FMI: The FaaS Message Interface

ROMAN BÖHRINGER

FaaS

On Premises	Colocation	Infrastructure as a Service (IaaS)	Platform as a Service (PaaS)	Function as a Service (FaaS)	Software as a Service (SaaS)
Data	Data	Data	Data	Data	Data
Functions	Functions	Functions	Functions	Functions	Functions
Application	Application	Application	Application	Application	Application
Runtime	Runtime	Runtime	Runtime	Runtime	Runtime
Operating System	Operating System	Operating System	Operating System	Operating System	Operating System
Virtualization	Virtualization	Virtualization	Virtualization	Virtualization	Virtualization
Data Center	Data Center	Data Center	Data Center	Data Center	Data Center

ETH zürich

EQUINIX

Amazon EC2

HEROKU

AWS Lambda

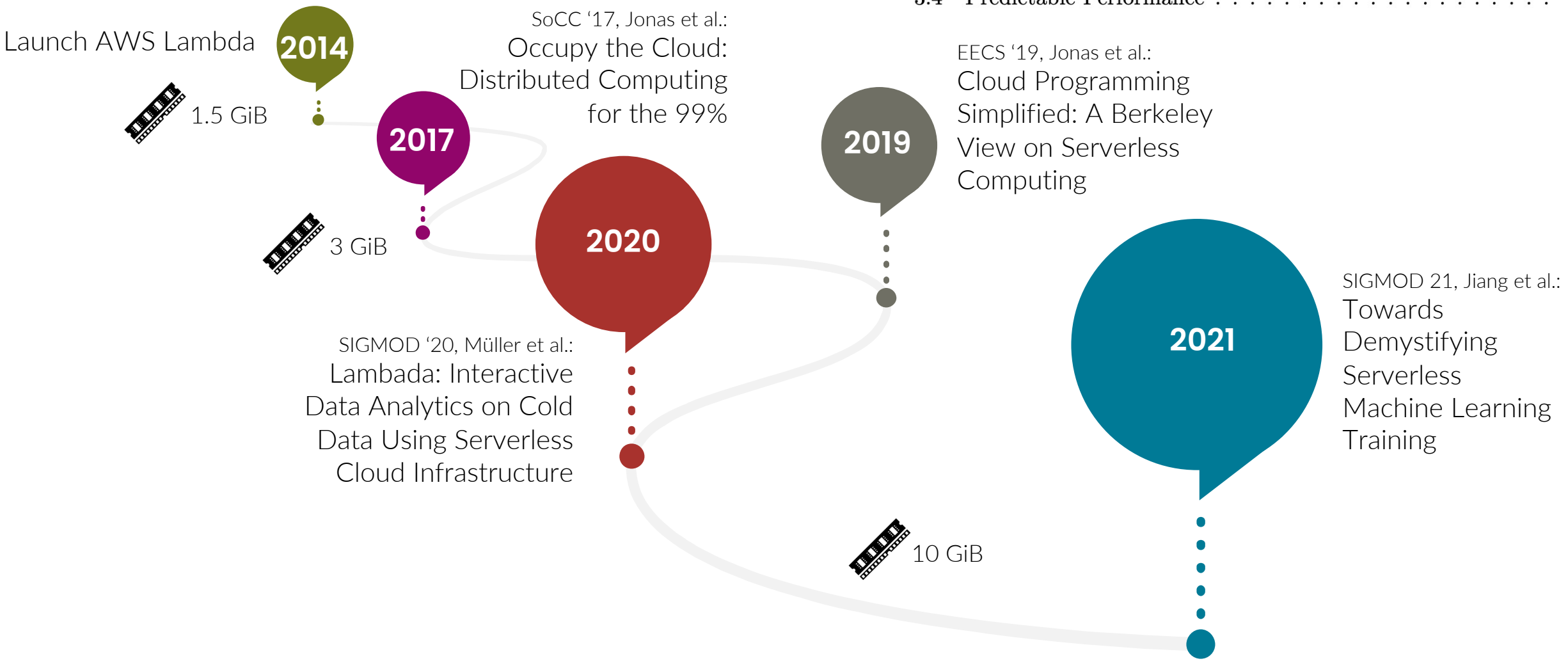
Google Docs

Managed by User

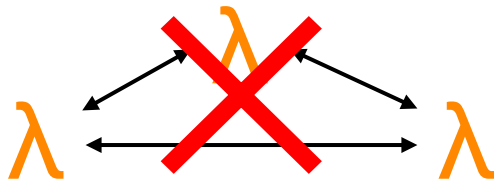
Managed by Provider

FaaS: Evolution

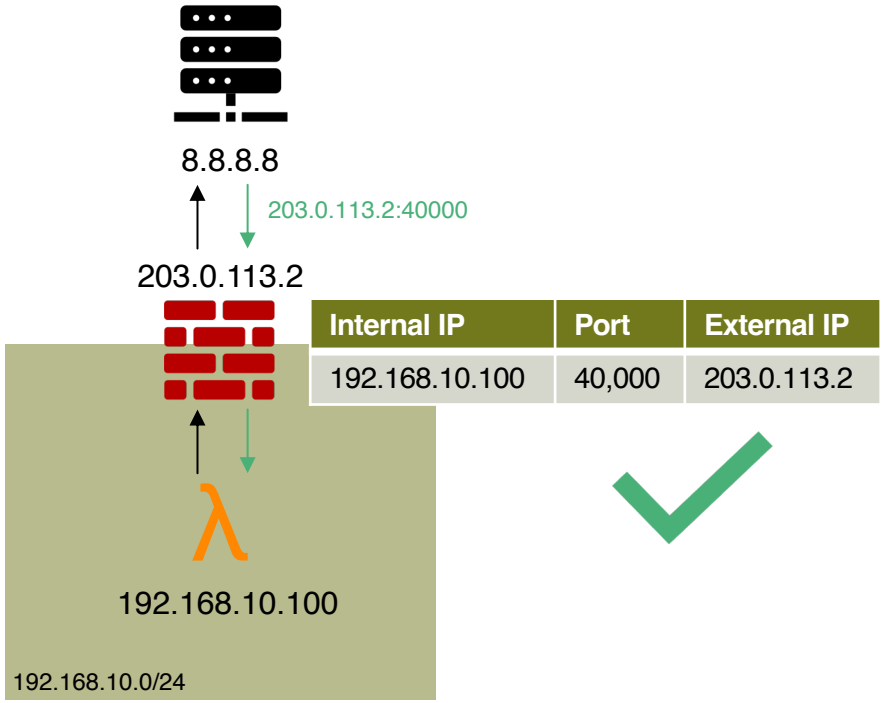
3 **Limitations of Today's Serverless Computing Platforms**
3.1 Inadequate storage for fine-grained operations . . . . . .
3.2 Lack of fine-grained coordination . . . . . .
3.3 Poor performance for standard communication patterns . .
3.4 Predictable Performance . . . . . .

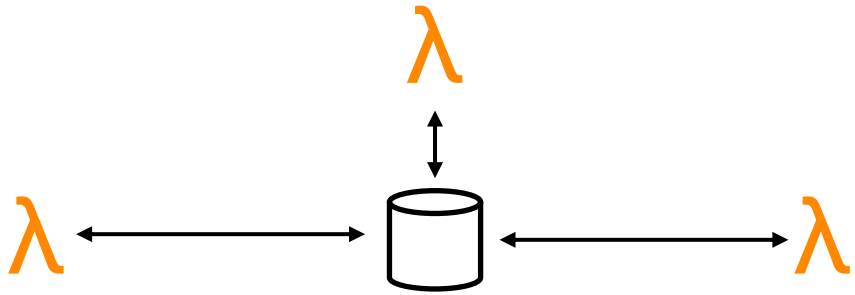


FaaS Communication



Network Address Translation



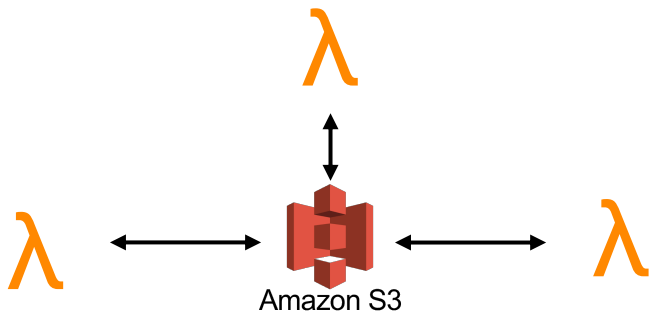


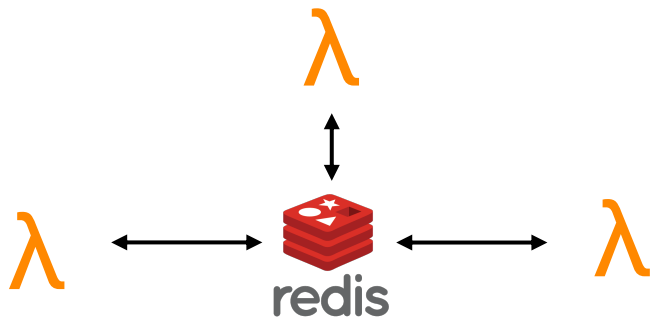

Amazon S3

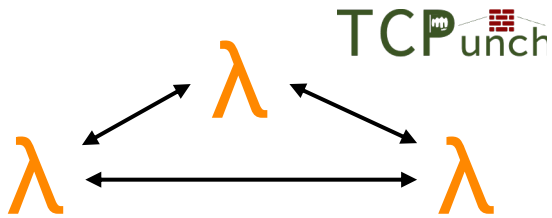

Amazon DynamoDB

...

