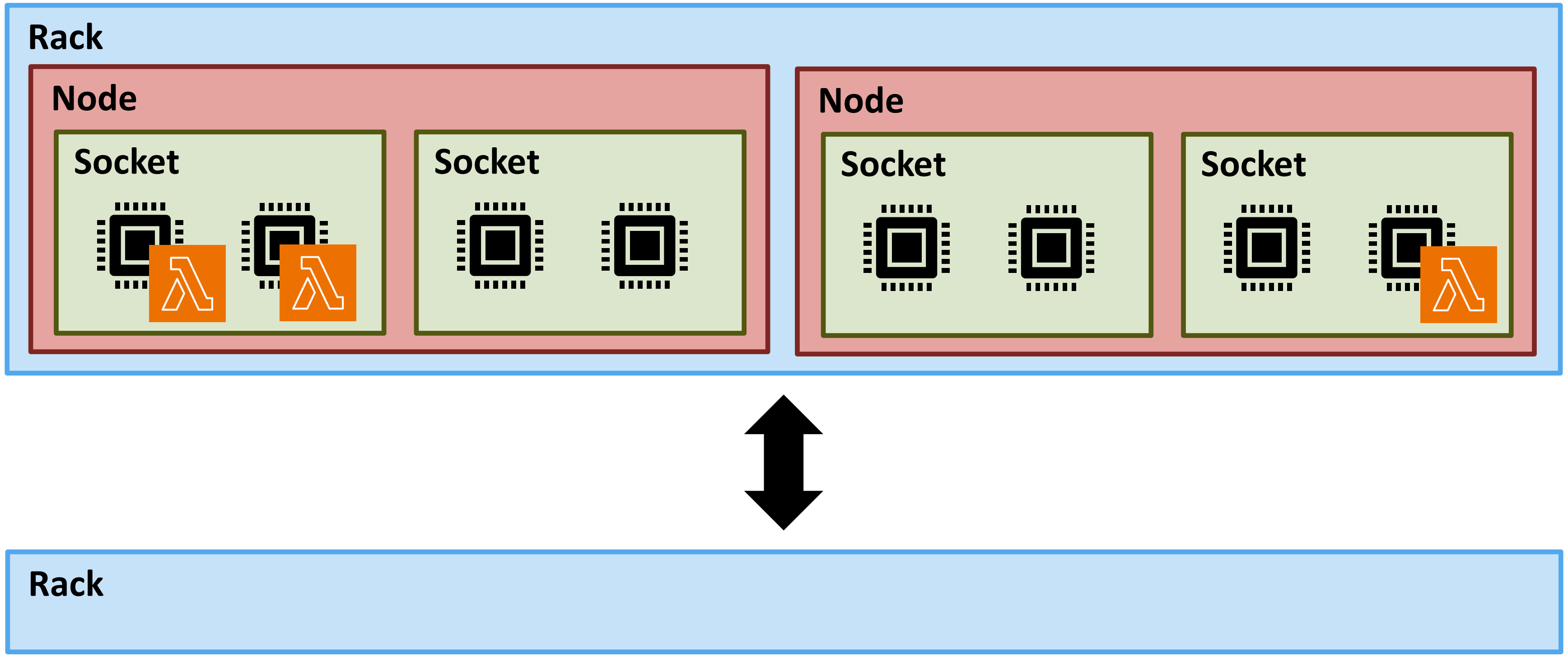
Communication in serverless: we need to do better!

- We need **high performance.**
- We need **division of labor.**
- We need **portable performance.**



Communication in serverless: we need to do better!

- We need **high performance.**
- We need **division of labor.**
- We need **portable performance.**

