Is It Time To Put Cold Starts In The Deep Freeze?

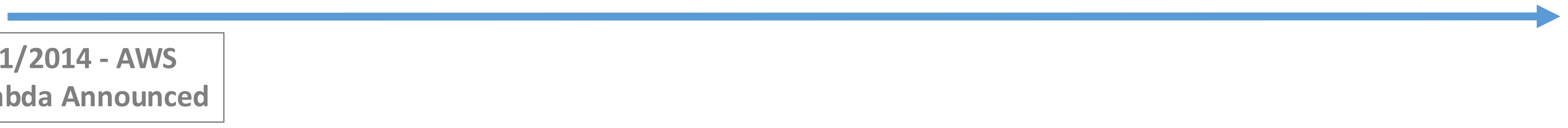
Carlos Segarra, Ivan Durev, Peter Pietzuch

Imperial College London

https://carlossegarra.com
<cs1620@ic.ac.uk>

SoCC 2024 – Redmond, WA

Cold Starts Have Dominated Serverless Research



Cold Starts Have Dominated Serverless Research

[ATC'18] Peeking Behind

[SoCC'17] PyWren

[CIDR'19] One Step Forward Two Steps Back

[ATC'18] SOCK

[ATC'18] SAND

[HotCloud'16] OpenLambda

11/2014 - AWS Lambda Announced

2018 – Cold-Start Survey Released

Cold Starts Have Dominated Serverless Research

[ATC'20] Serverless In the Wild

[ATC'18] Peeking Behind

[EuroSys'20] SEUSS

[CIDR'19] One Step Forward Two Steps Back

[SoCC'17] PyWren

[ATC'18] SOCK

[ASPLOS'20] Catalyzer

[ATC'18] SAND

[ATC'20] Faasm

[HotCloud'16] OpenLambda

[NSDI'20] Firecracker

11/2014 - AWS Lambda Announced

2018 – Cold-Start Survey Released

2020 - Azure Traces Released

Cold Starts Have Dominated Serverless Research

[ATC'20] Serverless In the Wild

[ATC'18] Peeking Behind

[EuroSys'20] SEUSS

[CIDR'19] One Step Forward Two Steps Back

[SoCC'17] PyWren

[ATC'18] SOCK

[ASPLOS'20] Catalyzer

[ASPLOS'22] IceBreaker

[ATC'18] SAND

[ATC'20] Faasm

[ATC'21] FaaSNet

[HotCloud'16] OpenLambda

[NSDI'20] Firecracker

11/2014 - AWS Lambda Announced

2018 – Cold-Start Survey Released

2020 - Azure Traces Released

2021 – Alibaba Traces Released

Cold Starts Have Dominated Serverless Research

[HotCloud'16] OpenLambda

[SoCC'17] PyWren

[ATC'18] Peeking Behind

[ATC'18] SOCK

[ATC'18] SAND

[CIDR'19] One Step Forward Two Steps Back

[ATC'20] Serverless In the Wild

[EuroSys'20] SEUSS

[ASPLOS'20] Catalyzer

[ATC'20] Faasm

[NSDI'20] Firecracker

[ATC'21] FaaSNet

[ASPLOS'22] IceBreaker

[SoCC'23] How Does It Function?

[OSDI'24] Sabre

[SoCC'24] Snapipilenie

[ASPLOS'24] RainbowCake

11/2014 - AWS Lambda Announced

2018 – Cold-Start Survey Released

2020 - Azure Traces Released

2021 – Alibaba Traces Released

2023 – Huawei Traces Released

Cold Starts Have Dominated Serverless Research

[ATC'20] Serverless In the Wild

[OSDI'24] Sabre

[ATC'18] Peeking Behind

[EuroSys'20] SEUSS

[SoCC'24] Snapipilenie

[CIDR'19] One Step Forward Two Steps Back

[SoCC'23] How Does It Function?

[SoCC'17] PyWren

[ATC'18] SOCK

[ASPLOS'20] Catalyzer

[ASPLOS'22] IceBreaker

[ATC'18] SAND

[ATC'20] Faasm

[ATC'21] FaaSNet

[ASPLOS'24] RainbowCake

[EuroSys'25] Serverless Cold Starts And Where to Find Them

[HotCloud'16] OpenLambda

[NSDI'20] Firecracker

11/2014 - AWS Lambda Announced

2018 – Cold-Start Survey Released

2020 - Azure Traces Released

2021 – Alibaba Traces Released

2023 – Huawei Traces Released

2025 – Huawei Traces Released

Cold Starts Have Dominated Serverless Research

[ATC'20] Serverless In the Wild

[ATC'18] Peeking

[OSDI'24] Sabre

pipilenie

lt

[SoCC'17]

Are cold starts that important?

[ATC'18] SAND

[ATC'20] Faasm

[ATC'21] FaaSNet

[ASPLOS'24] RainbowCake

[HotCloud'16] OpenLambda

[NSDI'20] Firecracker

[EuroSys'25] Serverless Cold Starts And Where to Find Them

11/2014 - AWS Lambda Announced

2018 – Cold-Start Survey Released

2020 - Azure Traces Released

2021 – Alibaba Traces Released

2023 – Huawei Traces Released

2025 – Huawei Traces Released

Why Do Cold Starts Matter?

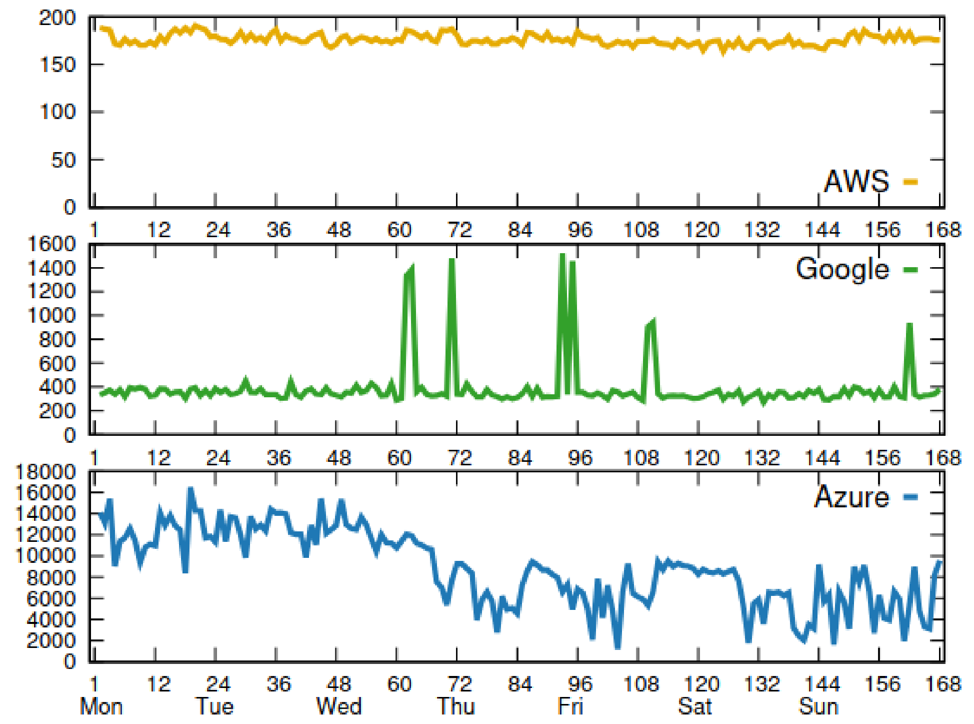


Figure 8: Coldstart latency (in ms) over 168 hours. All the measurements were started at right after midnight on a Sunday. Each data point is the median of all coldstart latencies collected in a given hour. For clarity, the y-axes use different ranges for each service.

[ATC'18] Peeking Behind the Curtains of Serverless Platforms

Why Do Cold Starts Matter?

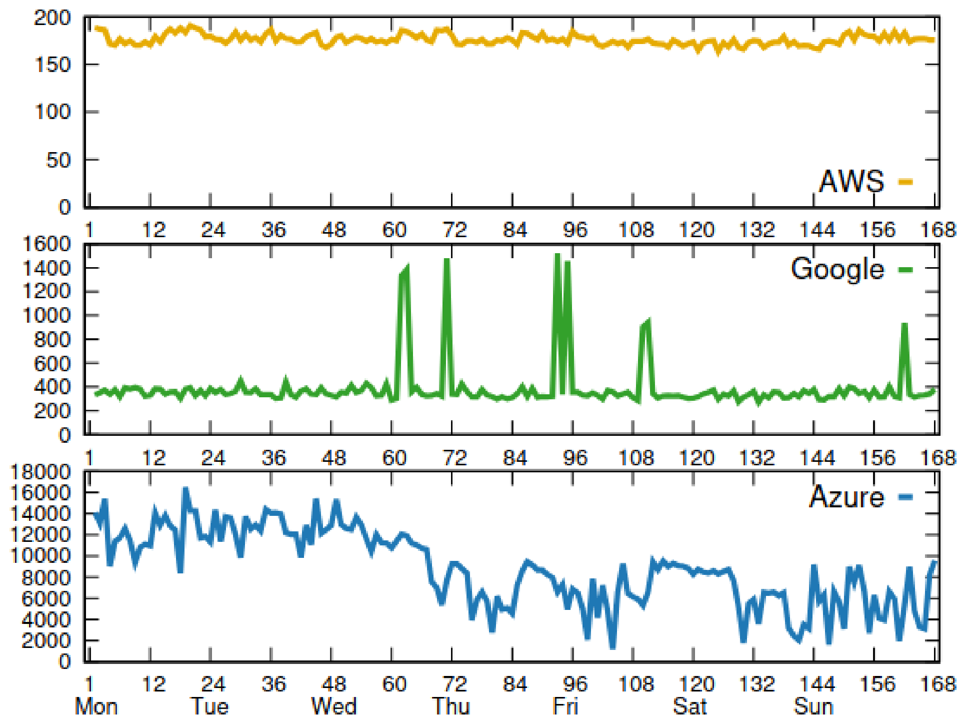


Figure 8: Coldstart latency (in ms) over 168 hours. All the measurements were started at right after midnight on a Sunday. Each data point is the median of all coldstart latencies collected in a given hour. For clarity, the y-axes use different ranges for each service.

Cold start times range between 100s of ms to 10s of secs

Why Do Cold Starts Matter?

Cold start times range between 100s of ms to 10s of secs

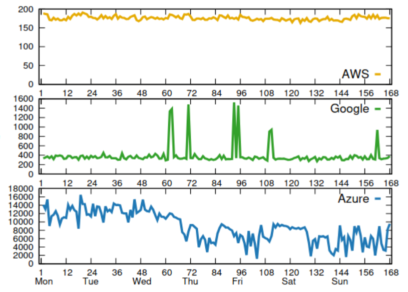


Figure 8: Coldstart latency (in ms) over 168 hours. All the measurements were started at right after midnight on a Sunday. Each data point is the median of all coldstart latencies collected in a given hour. For clarity, the y-axes use different ranges for each service.

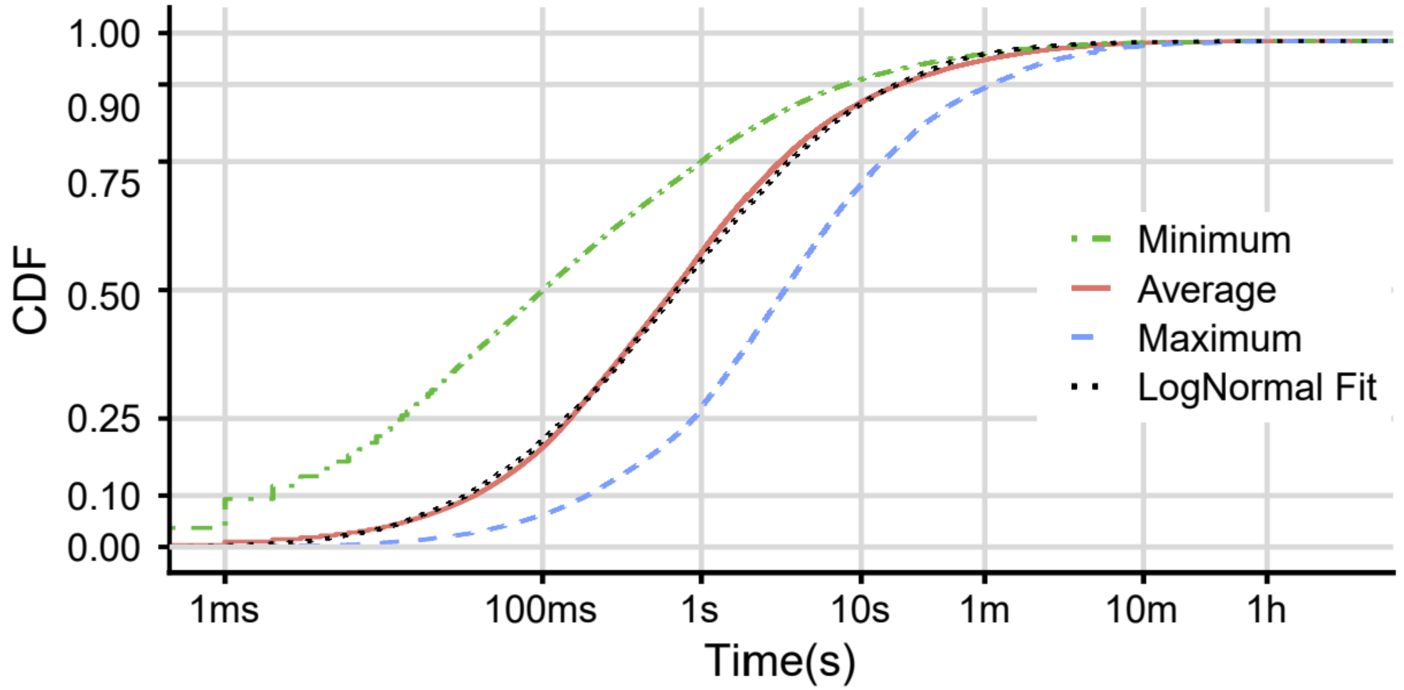


Figure 7: Distribution of function execution times. Min, avg, and max are separate CDFs, and use independent sorting.

[ATC'20] Serverless in the Wild: Characterizing and Optimizing the Serverless Workload at a Large Cloud Provider

Why Do Cold Starts Matter?

Cold start times range between 100s of ms to 10s of secs

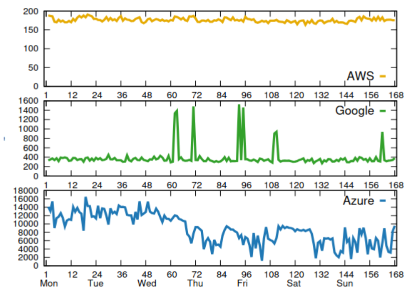


Figure 8: Coldstart latency (in ms) over 168 hours. All the measurements were started at right after midnight on a Sunday. Each data point is the median of all coldstart latencies collected in a given hour. For clarity, the y-axes use different ranges for each service.

Functions are short-lived! (50/90% shorter than 1/10s)

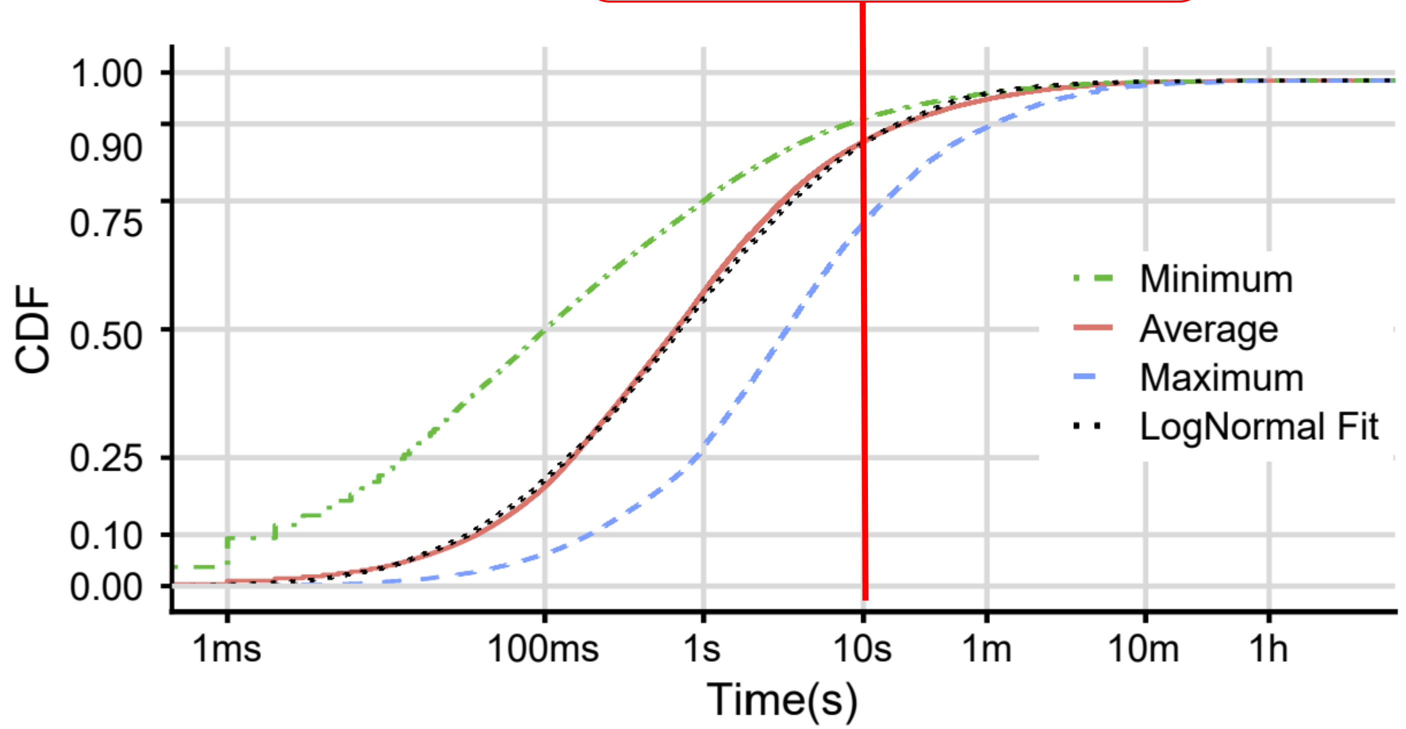


Figure 7: Distribution of function execution times. Min, avg, and max are separate CDFs, and use independent sorting.