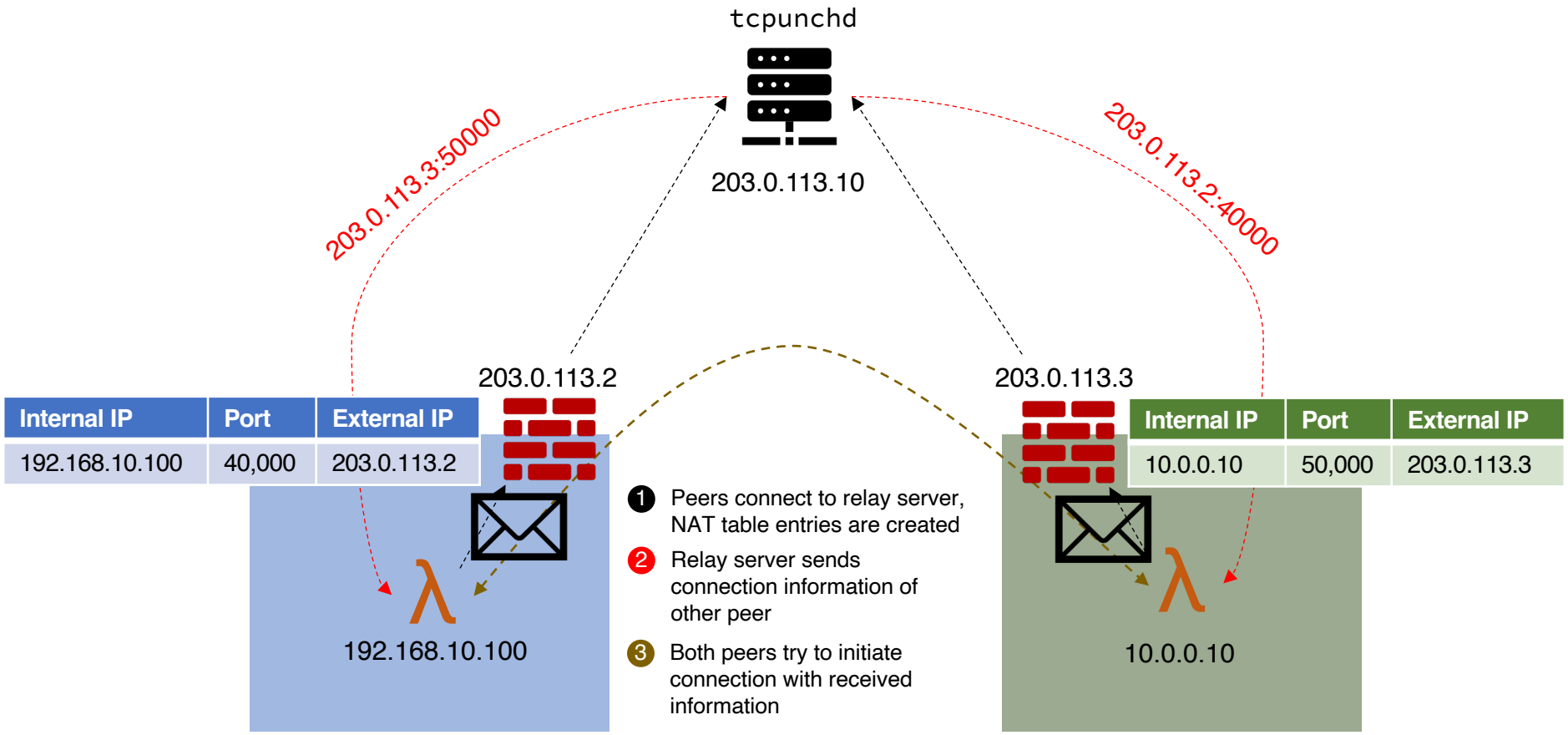
Analyzed Communication Channels





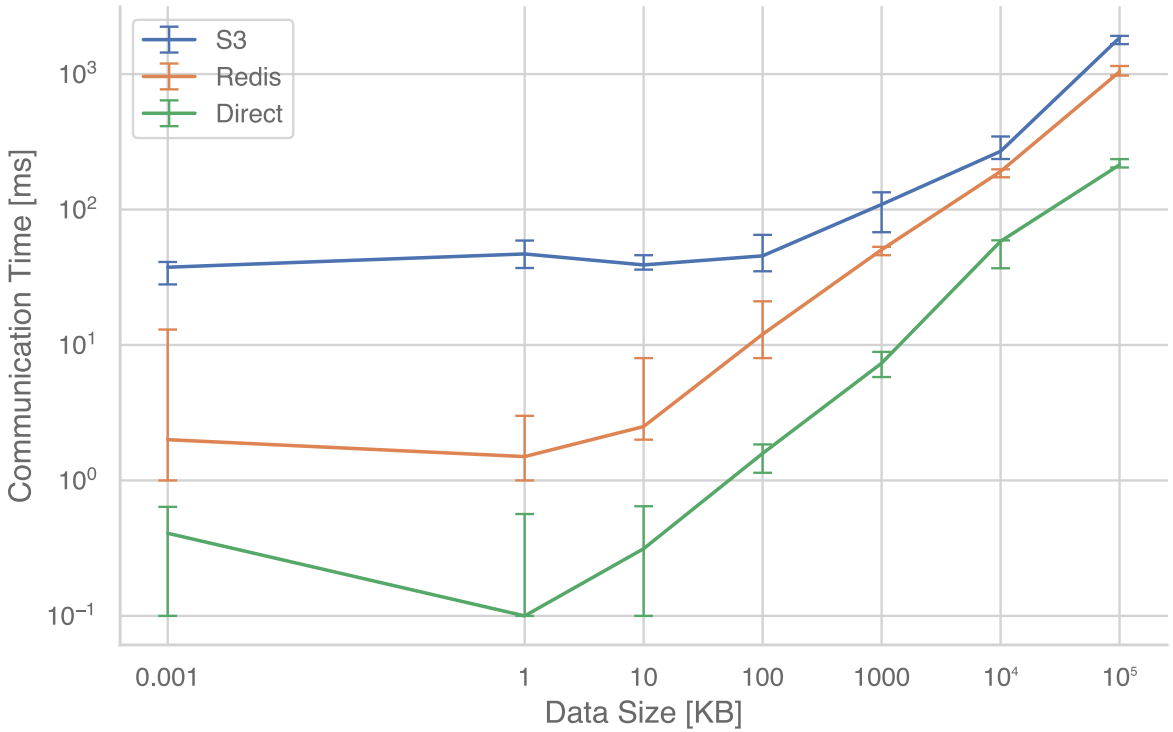


TCP NAT Hole Punching

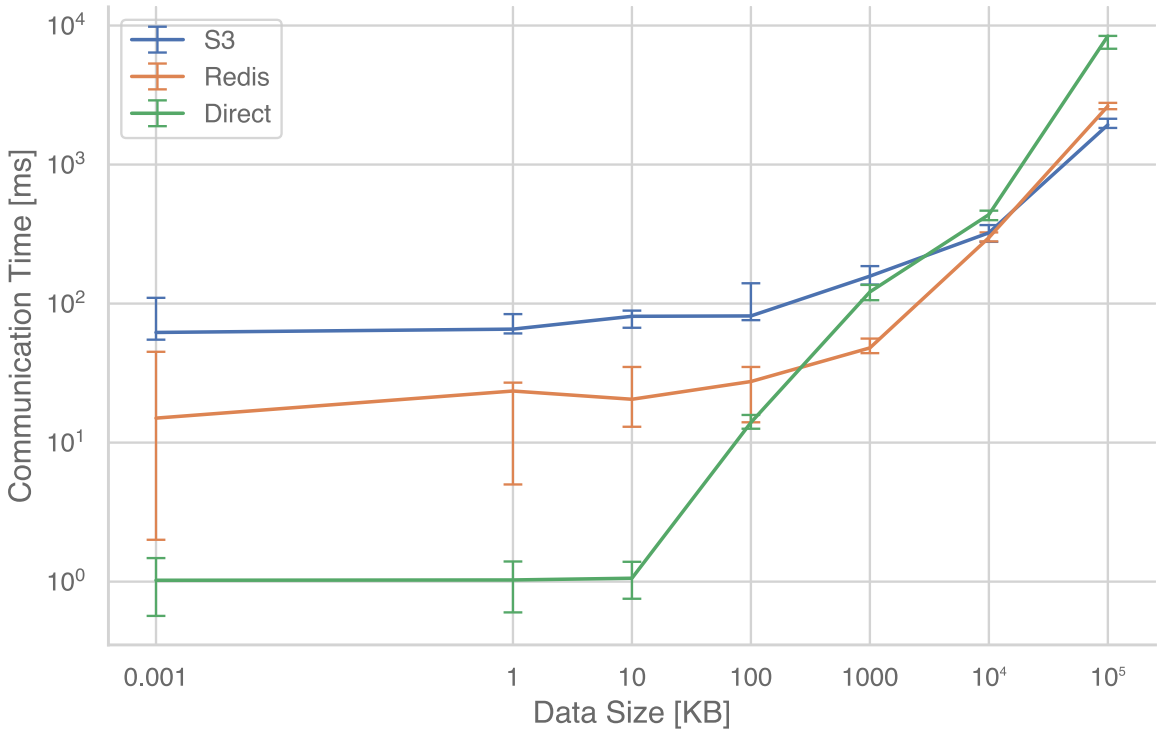


Communication Channel Benchmarks

Peer-to-Peer

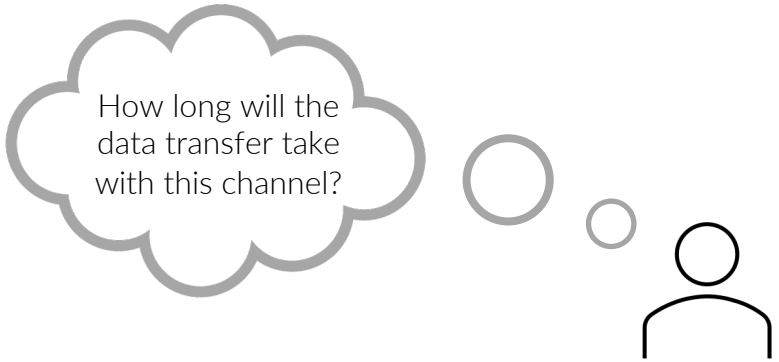


1 Producer, 8 Consumers



Platform: AWS Lambda functions with 2 GiB RAM

Performance and Cost Model



Bandwidth-Latency model with channel-specific bandwidth function based on number of clients

Experiment	Communication Channel	R^2 Score
1 Producer 1 Consumer Varying Data Size	S3	0.97
	ElastiCache Redis	0.987
	Direct Communication	0.969