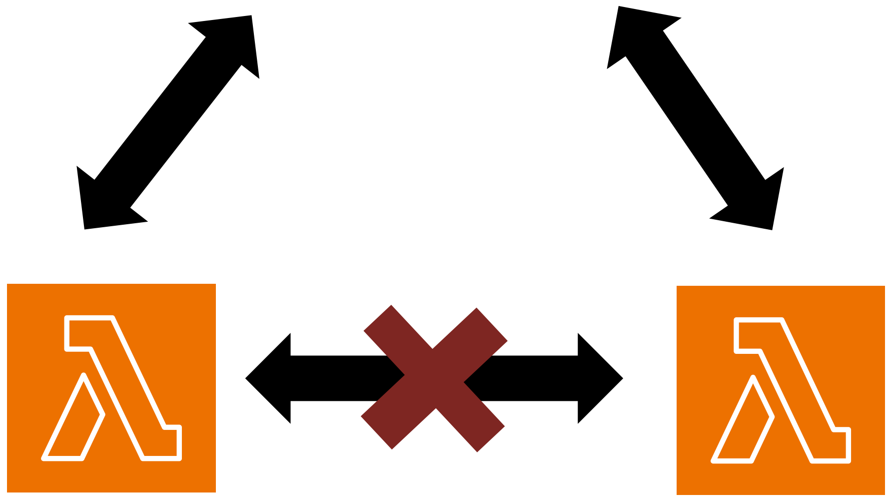


Communication in serverless

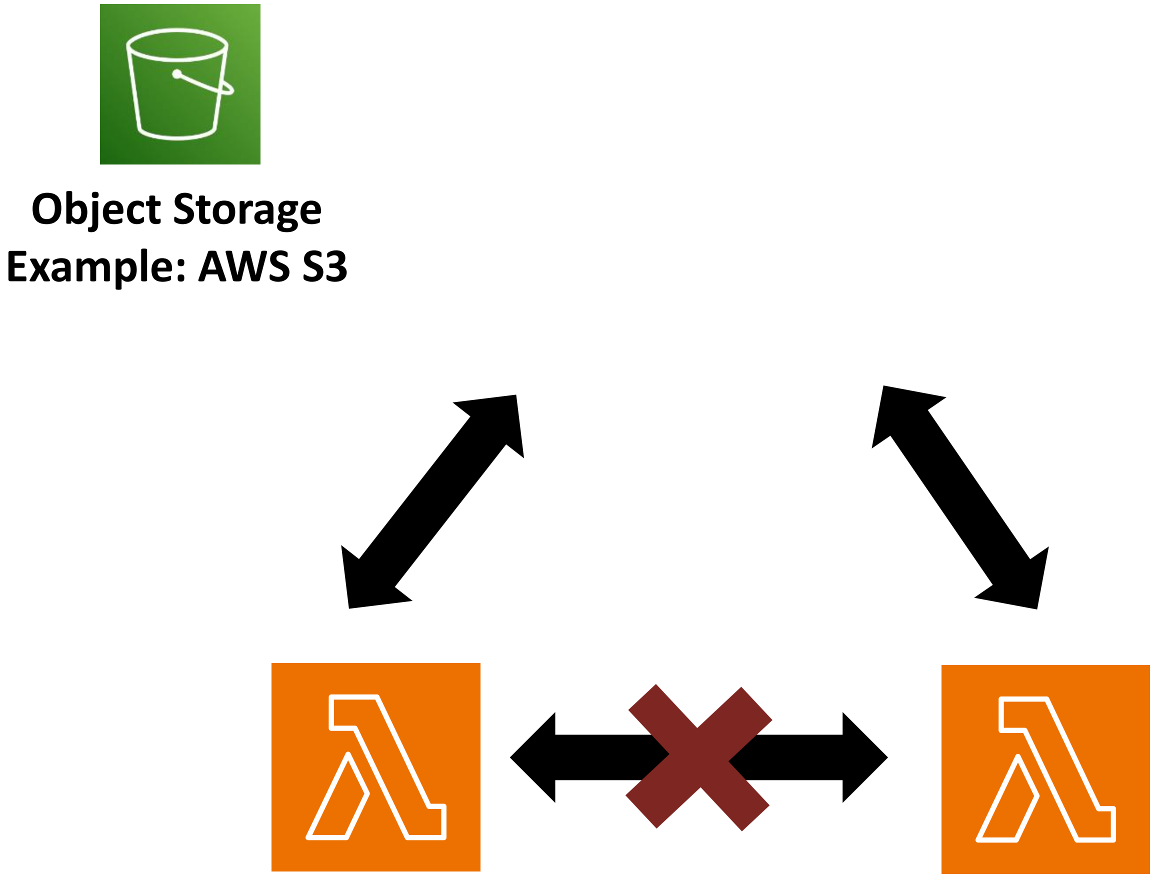
Communication in serverless

Communication in serverless

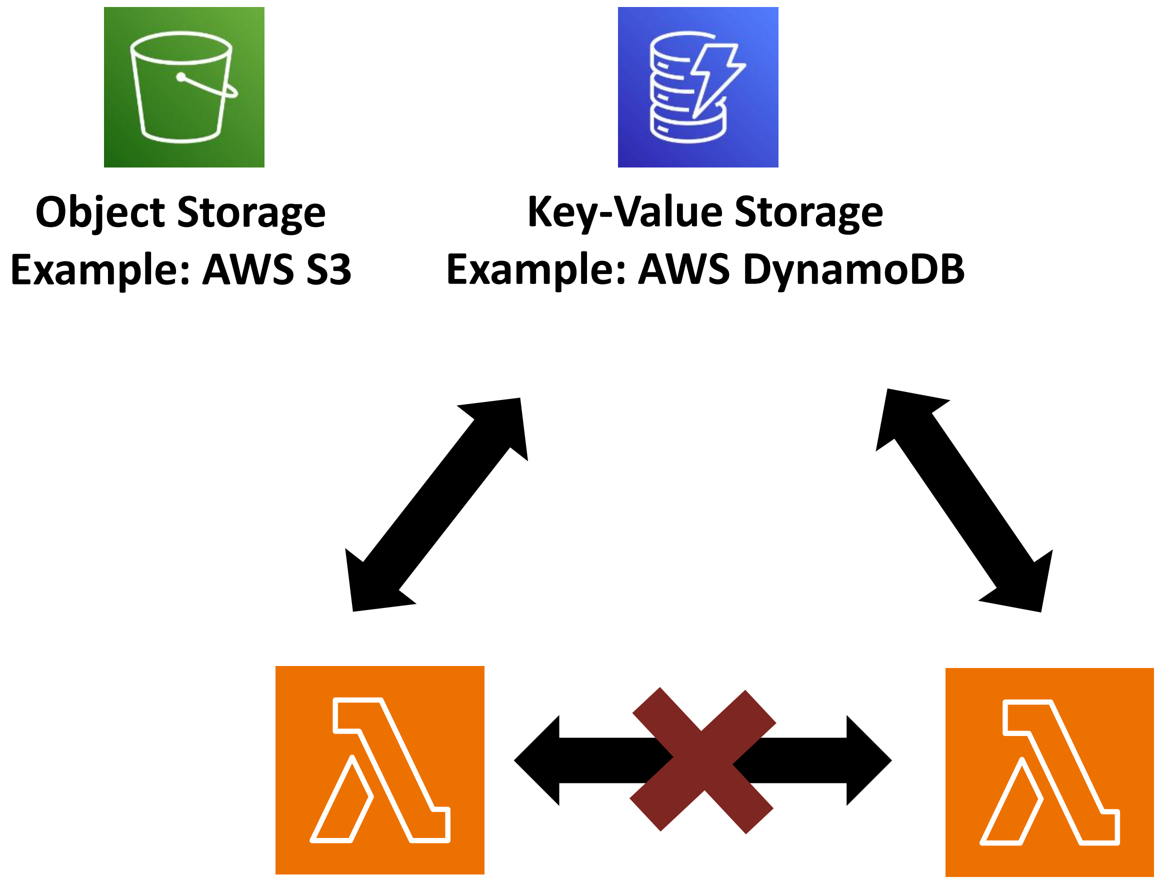
Communication in serverless



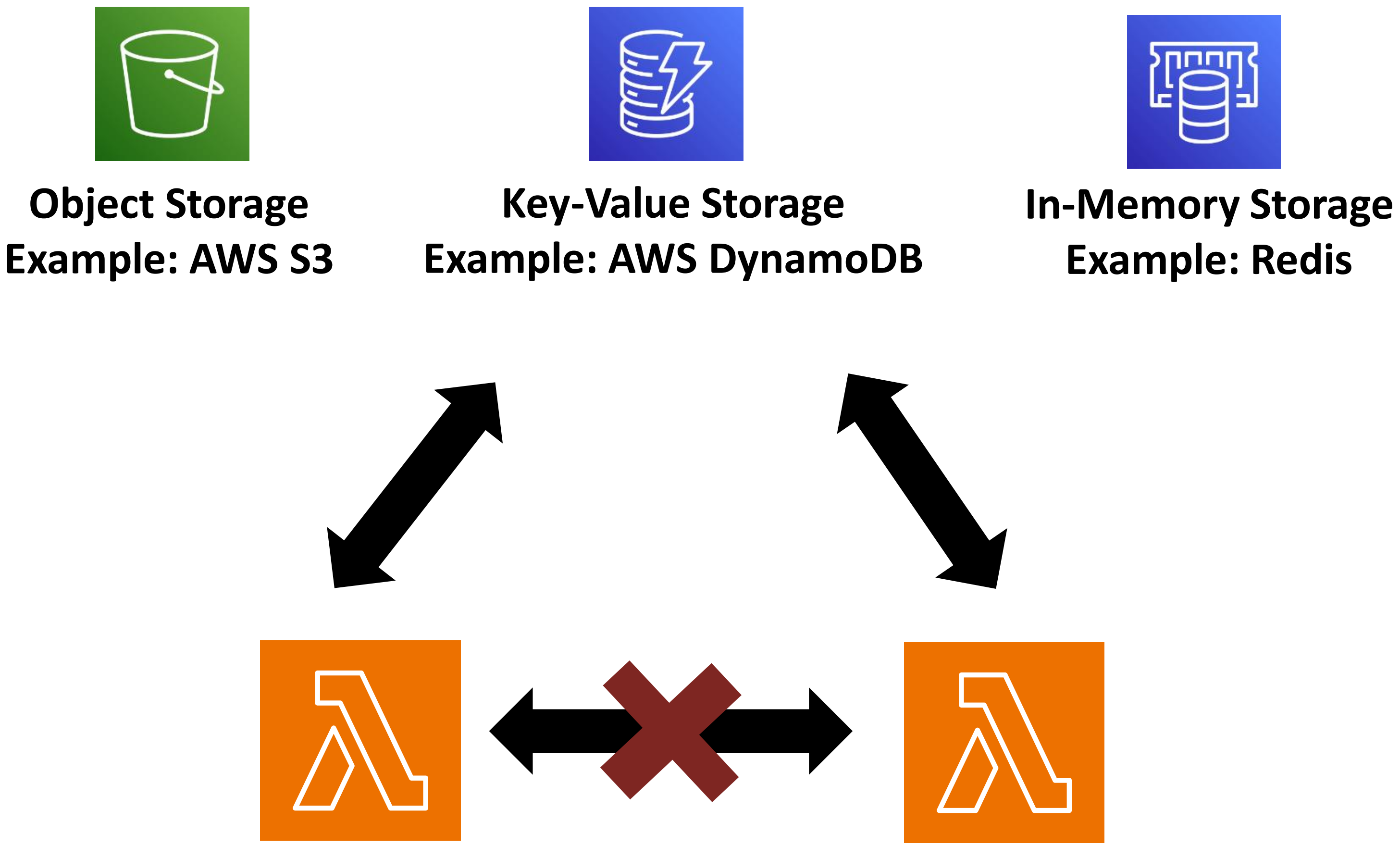
Communication in serverless



Communication in serverless

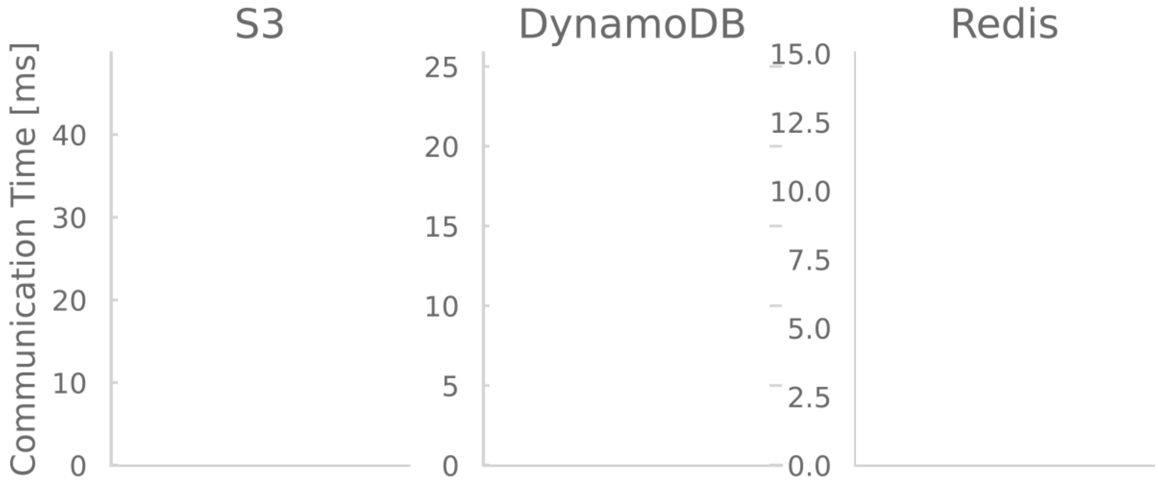


Communication in serverless

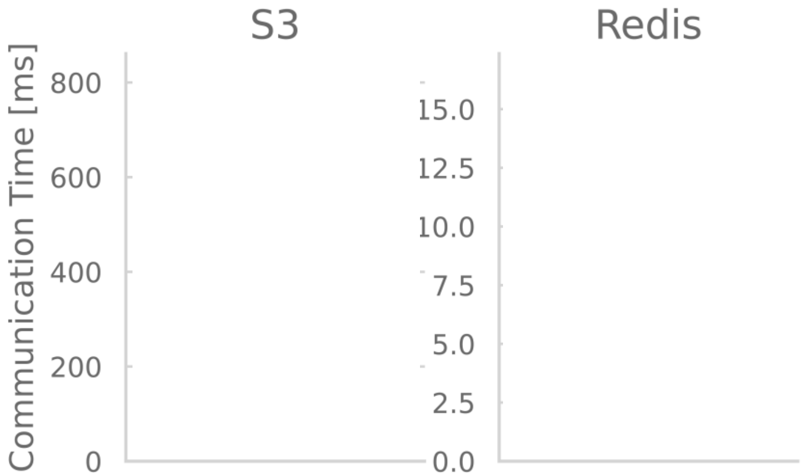


Performance of communication channels

1 B

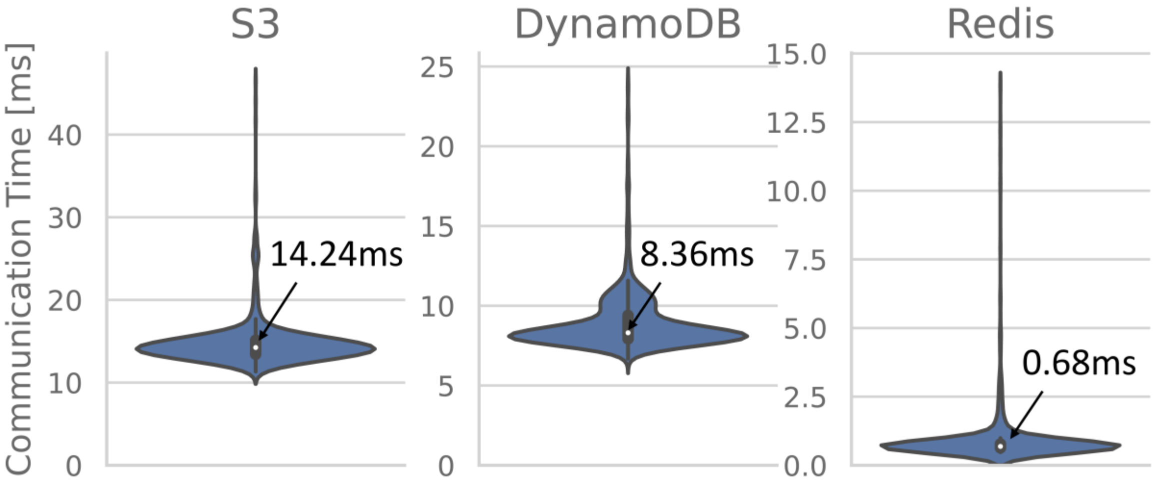


1 MB

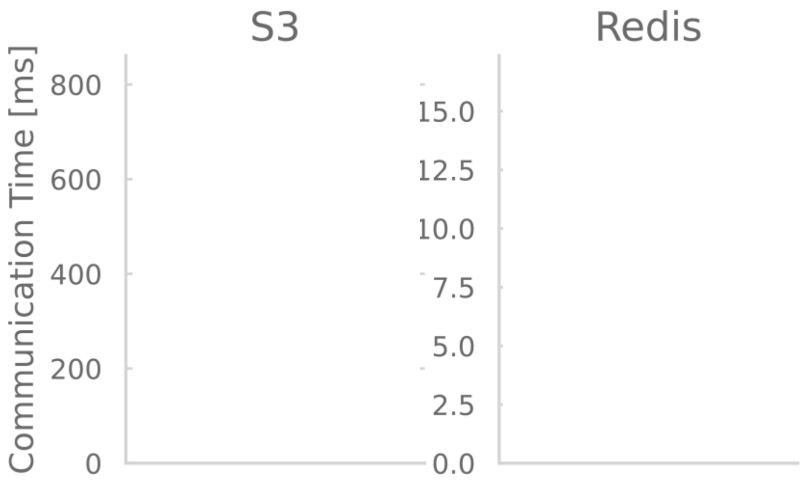


Performance of communication channels

1 B

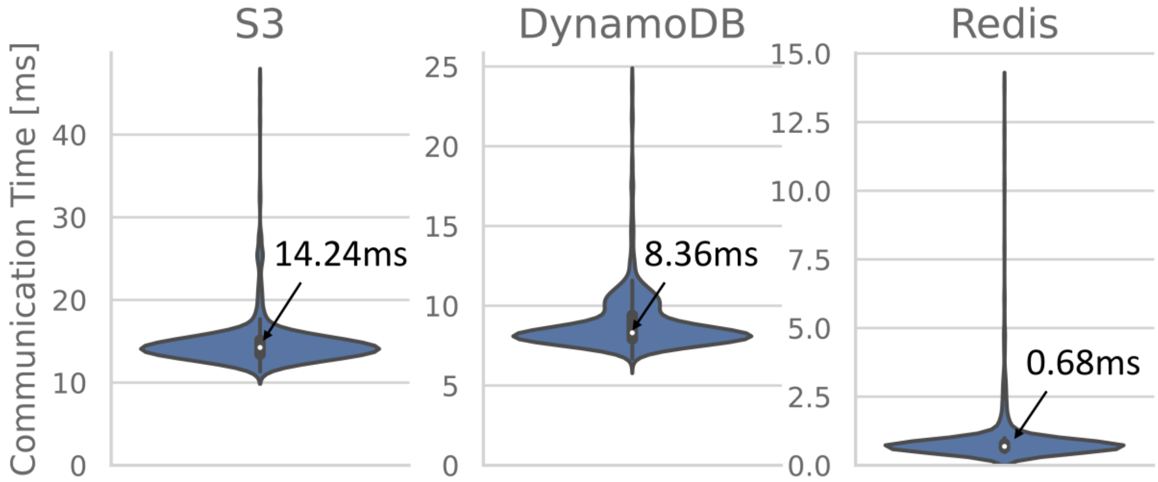


1 MB

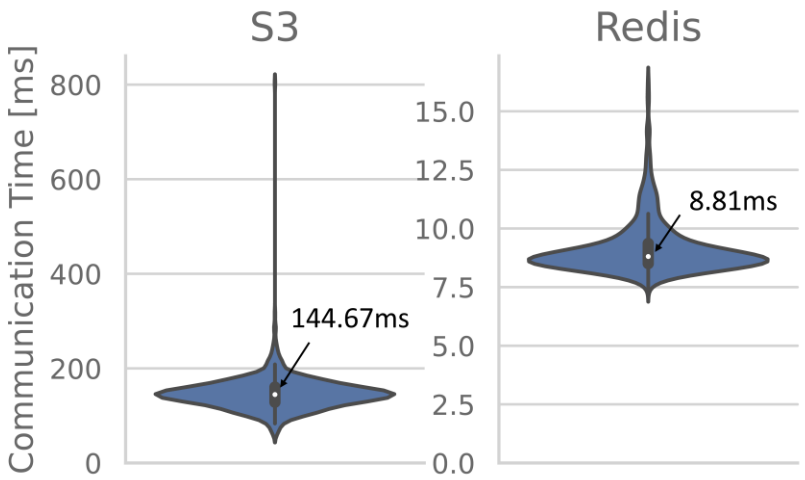


Performance of communication channels

1 B



1 MB



NAT Hole Punching