[ATC'20] Serverless in the Wild: Characterizing and Optimizing the Serverless Workload at a Large Cloud Provider

Why Do Cold Starts Matter?

Cold start times range between 100s of ms to 10s of secs

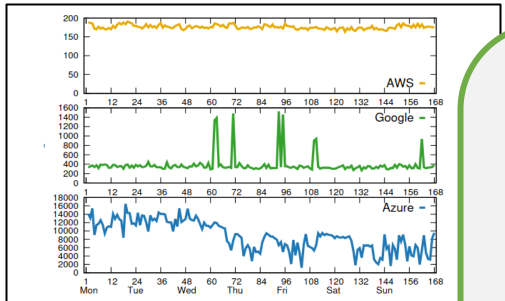


Figure 8: Coldstart latency (in ms) over 168 hours. All the measurements were started at right after midnight on a Sunday. Each data point is the median of all coldstart latencies collected in a given hour. For clarity, the y-axes use different ranges for each service.

Functions are short-lived! (50/90% shorter than 1/10s)

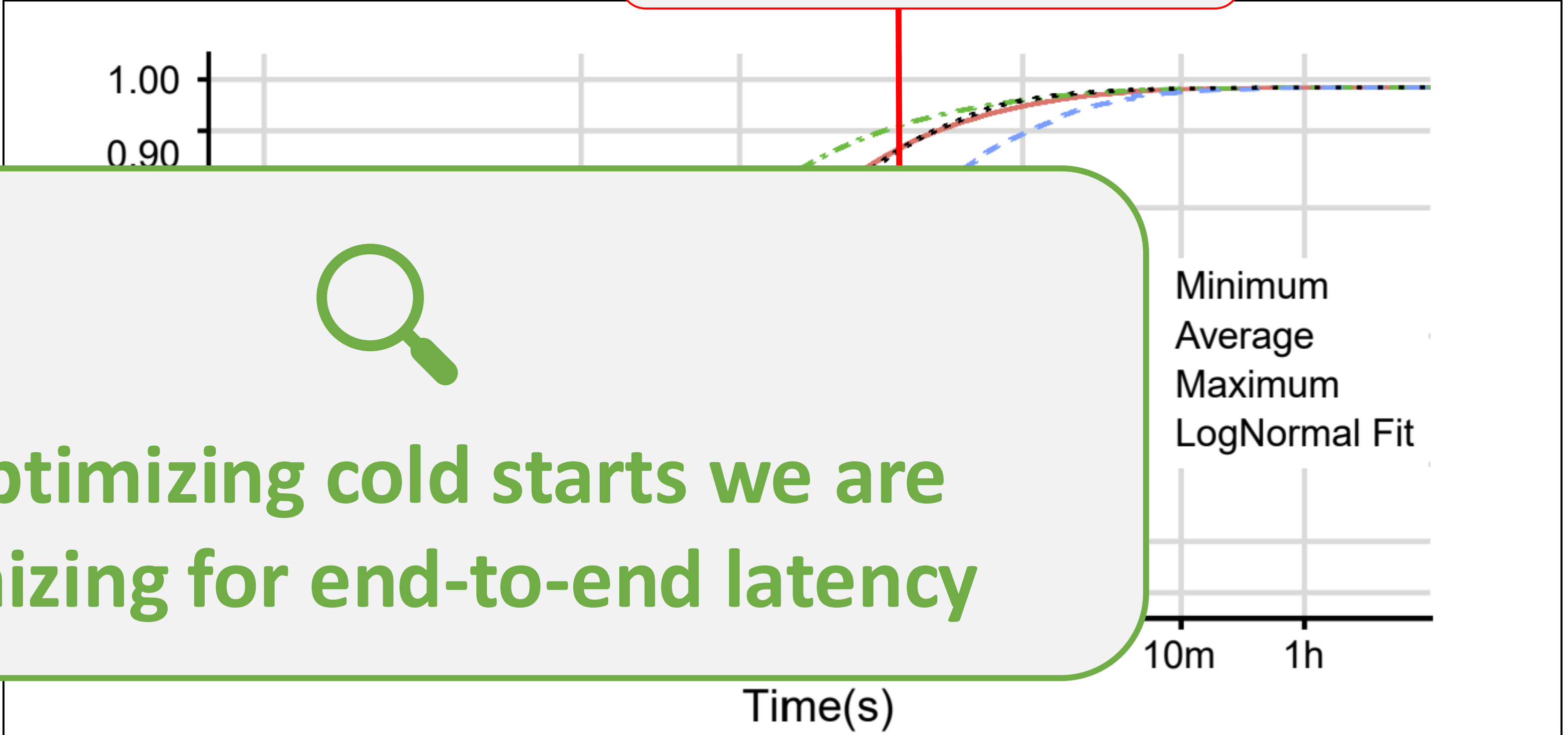


Figure 7: Distribution of function execution times. Min, avg, and max are separate CDFs, and use independent sorting.

By optimizing cold starts we are optimizing for end-to-end latency

[ATC'20] Serverless in the Wild: Characterizing and Optimizing the Serverless Workload at a Large Cloud Provider

Why Do Cold Starts NOT Matter?

	% non-interactive invocations
Azure [ATC'20]	
Meta [SOSP'23]	

Why Do Cold Starts NOT Matter?

	% non-interactive invocations
Azure [ATC'20]	**> 64%**
Meta [SOSP'23]	

Why Do Cold Starts NOT Matter?

	% non-interactive invocations
Azure [ATC'20]	**> 64%**
Meta [SOSP'23]	**> 85%**

Why Do Cold Starts NOT Matter?

	% non-interactive invocations
Azure [ATC'20]	**> 64%**
Meta [SOSP'23]	**> 85%**

If functions are triggered by non-interactive events, is end-to-end execution time the right metric to reduce?

Why Do Cold Starts NOT Matter?

	% non-interactive invocations
Azure [ATC'20]	**> 64%**
Meta [SOSP'23]	**> 85%**

35 papers	% latency insensitive
Functions	
Workflows	

If functions are triggered by non-interactive events, is end-to-end execution time the right metric to reduce?

Why Do Cold Starts NOT Matter?

	% non-interactive invocations
Azure [ATC'20]	**> 64%**
Meta [SOSP'23]	**> 85%**

35 papers	% latency insensitive
Functions	
Workflows	

If functions are triggered by non-interactive events, is end-to-end execution time the right metric to reduce?

Why Do Cold Starts NOT Matter?

	% non-interactive invocations
Azure [ATC'20]	**> 64%**
Meta [SOSP'23]	**> 85%**

35 papers	% latency insensitive
Functions	**> 80%**
Workflows	

If functions are triggered by non-interactive events, is end-to-end execution time the right metric to reduce?

Why Do Cold Starts NOT Matter?

	% non-interactive invocations
Azure [ATC'20]	**> 64%**
Meta [SOSP'23]	**> 85%**

35 papers	% latency insensitive
Functions	**> 80%**
Workflows	**> 70%**

If functions are triggered by non-interactive events, is end-to-end execution time the right metric to reduce?

Why Do Cold Starts NOT Matter?

	% non-interactive invocations
Azure [ATC'20]	**> 64%**
Meta [SOSP'23]	**> 85%**

35 papers	% latency insensitive
Functions	**> 80%**
Workflows	**> 70%**

If functions are triggered by non-interactive events, is end-to-end execution time the right metric to reduce?

If functions are latency insensitive, why reduce end-to-end execution time?

Why Do Cold Starts NOT Matter?

	% worker utilization
Huawei [SIGCOMM'24]	**~ 50%**
Meta [SOSP'23]	**~ 60%**

	% non-interactive invocations
Azure [ATC'20]	**> 64%**
Meta [SOSP'23]	**> 85%**

35 papers	% latency insensitive
Functions	**> 80%**
Workflows	**> 70%**

If functions are triggered by non-interactive events, is end-to-end execution time the right metric to reduce?

If functions are latency insensitive, why reduce end-to-end execution time?

Why Do Cold Starts NOT Matter?

	% worker utilization
Huawei [SIGCOMM'24]	**~ 50%**
Meta [SOSP'23]	**~ 60%**

Could low serverless utilization be a consequence of cold-start optimizations?

	% non-interactive invocations
Azure [ATC'20]	**> 64%**
Meta [SOSP'23]	**> 85%**

35 papers	% latency insensitive
Functions	**> 80%**
Workflows	**> 70%**

If functions are triggered by non-interactive events, is end-to-end execution time the right metric to reduce?

If functions are latency insensitive, why reduce end-to-end execution time?

Is It Time to Put Cold Starts in the Deep Freeze?

Serverless functions are triggered by non-interactive events

Serverless functions are latency insensitive

Is It Time to Put Cold Starts in the Deep Freeze?

Serverless functions are triggered by non-interactive events

Serverless functions are latency insensitive

! Much serverless usage today is for non-interactive, latency-insensitive workloads

Is It Time to Put Cold Starts in the Deep Freeze?

Serverless functions are triggered by non-interactive events

Serverless functions are latency insensitive

BATCH

!

Much serverless usage today is for non-interactive, latency-insensitive workloads

Is It Time to Put Cold Starts in the Deep Freeze?

Serverless functions are triggered by non-interactive events

Serverless functions are latency insensitive

BATCH

! Much serverless usage today is for non-interactive, latency-insensitive workloads

? Serverless infrastructure lagging behind research and/or need even more cold-start optimizations?

[SoCC'23] Serverless Gap [SOSP'24] Dirigent

Is It Time to Put Cold Starts in the Deep Freeze?

Serverless functions are triggered by non-interactive events

Serverless functions are latency insensitive

BATCH

! Much serverless usage today is for non-interactive, latency-insensitive workloads

? Serverless infrastructure lagging behind research and/or need even more cold-start optimizations?

[SoCC'23] Serverless Gap [SOSP'24] Dirigent

? Non-interactive, latency-insensitive, batch workloads are a good fit for serverless?

Is It Time to Put Cold Starts in the Deep Freeze?

Serverless functions are triggered by non-interactive events

Serverless functions are latency insensitive

BATCH

! Much serverless usage today is for non-interactive, latency-insensitive workloads

? Serverless infrastructure lagging behind research and/or need even more cold-start optimizations?

[SoCC'23] Serverless Gap [SOSP'24] Dirigent

Batch workloads are a good fit for serverless, and we should also optimize for them!

Is It Time to Put Cold Starts in the Deep Freeze?

Serverless functions are triggered by non-interactive events

Serverless functions are latency insensitive

! Much serverless usage today is for non-interactive, latency-insensitive workloads

BATCH

? Serverless infrastructure lagging behind research and/or need even more cold-start optimizations?

[SoCC'23] Serverless Gap [SOSP'24] Dirigent

Batch workloads are a good fit for serverless, and we should also optimize for them!

Utilization

Is It Time to Put Cold Starts in the Deep Freeze?

Serverless functions are triggered by non-interactive events

Serverless functions are latency insensitive

BATCH

! Much serverless usage today is for non-interactive, latency-insensitive workloads

? Serverless infrastructure lagging behind research and/or need even more cold-start optimizations?

[SoCC'23] Serverless Gap [SOSP'24] Dirigent

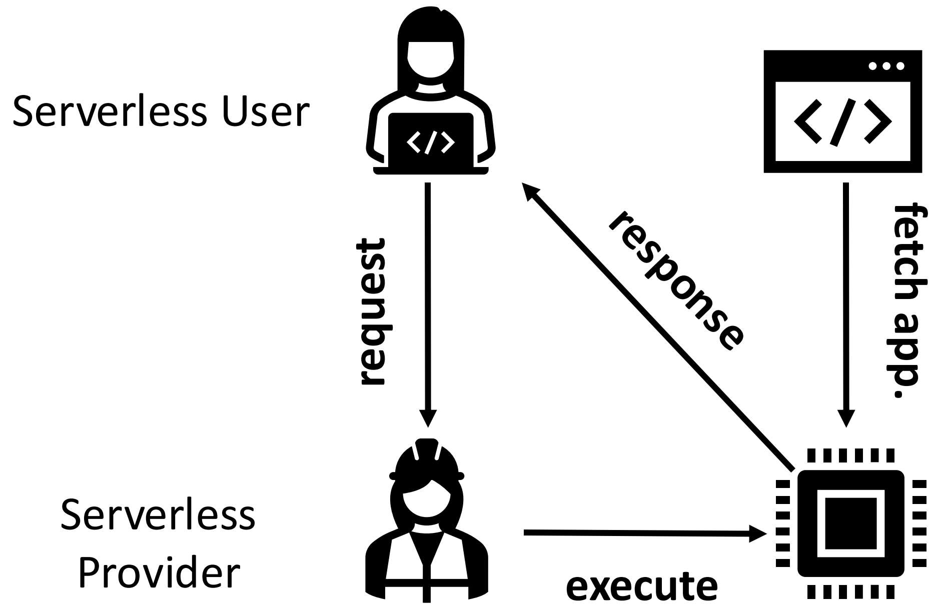
Batch workloads are a good fit for serverless, and we should also optimize for them!

Utilization

Throughput

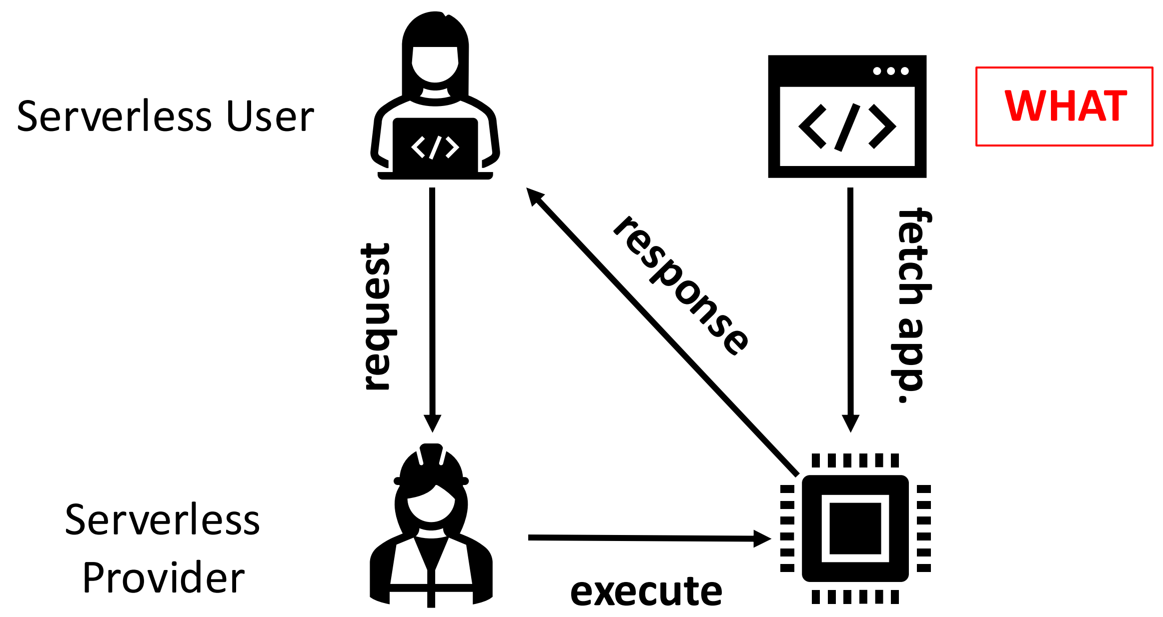
Batch Workloads Fit The Serverless Model