$$c(p_{faas}, p_{vm}, \dots) = c_{faas} + c_{channel}$$

$$c_{obj} = c_{up} + c_{down}$$

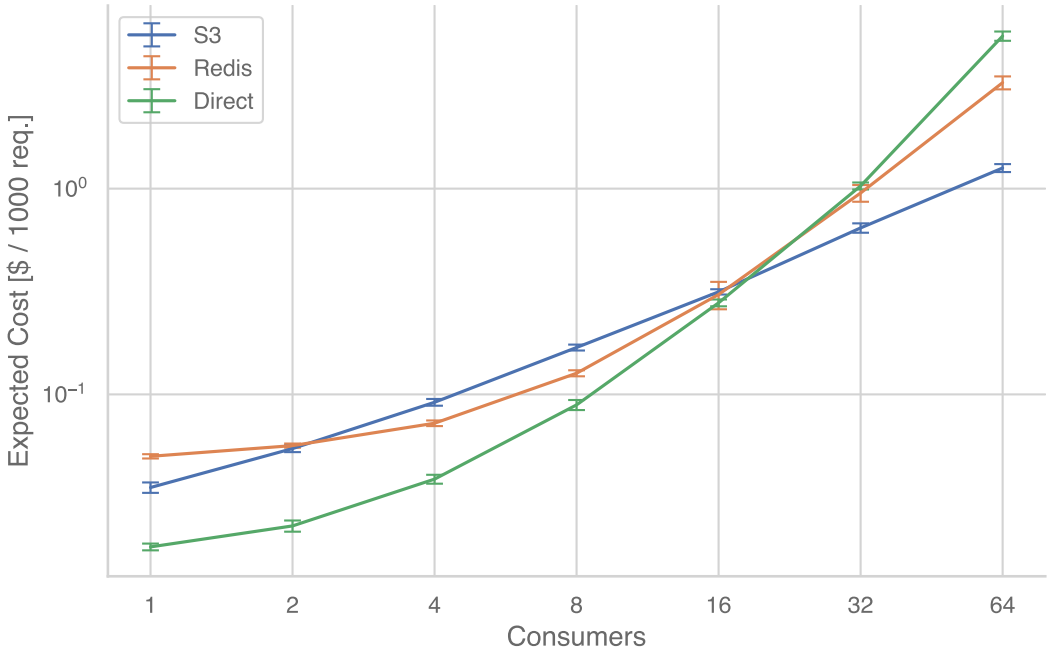
$$c_{mem} = c_{trans} + c_{infra}$$

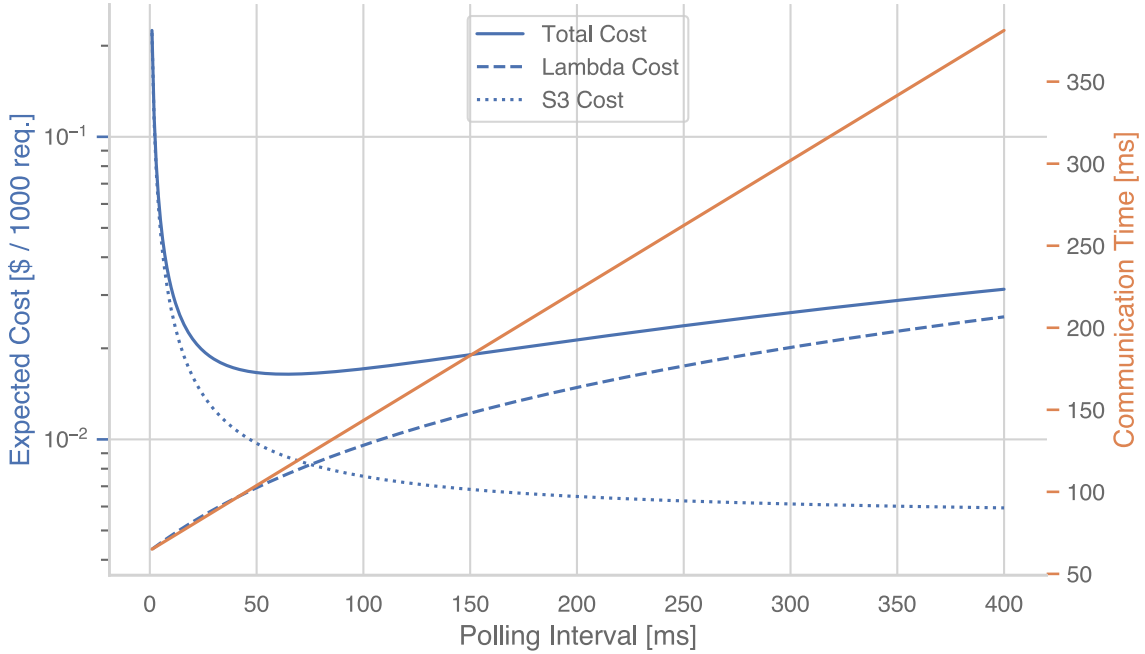
$$c_{direct} = c_{trans} + c_{infra}$$

Datapoint	Value	Granularity	Example
p_{faas}	0.000016667$	second	Lambda GiB-second
p_{vm}	0.0134$	hour	t2.micro EC2 instance
p_{mem}	0.038$	hour	cache.t3.small Redis node
p_d	0.00043$	1000 req.	S3 GET request
p_u	0.0054$	1000 req.	S3 PUT request
$p_{t.obj}, p_{t.mem}, p_{t.direct}$	0$	GB	Intraregional data transfers