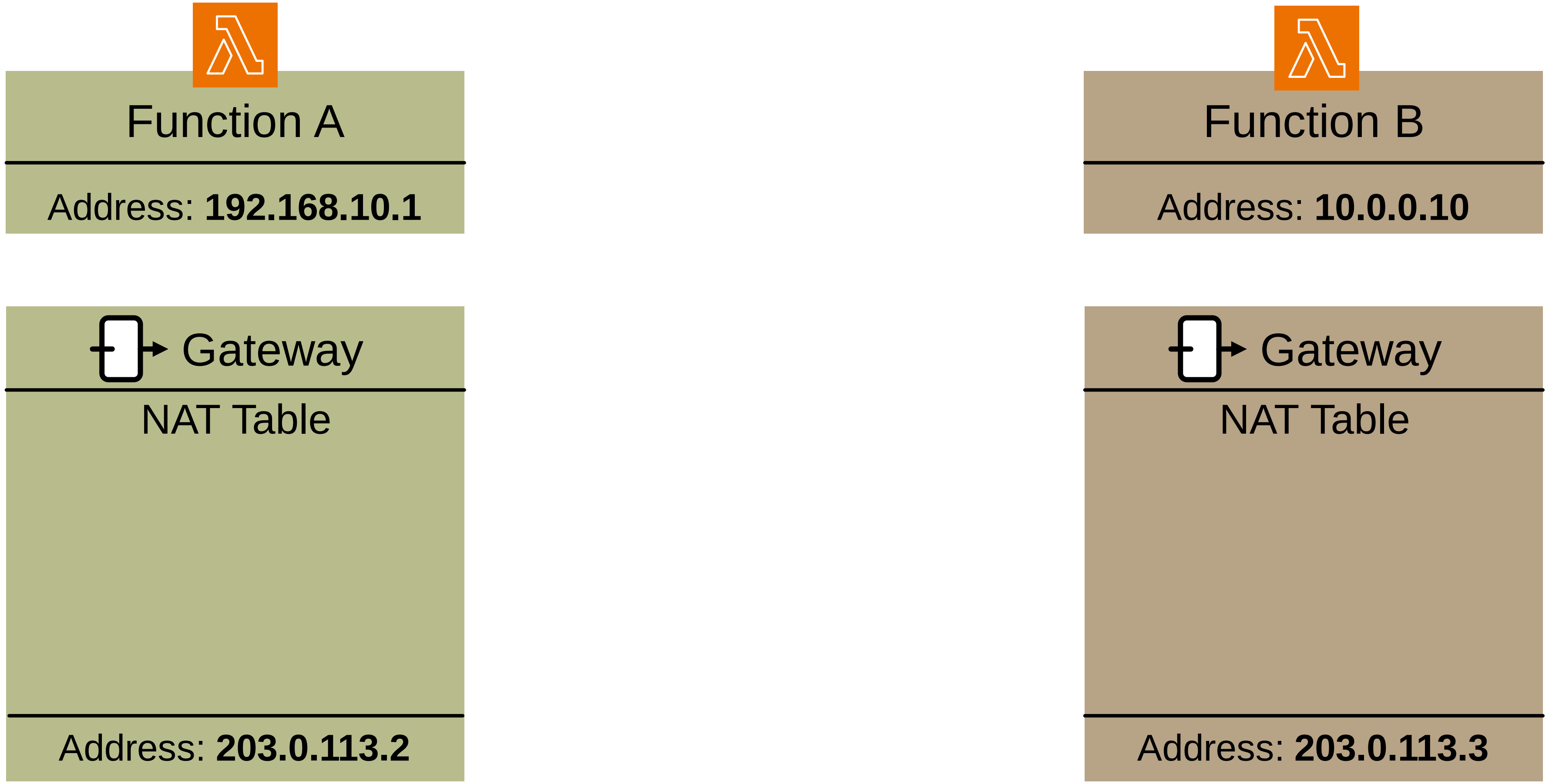
Function A

Address: **192.168.10.1**

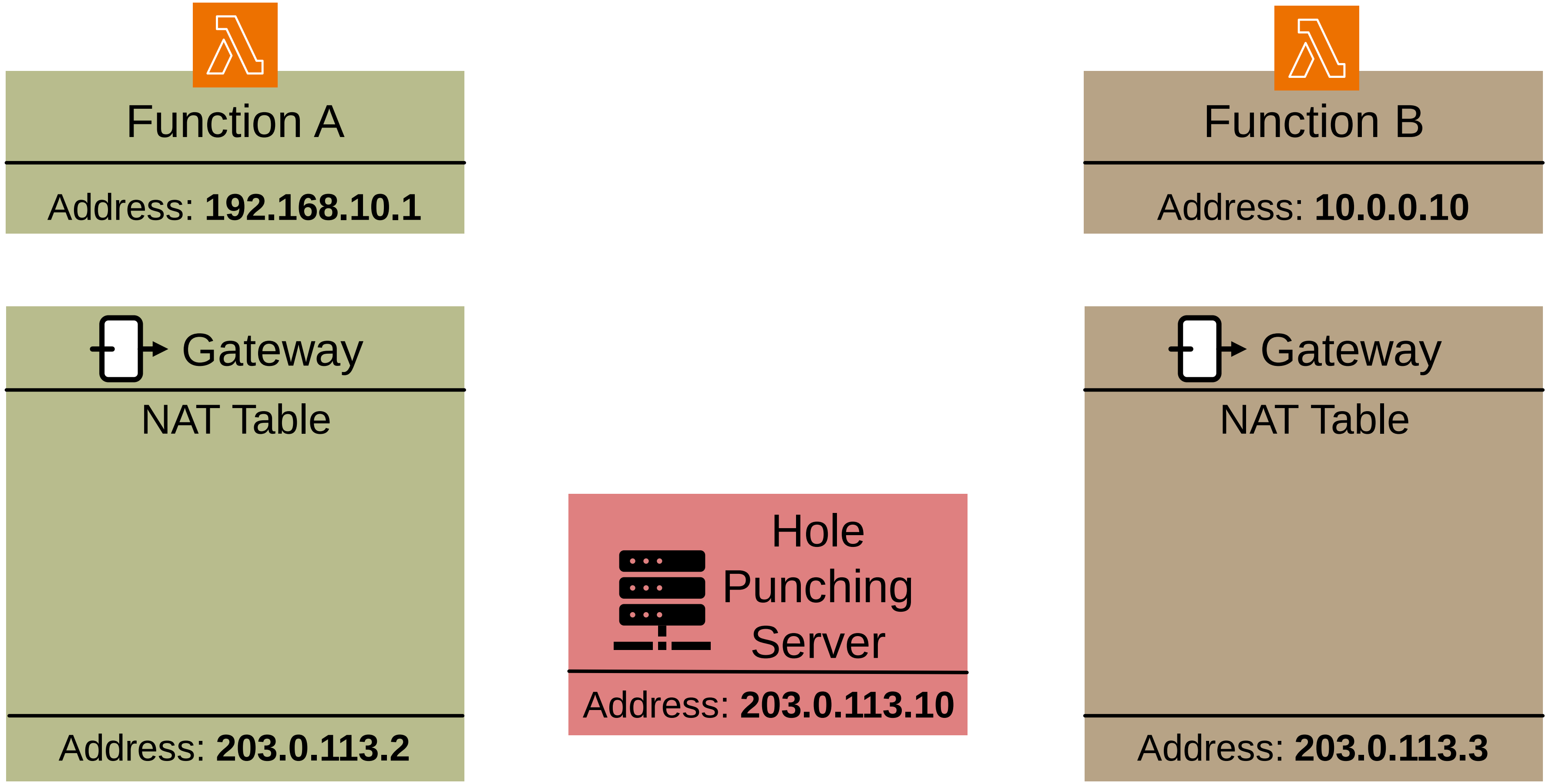
Function B

Address: **10.0.0.10**

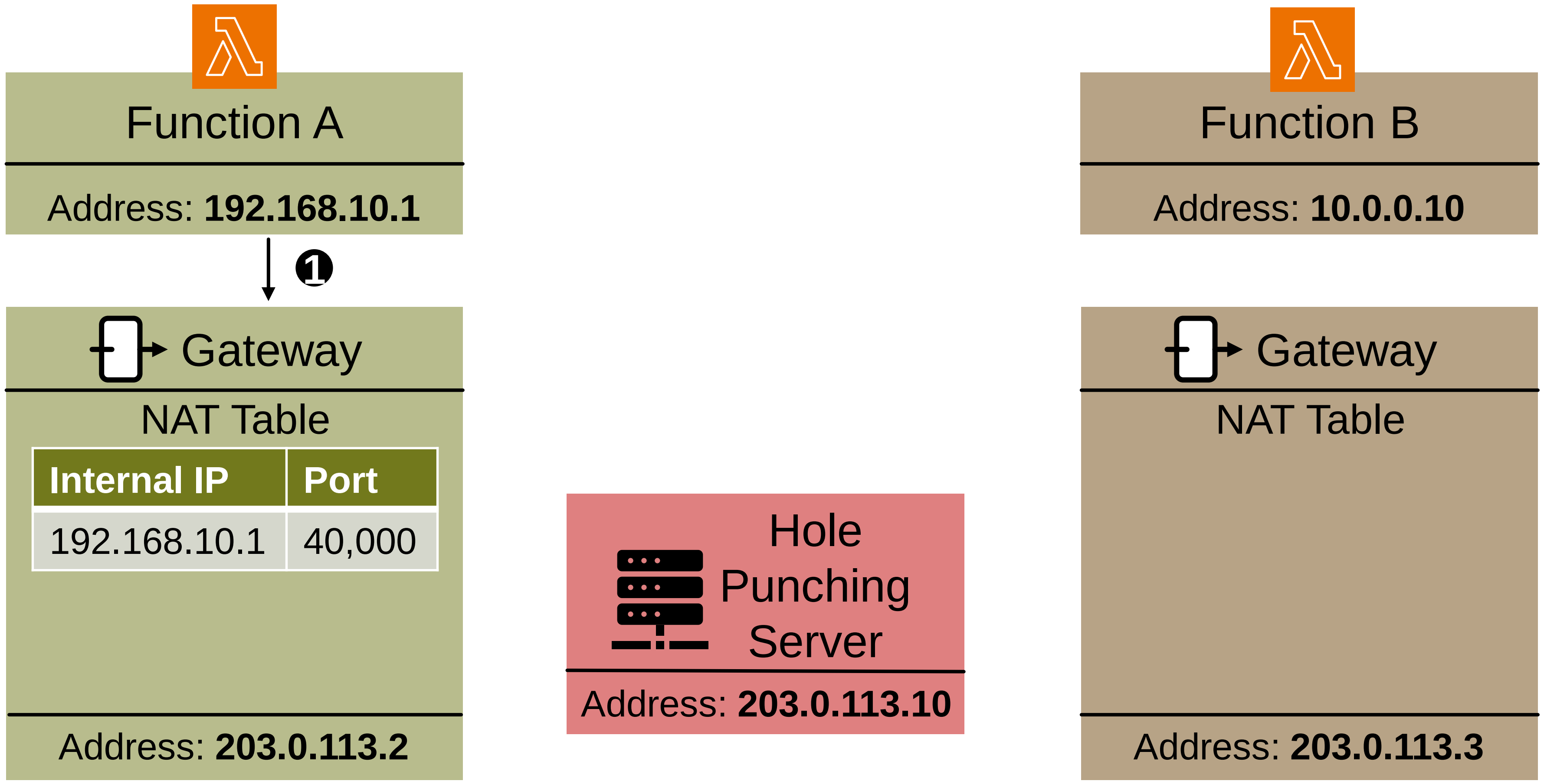
NAT Hole Punching



NAT Hole Punching

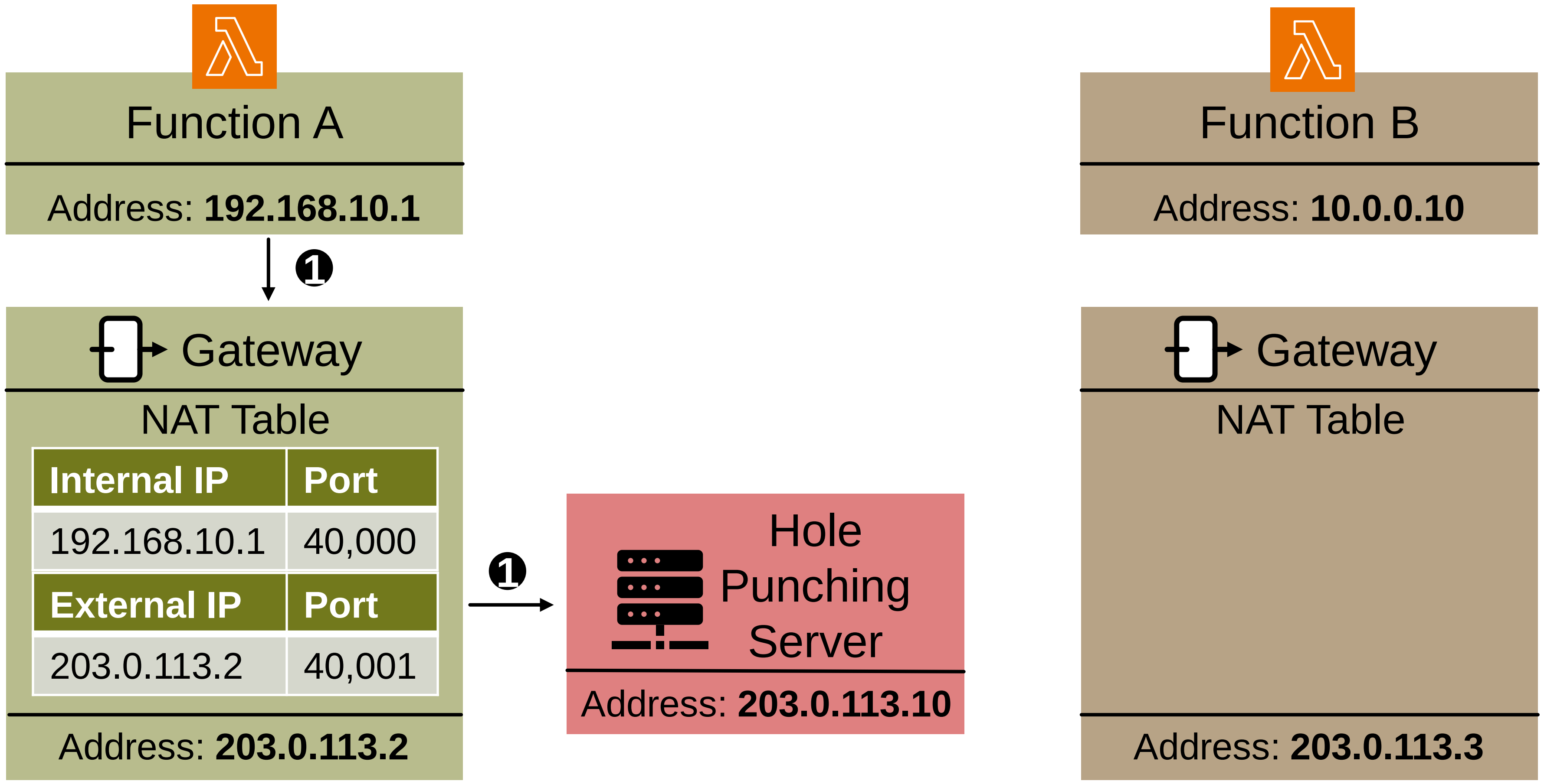


NAT Hole Punching



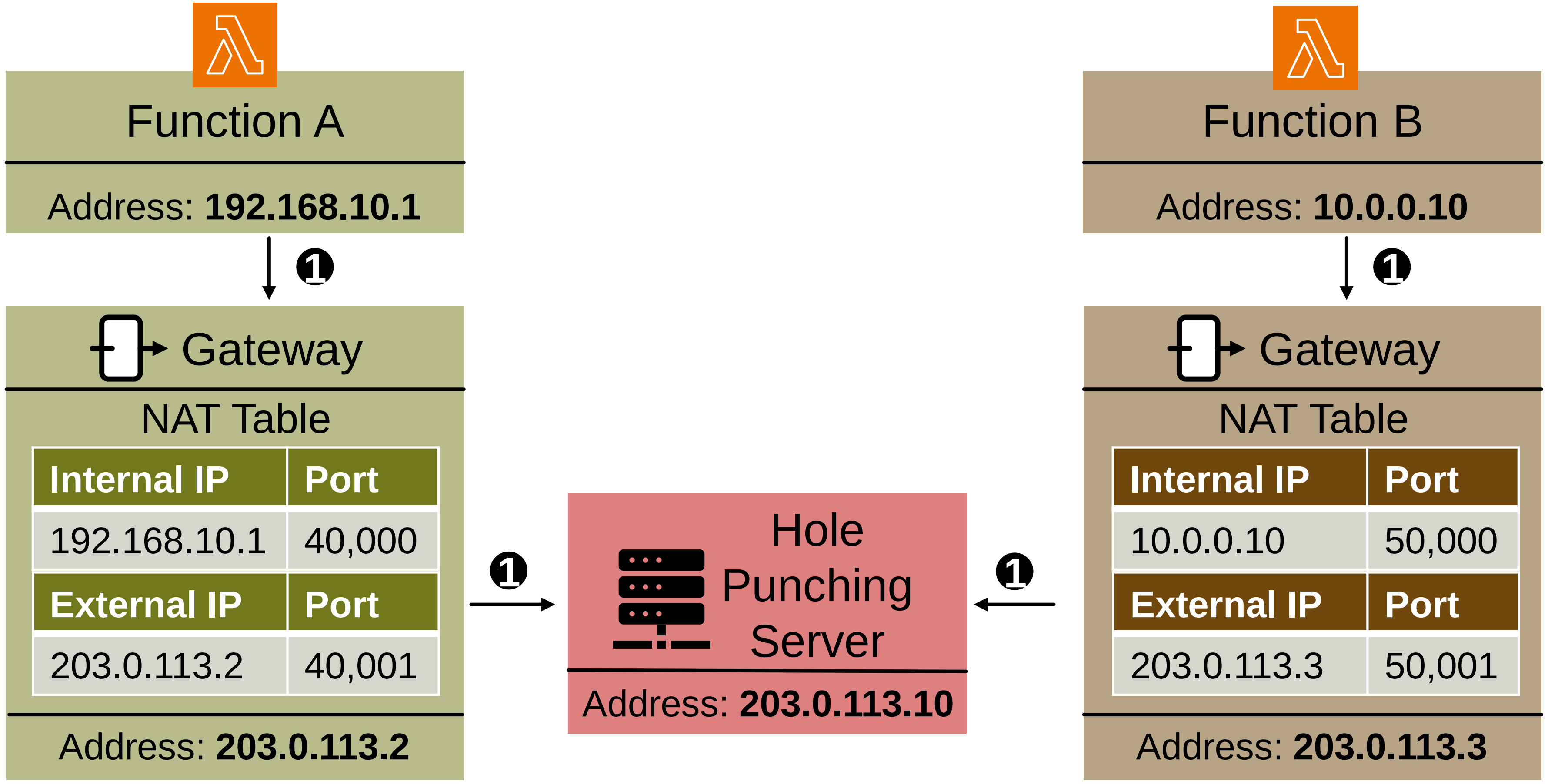
❶ Initial communication to the hole puncher.

NAT Hole Punching



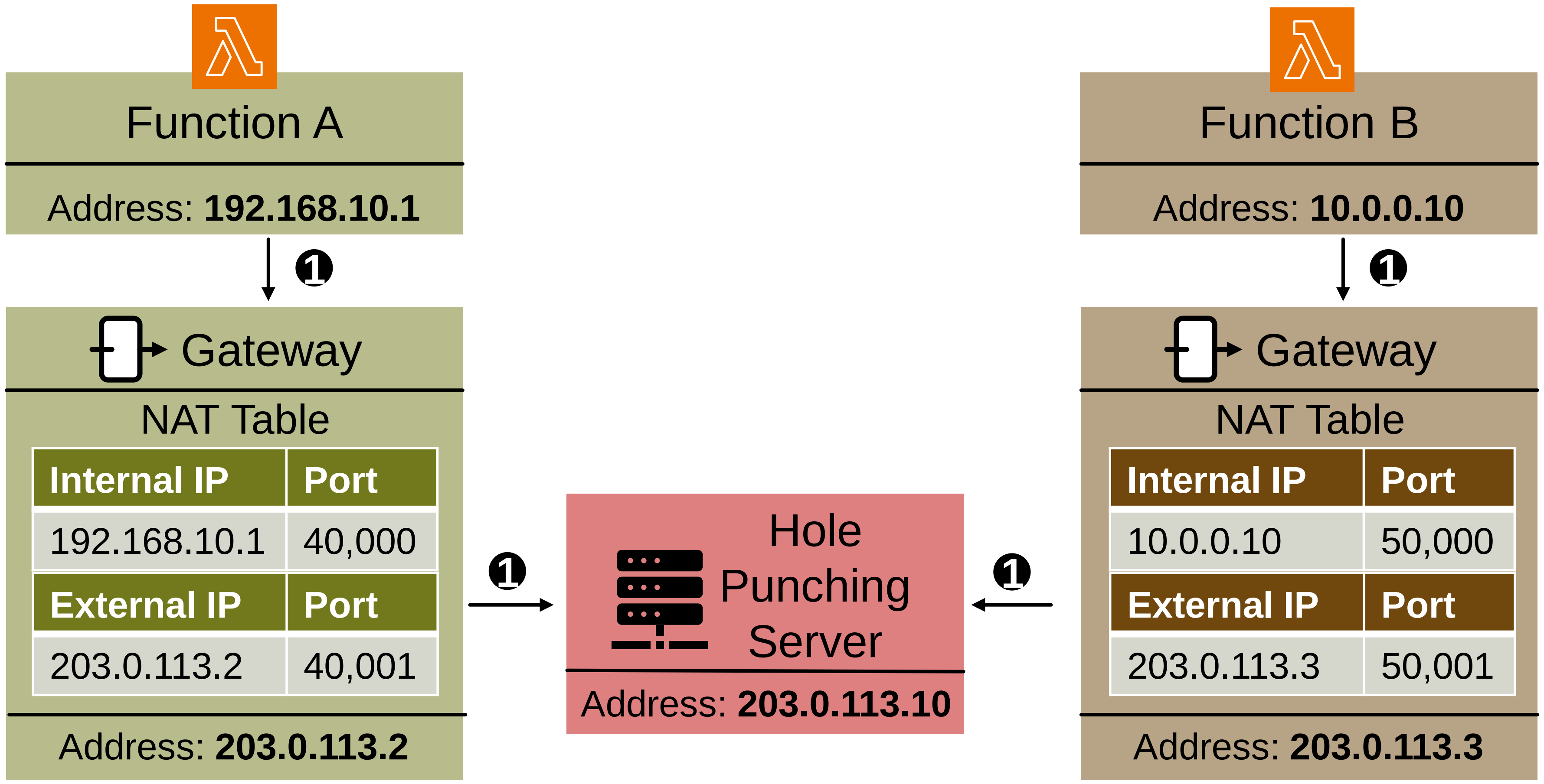
❶ Initial communication to the hole puncher.

NAT Hole Punching

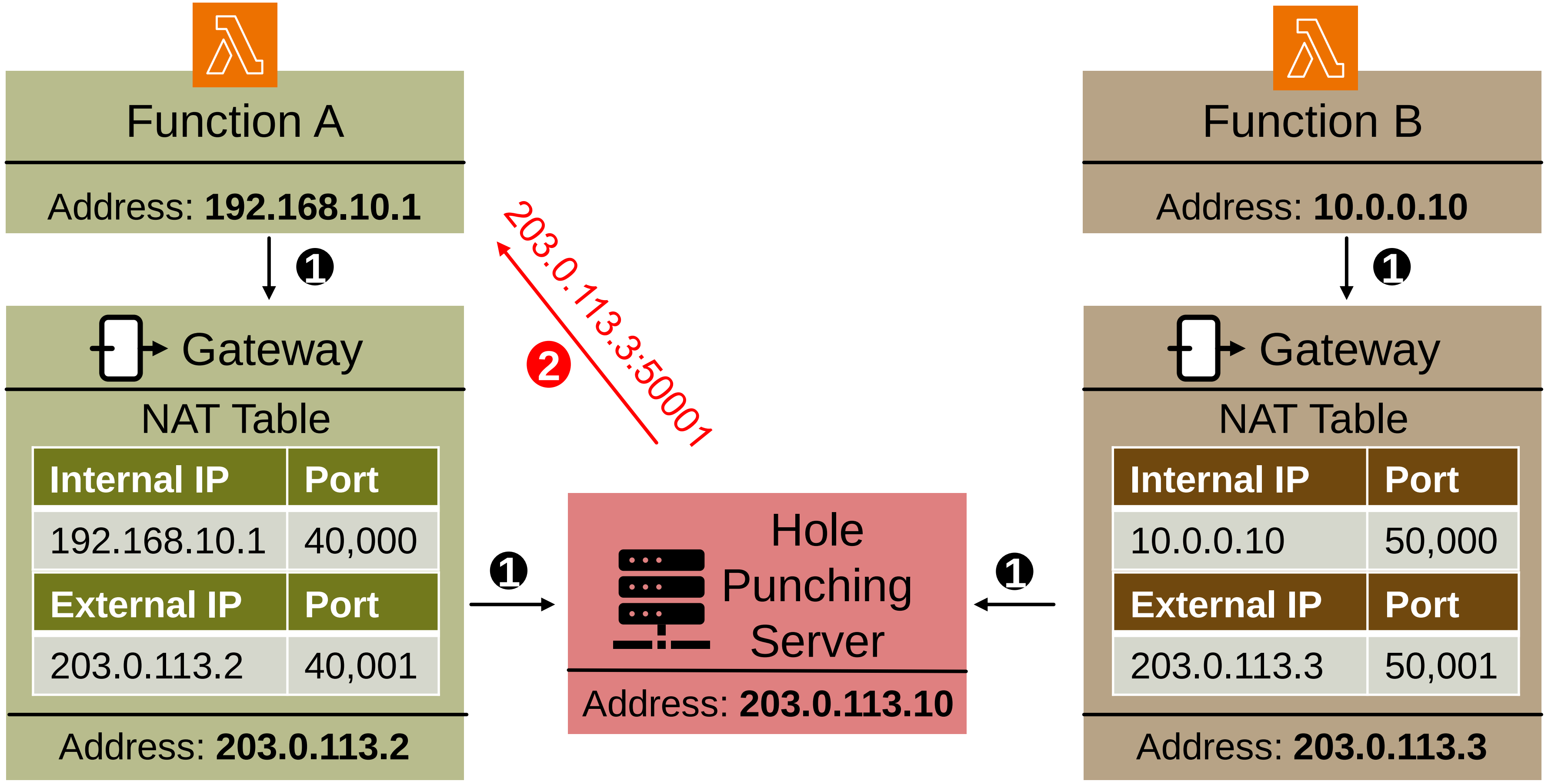


❶ Initial communication to the hole puncher.

NAT Hole Punching

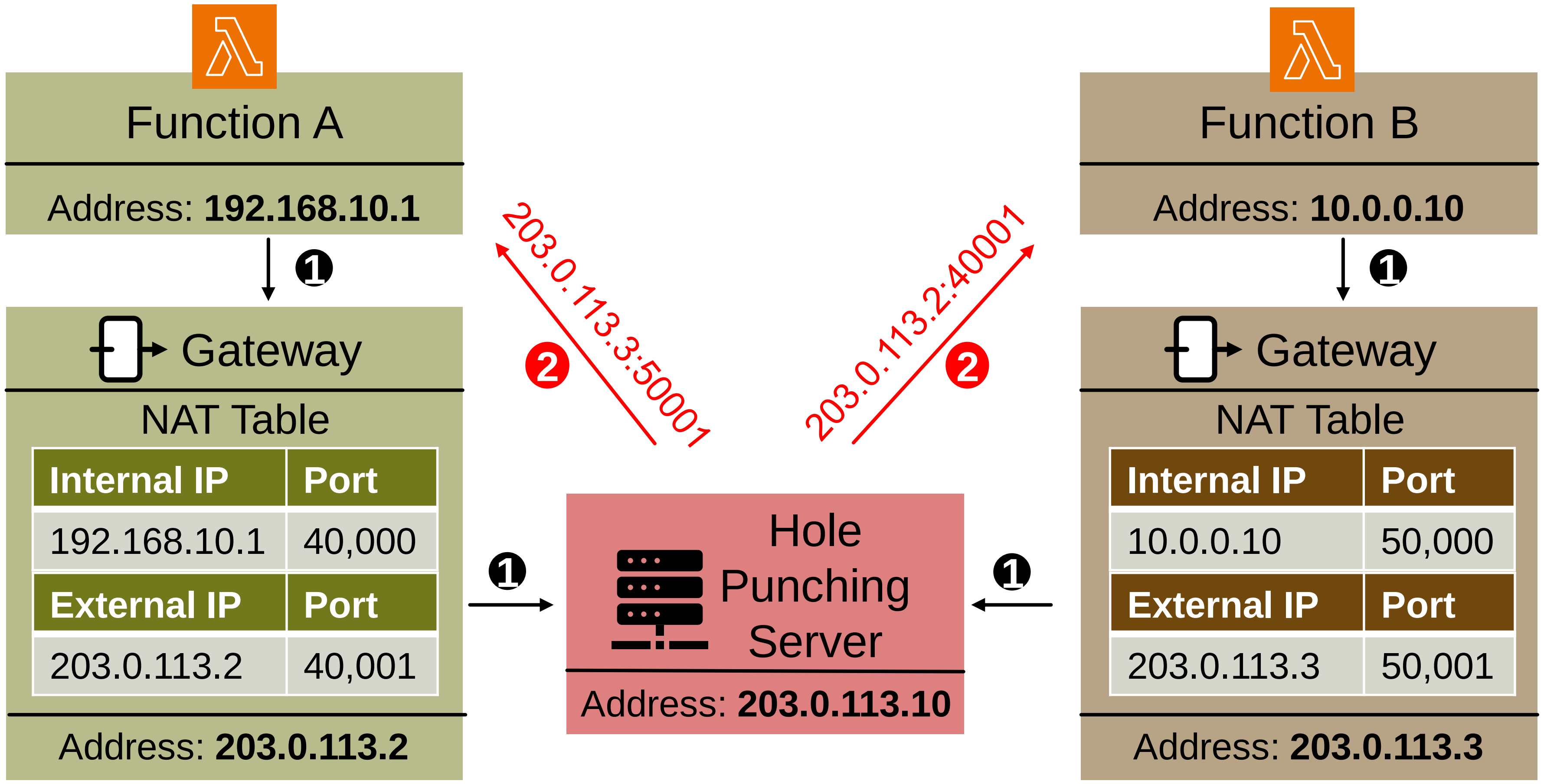


NAT Hole Punching



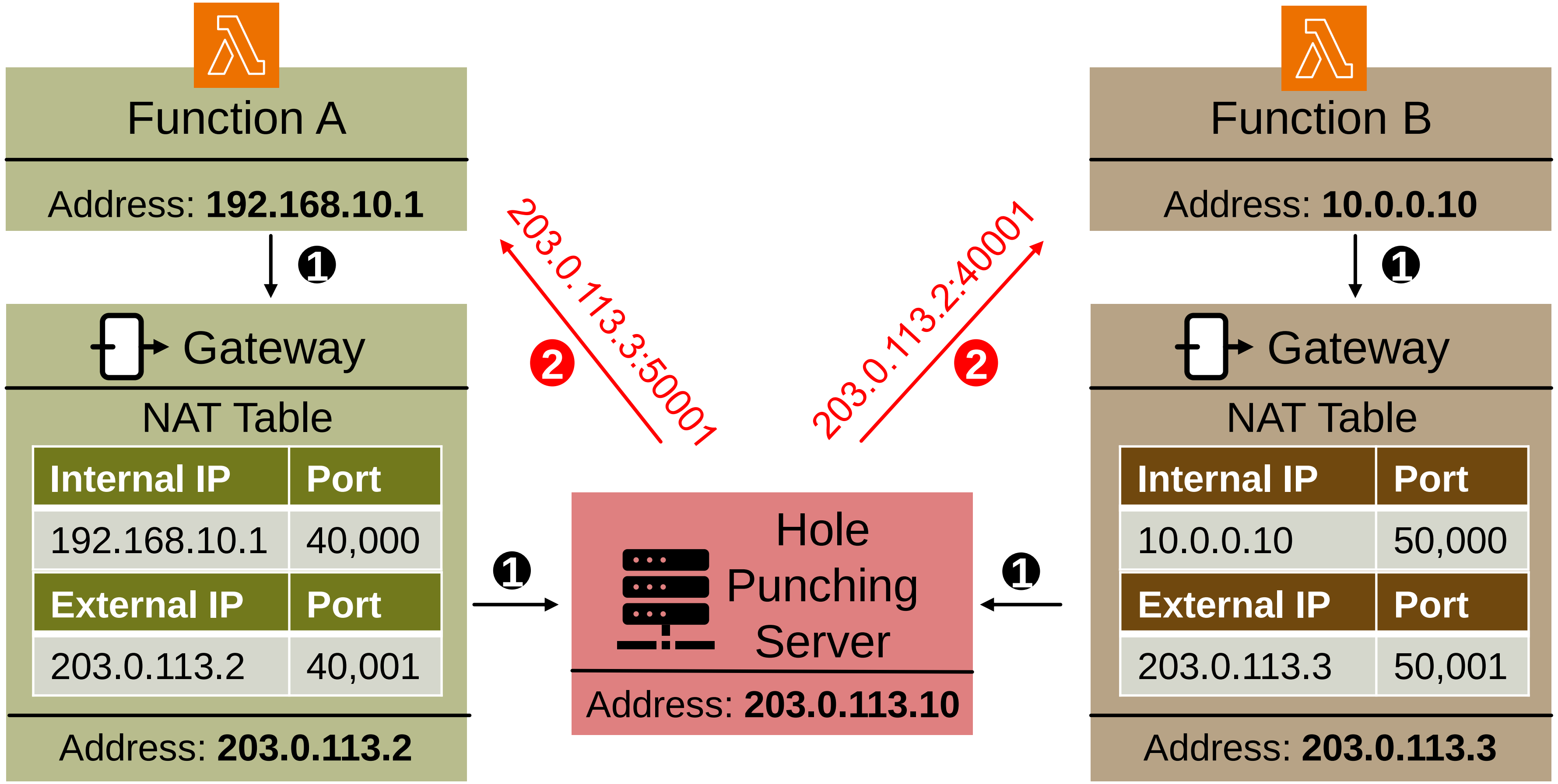
❷ Hole punching server sends connection information of the other function.