Traditional Serverless

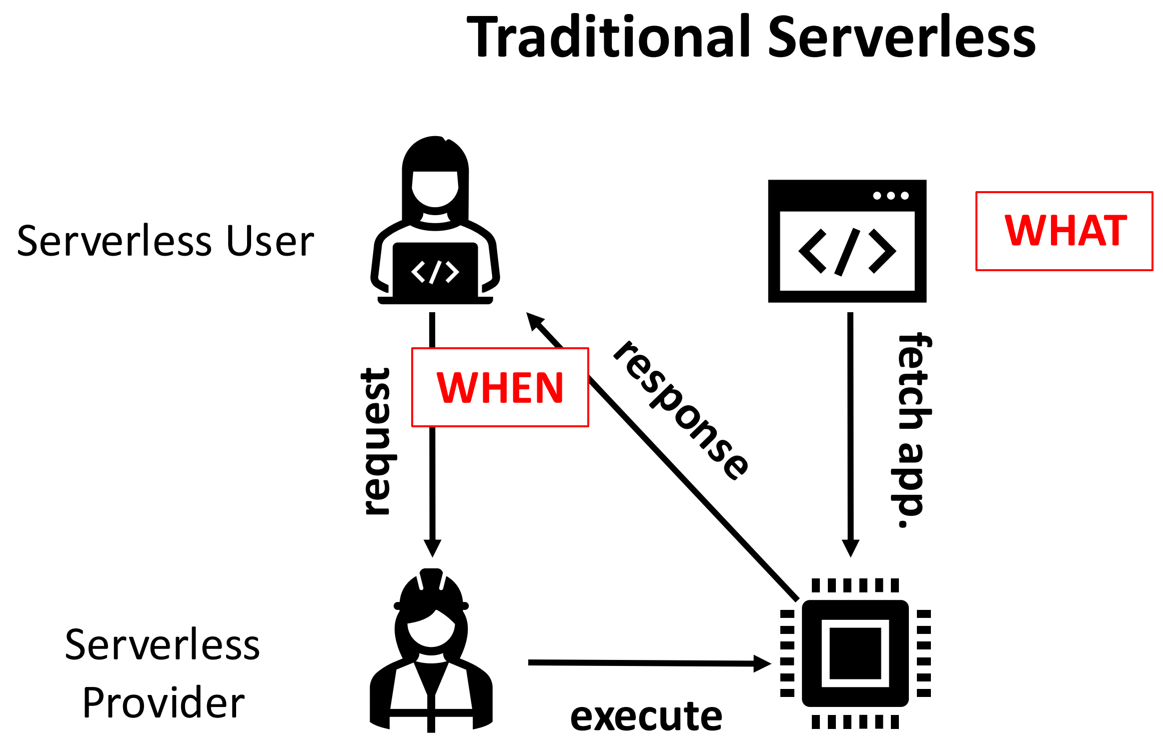


Batch Workloads Fit The Serverless Model

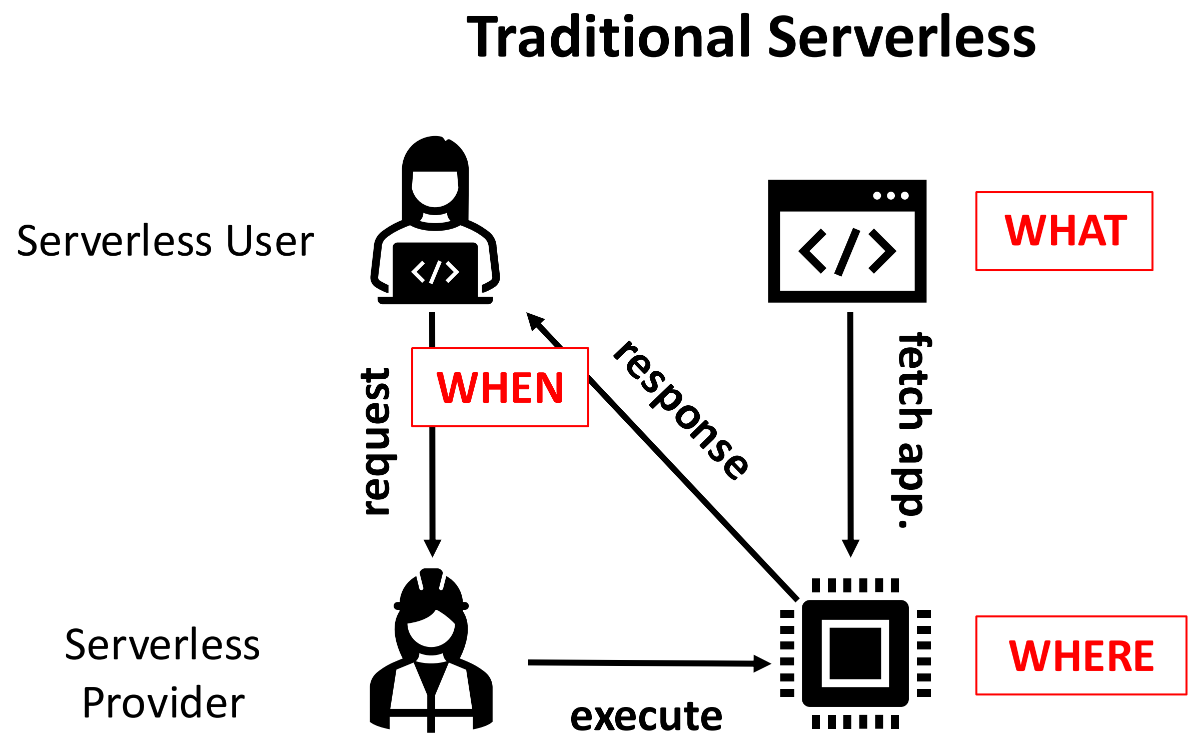
Traditional Serverless



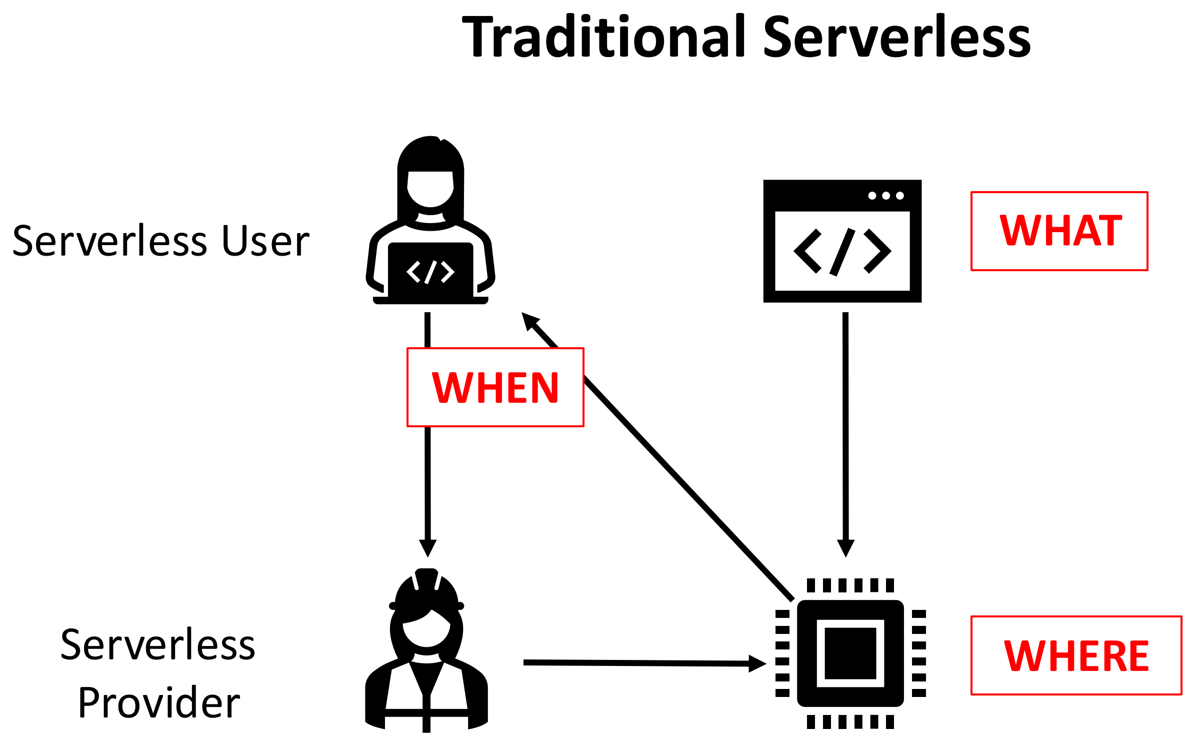
Batch Workloads Fit The Serverless Model

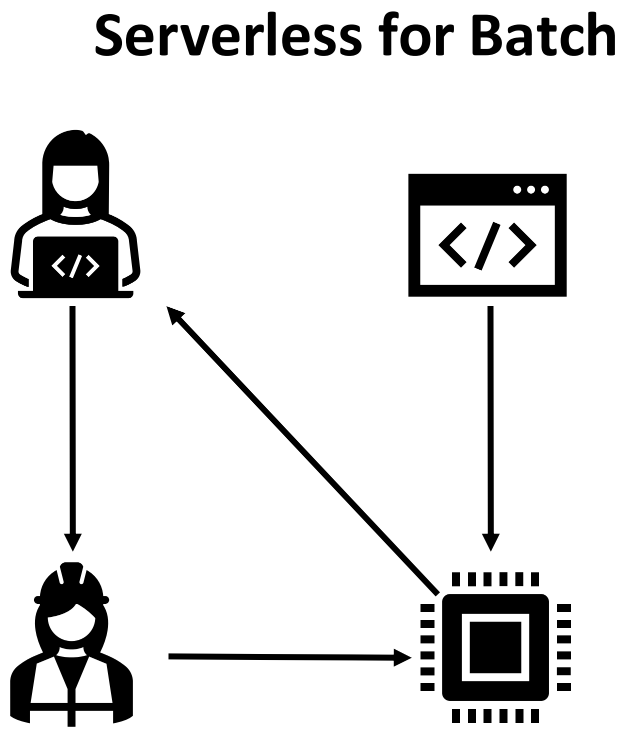


Batch Workloads Fit The Serverless Model

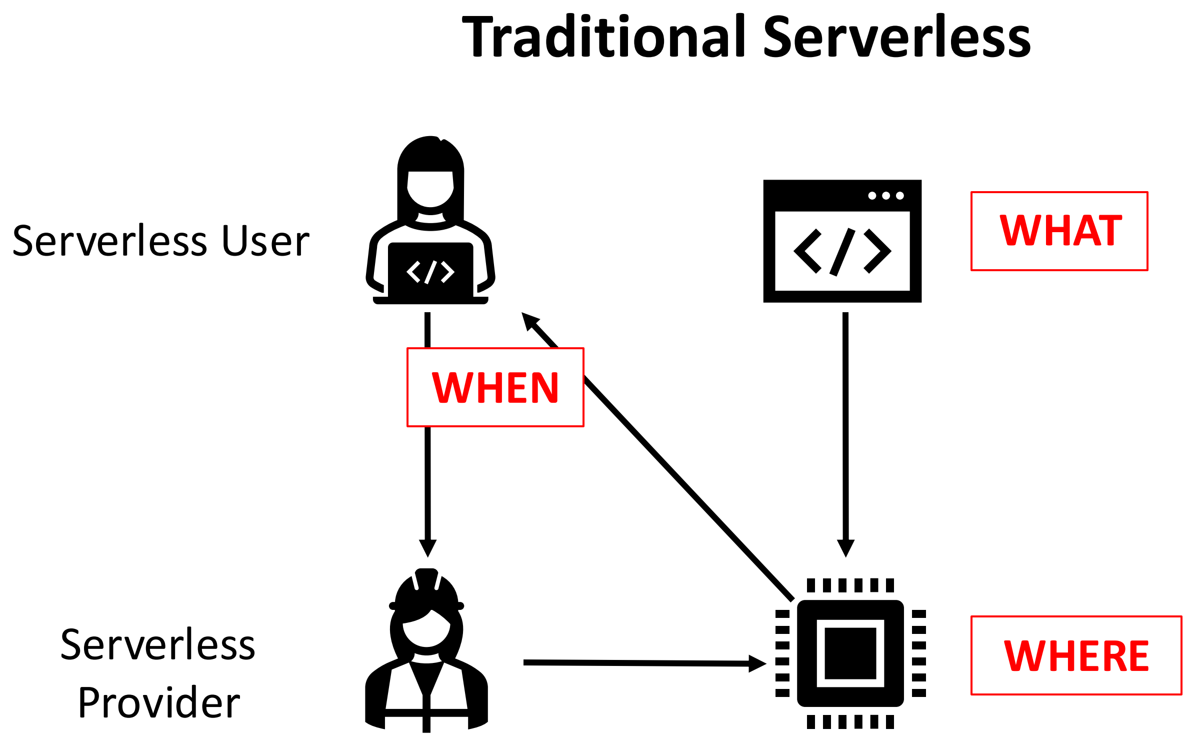


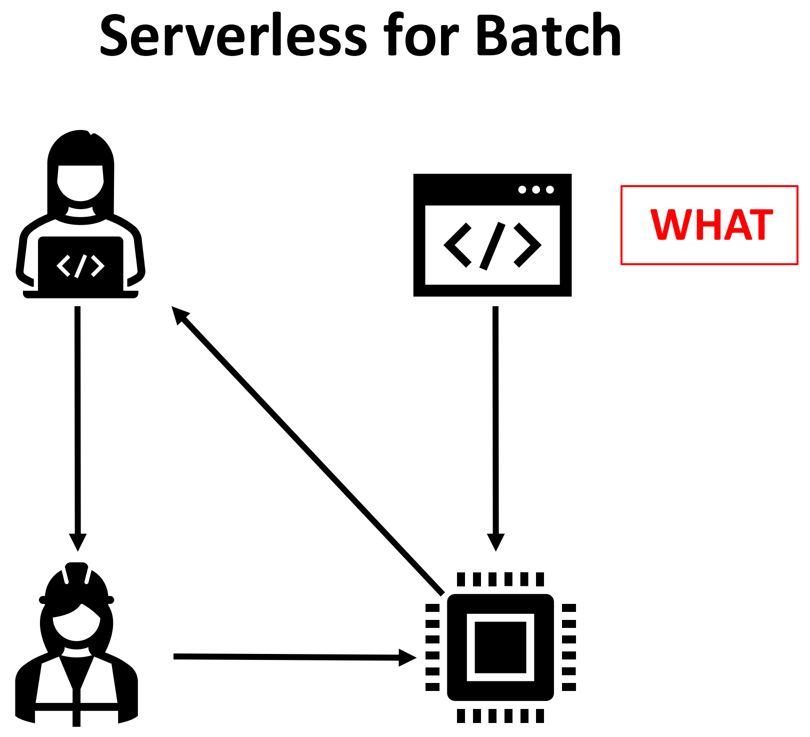
Batch Workloads Fit The Serverless Model



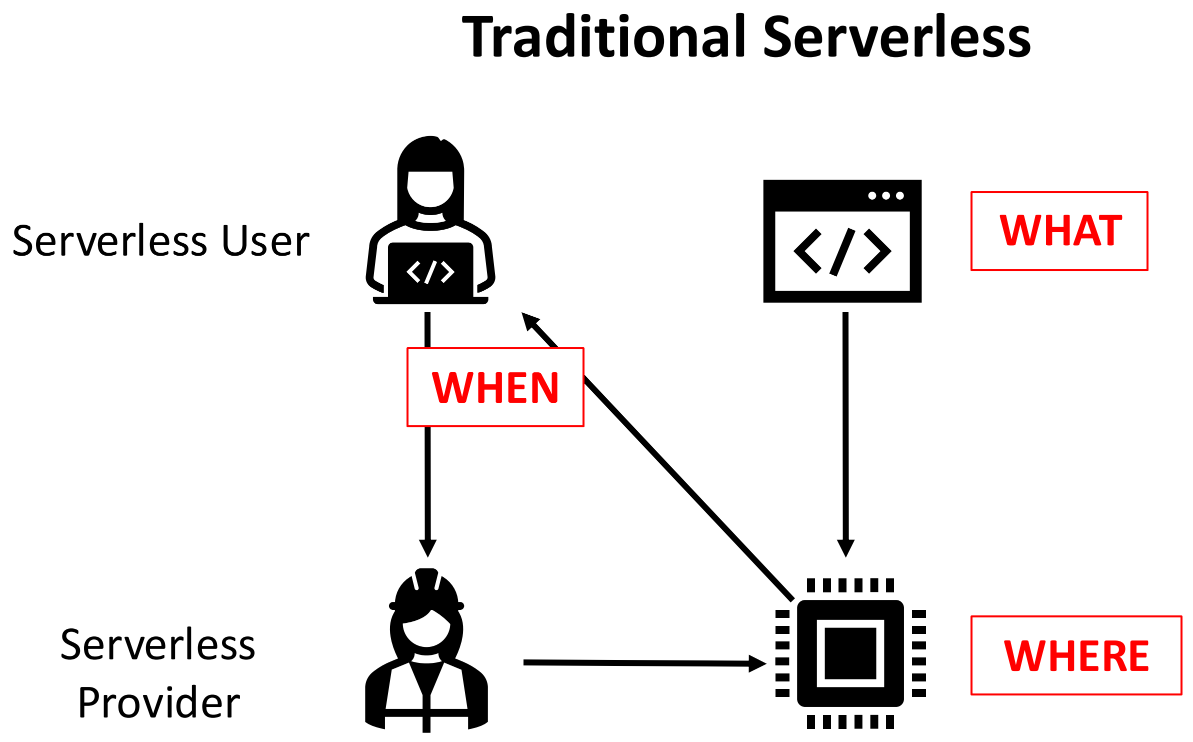


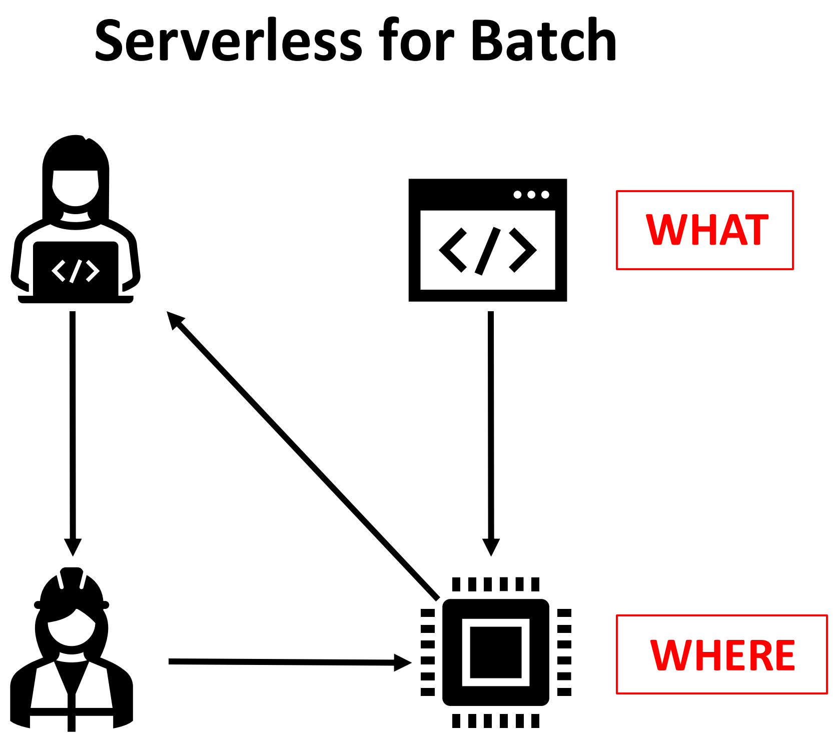
Batch Workloads Fit The Serverless Model