Cost Model Computations

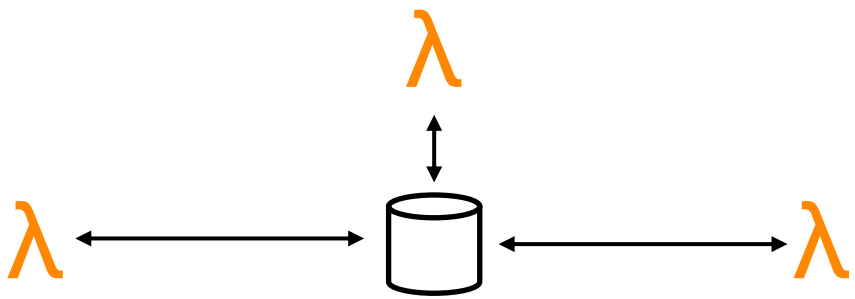
Cost when varying consumers

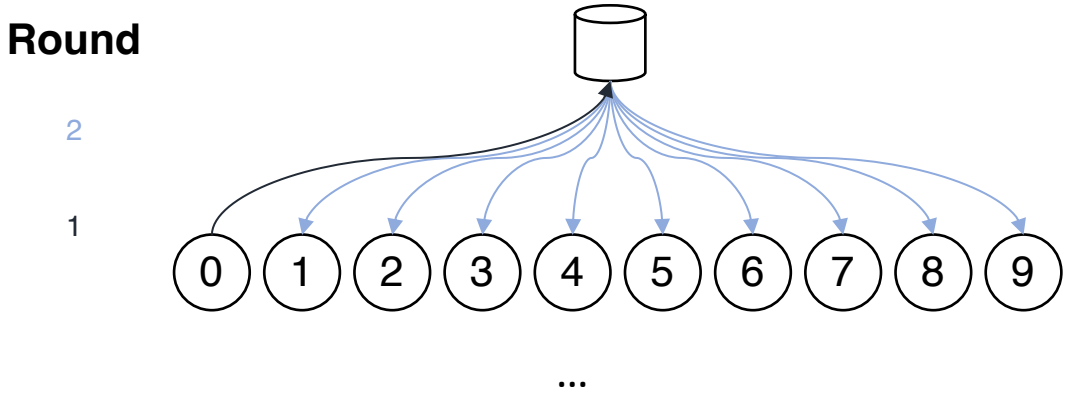


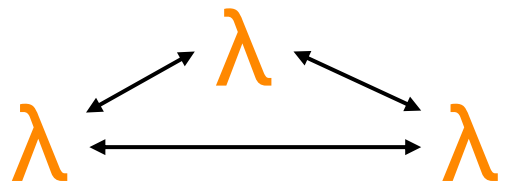


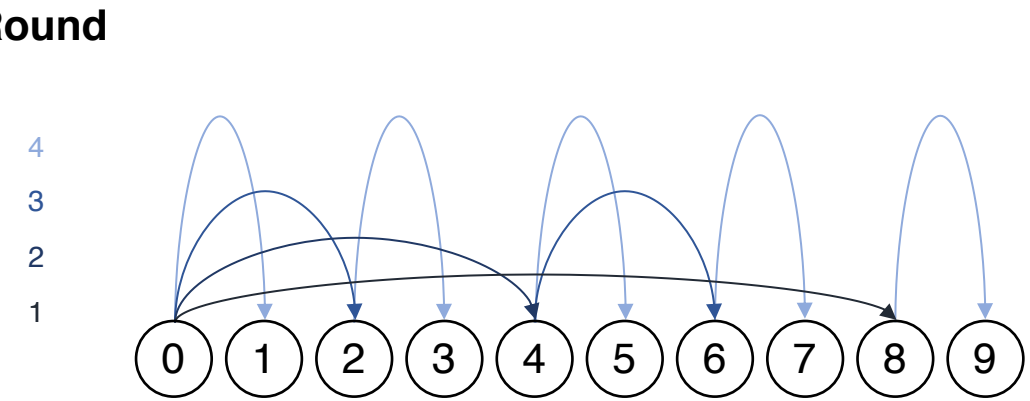
Simulated data size: 10 MB

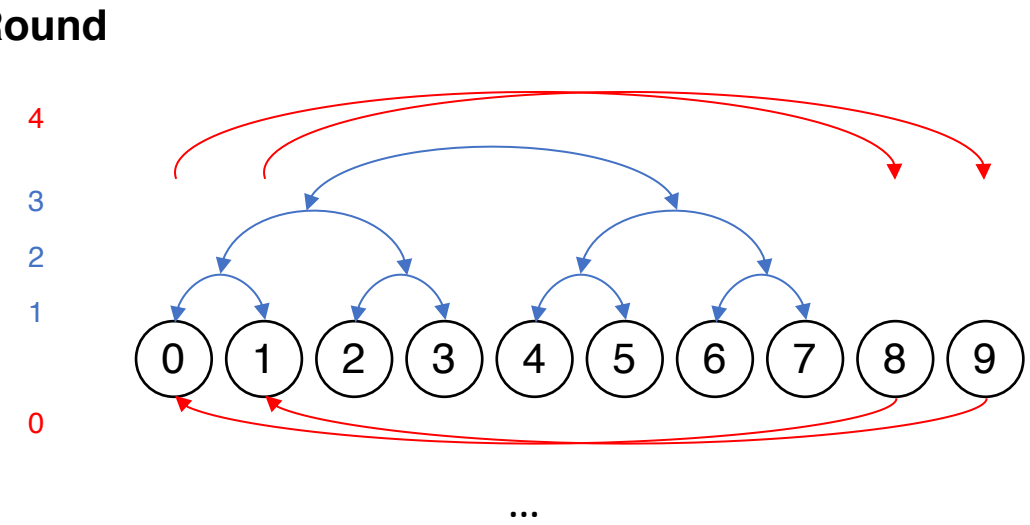
Design / Modeling: Collectives



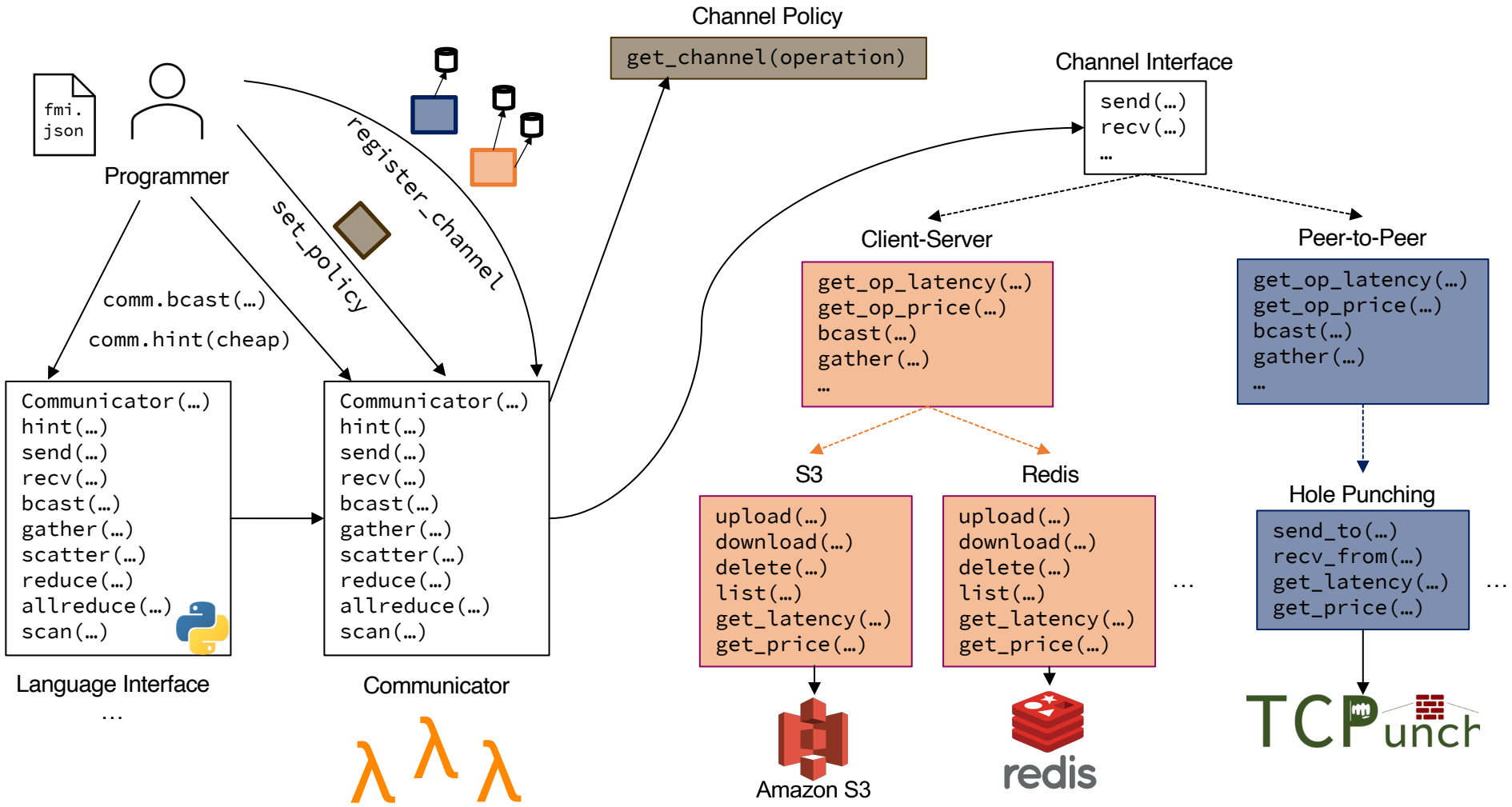








FMI <λ>



fmi.opencore.ch

```
#include <fmi.h>
...
FMI::Communicator comm(

FMI::Comm::Data<double>
if (peer_id == 0) {
    d = 42.0;
}
comm.bcast(d, 0);
assert(d == 42.0);
FMI::Comm::Data<std::ve
FMI::Comm::Data<std::ve
comm.scatter(vec, recv,
assert(recv.get()[0] ==
FMI::Utils::Function<in
  return a + b; }, true
FMI::Comm::Data<int> id
FMI::Comm::Data<int> id
comm.allreduce(id, id_s
assert(id_sum == 3);
```

Installation / Compilation

C++

To facilitate compiling, there is a Docker image with all necessary dependencies:

```
docker pull ghcr.io/opencorech/fmi-build-cpp:latest
```

Python

AWS Lambda Layer

The simplest way to use the library are the AWS Lambda layers, available under the following ARNs:

- `arn:aws:lambda:eu-central-1:386971375191:layer:fmi-python36:1`
- `arn:aws:lambda:eu-central-1:386971375191:layer:fmi-python37:1`
- `arn:aws:lambda:eu-central-1:386971375191:layer:fmi-python38:1`
- `arn:aws:lambda:eu-central-1:386971375191:layer:fmi-python39:1`

If you are using CloudFormation, you simply have to reference the ARN with `Layers:`, a minimal example for Python 3.9 looks like this:

```yaml
AWSTemplateFormatVersion: '2010-09-09'
Transform: 'AWS::Serverless-2016-10-31'

Resources:
  lambdaml:
    Type: AWS::Serverless::Function
    Properties:
      FunctionName: fmi-example
      Runtime: python3.9
      Layers:
        - arn:aws:lambda:eu-central-1:386971375191:layer:fmi-python39:1
```

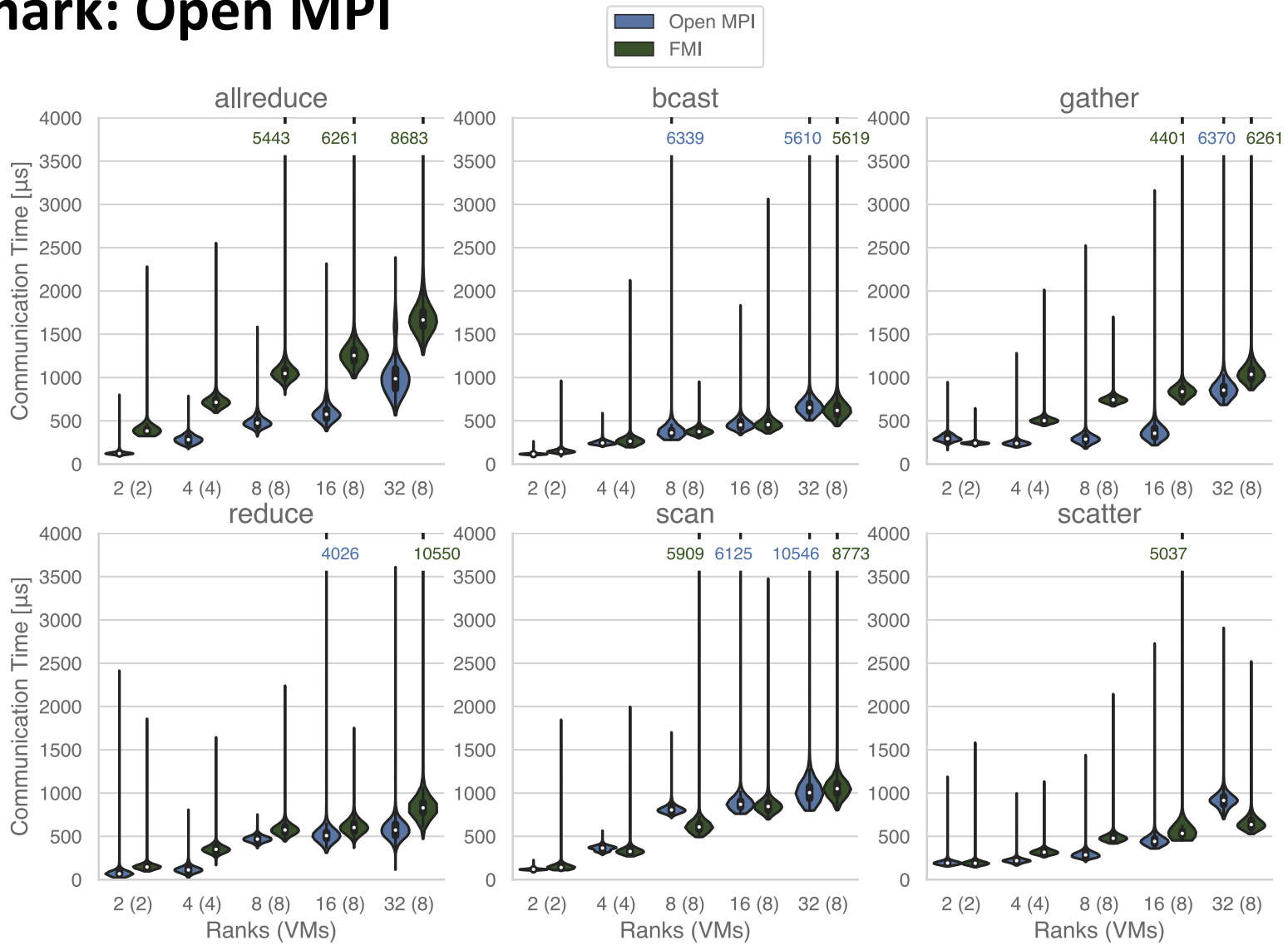
When using the AWS Management Console, you can add it under `Layers` -> `Add a layer` -> `Specify an ARN`:

Choose a layer

```
,
name)

pes.int))

types.int))

list, 3))

fmi.op.sum),
types.int))

ambda a, b:
,
nt))
```