NAT Hole Punching

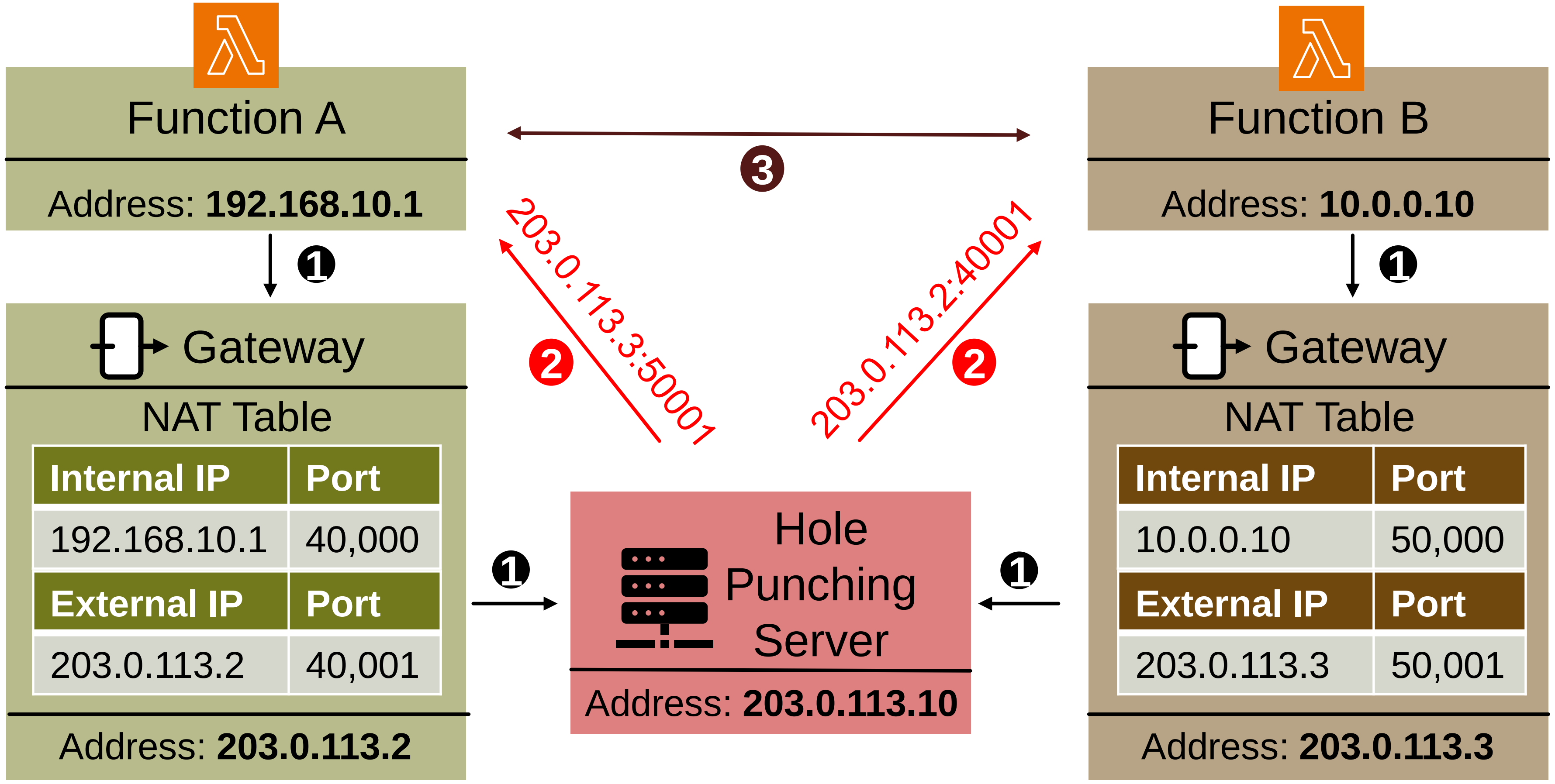


2 Hole punching server sends connection information of the other function.

NAT Hole Punching



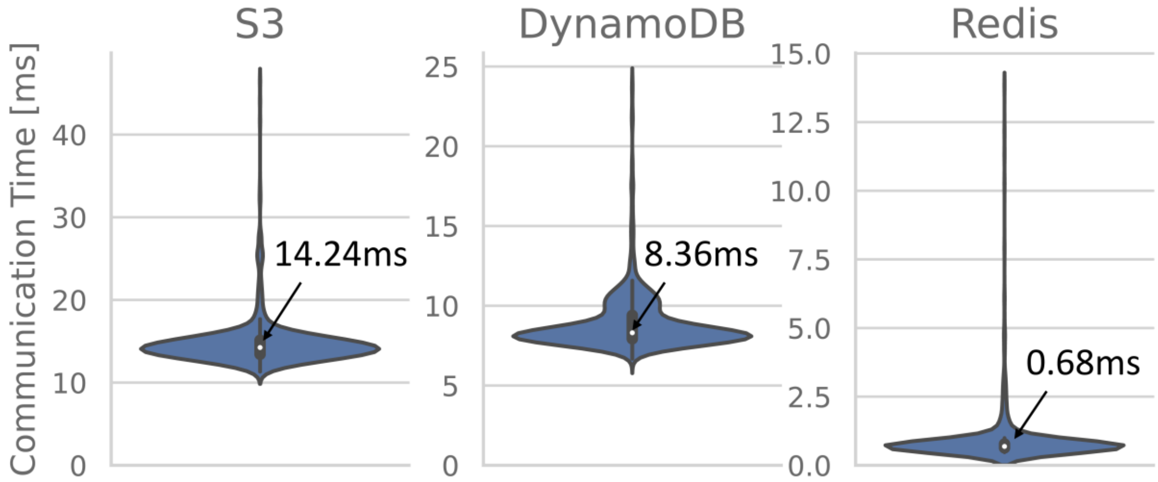
NAT Hole Punching



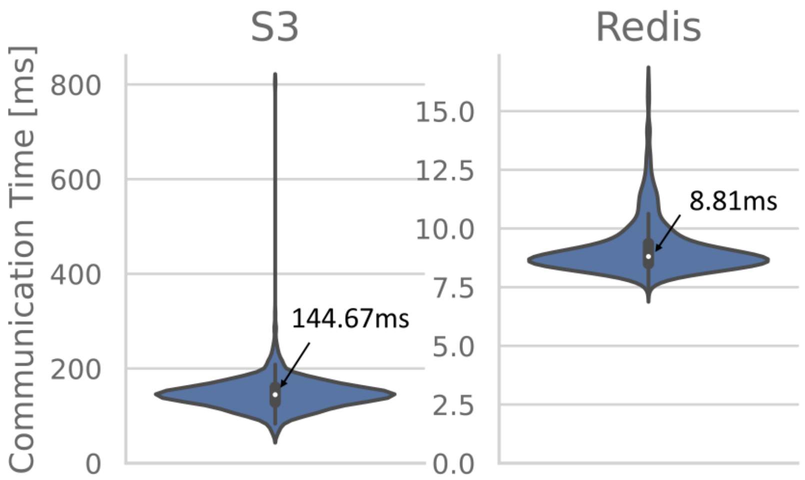
❸ Both functions initiate a connection with the new information.