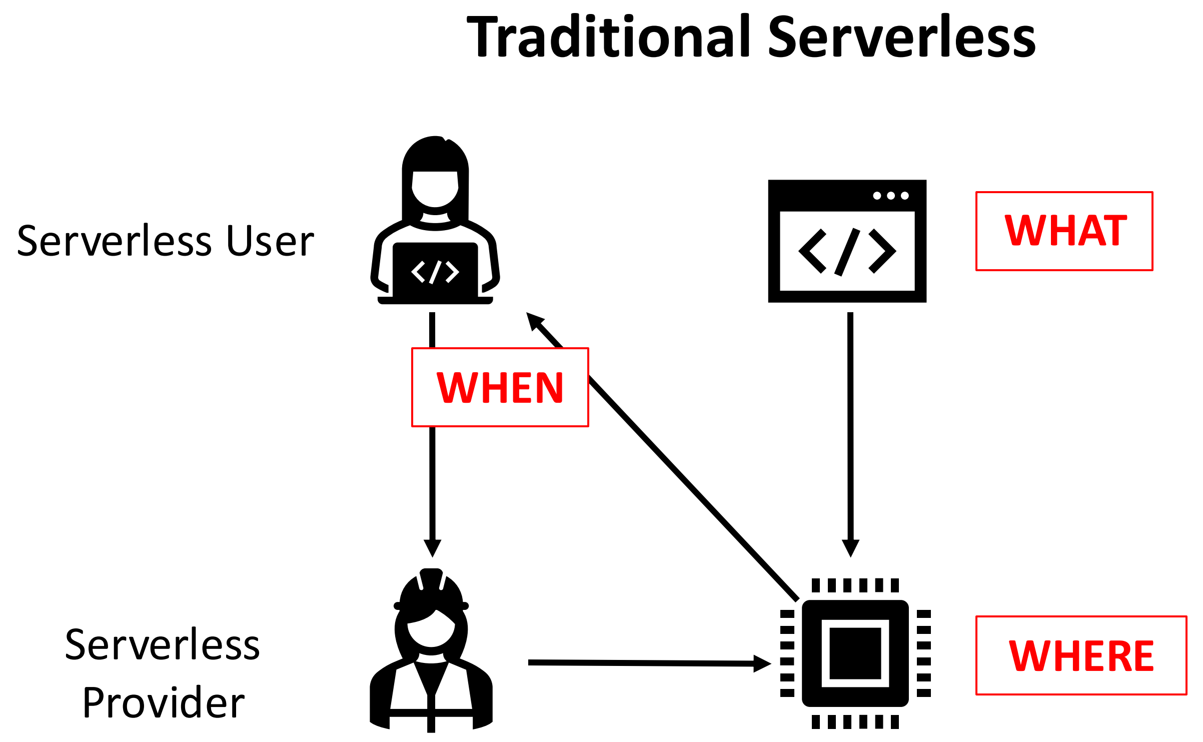


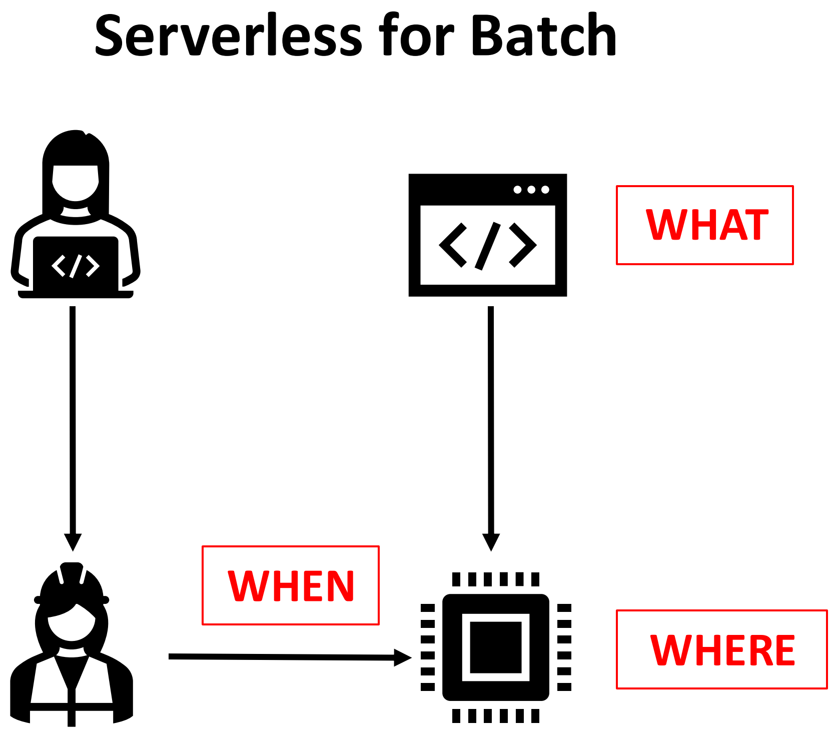
Batch Workloads Fit The Serverless Model



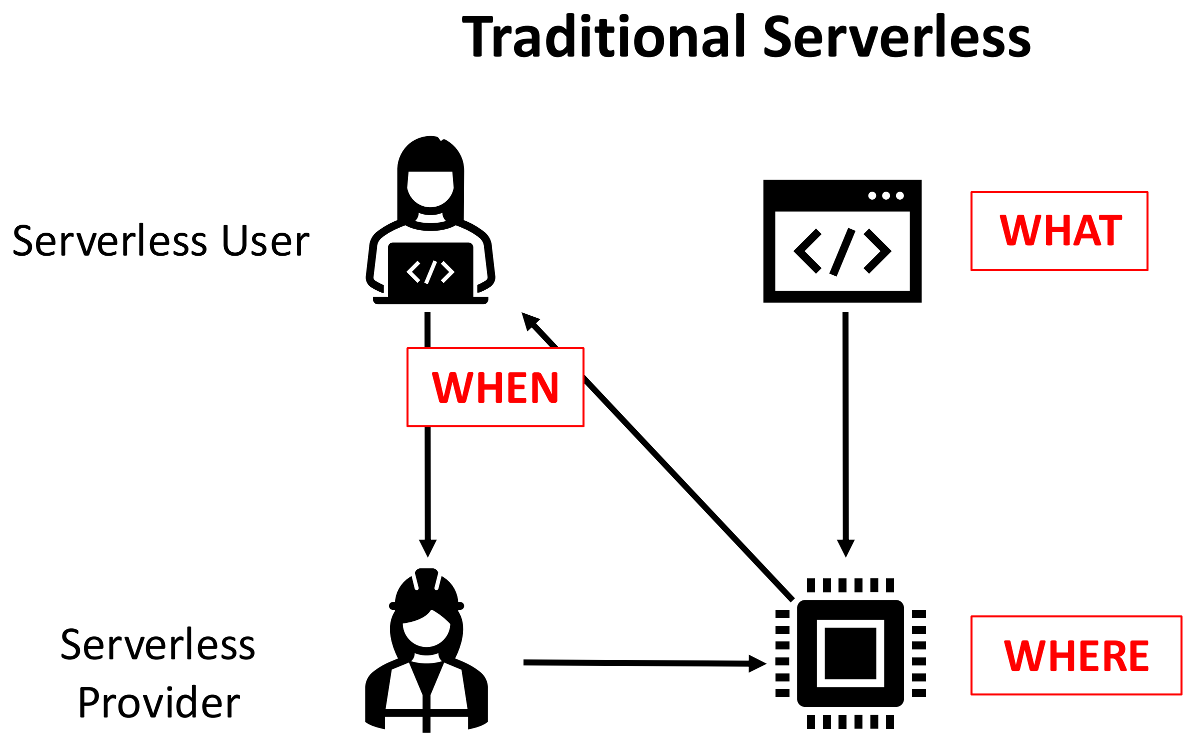


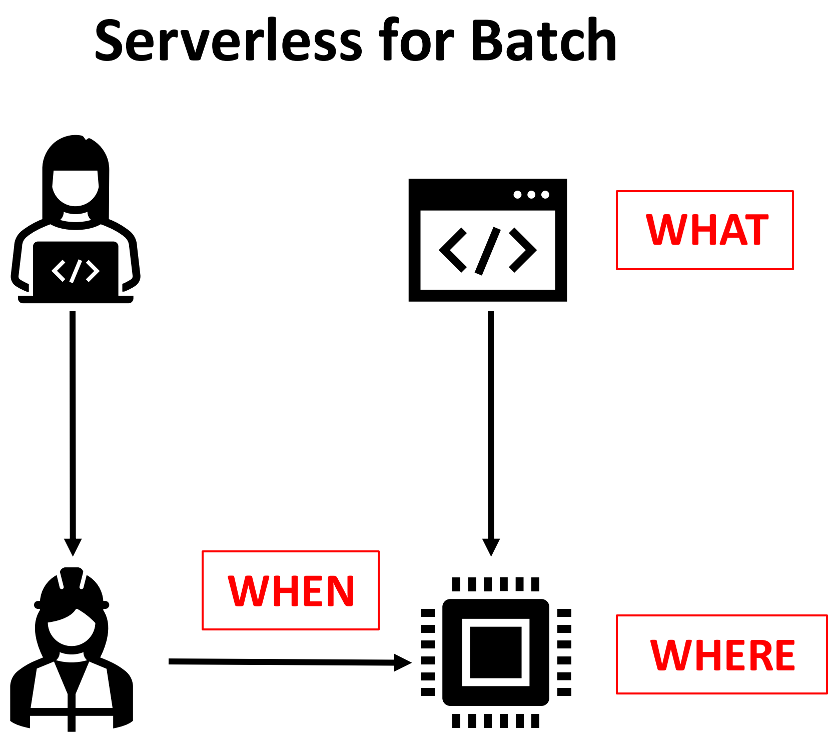
Batch Workloads Fit The Serverless Model





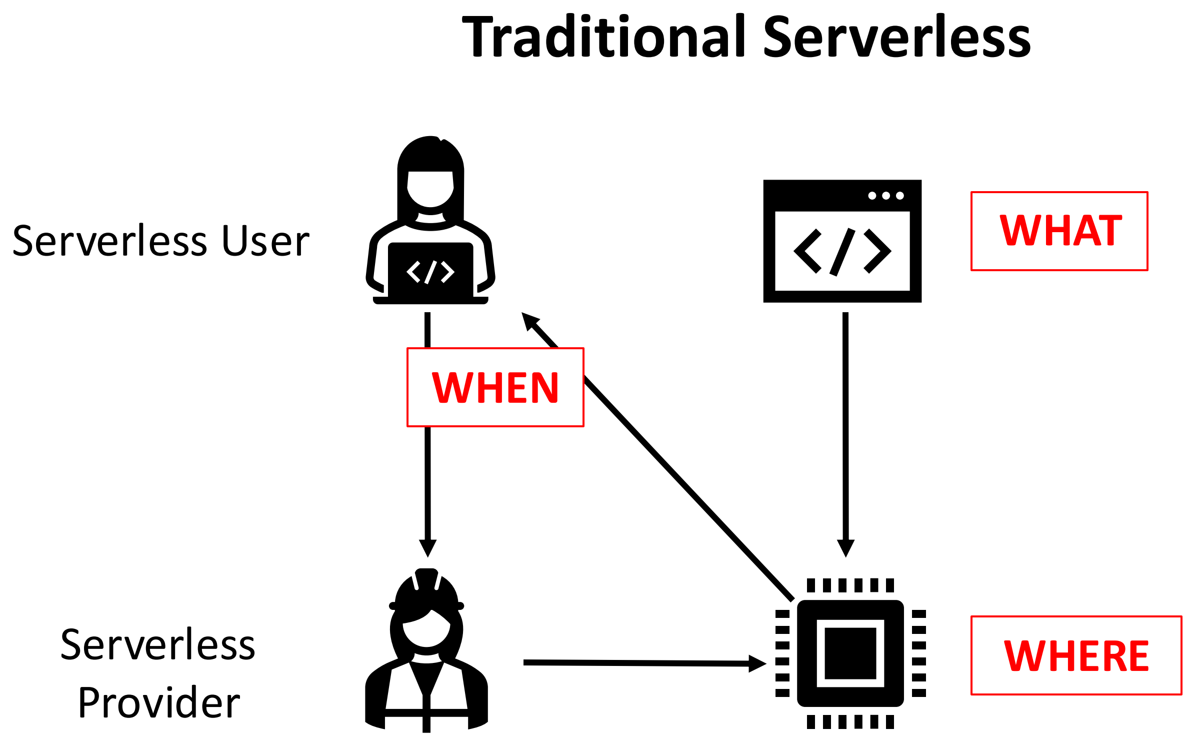
Batch Workloads Fit The Serverless Model

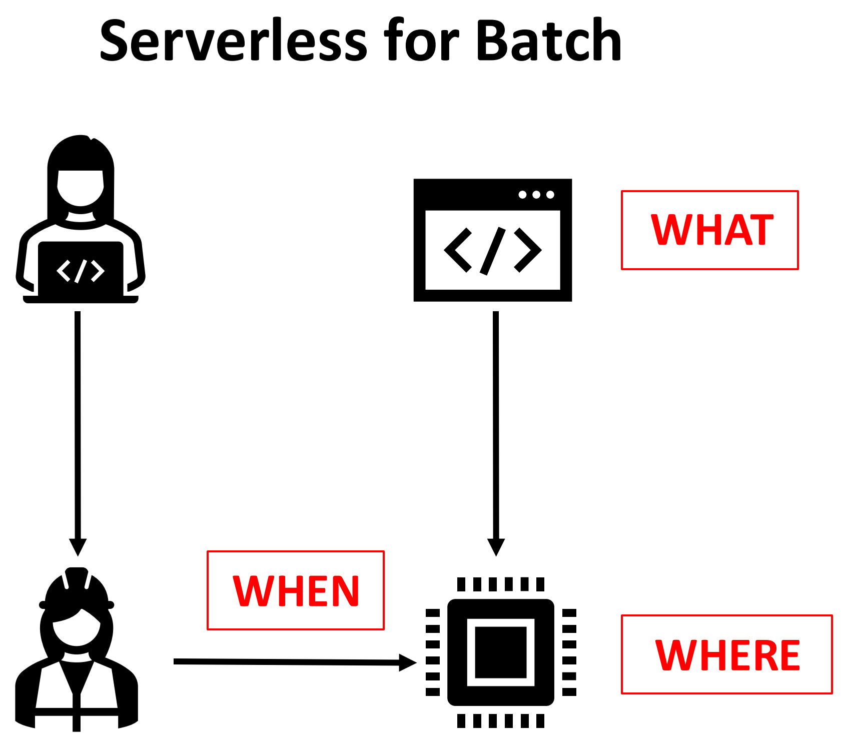




Serverless for non-interactive batch workloads can improve **resource utilization** by further delegating control to the serverless provider

Batch Workloads Fit The Serverless Model





Serverless for non-interactive batch workloads can improve **energy efficiency** by further delegating control to the serverless provider