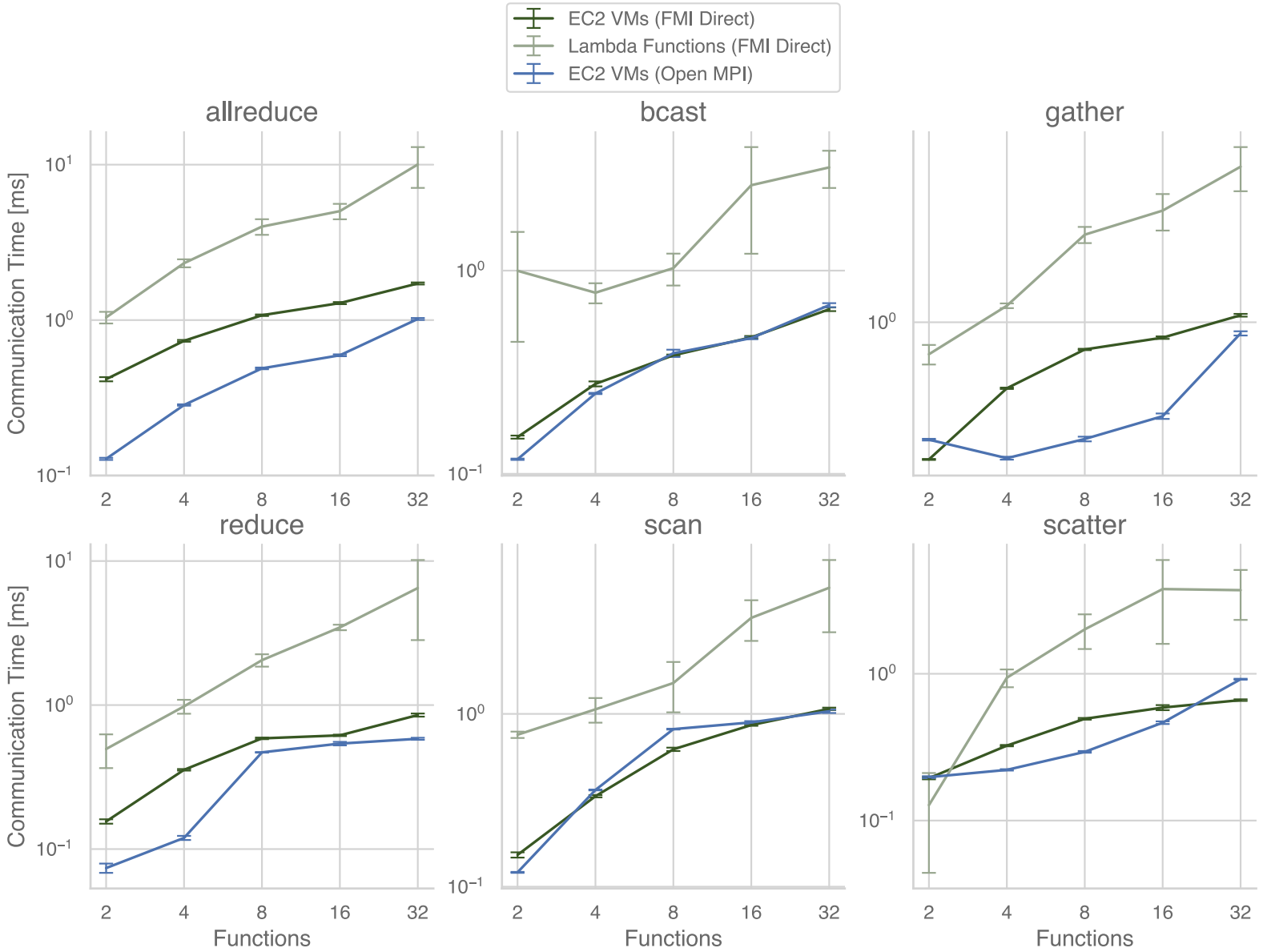
fmi.opencore.ch

IaaS Benchmark: Open MPI

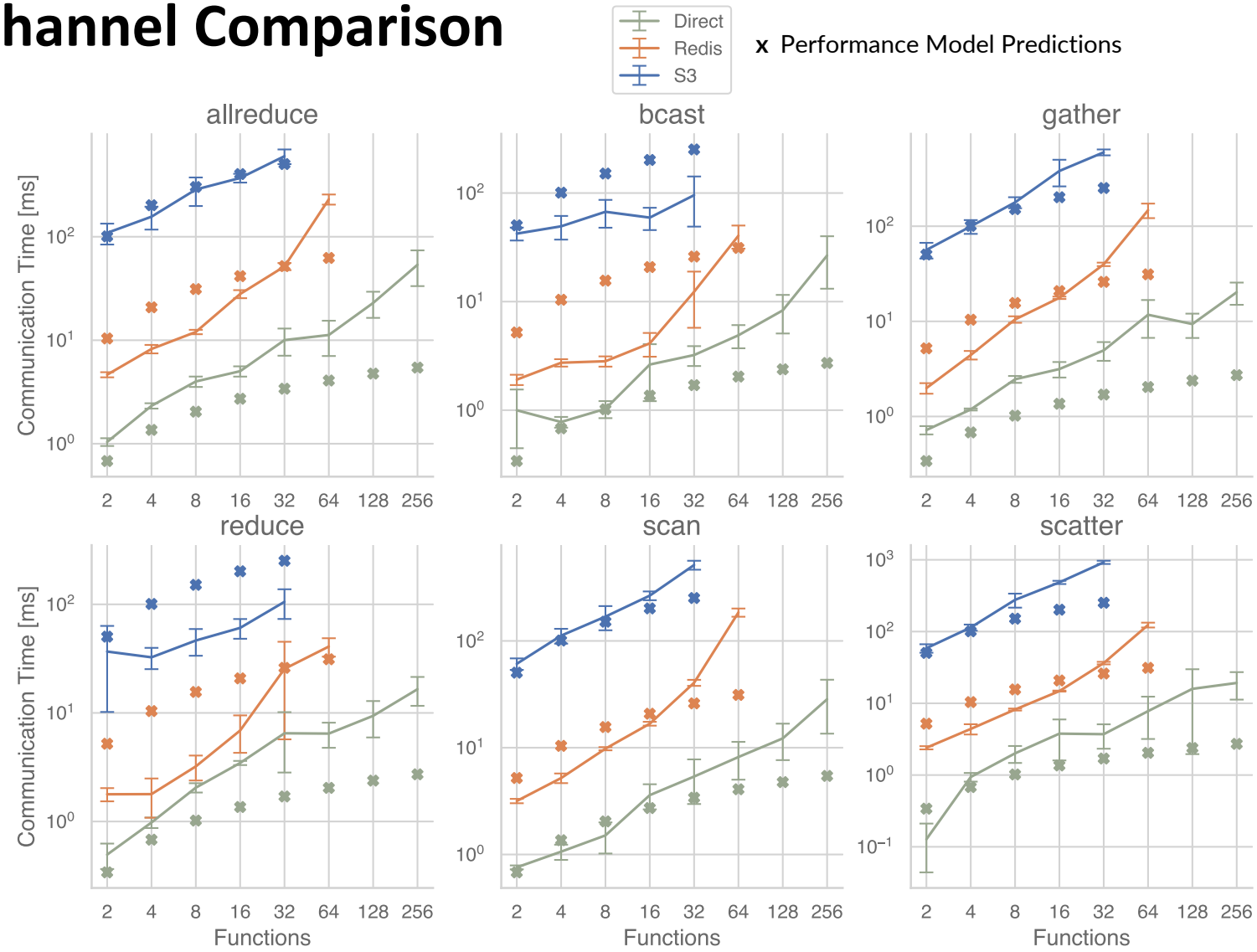


Environment: AWS EC2 t2.xlarge (16 GiB RAM, 4 vCPUs) VMs, 2 / 4 ranks per VM for 16 / 32 ranks.
Data Sizes: 1 integer for allreduce, bcast, reduce, and scan. 5,000 integers (20 KB) for gather / scatter

Case Study: IaaS vs. FaaS Collectives



Communication Channel Comparison

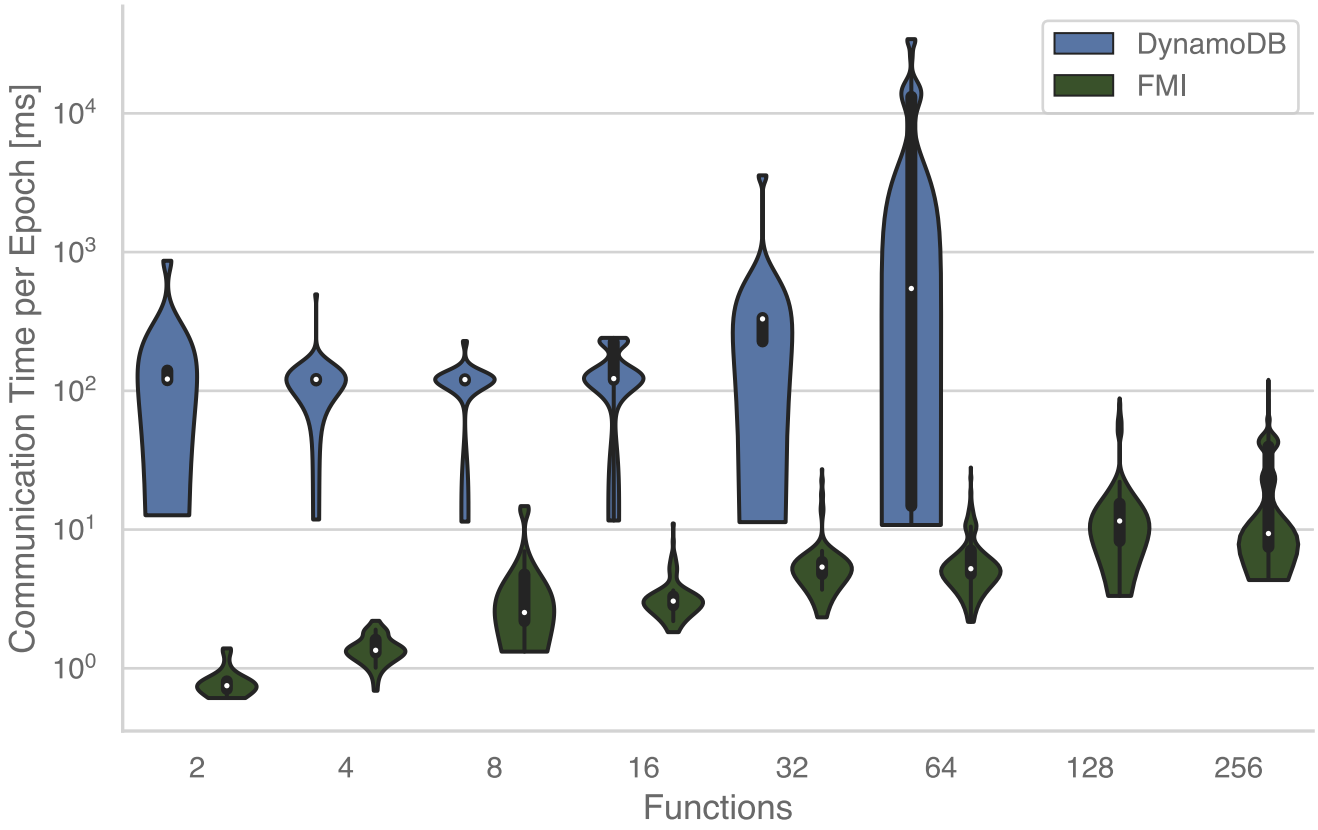


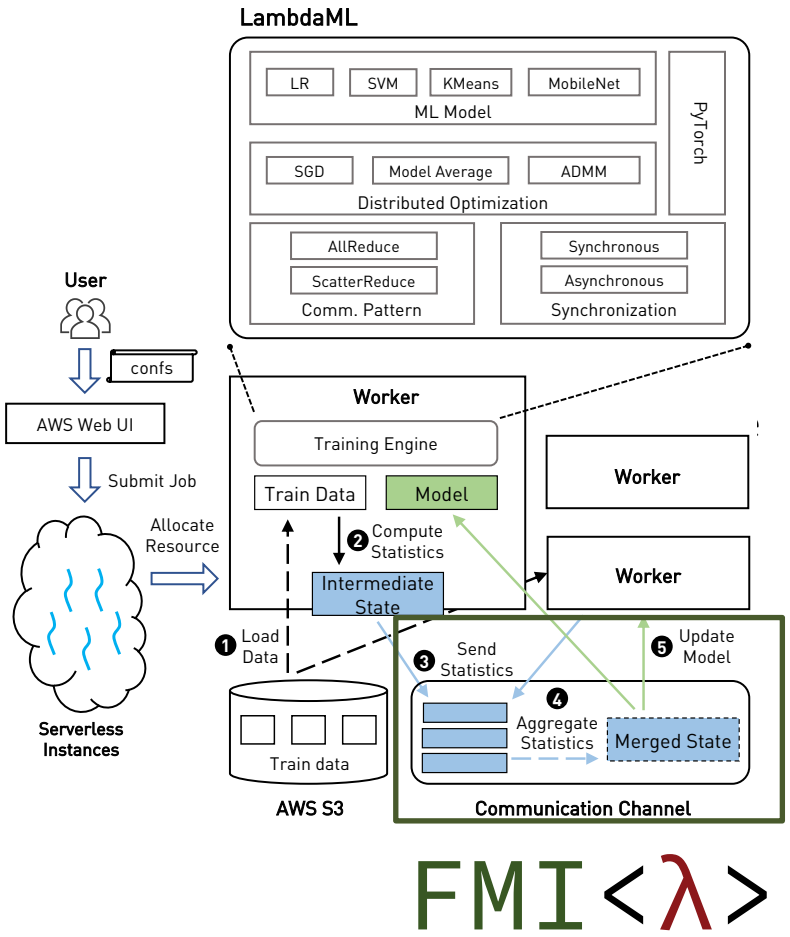
Environment: AWS Lambda functions with 1 GiB RAM, t2.micro (1 GiB RAM, 1 vCPU) hole punching server, cache.t2.small (1.55 GiB RAM, 1 vCPU) Redis instance
Data Sizes: 1 integer for allreduce, bcast, reduce, and scan. 5,000 integers (20 KB) for gather / scatter