Communication channels

1 B

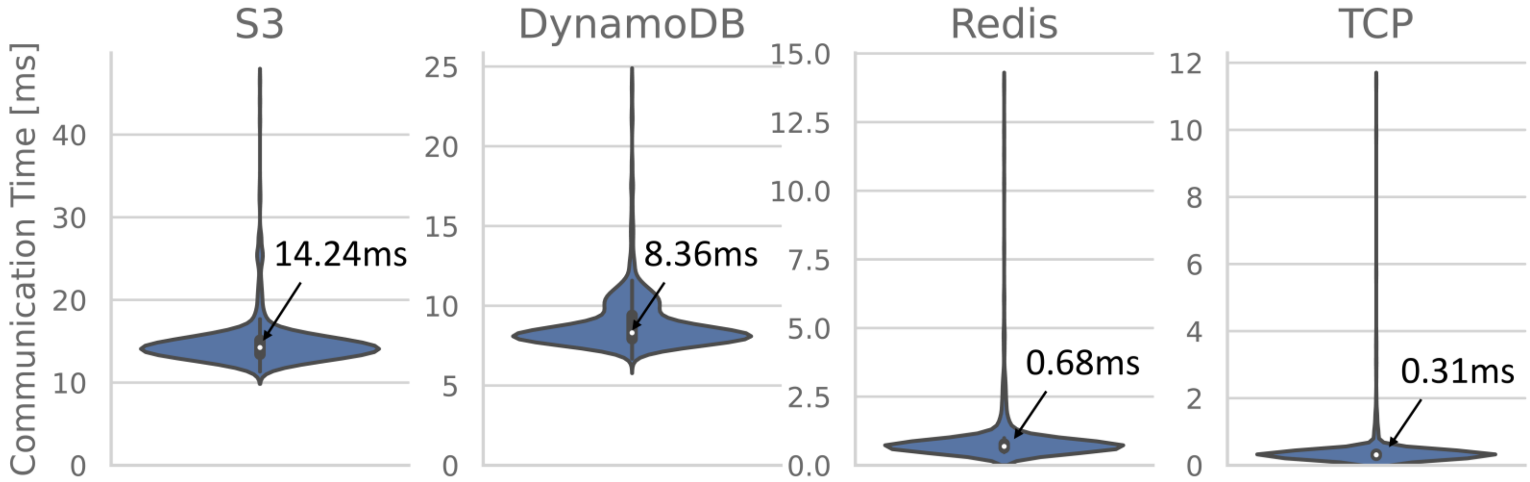


1 MB

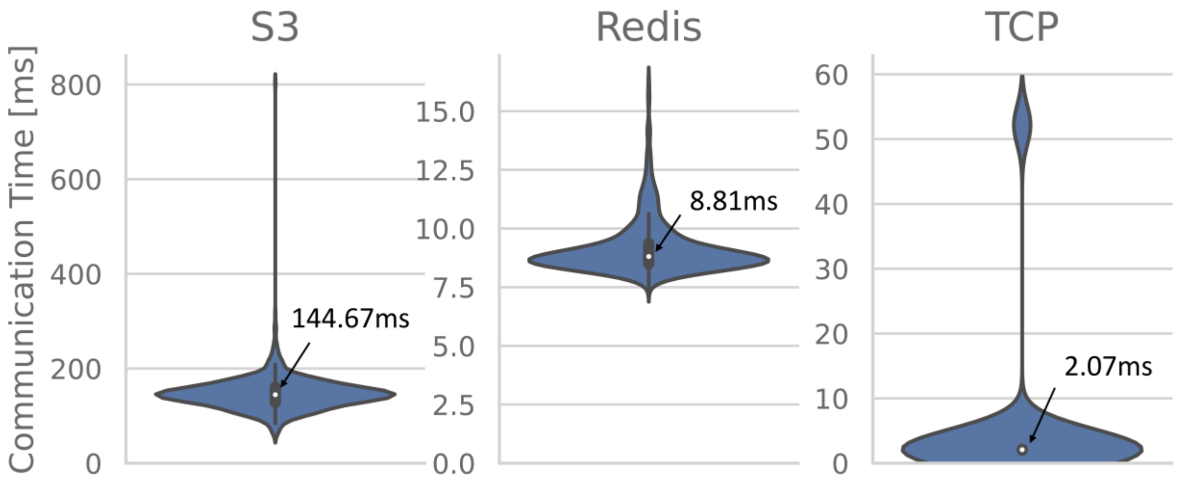


Communication channels

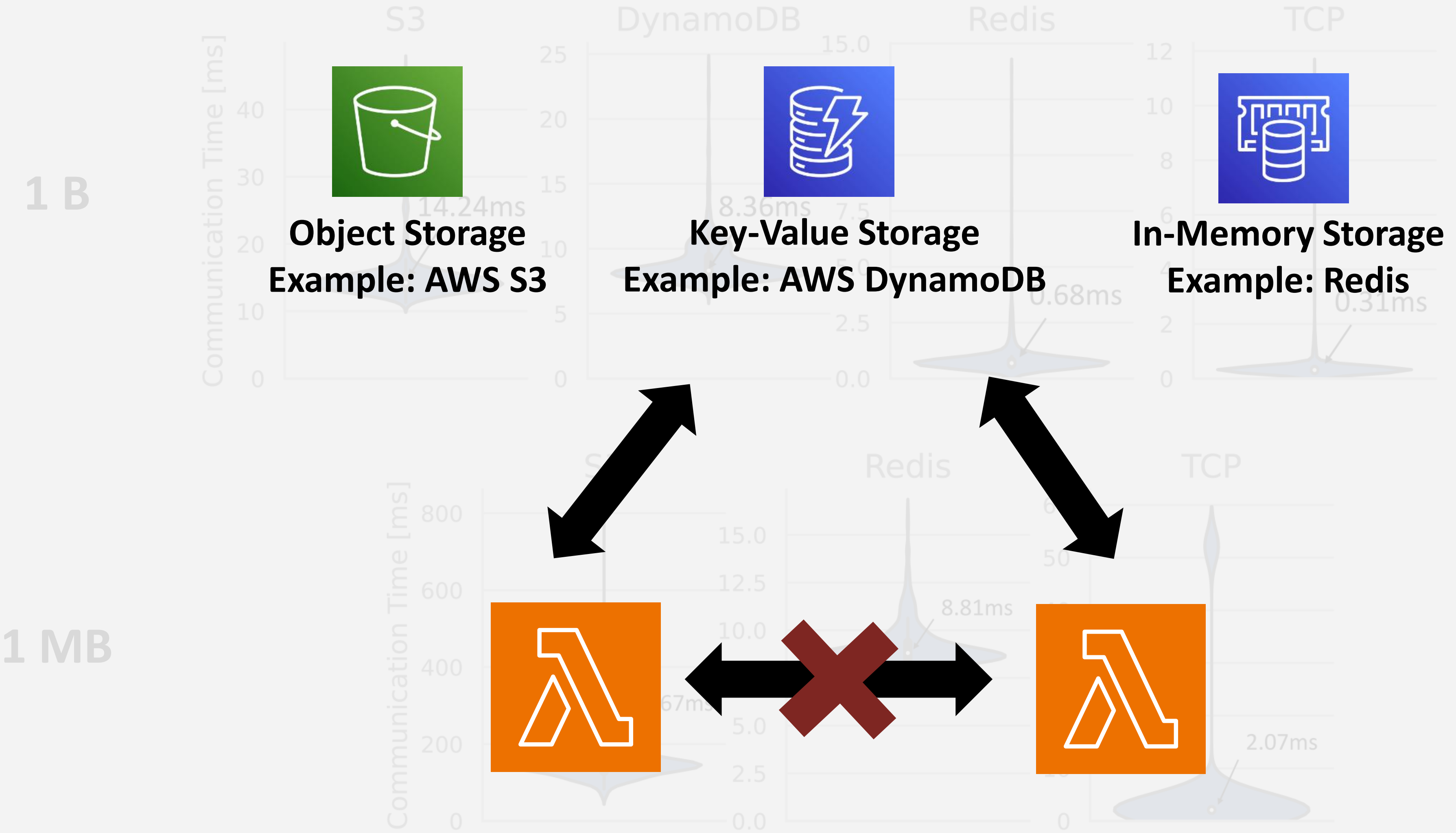
1 B



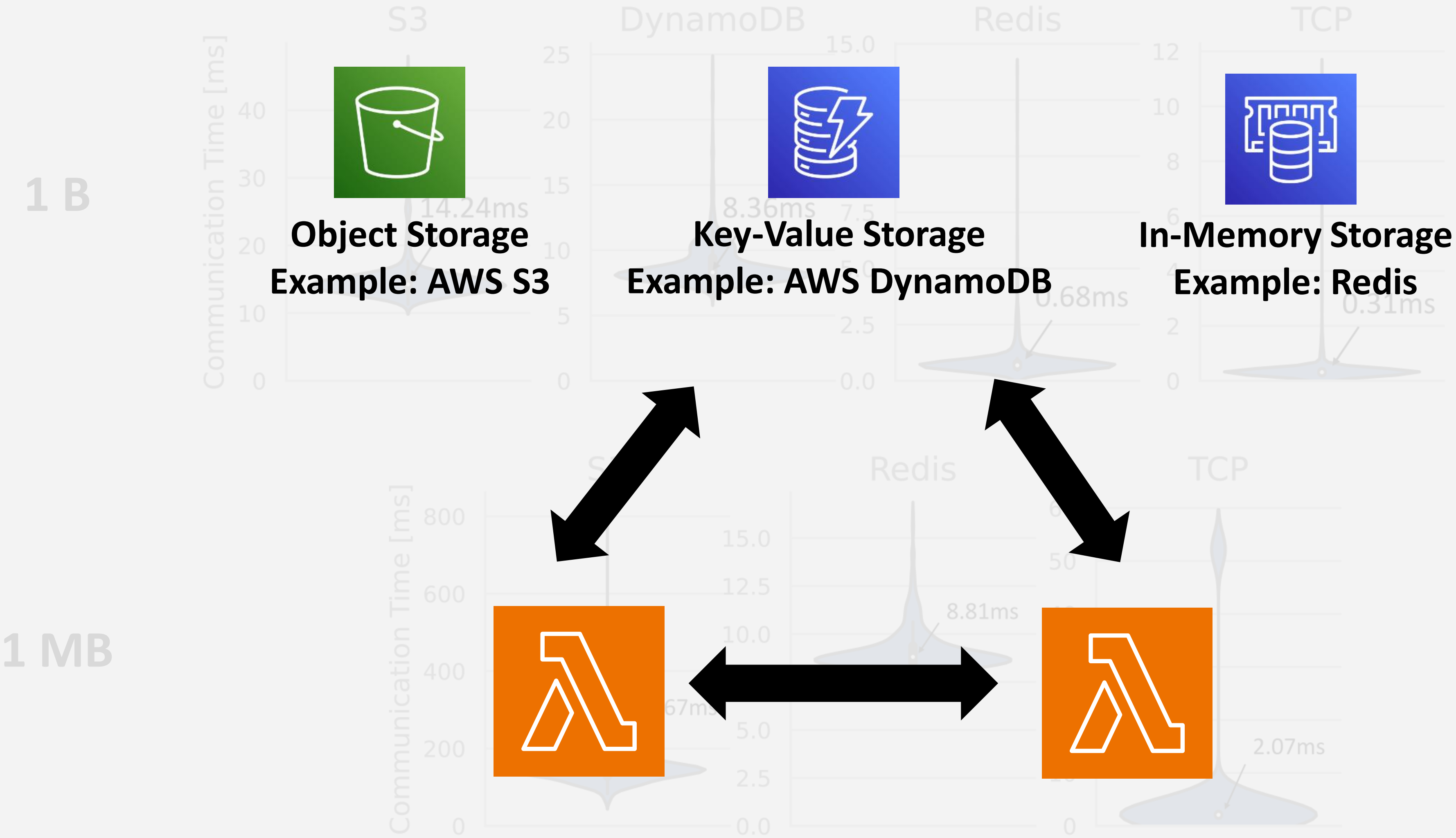
1 MB



Communication channels

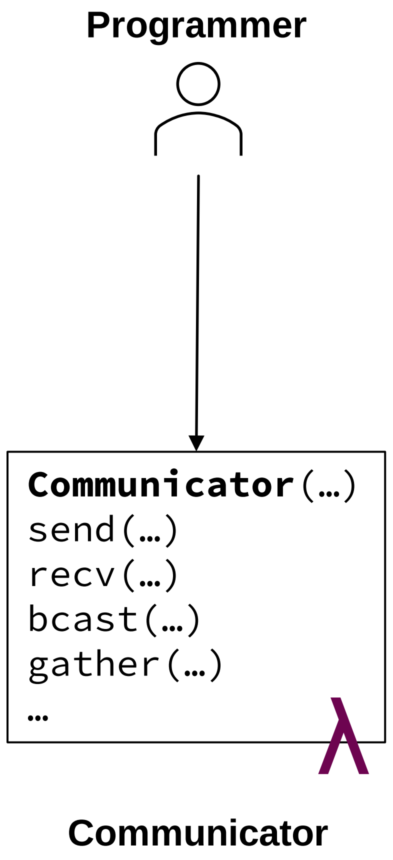


Communication channels

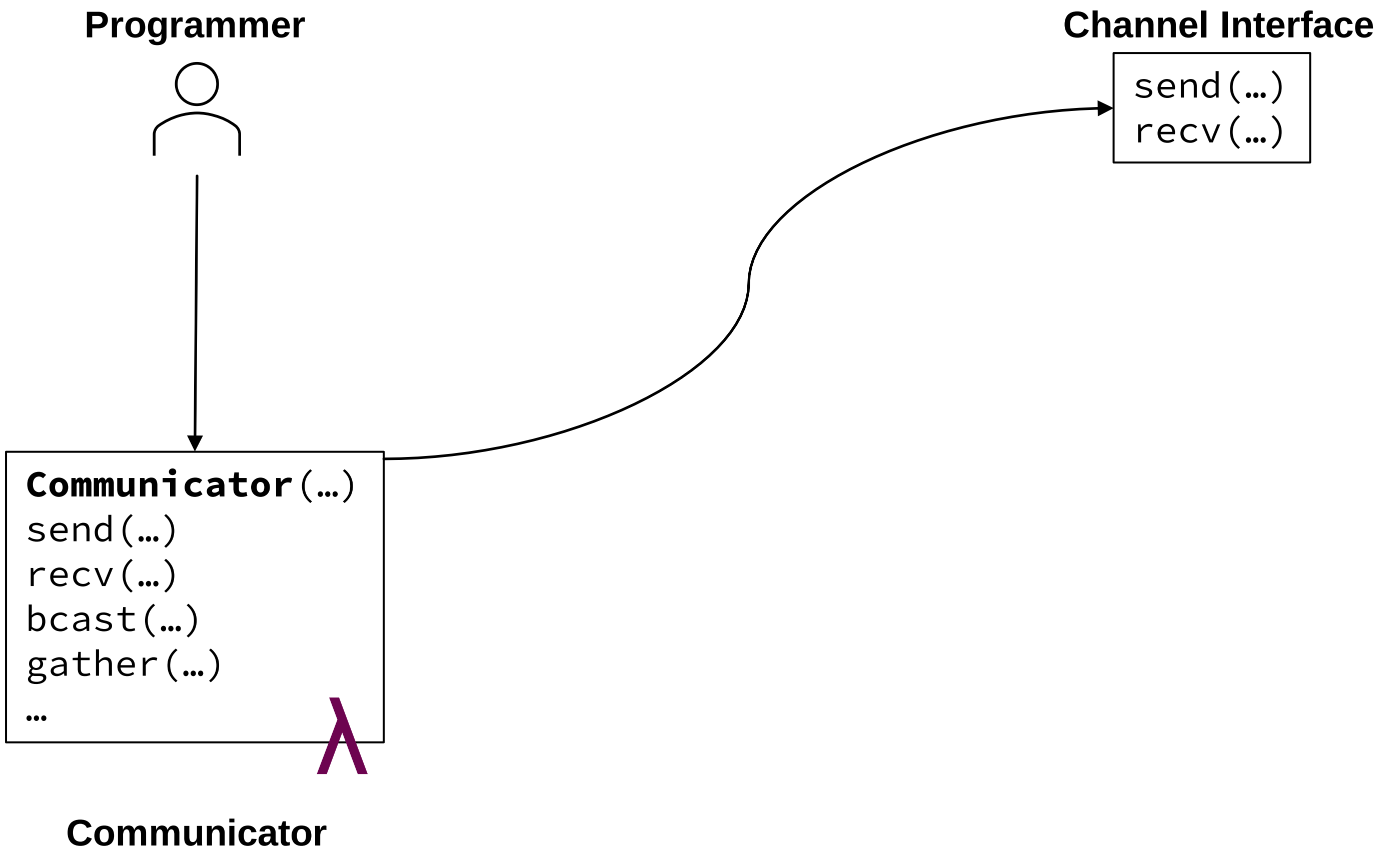


Function Message Interface (FMI): MPI for serverless

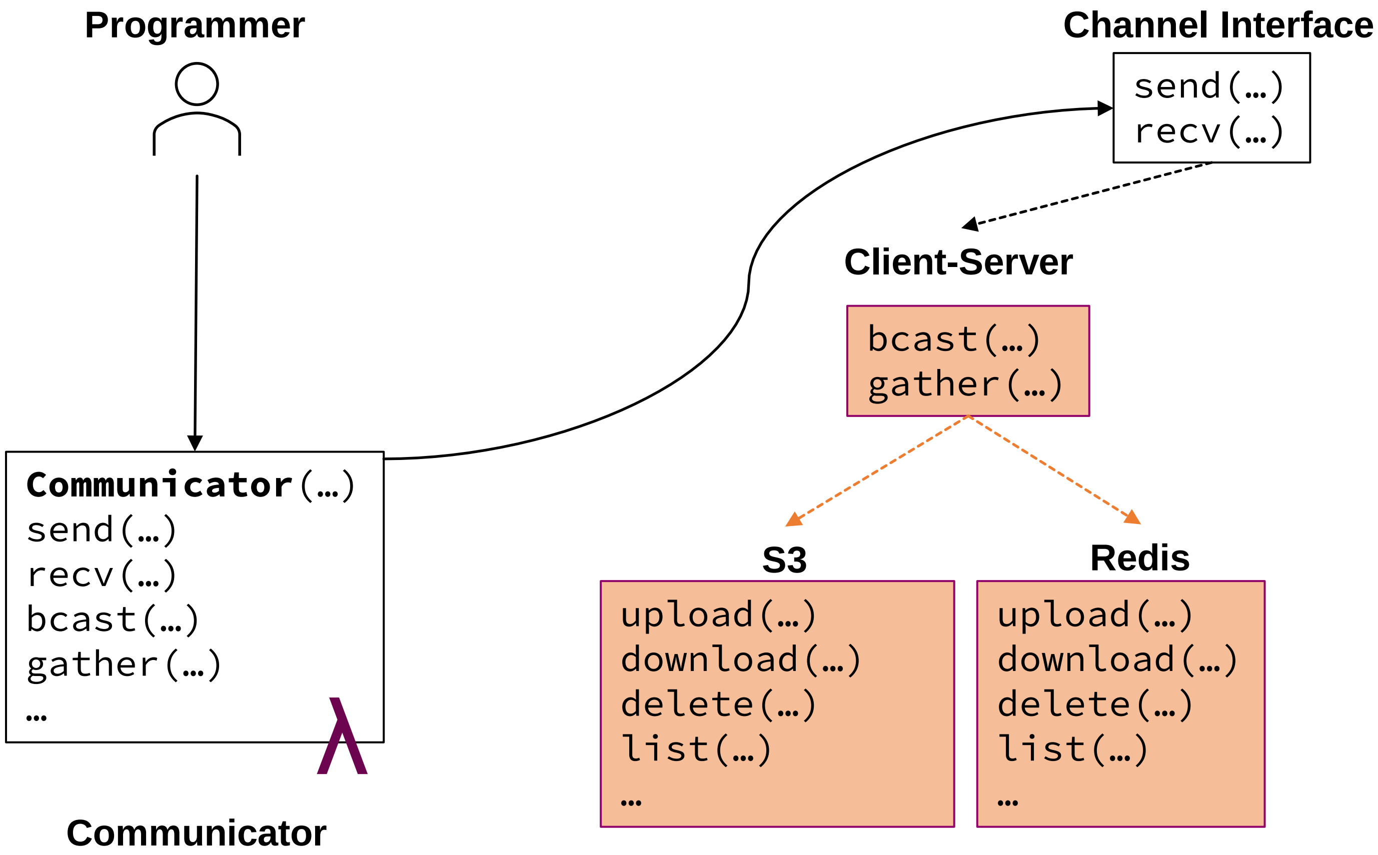
Function Message Interface (FMI): MPI for serverless



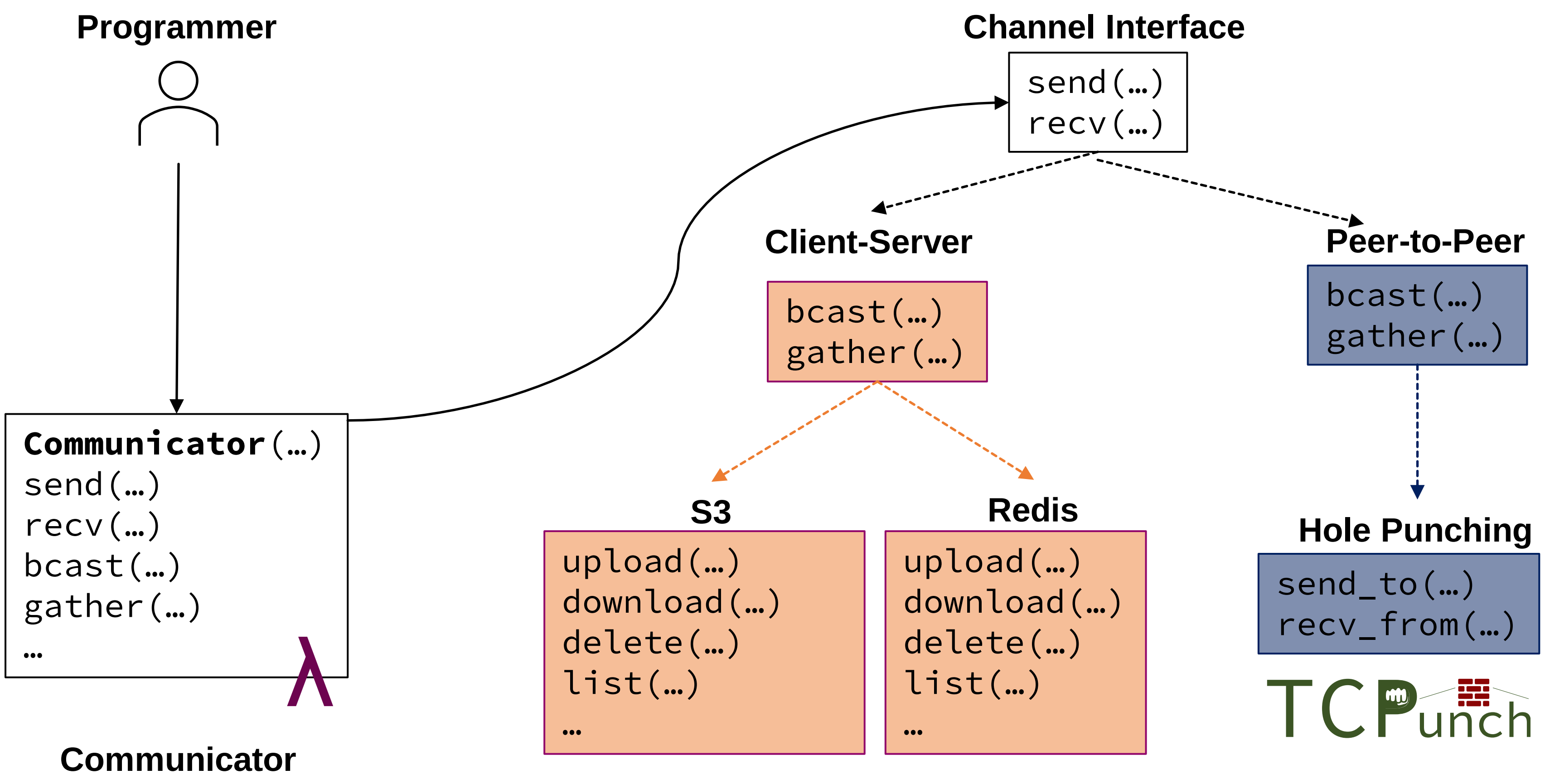
Function Message Interface (FMI): MPI for serverless



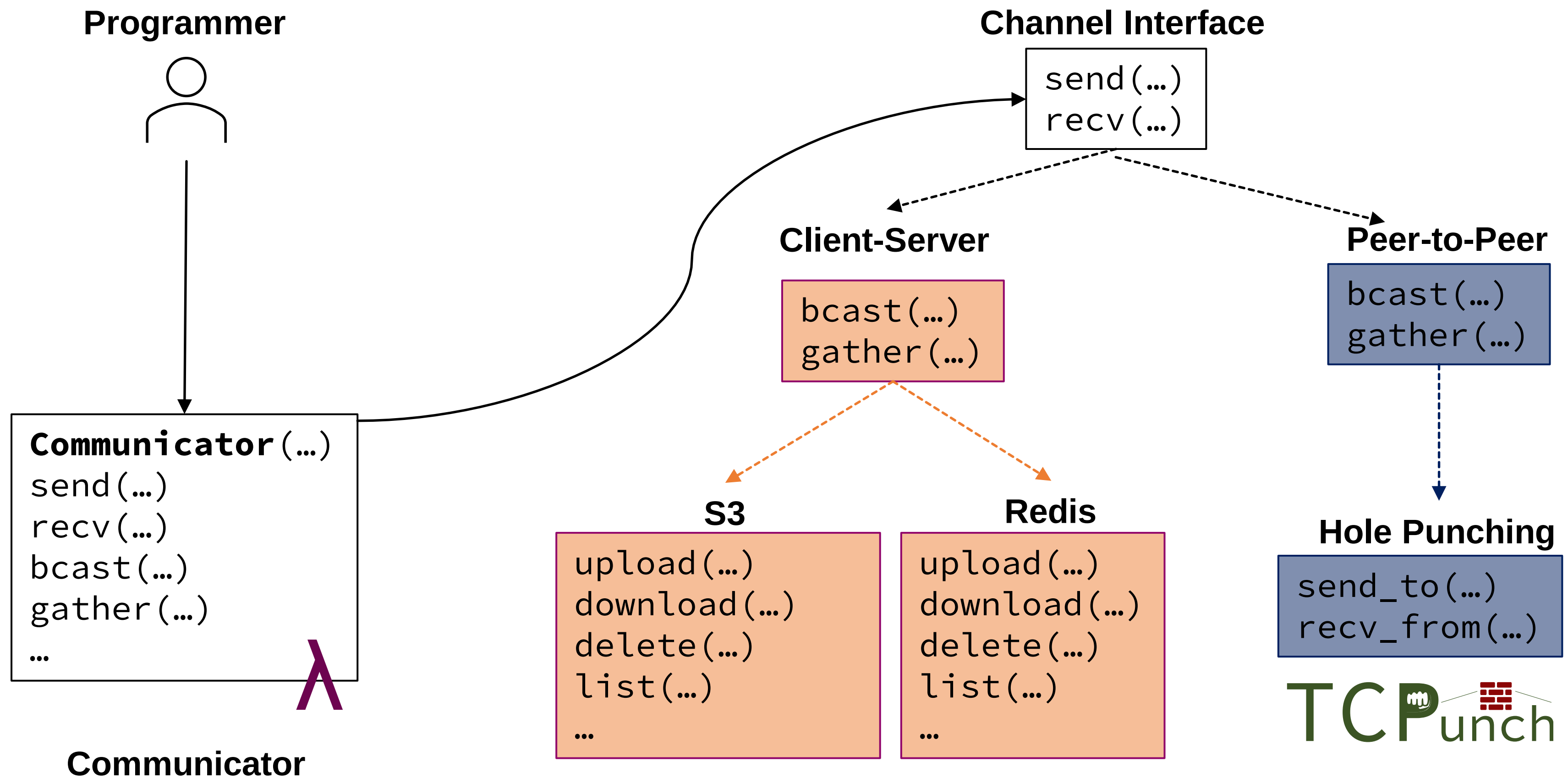
Function Message Interface (FMI): MPI for serverless



Function Message Interface (FMI): MPI for serverless



Function Message Interface (FMI): MPI for serverless



Serverless Cloud: communicate through **mediated** and **direct** channels.

HPC System: map FMI calls to an existing MPI library.