The Serverless Model Fits Batch Workloads

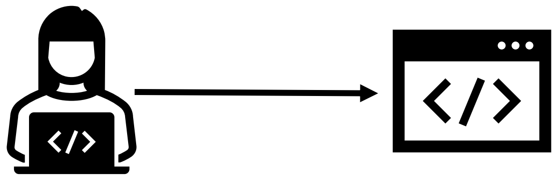
Traditional Serverless

Serverless for Batch

The Serverless Model Fits Batch Workloads

Traditional Serverless

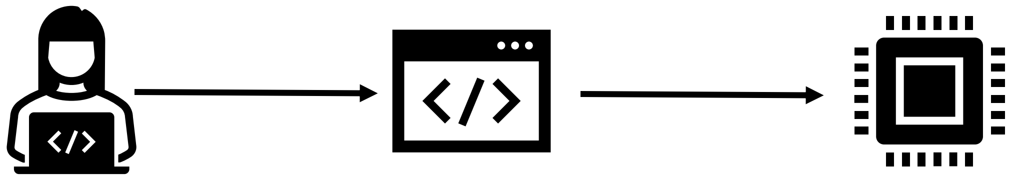
Serverless for Batch



The Serverless Model Fits Batch Workloads

Traditional Serverless

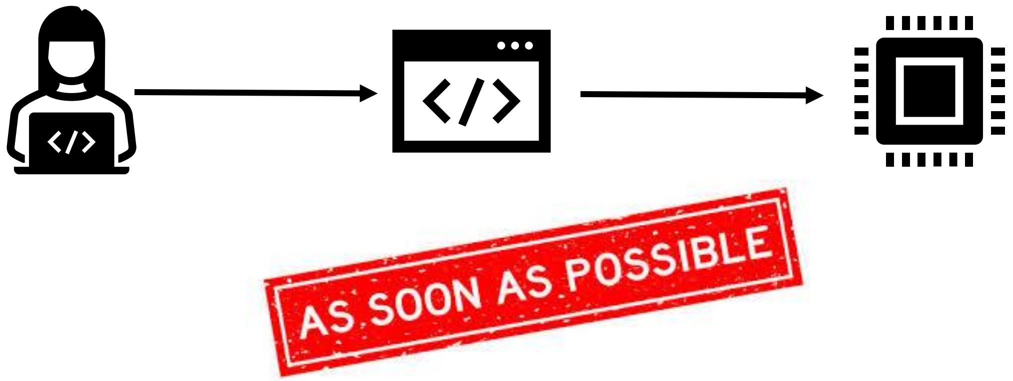
Serverless for Batch



The Serverless Model Fits Batch Workloads

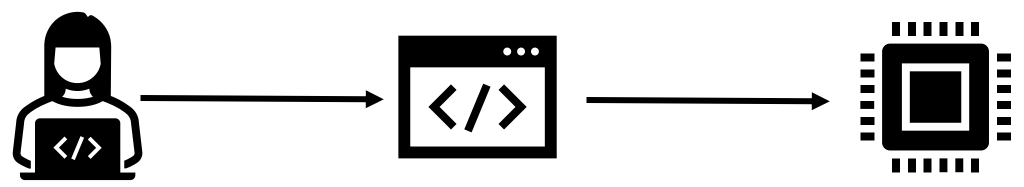
Traditional Serverless

Serverless for Batch

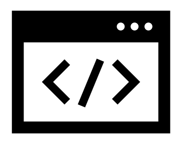


The Serverless Model Fits Batch Workloads

Traditional Serverless

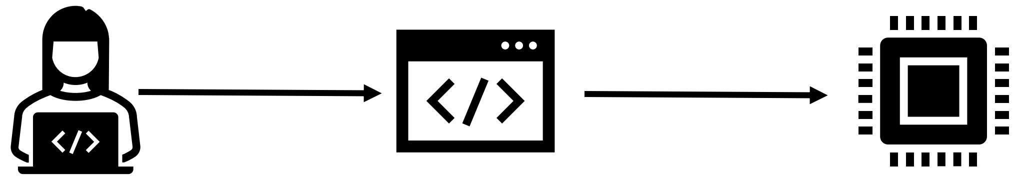


Serverless for Batch

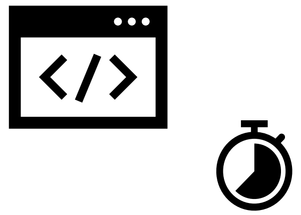


The Serverless Model Fits Batch Workloads

Traditional Serverless

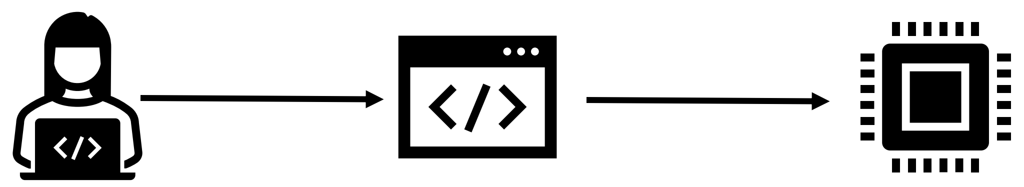


Serverless for Batch

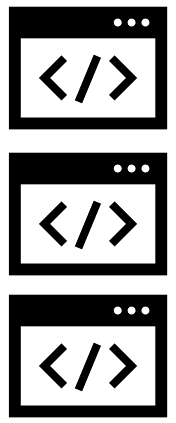


The Serverless Model Fits Batch Workloads

Traditional Serverless

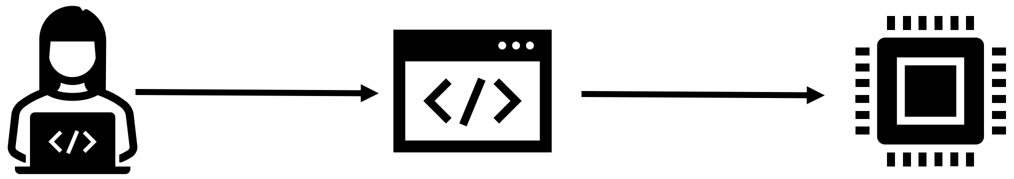


Serverless for Batch

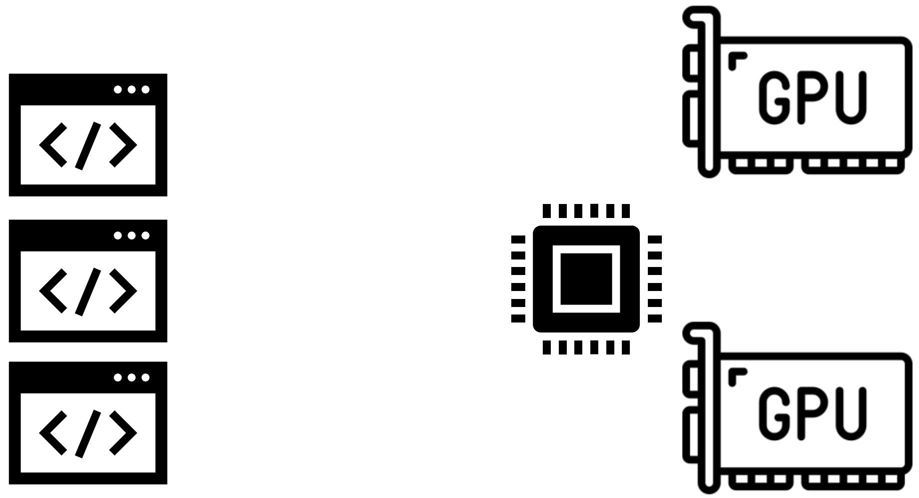


The Serverless Model Fits Batch Workloads

Traditional Serverless

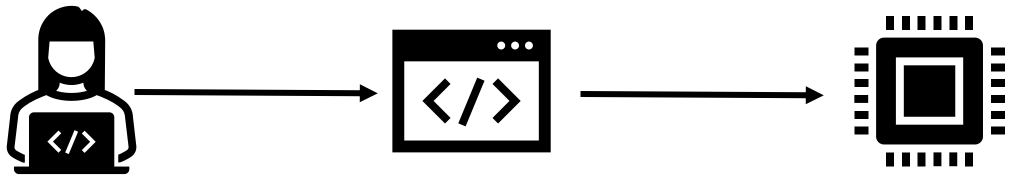


Serverless for Batch

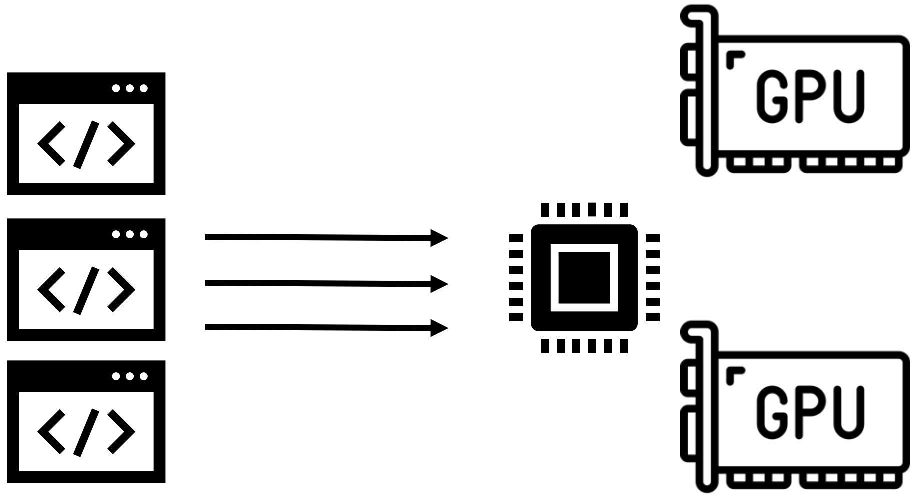


The Serverless Model Fits Batch Workloads

Traditional Serverless

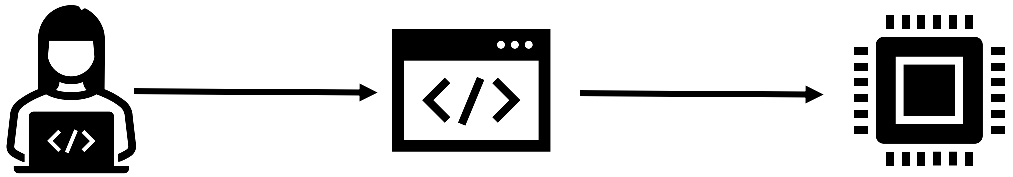


Serverless for Batch

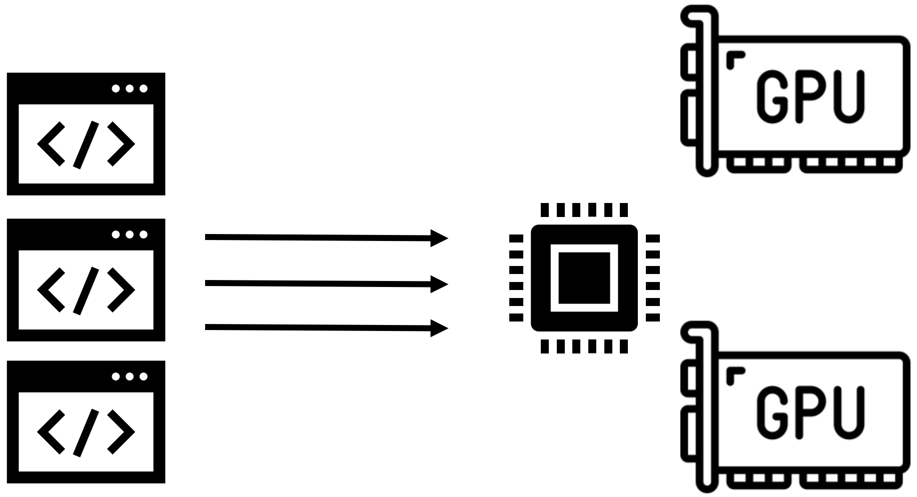


The Serverless Model Fits Batch Workloads

Traditional Serverless



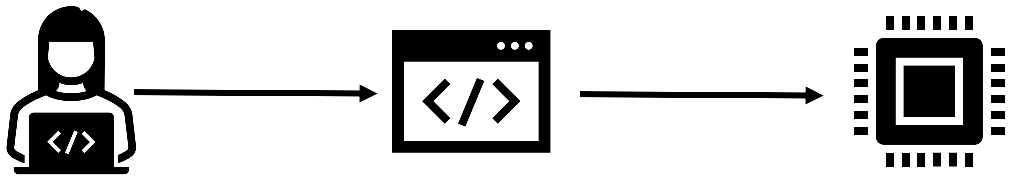
Serverless for Batch



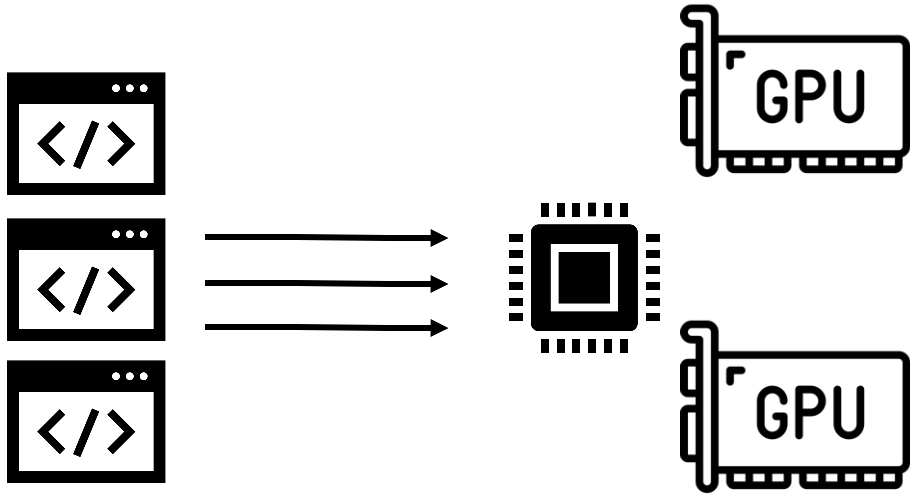
+ Efficiency

The Serverless Model Fits Batch Workloads

Traditional Serverless



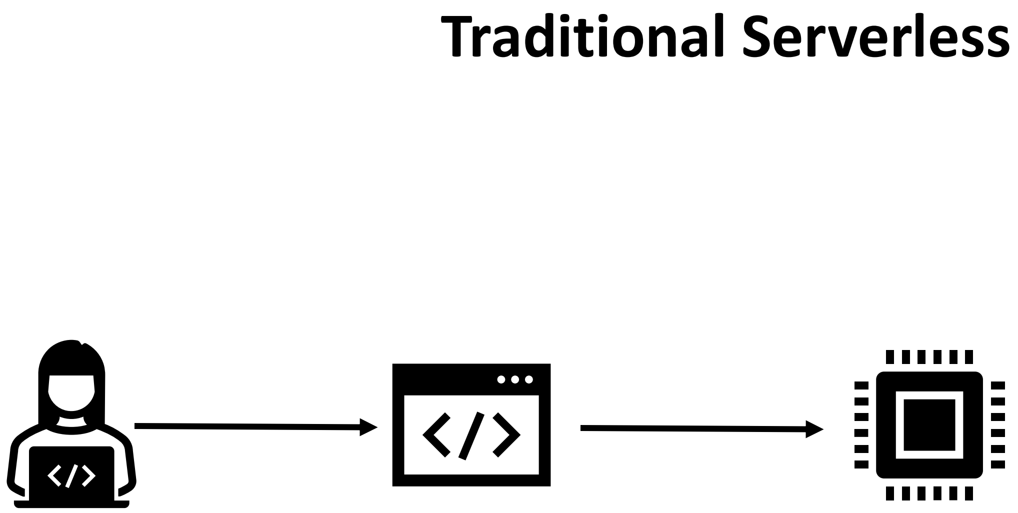
Serverless for Batch

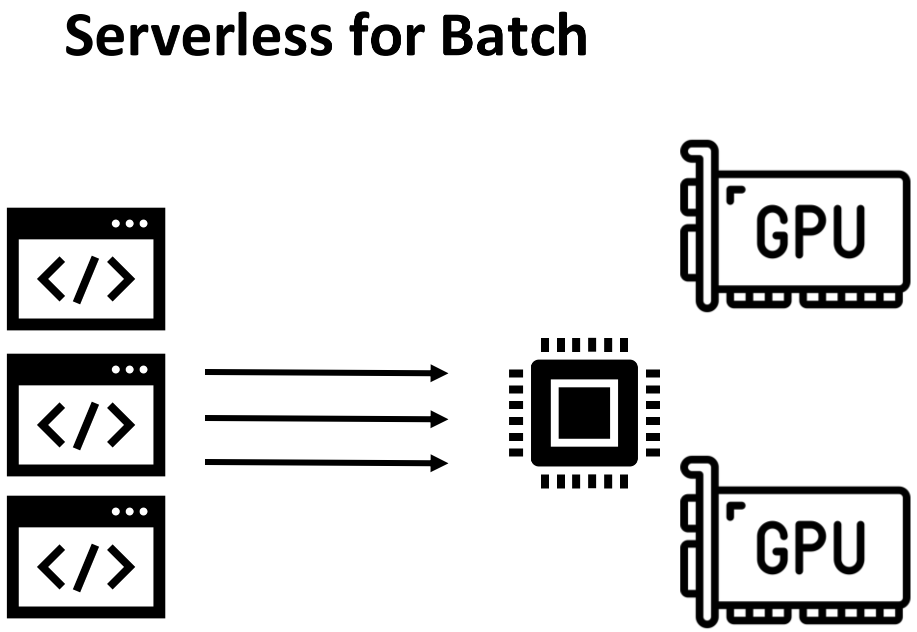


+ Efficiency

- Cost

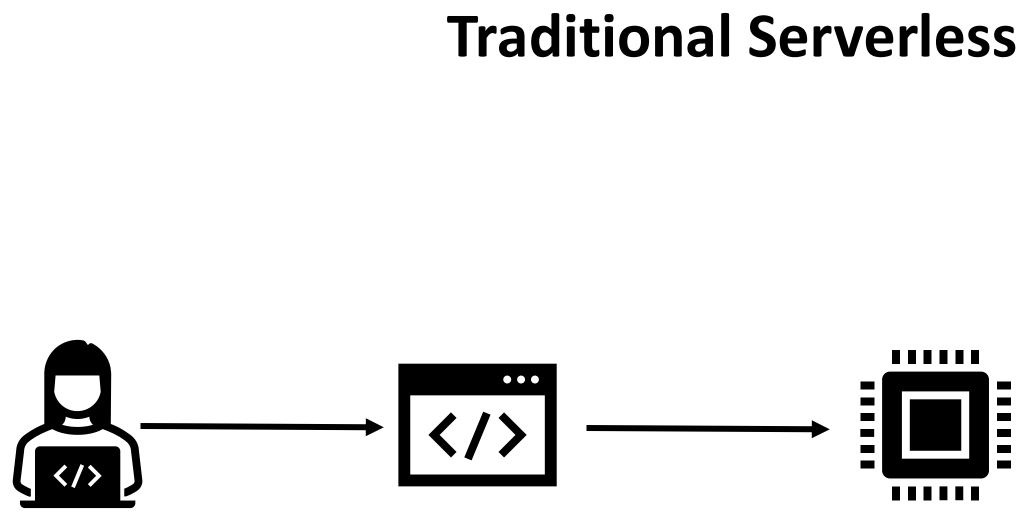
The Serverless Model Fits Batch Workloads

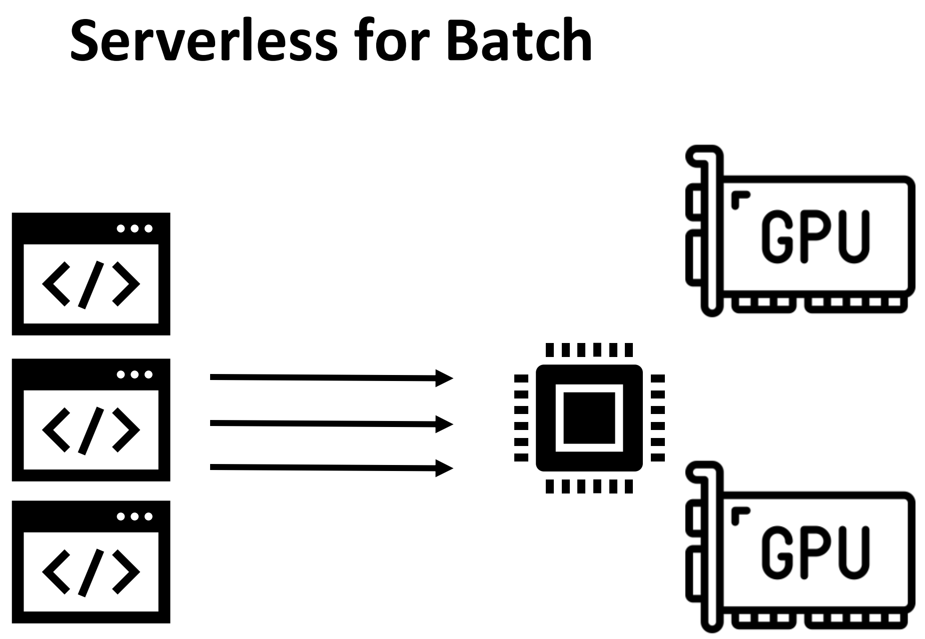




Serverless for batch workloads can improve **throughput** by leveraging accelerators

The Serverless Model Fits Batch Workloads





Serverless for batch workloads can improve **throughput** by **transparently** leveraging accelerators

[ASPLOS'22] Molecule

DFaaS: Delayable FaaS

How can we better support batch workloads with a serverless model?