FaaS Benchmark: LambdaML

SIGMOD 21, Jiang et al.:
Towards Demystifying Serverless Machine Learning Training

Unmodified LambdaML only runs up to 64 functions
FMI can scale to 256 functions with the smallest hole punching server





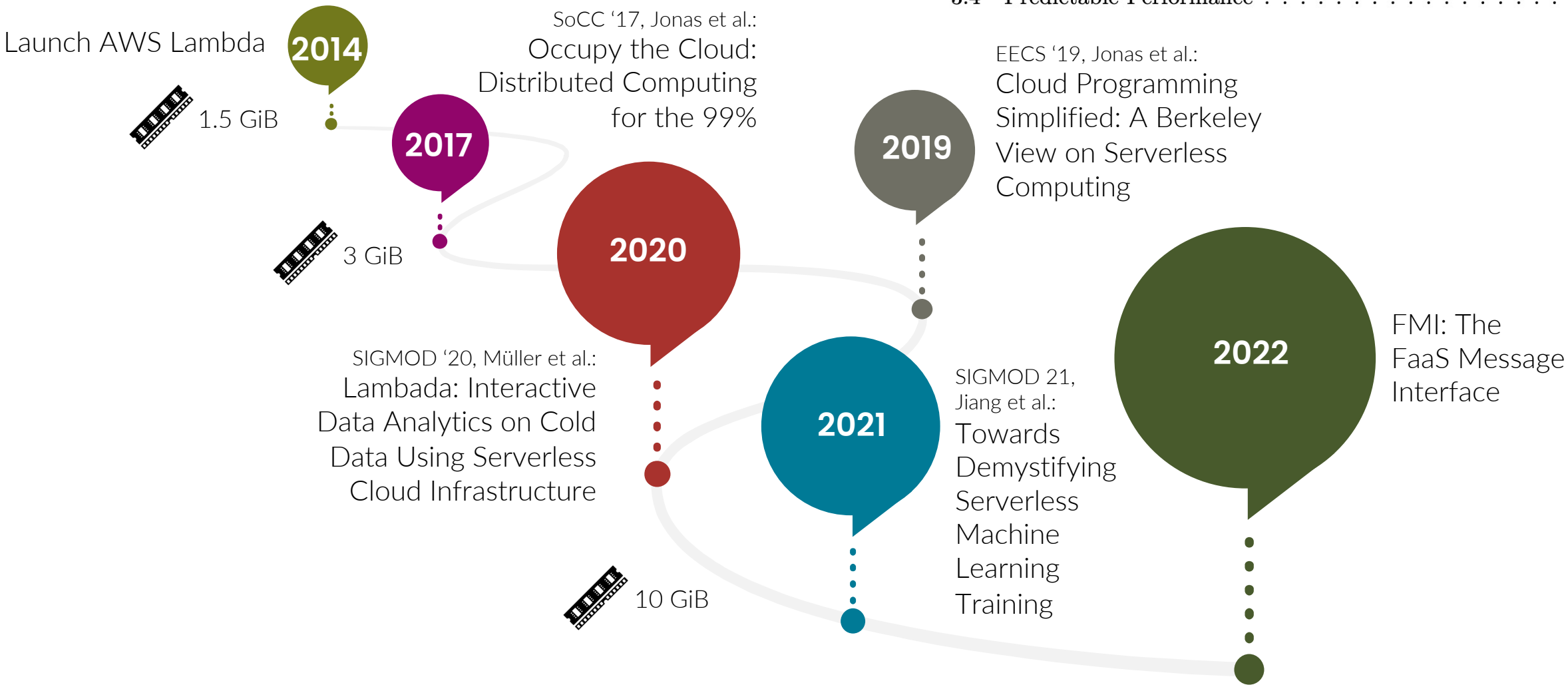
FMI <λ>

Environment: AWS Lambda functions with 1 GiB RAM, t2.micro (1 GiB RAM, 1 vCPU) hole punching server.
Code: Unmodified LambdaML for DynamoDB, only 3 lines changed for FMI.
Model: Distributed K-Means with 3 centroids and 28 dimensions. One allreduce (sum) per epoch.

FaaS: Evolution

3 Limitations of Today's Serverless Computing Platforms

3.1 Inadequate storage for fine-grained operations . . . . . .
3.2 Lack of fine-grained coordination . . . . . .
3.3 Poor performance for standard communication patterns . . .
3.4 Predictable Performance . . . . . .