FMI in Python

```python
import fmi

comm = fmi.Communicator(
  peer_id=node_id,
  num_peers=num_nodes,
  config_path="config/fmi.json",
  comm_name="FMI_WORLD",
  faas_memory=function_memory
)
dtype = fmi.types(fmi.datatypes.int)

if my_id == 0:
    comm.bcast(42, 0, dtype)
else:
    assert comm.bcast(None, 0, dtype) == 42
```

FMI on AWS Lambda

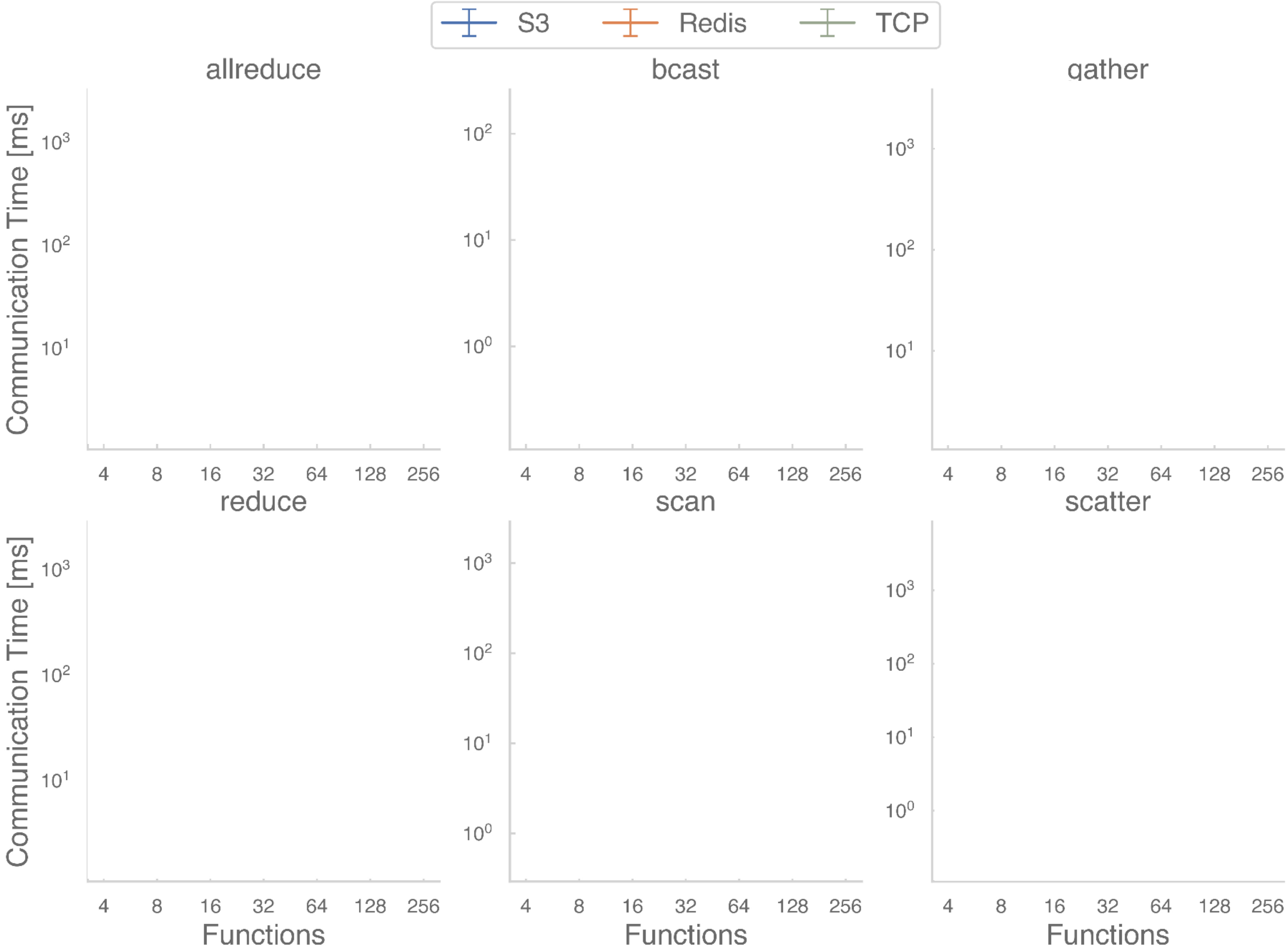
FMI Benchmark

- C++ functions, GCC 9.5
- AWS Lambda Functions with 2048 MB memory
- AWS ElastiCache Redis, cache.t3.small
- S3 with 20 ms polling interval

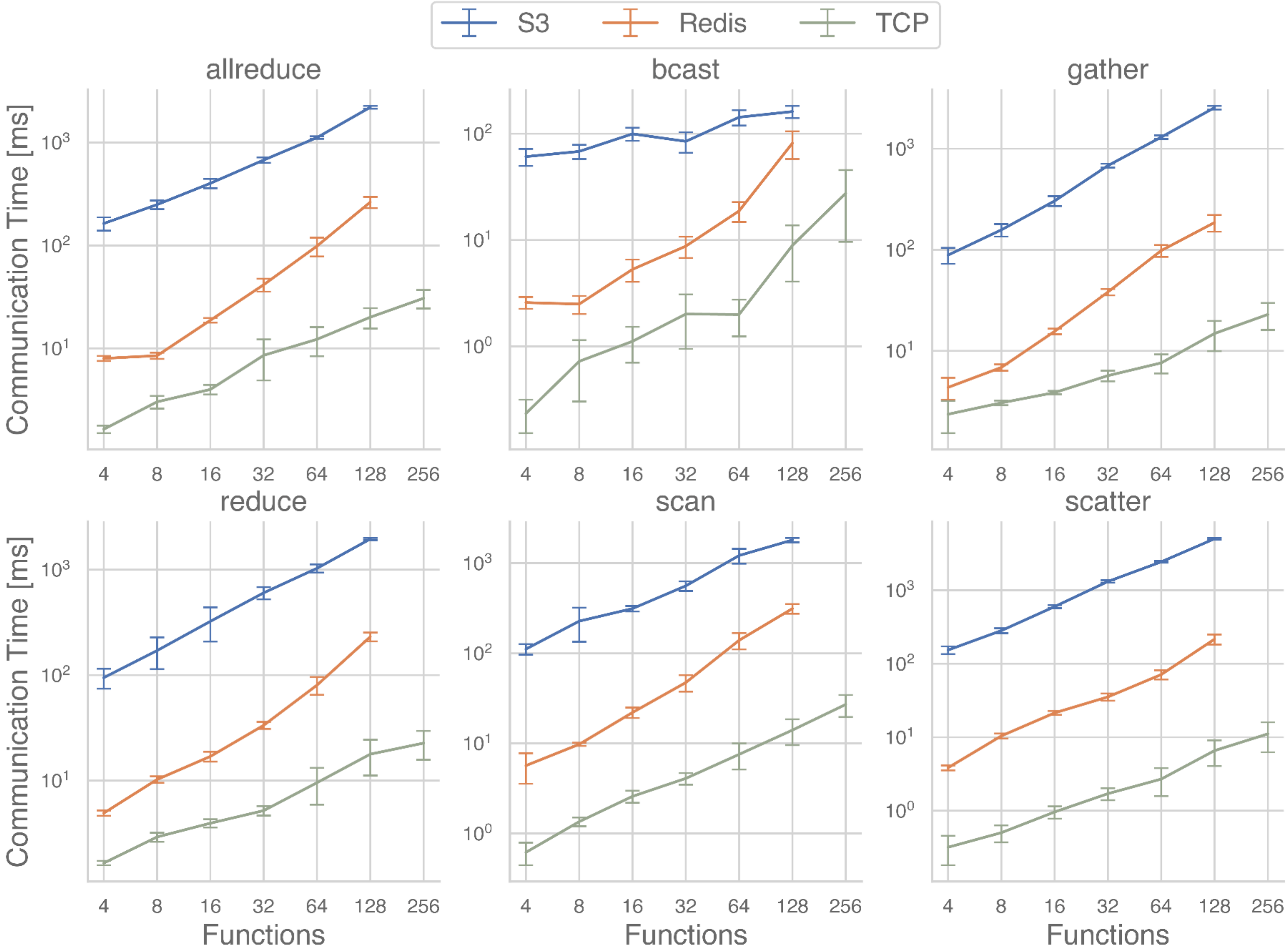
Collectives

- reduce and allreduce, adding one integer
- broadcast, 4 bytes
- scatter and gather, send 20,000 bytes
- scan, accumulating integers

FMI on AWS Lambda



FMI on AWS Lambda



LambdaML

Towards Demystifying Serverless Machine Learning Training

Jiawei Jiang[*,†], Shaoduo Gan[*,†], Yue Liu[†], Fanlin Wang[†]
Gustavo Alonso[†], Ana Klimovic[†], Ankit Singla[†], Wentao Wu[#], Ce Zhang[†]
[†]Systems Group, ETH Zürich [#]Microsoft Research, Redmond

SIGMOD, 2021

LambdaML

Towards Demystifying Serverless Machine Learning Training

Jiawei Jiang[*,†], Shaoduo Gan[*,†], Yue Liu[†], Fanlin Wang[†]
Gustavo Alonso[†], Ana Klimovic[†], Ankit Singla[†], Wentao Wu[#], Ce Zhang[†]
[†]Systems Group, ETH Zürich [#]Microsoft Research, Redmond

SIGMOD, 2021

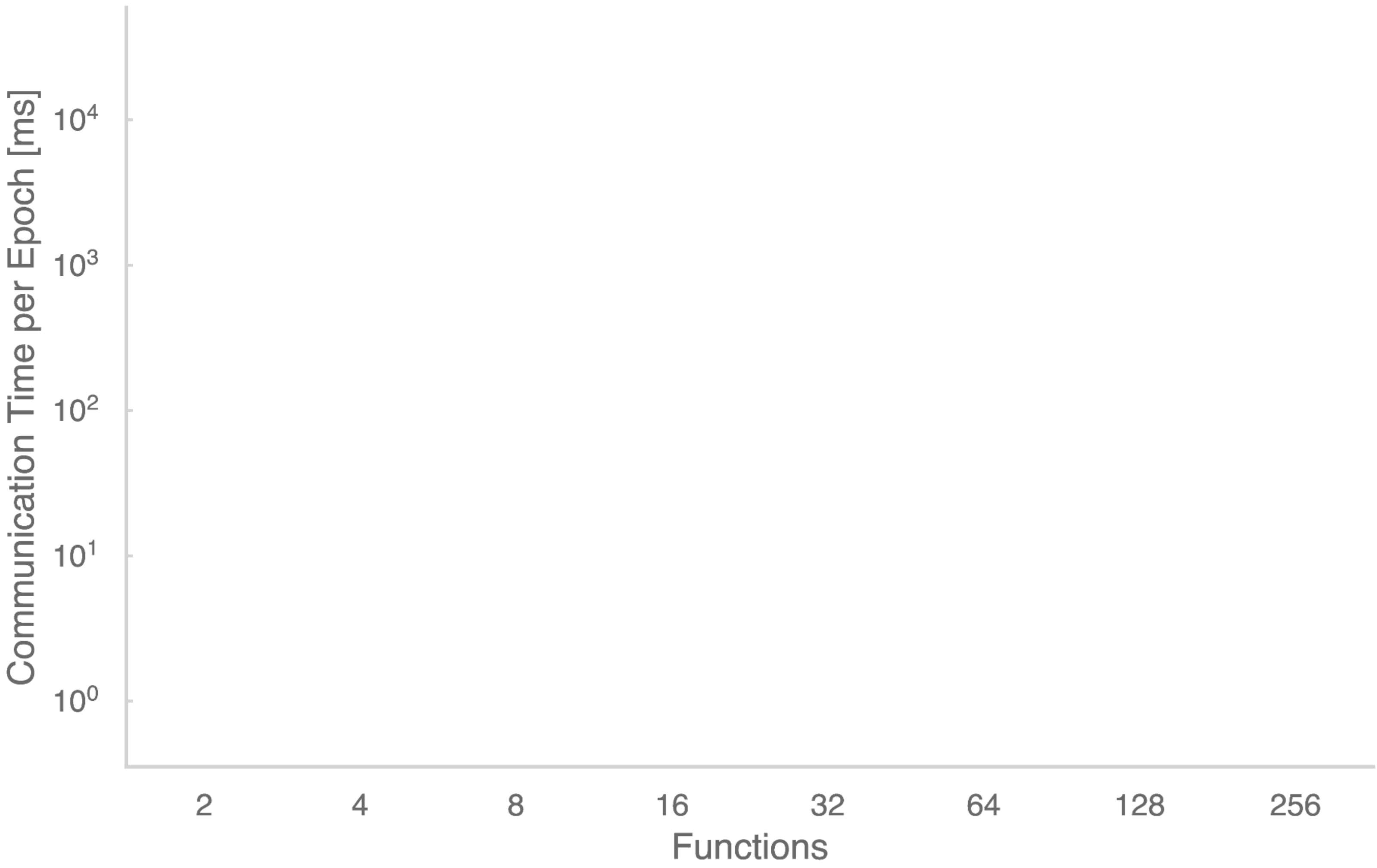
K-Means Benchmark

- HIGGS Dataset, 1 MB per function.
- 10 epochs.
- Lambda functions with 1024 MB memory.
- DynamoDB with autoscaling.

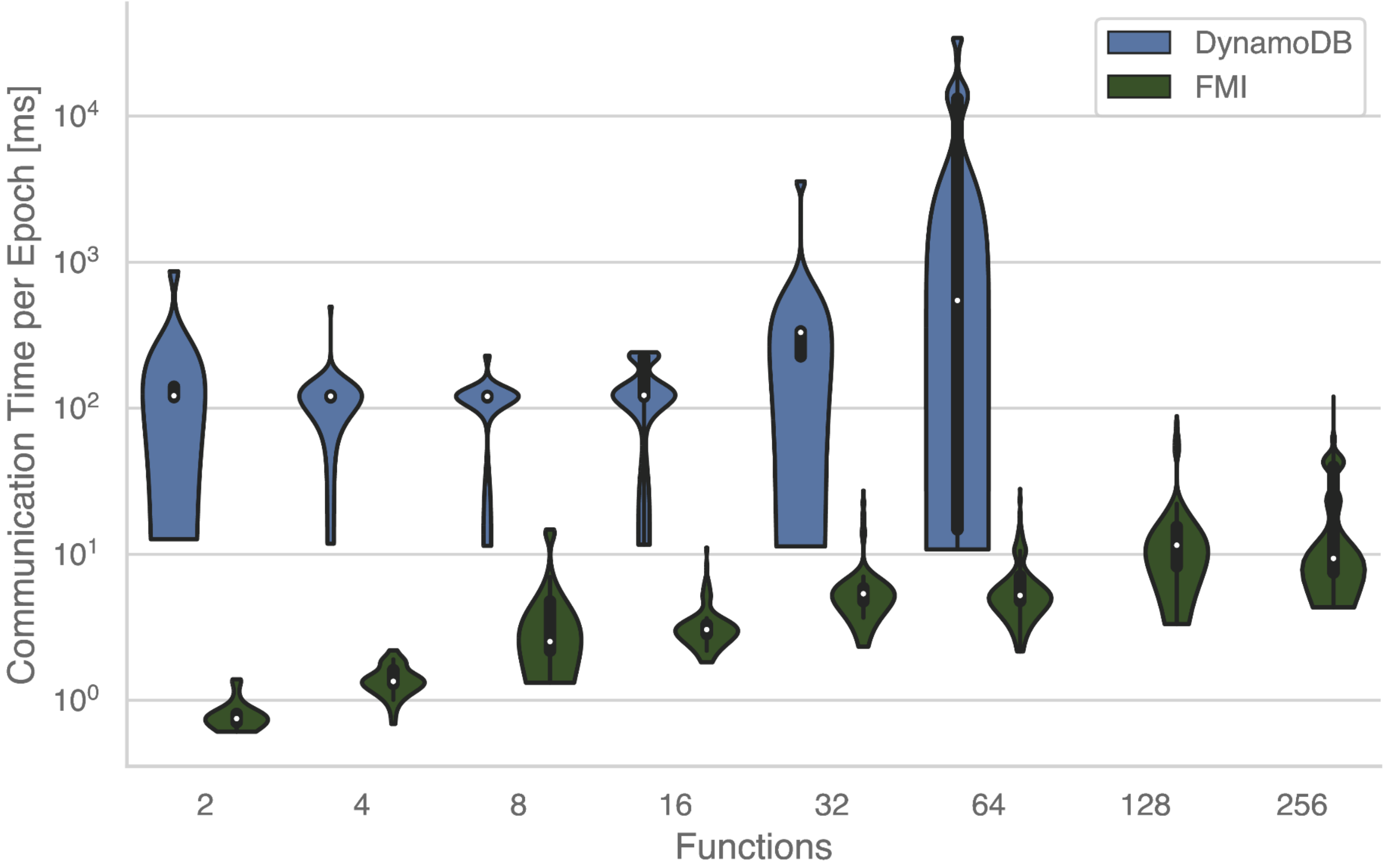
K-Means Benchmark with FMI

- Direct TCP communication.
- Replace reduce algorithm.