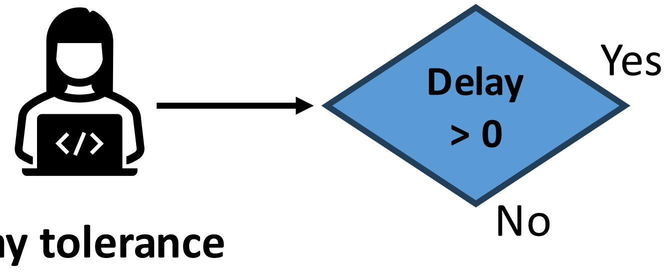
DFaaS: Delayable FaaS

How can we better support batch workloads with a serverless model?

delay tolerance

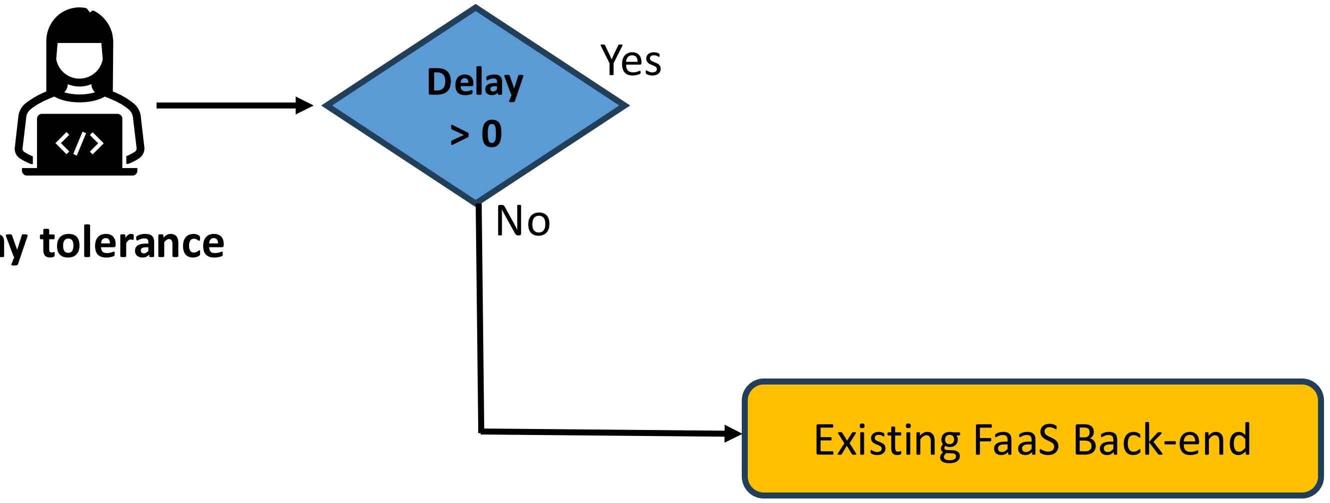
DFaaS: Delayable FaaS

How can we better support batch workloads with a serverless model?



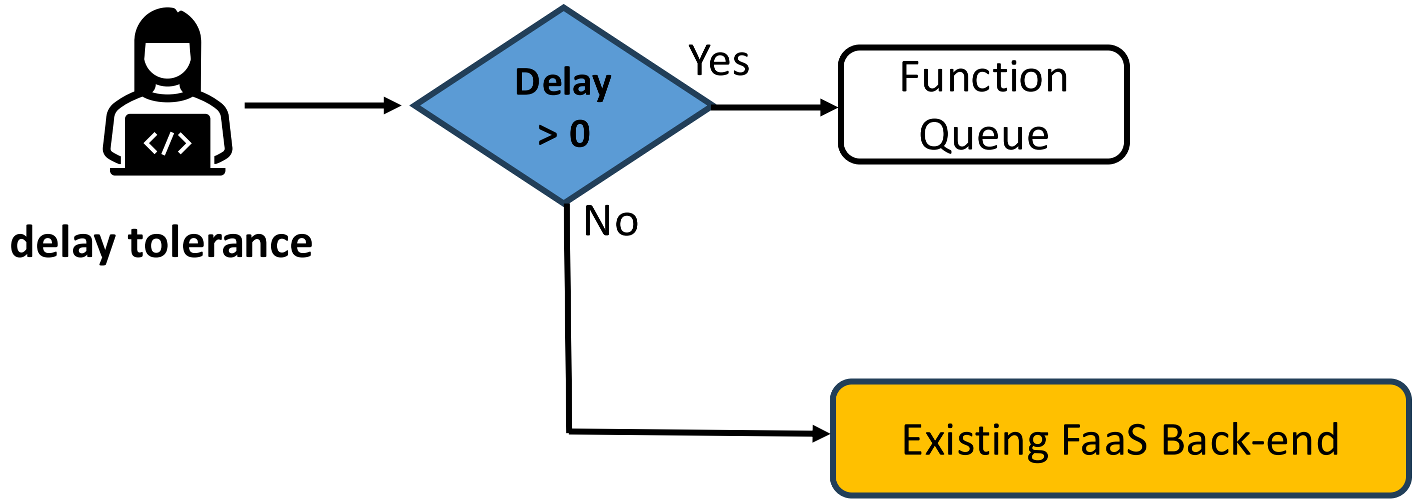
DFaaS: Delayable FaaS

How can we better support batch workloads with a serverless model?



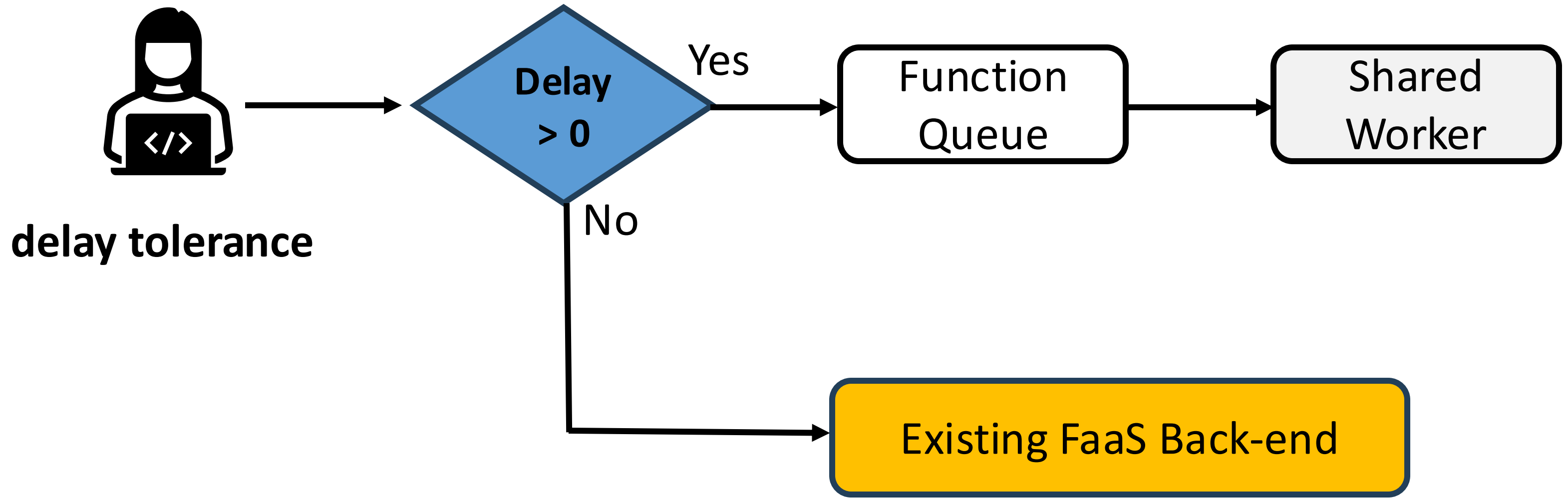
DFaaS: Delayable FaaS

How can we better support batch workloads with a serverless model?



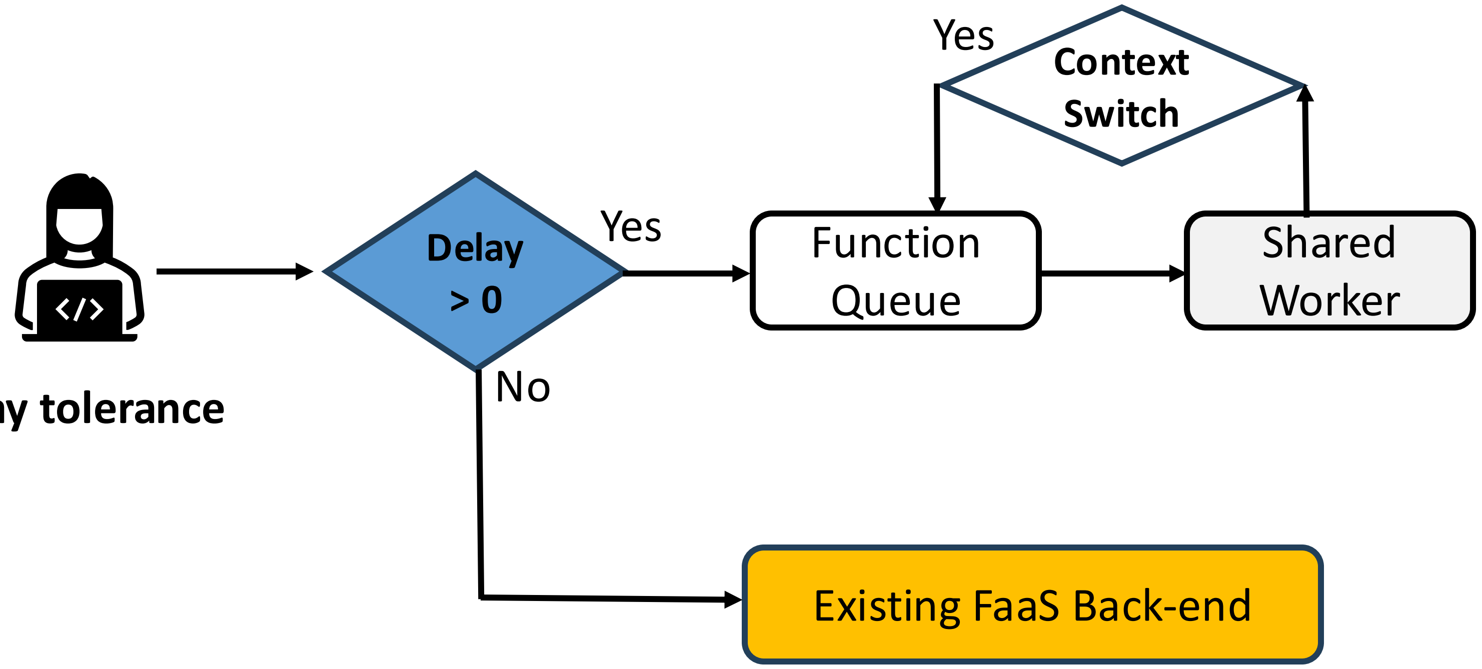
DFaaS: Delayable FaaS

How can we better support batch workloads with a serverless model?



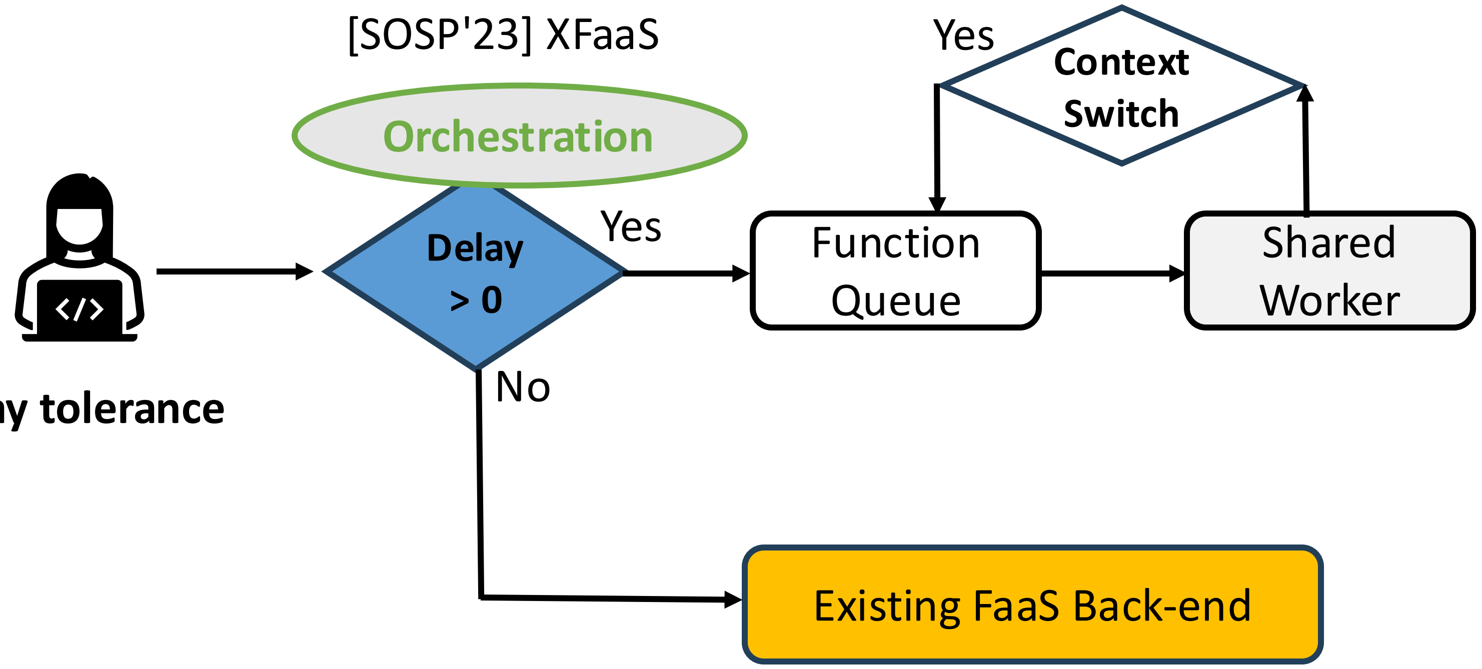
DFaaS: Delayable FaaS

How can we better support batch workloads with a serverless model?