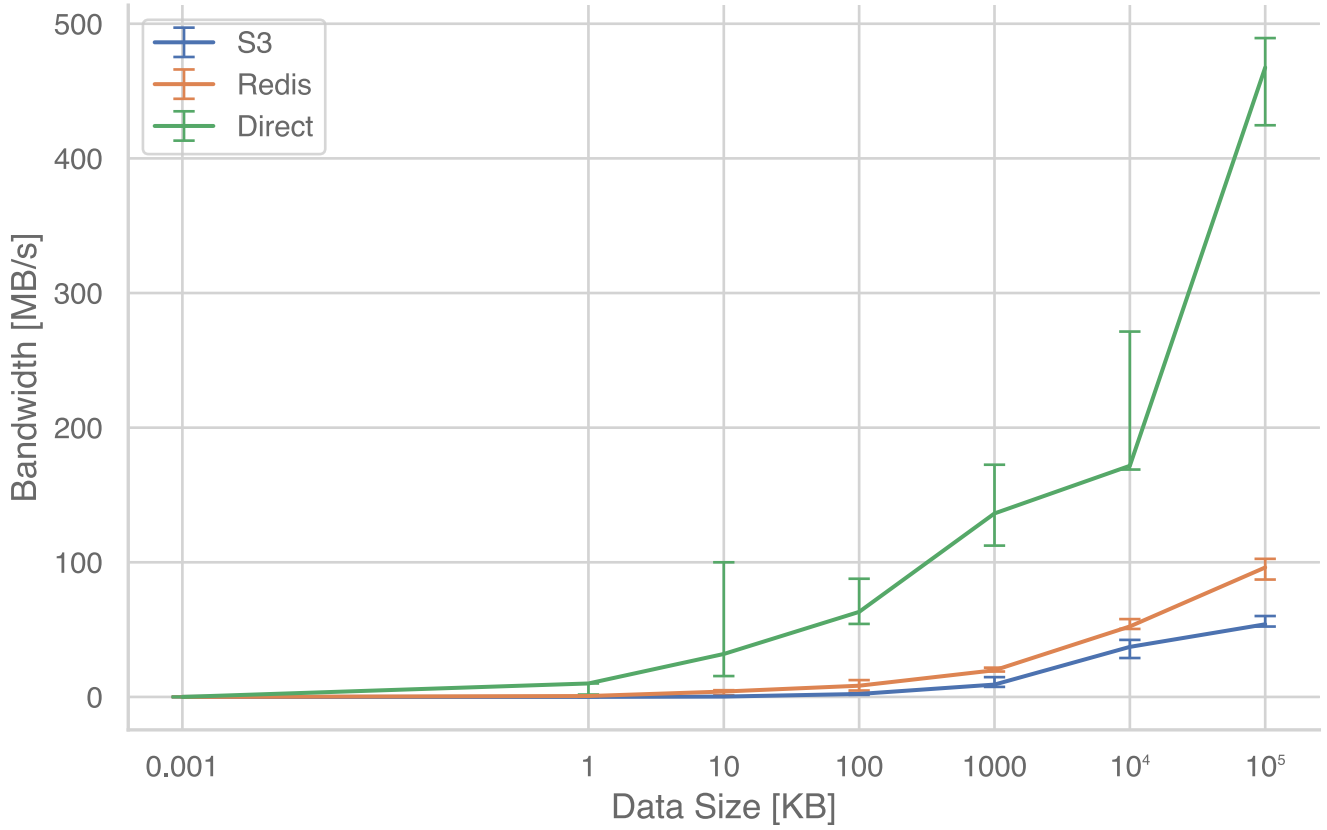
FMI: The FaaS Message Interface

FMI <λ>

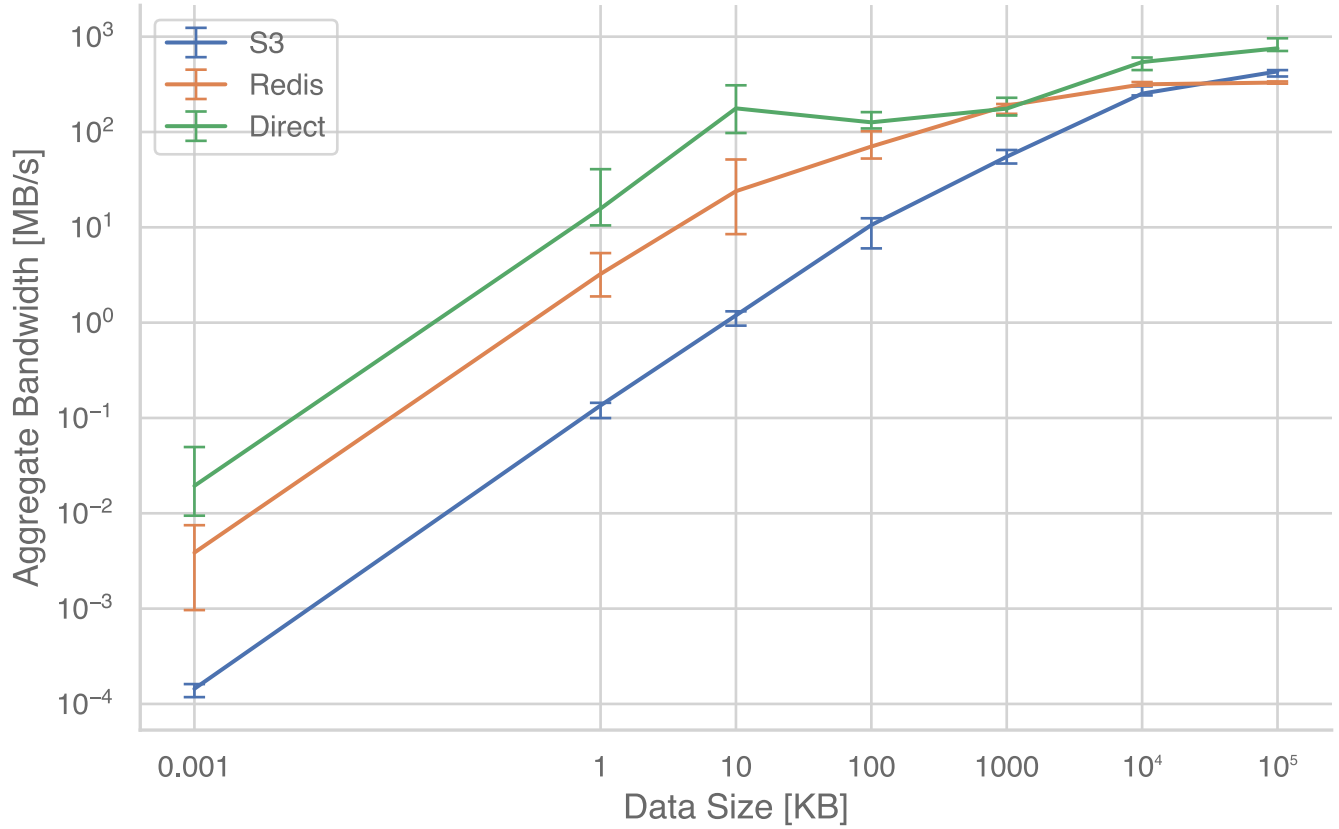
ROMAN BÖHRINGER

Backup Slides

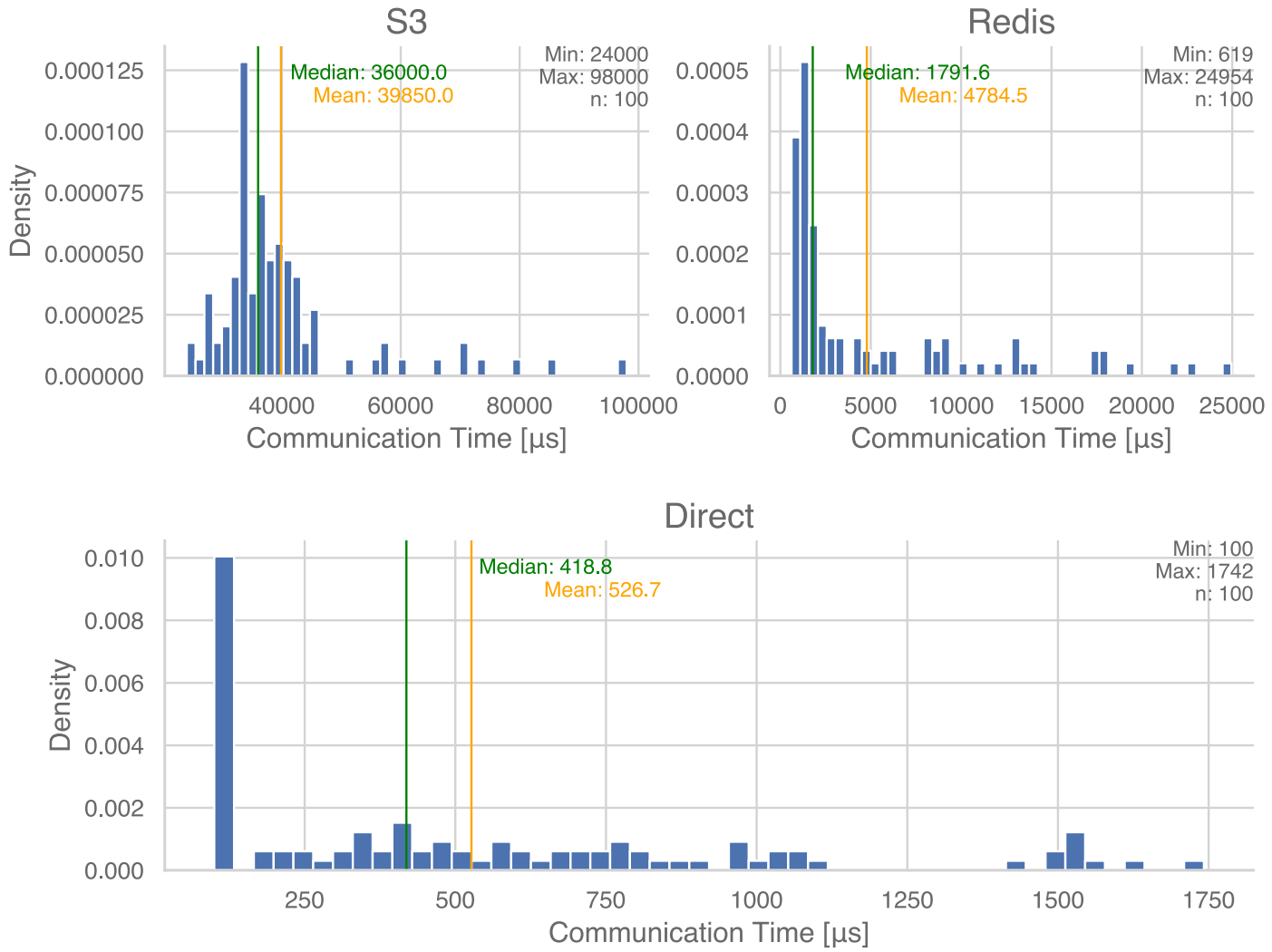
Microbenchmarks: Peer-to-Peer Bandwidth



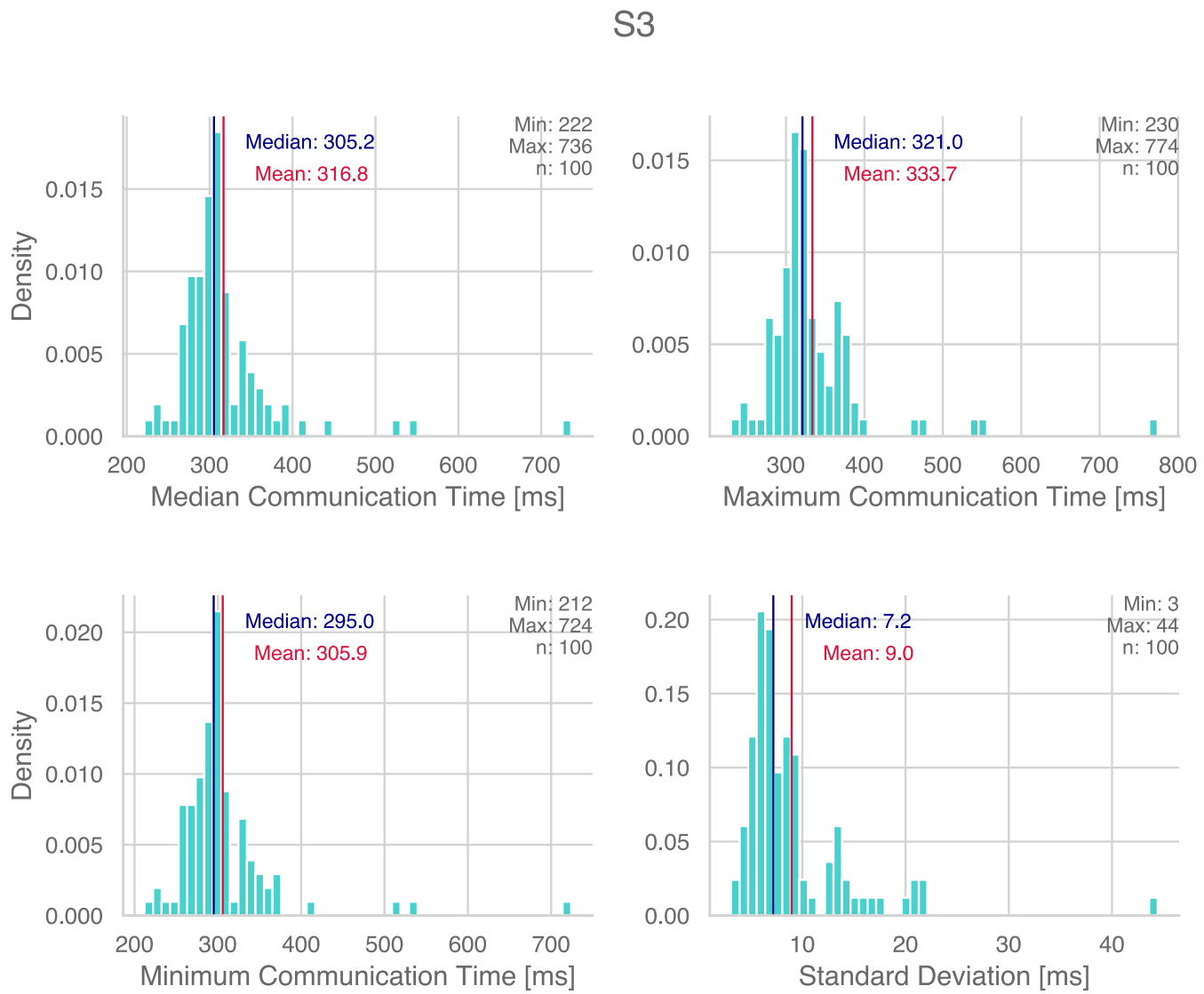
Microbenchmarks: 1 Producer, 8 Consumers Bandwidth



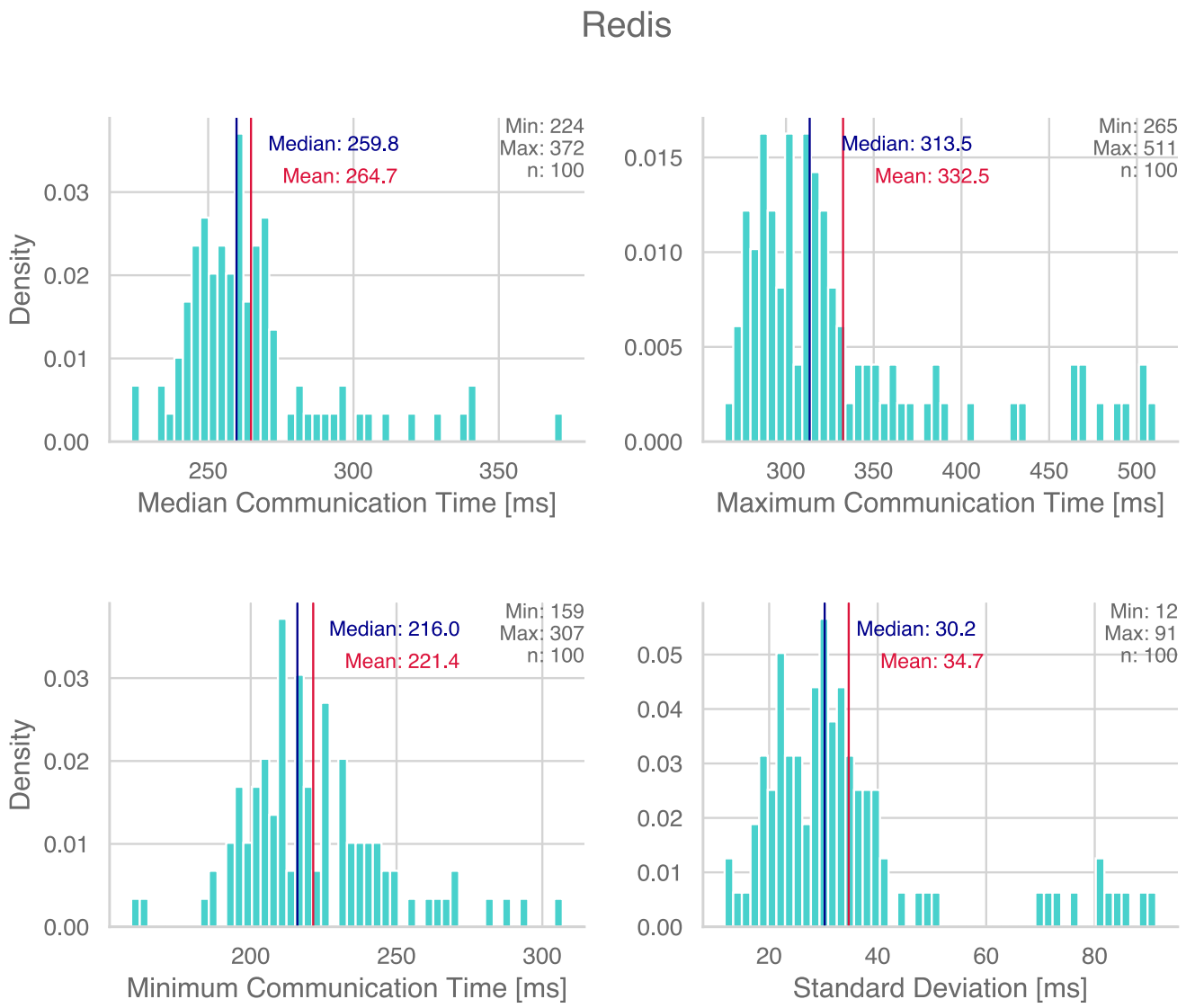
Microbenchmarks: Peer-to-Peer Latency Variance