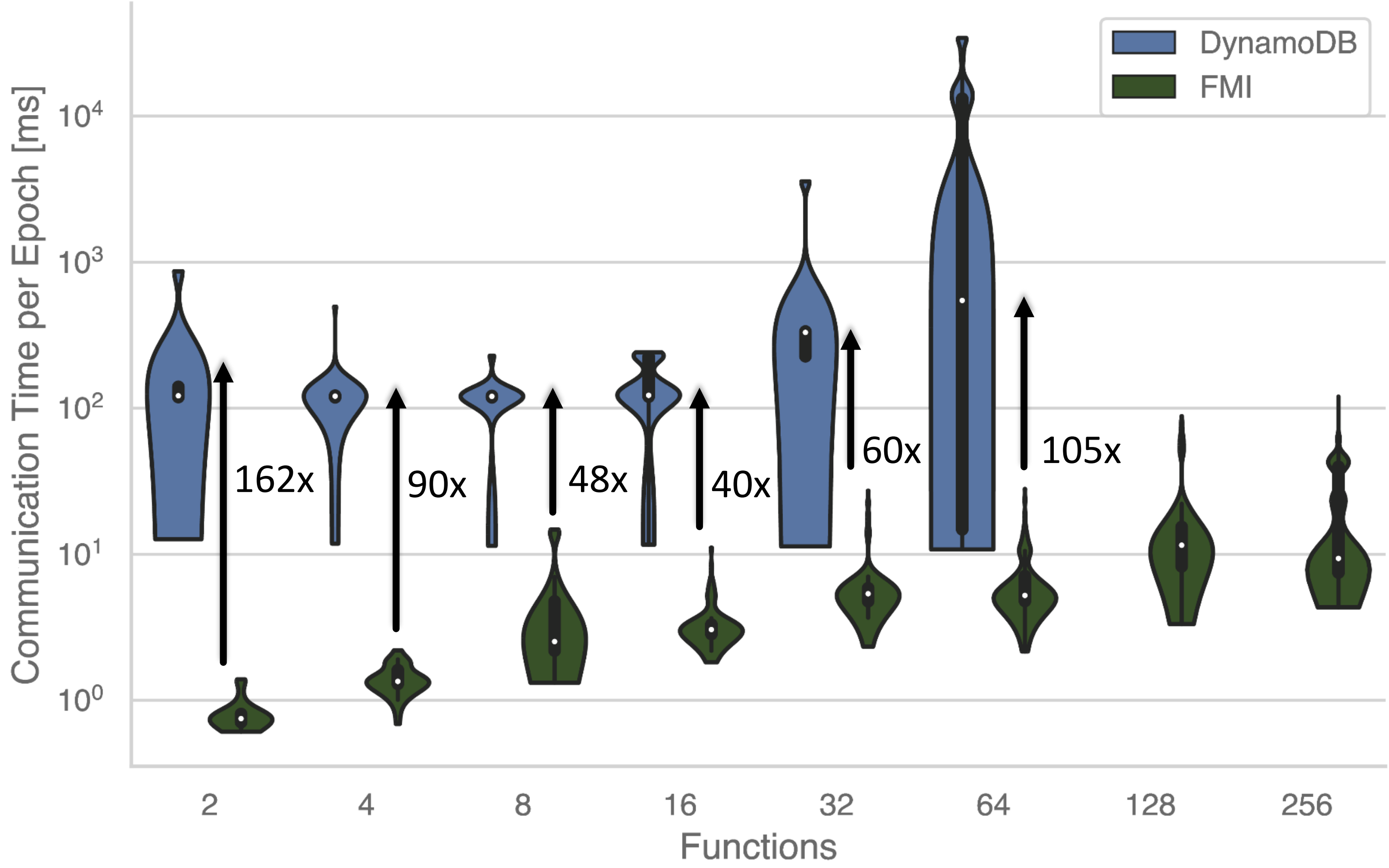
LambdaML Performance



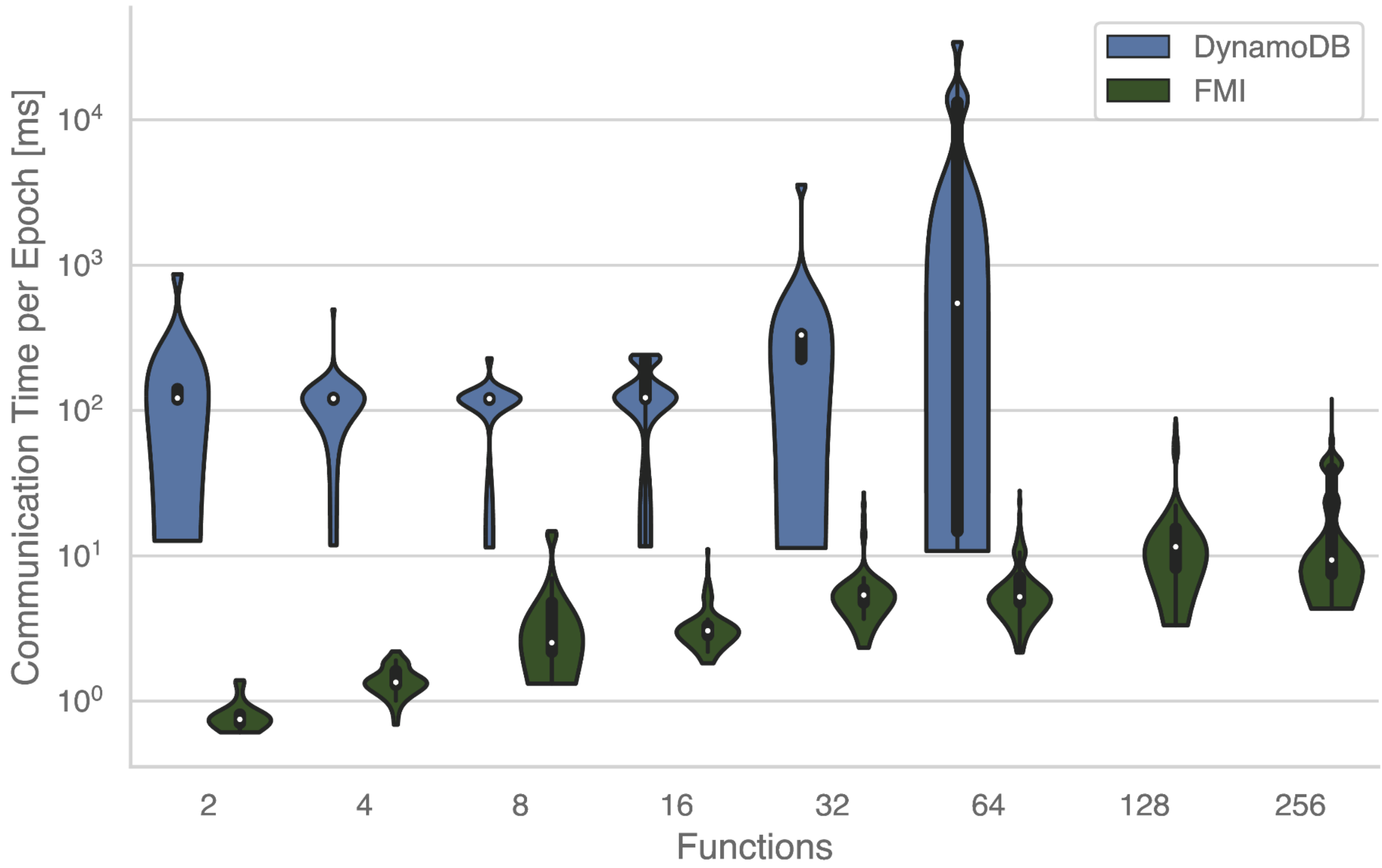
LambdaML Performance



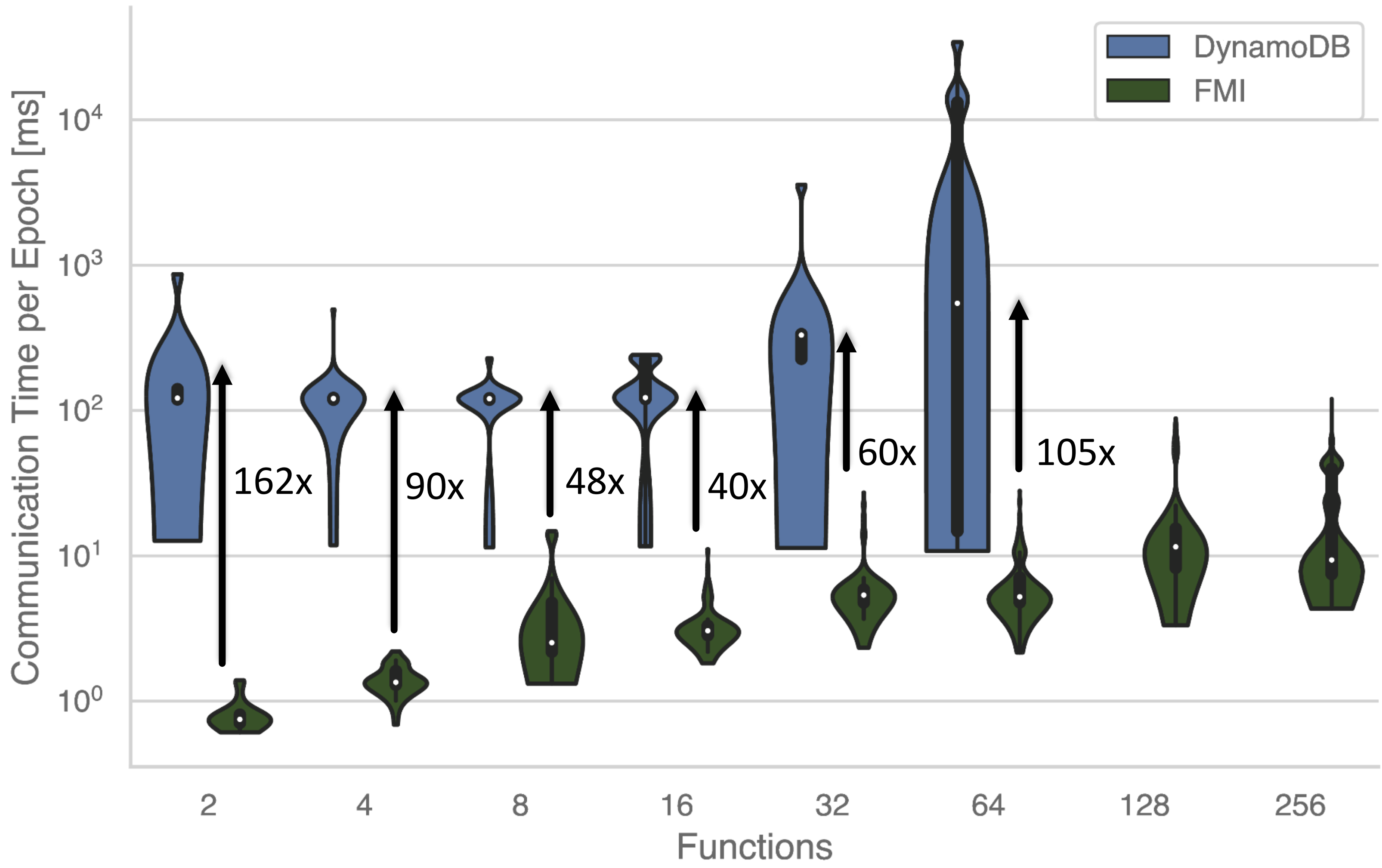
LambdaML Performance



LambdaML Performance



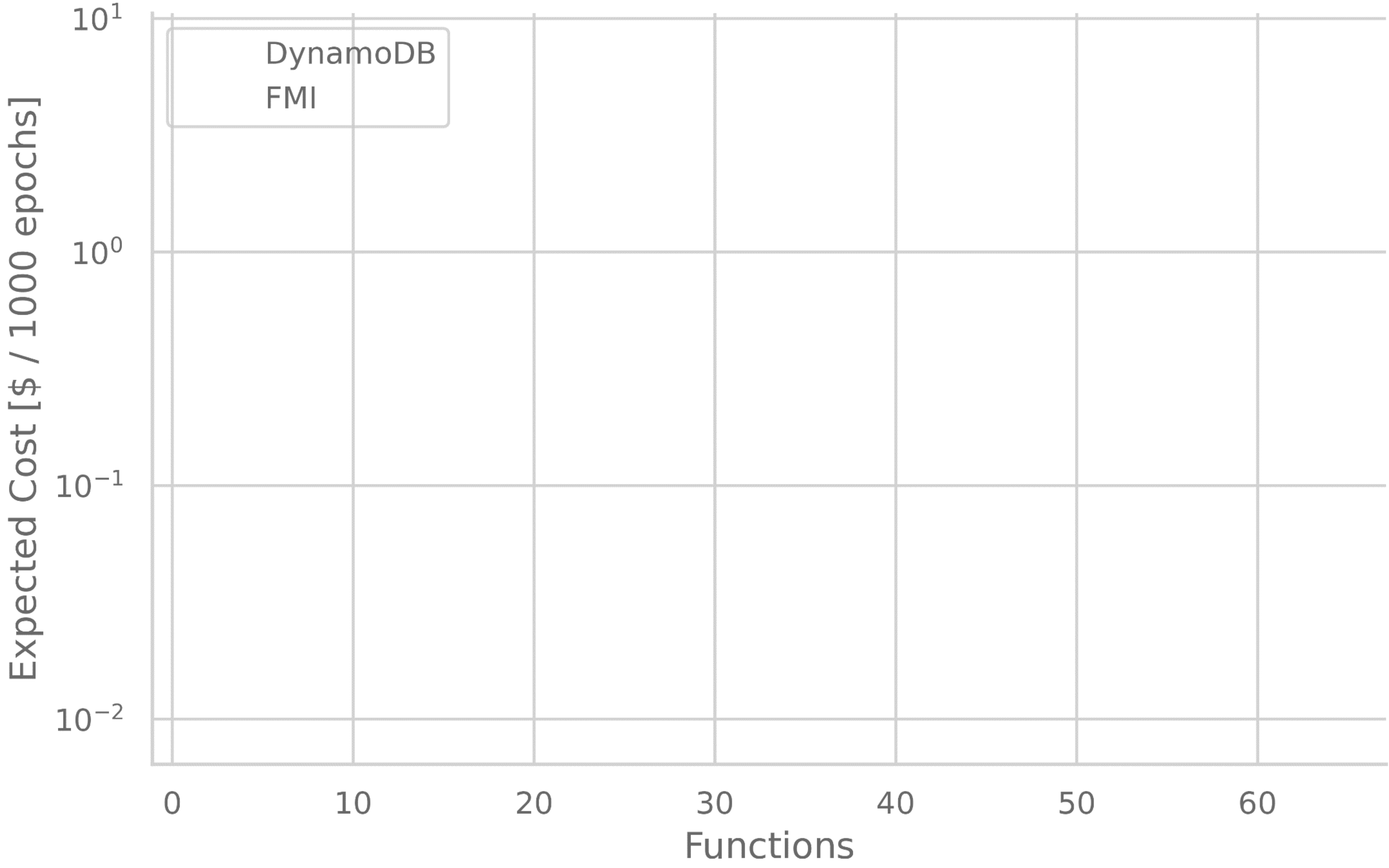
LambdaML Performance



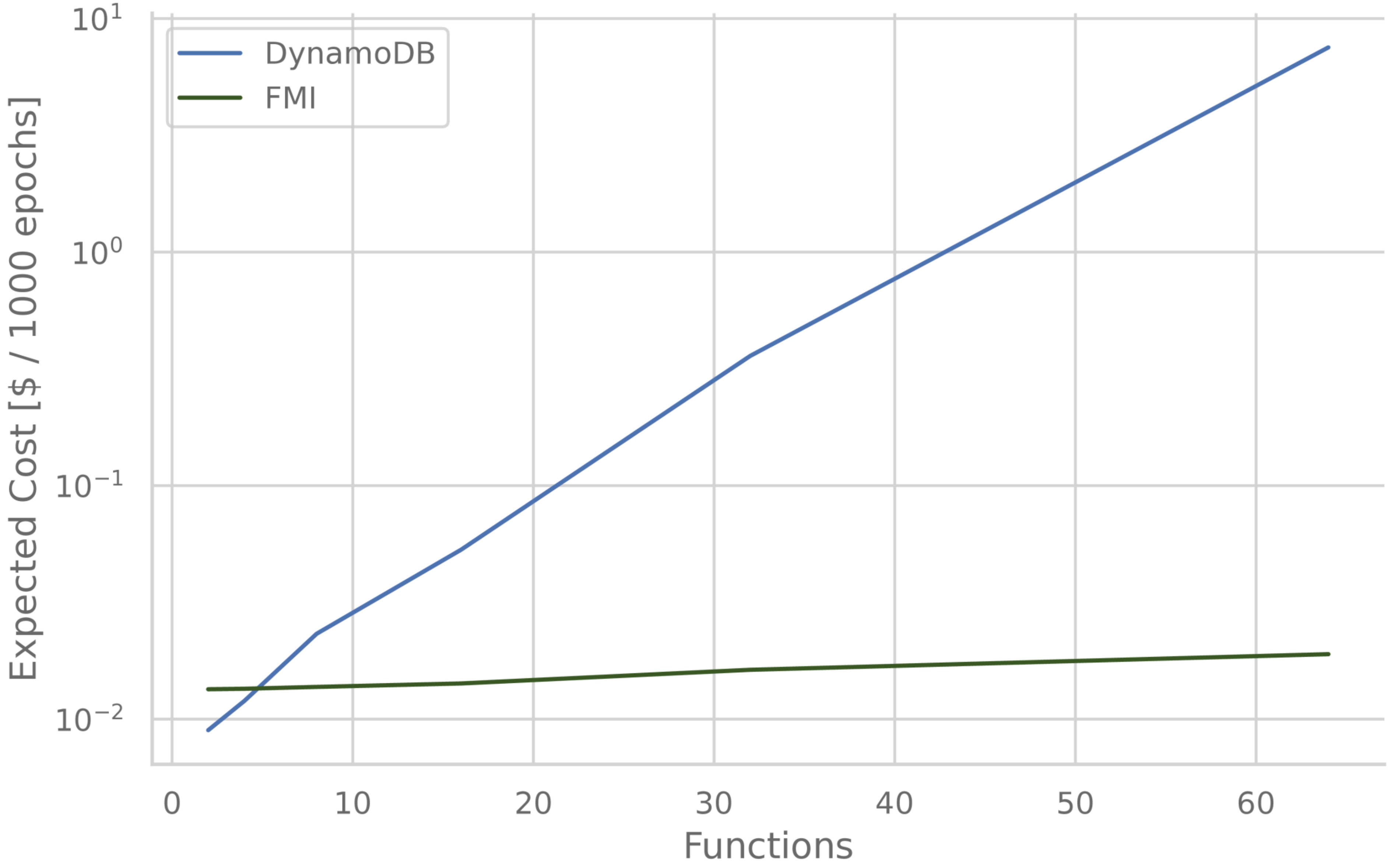
LambdaML Cost

DynamoDB Cost = (Compute Cost + Storage Cost) * Time
FMI TCP Cost = (Compute Cost + Hole Puncher Cost) * Time