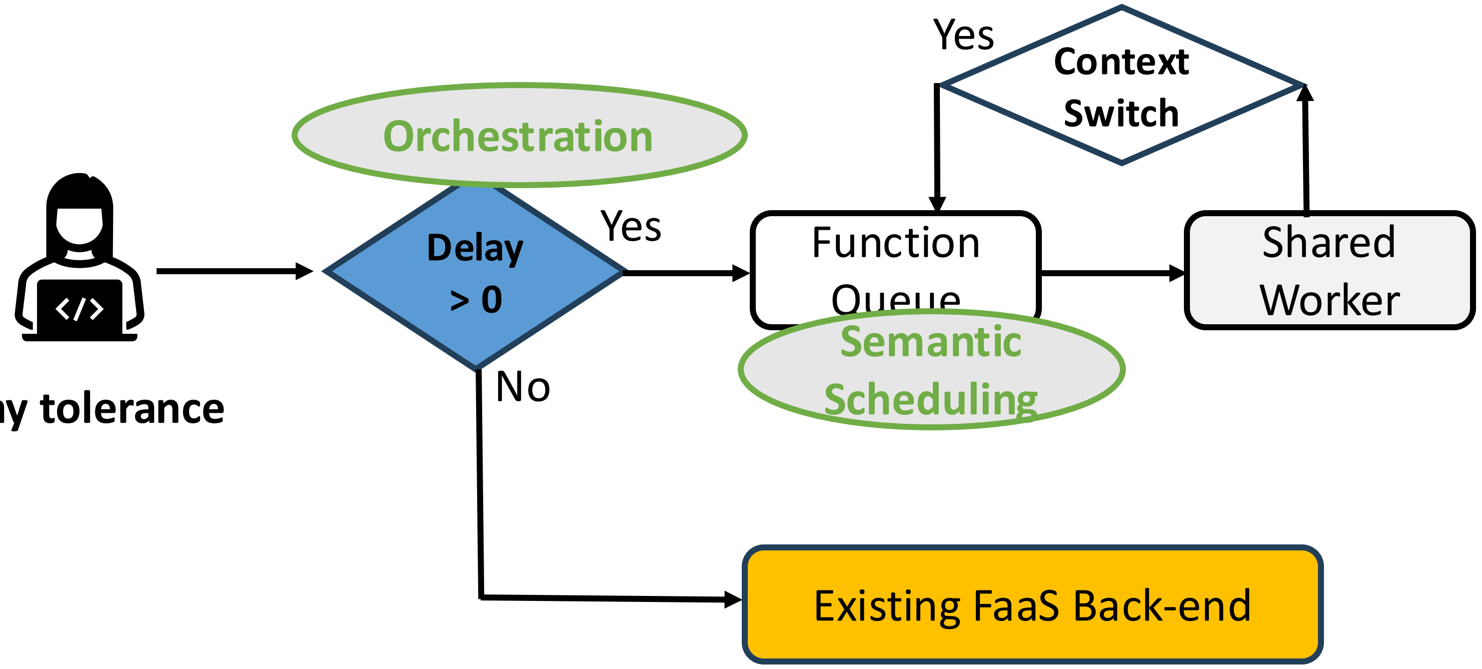
DFaaS: Delayable FaaS

How can we better support batch workloads with a serverless model?



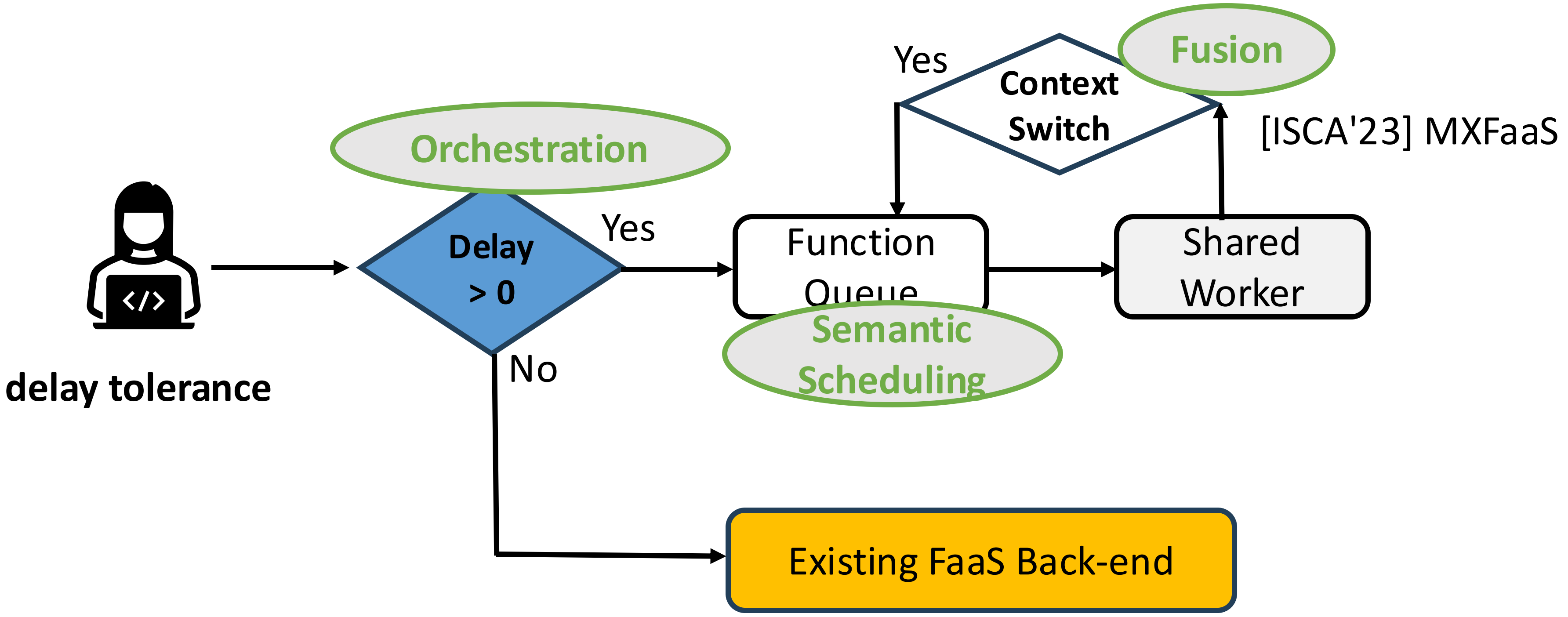
DFaaS: Delayable FaaS

How can we better support batch workloads with a serverless model?



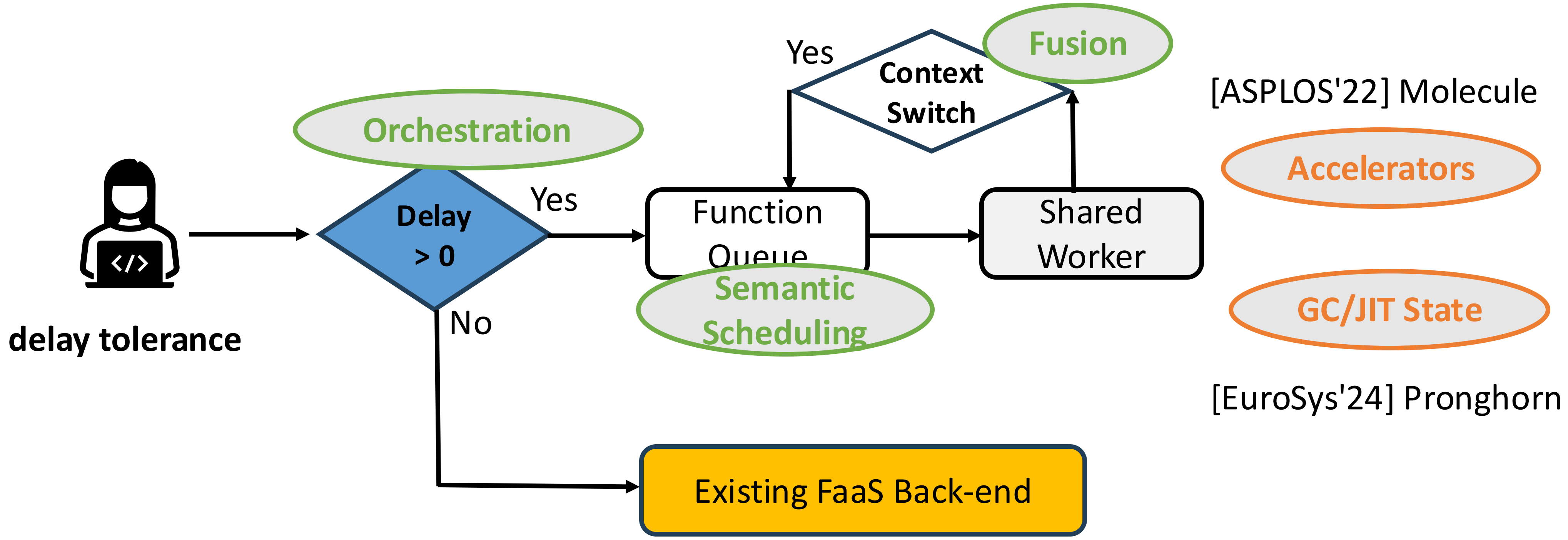
DFaaS: Delayable FaaS

How can we better support batch workloads with a serverless model?



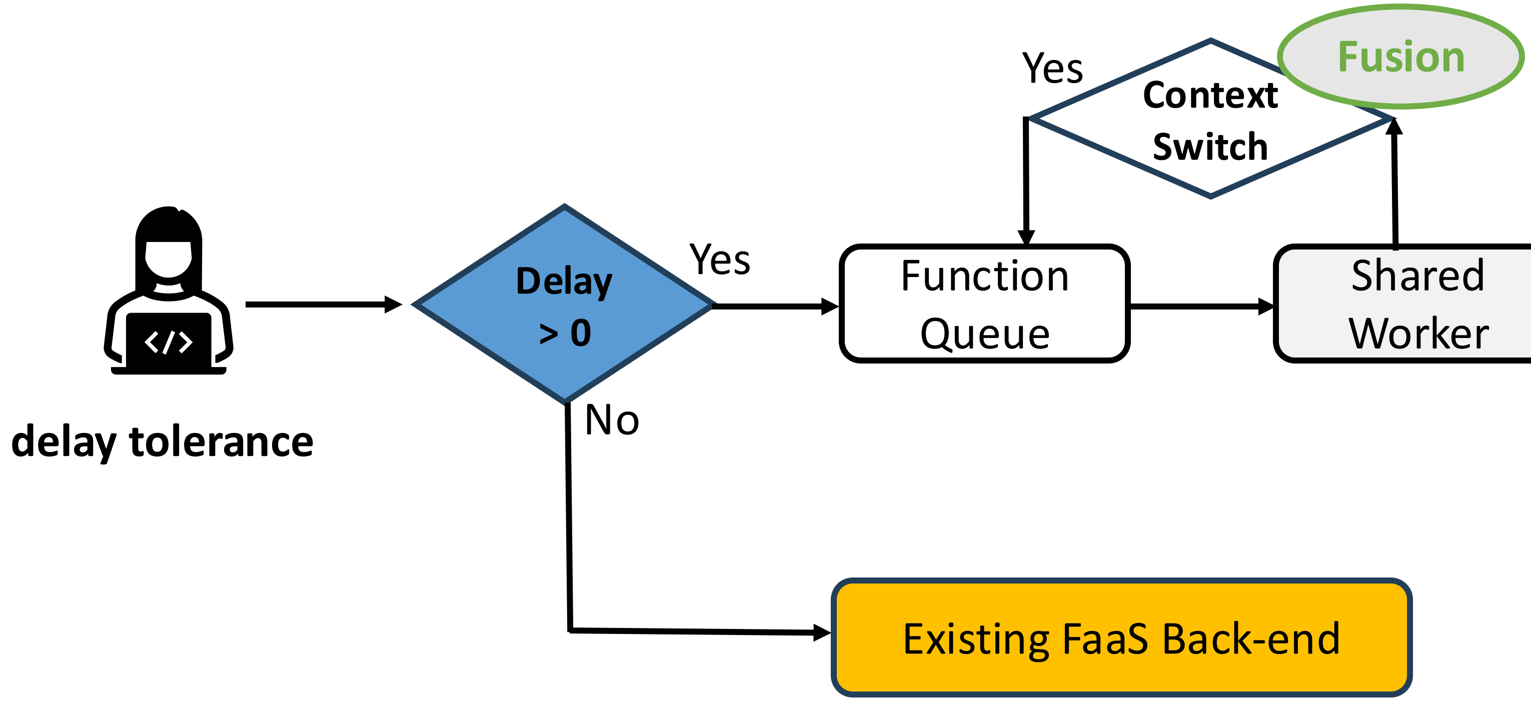
DFaaS: Delayable FaaS

How can we better support batch workloads with a serverless model?

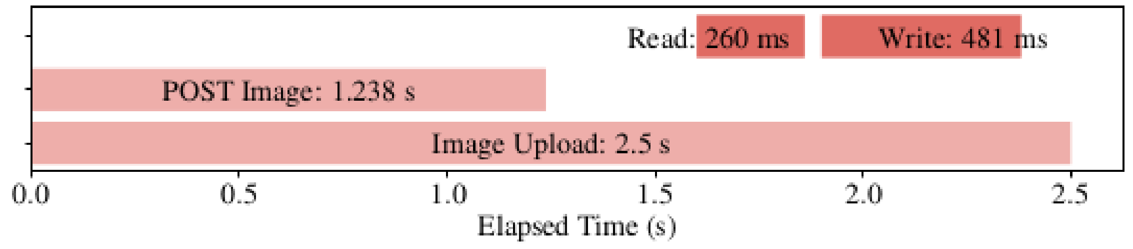


DFaaS: Delayable FaaS

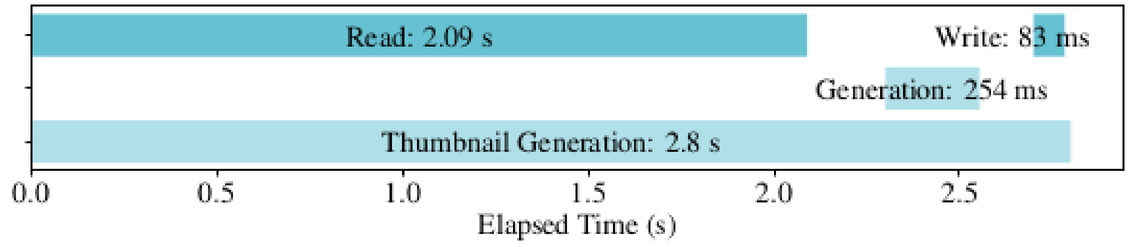
How can we better support batch workloads with a serverless model?