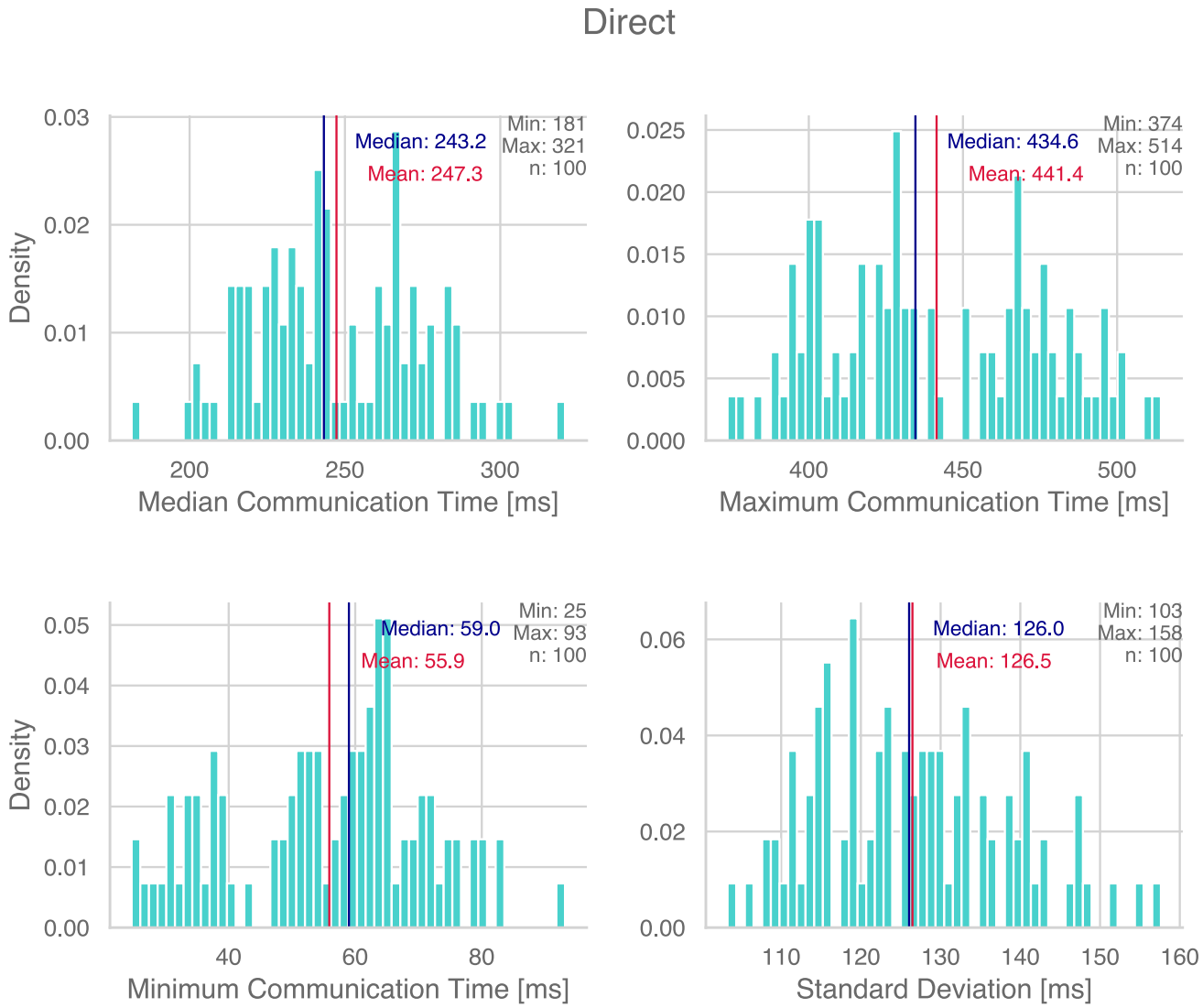
Microbenchmarks: 1 Producer, 8 Consumers Variance (100 MB)



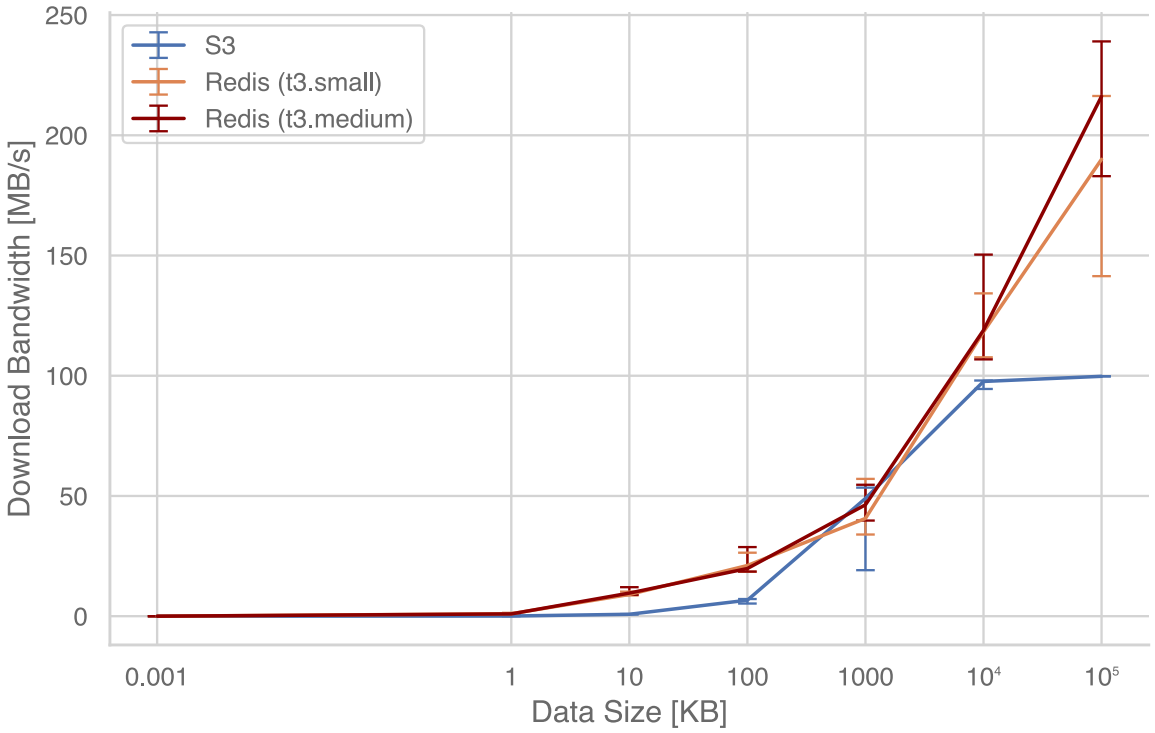
Microbenchmarks: 1 Producer, 8 Consumers Variance (100 MB)

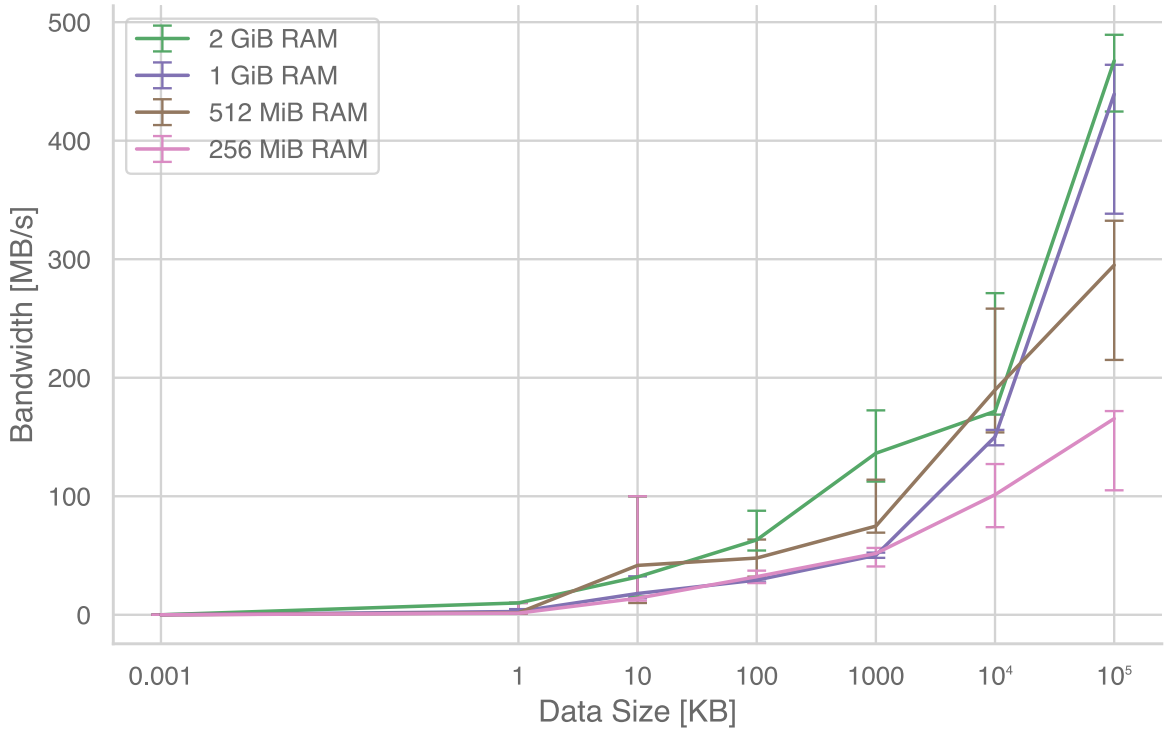


Microbenchmarks: 1 Producer, 8 Consumers Variance (100 MB)



Microbenchmarks: Impact of System Parameters





Cost Model: Impact of System Parameters