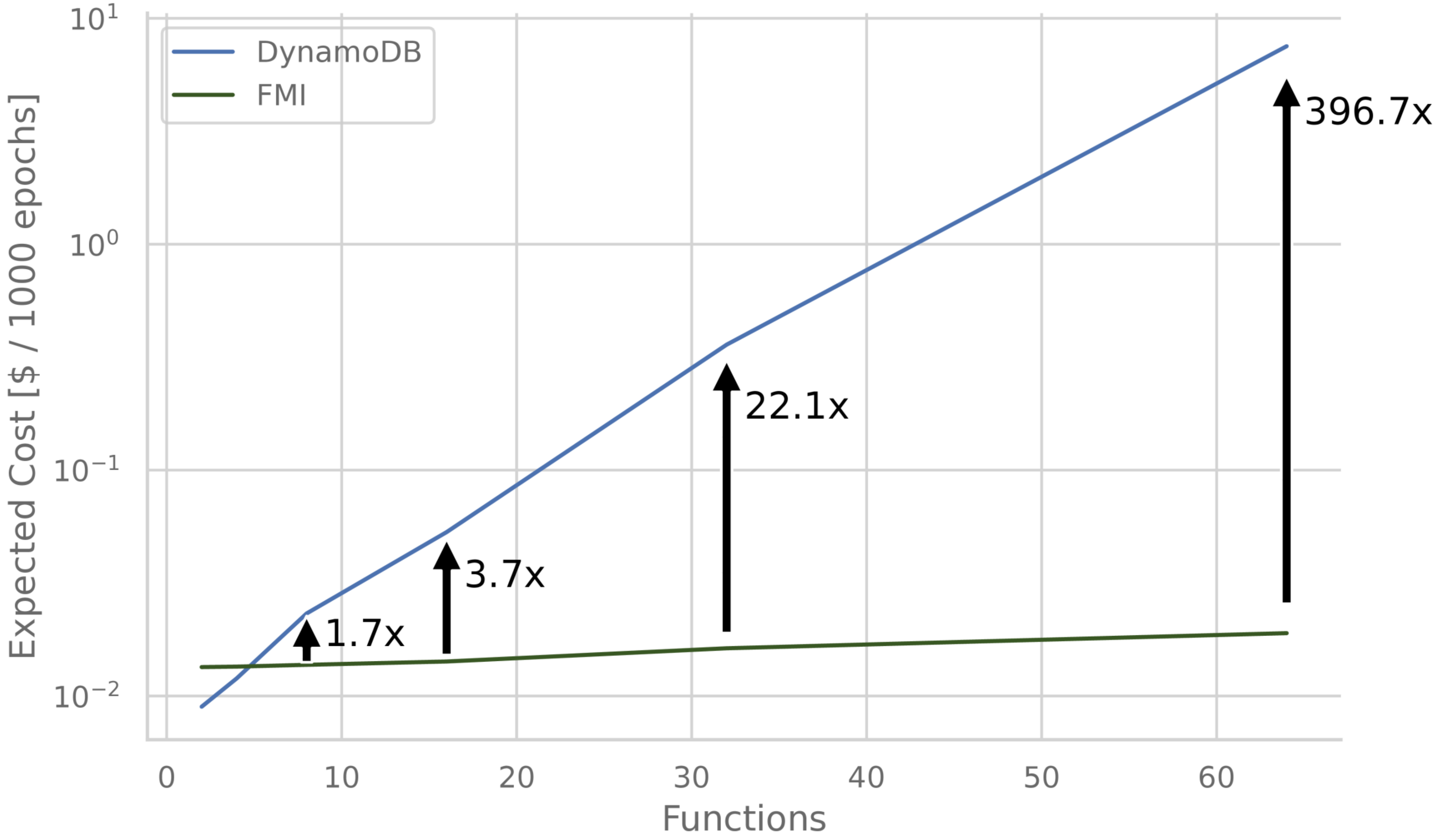
LambdaML Cost



LambdaML Cost



LambdaML Cost



Serverless Solutions for HPC

Serverless Solutions for HPC

spcl/fmi

spcl/ TCPunch

Serverless Solutions for HPC

spcl/fmi

spcl/ TCPunch

spcl/serverless-benchmarks

spcl/rFaaS

spcl/PraaS

Conclusions

More of SPCL's research:

youtube.com/@spcl — 150+ Talks

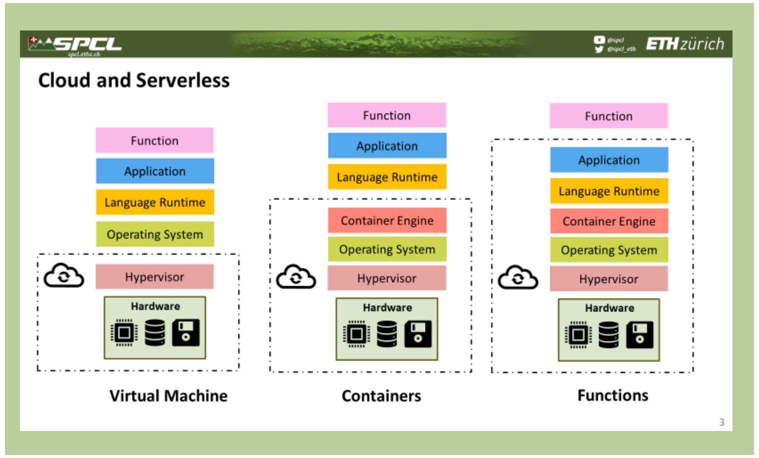
twitter.com/spcl_eth — 1.2K+ Followers

github.com/spcl — 2K+ Stars

… or spcl.ethz.ch

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

Conclusions



More of SPCL's research:

youtube.com/@spcl — 150+ Talks

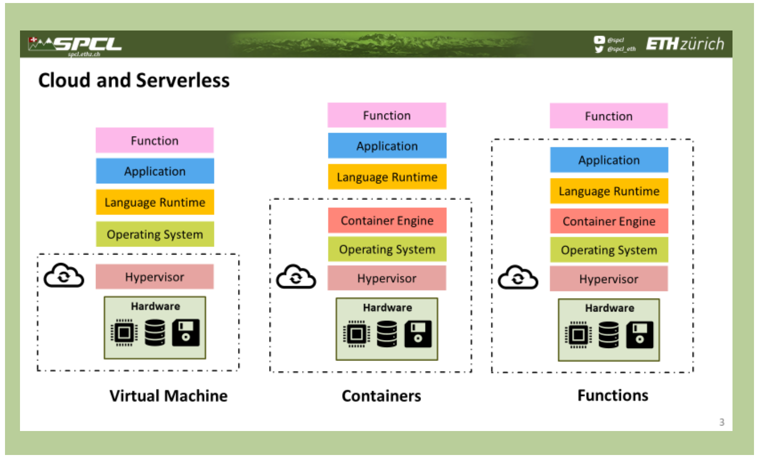
twitter.com/spcl_eth — 1.2K+ Followers

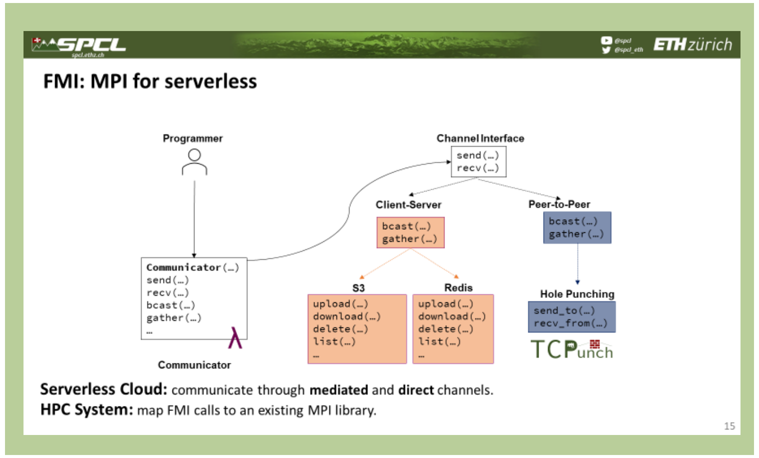
github.com/spcl — 2K+ Stars

… or spcl.ethz.ch

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

Conclusions





More of SPCL's research:

youtube.com/@spcl — 150+ Talks

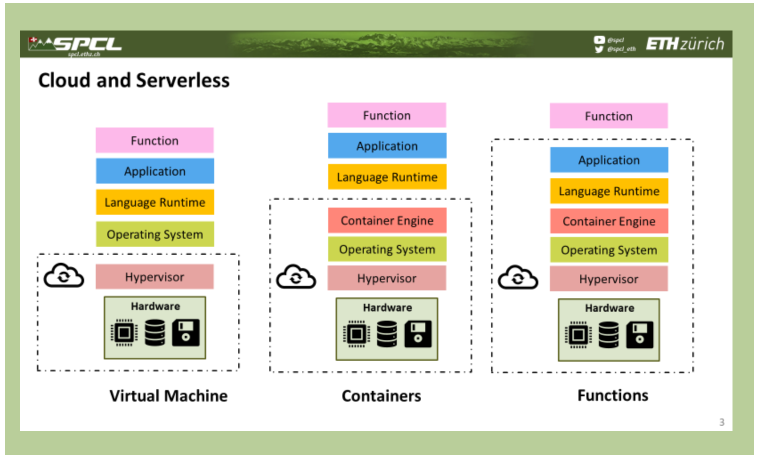
twitter.com/spcl_eth — 1.2K+ Followers

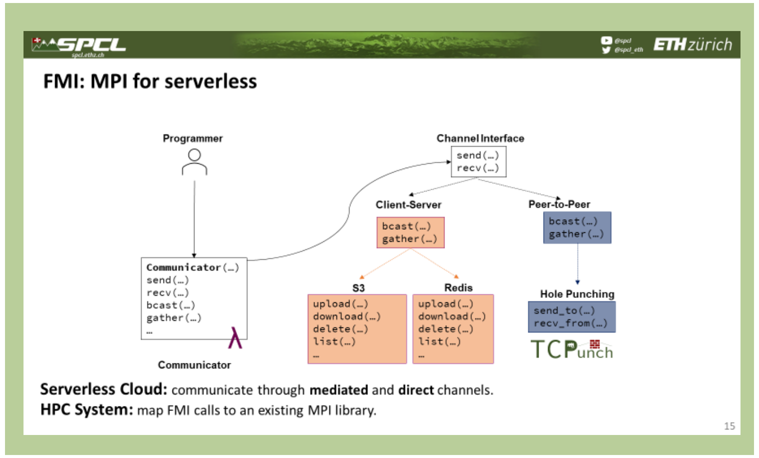
github.com/spcl — 2K+ Stars

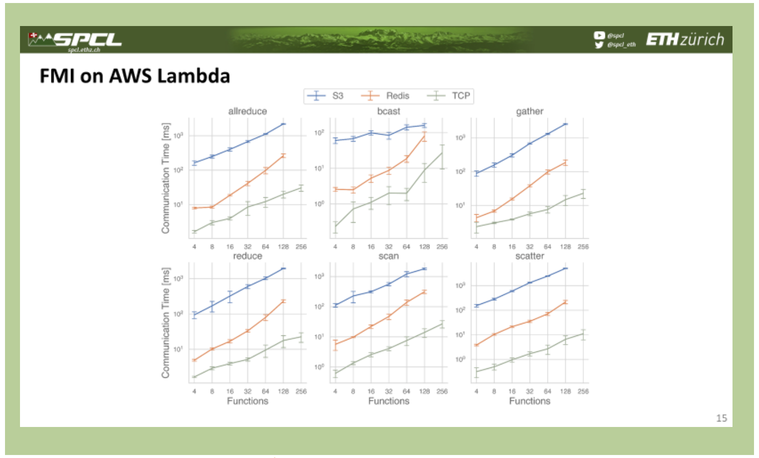
… or spcl.ethz.ch

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

Conclusions







More of SPCL's research:

youtube.com/@spcl — 150+ Talks

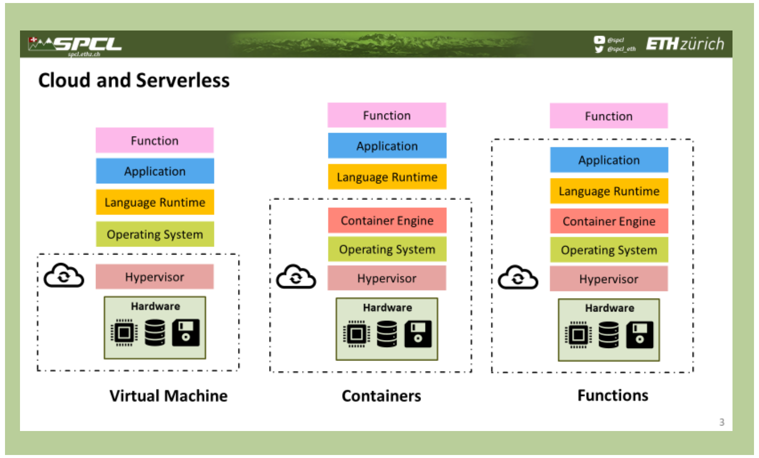
twitter.com/spcl_eth — 1.2K+ Followers

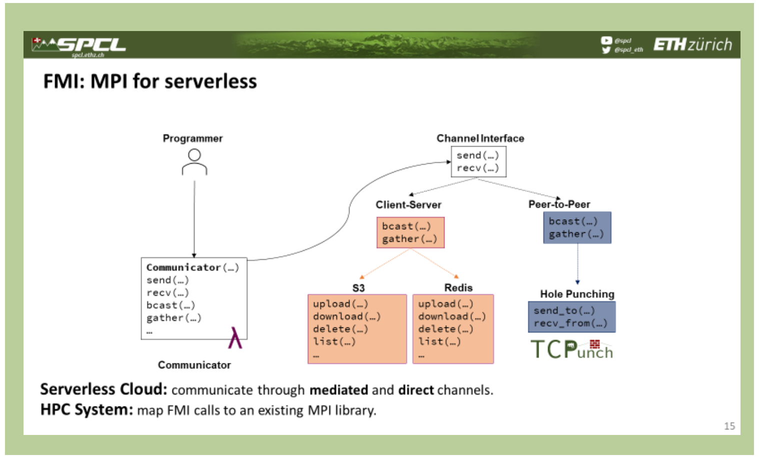
github.com/spcl — 2K+ Stars

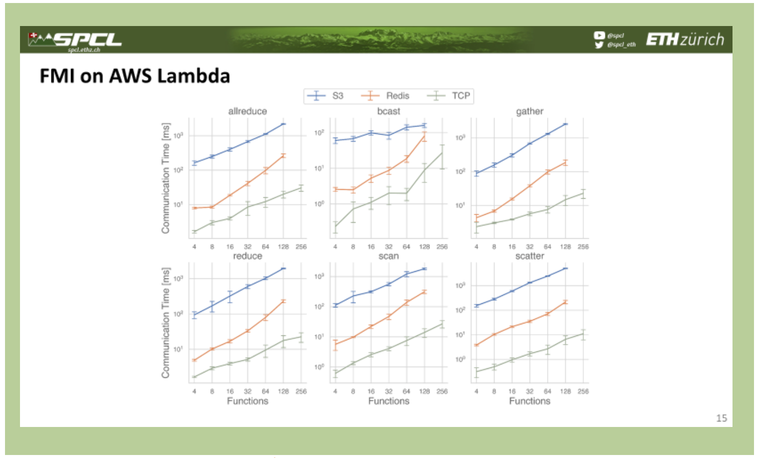
… or spcl.ethz.ch

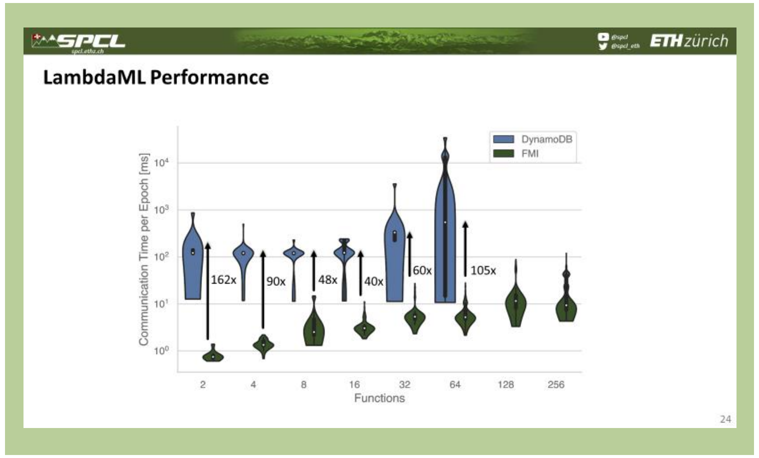
This work has received funding from the European Research Council (ERC) and from Amazon Web Services through the AWS Cloud Credits for Research program.

Conclusions









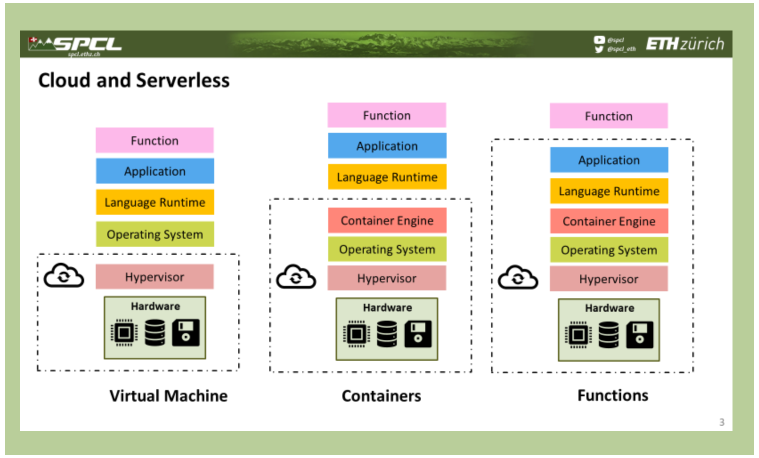
More of SPCL's research:

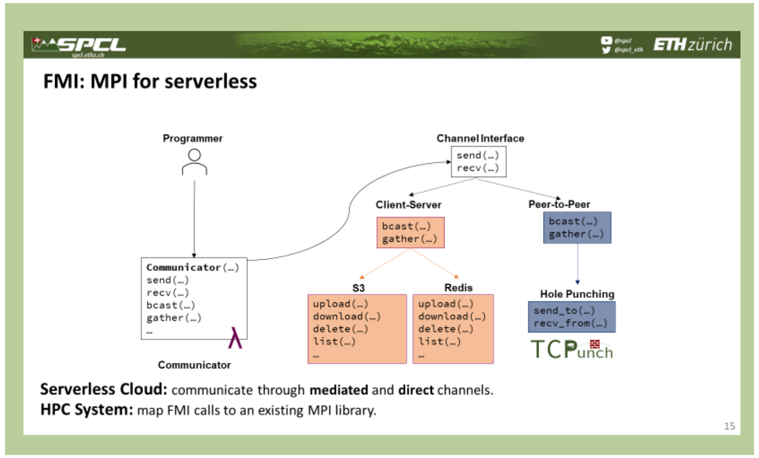
- youtube.com/@spcl — 150+ Talks
- twitter.com/spcl_eth — 1.2K+ Followers
- github.com/spcl — 2K+ Stars

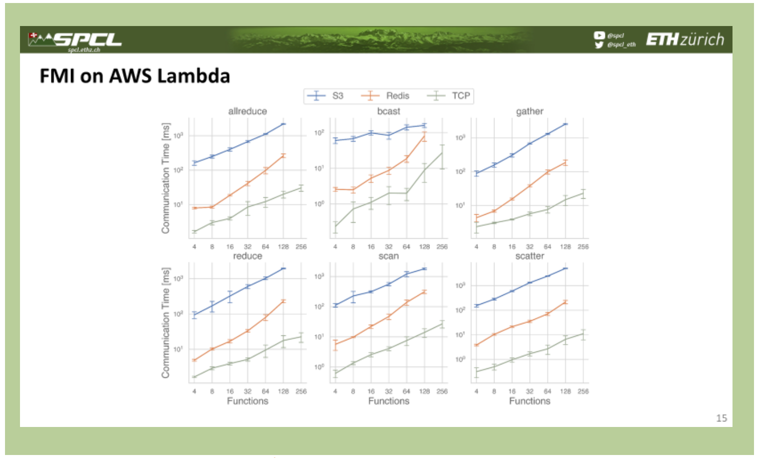
… or spcl.ethz.ch

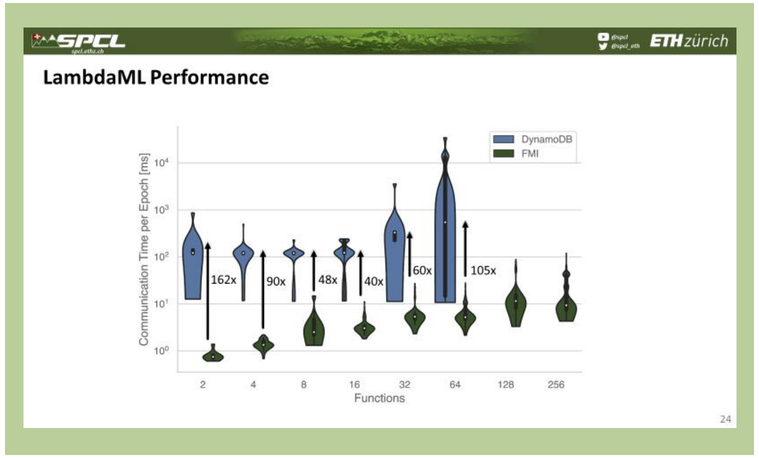
This work has received funding from the European Research Council (ERC) and from Amazon Web Services through the AWS Cloud Credits for Research program.

Conclusions









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

Paper **Paper artifact**

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