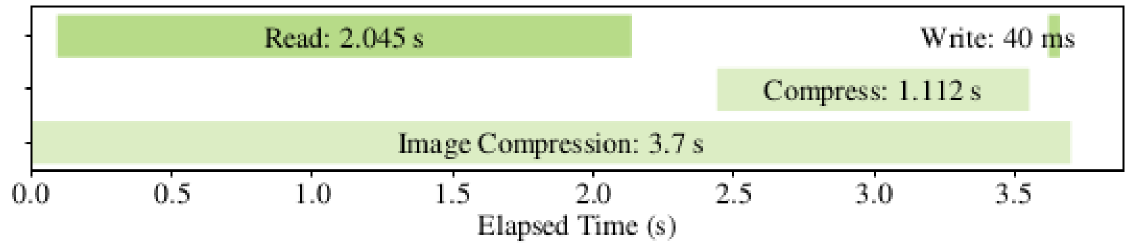
DFaaS: Function Fusion



(a) Image upload

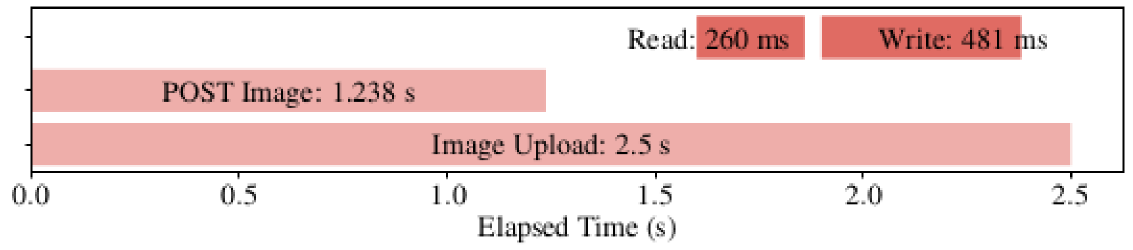


(b) Thumbnail generation

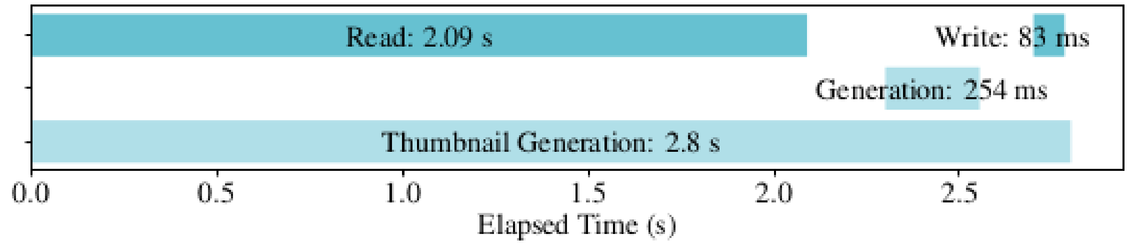


(c) Compression

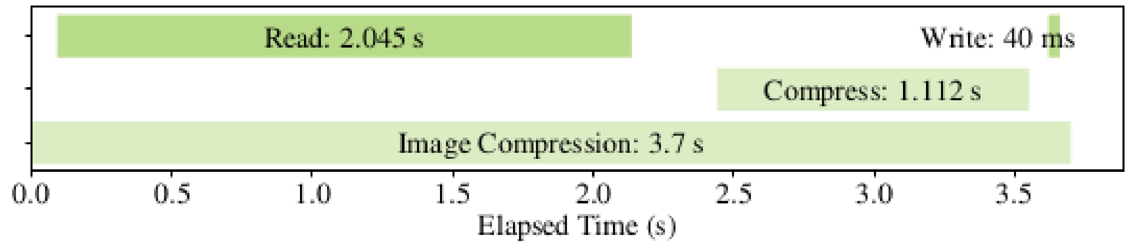
DFaaS: Function Fusion



(a) Image upload

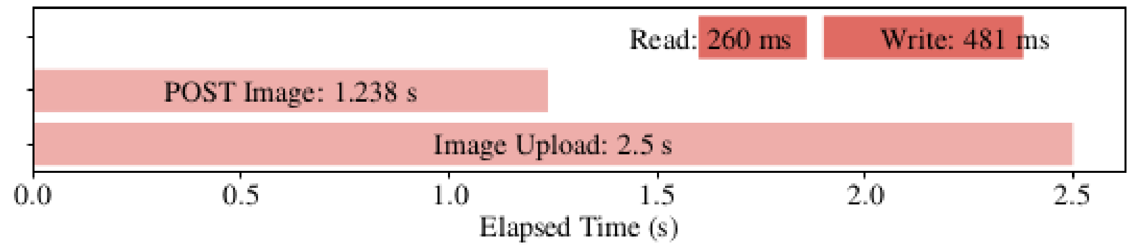


(b) Thumbnail generation

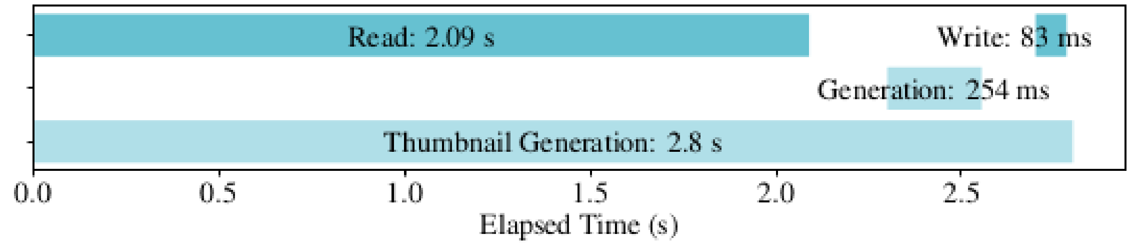


(c) Compression

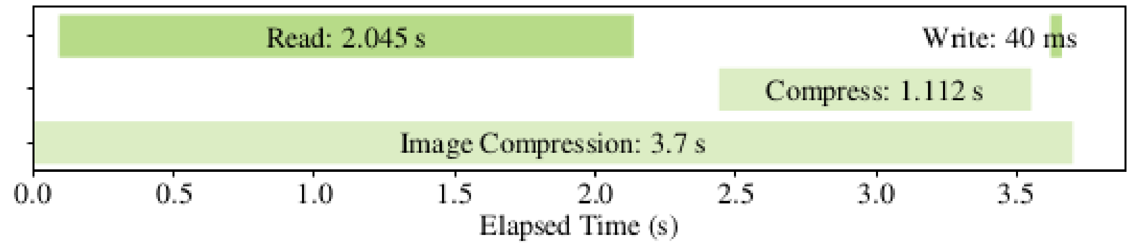
DFaaS: Function Fusion



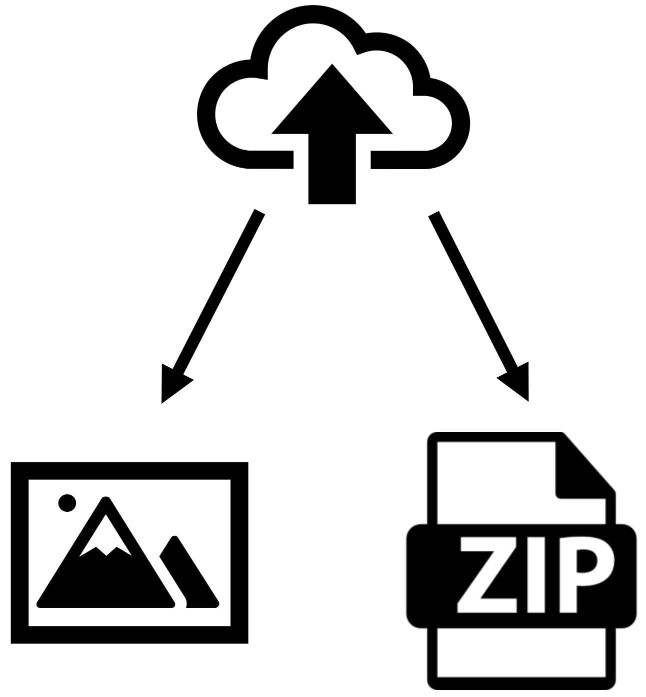
(a) Image upload



(b) Thumbnail generation

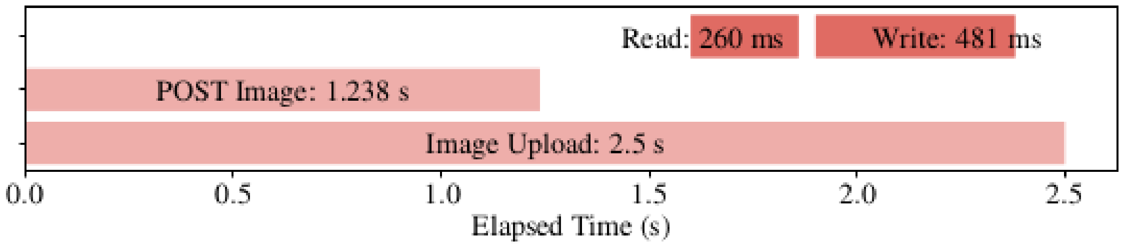


(c) Compression



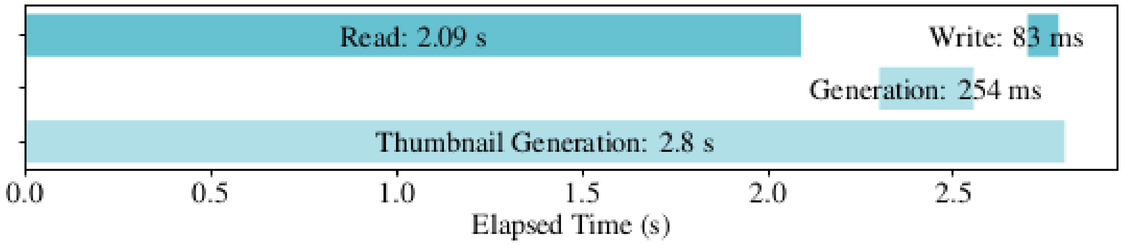
DFaaS: Function Fusion

COMPUTE



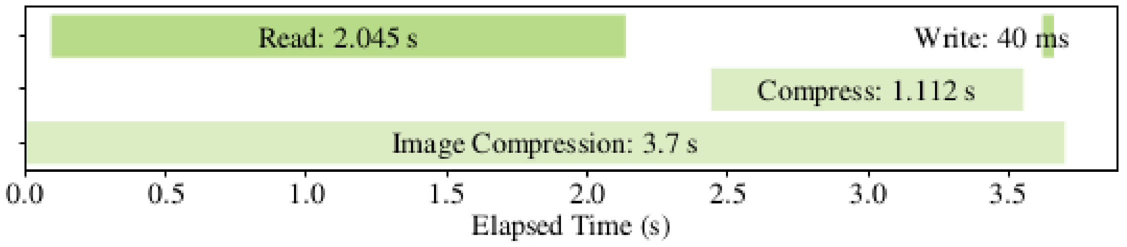
(a) Image upload

COMPUTE

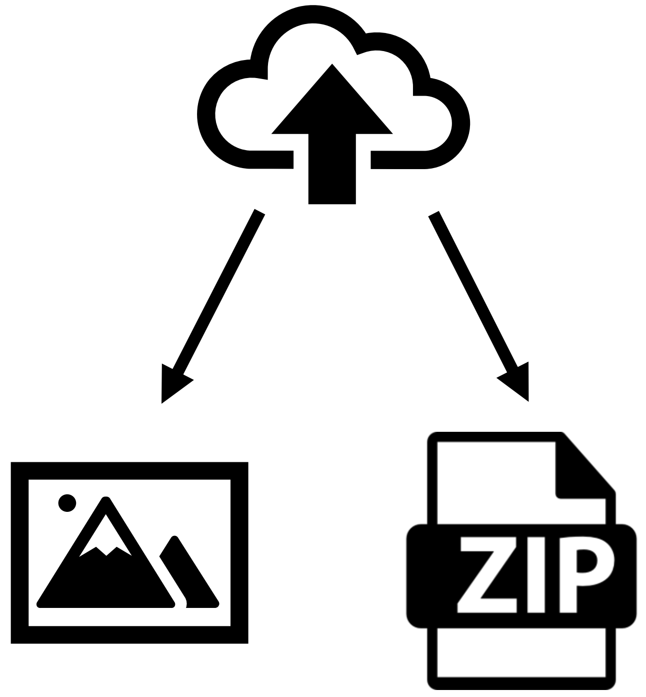


(b) Thumbnail generation

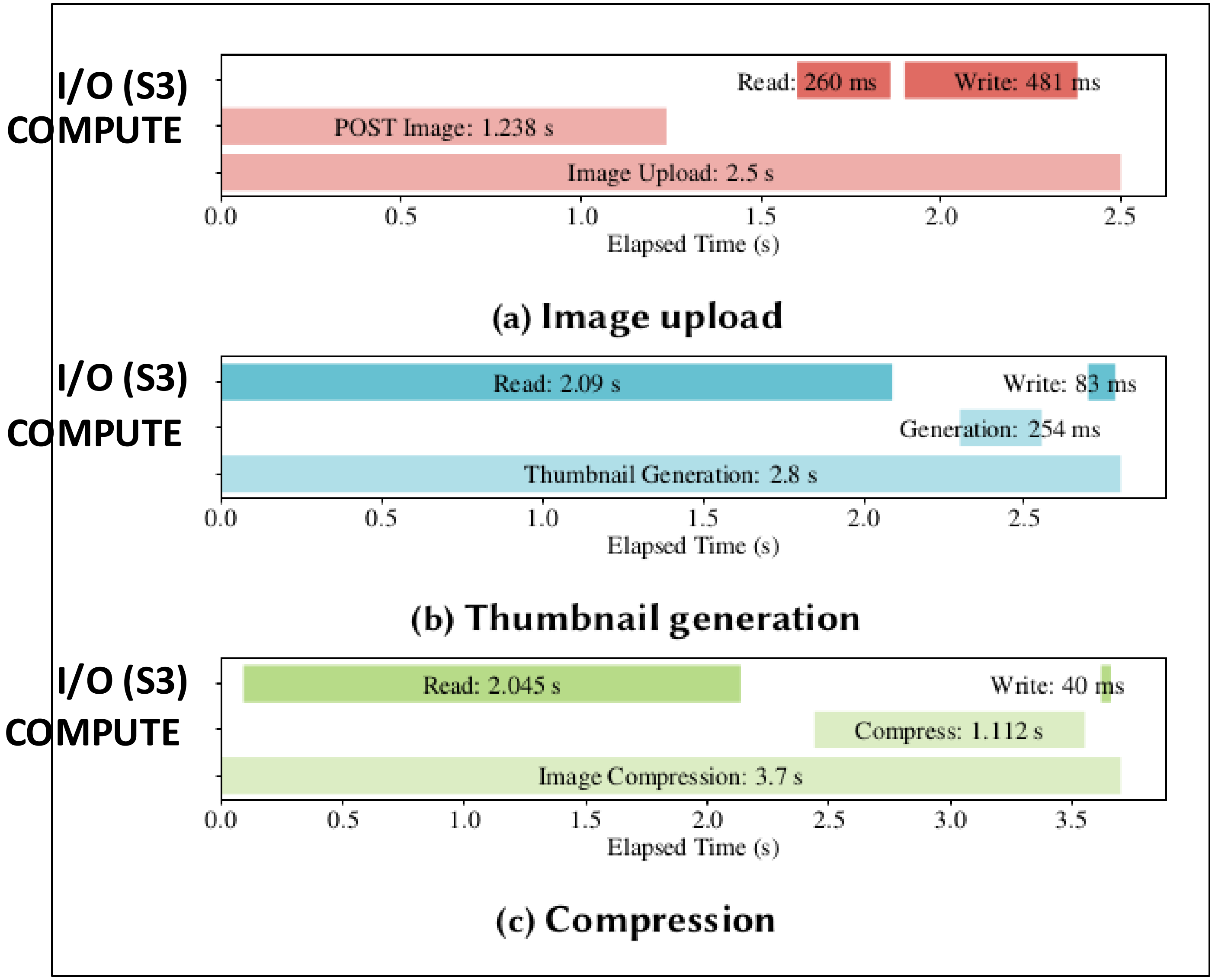
COMPUTE

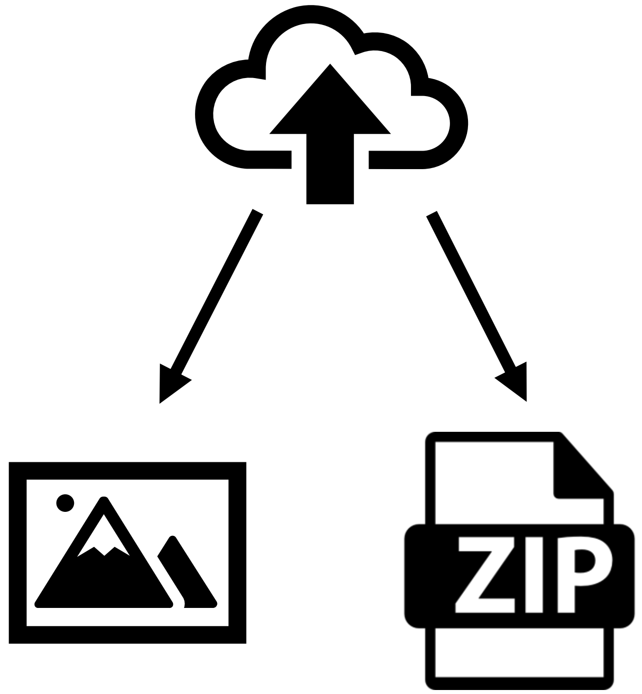


(c) Compression

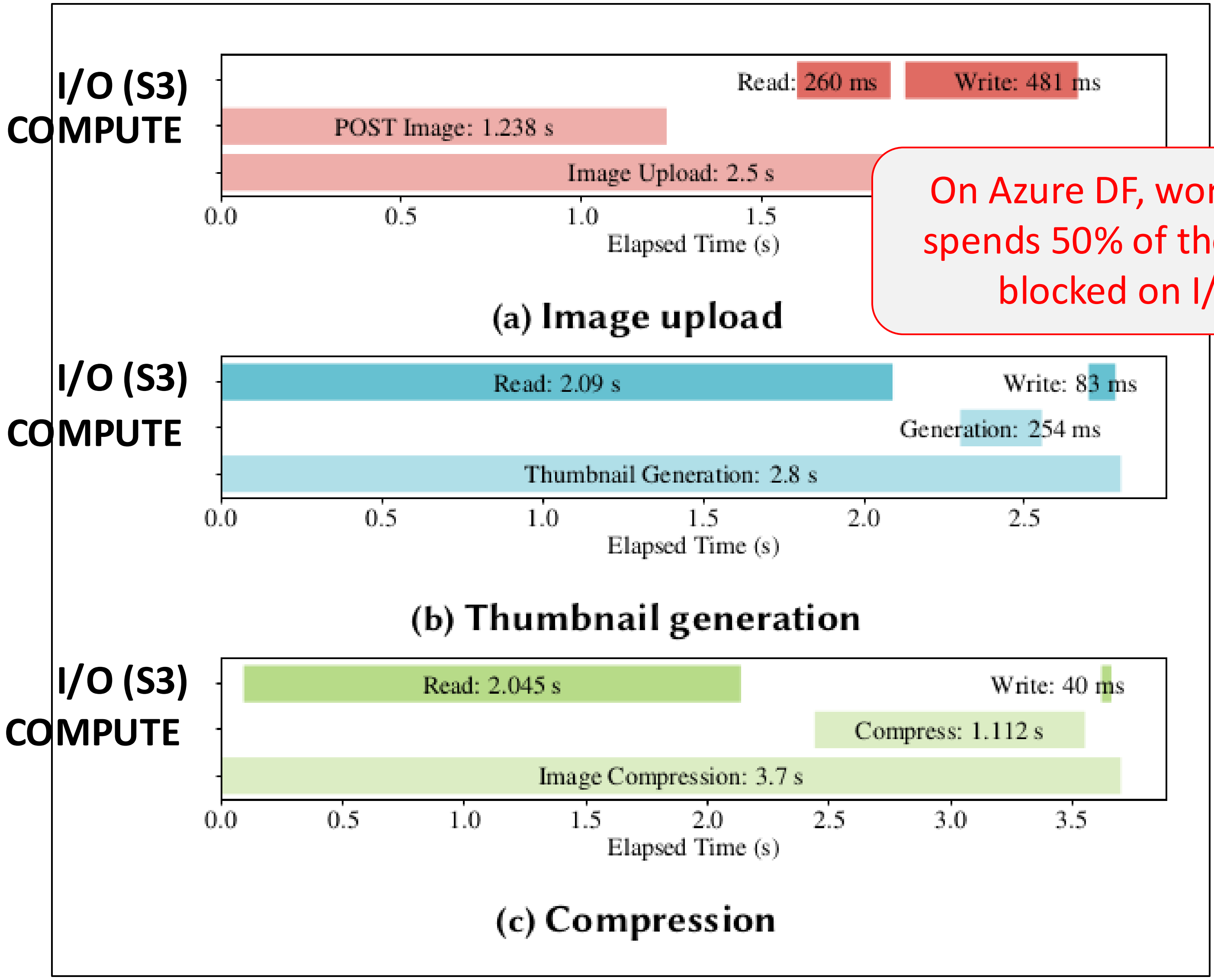


DFaaS: Function Fusion

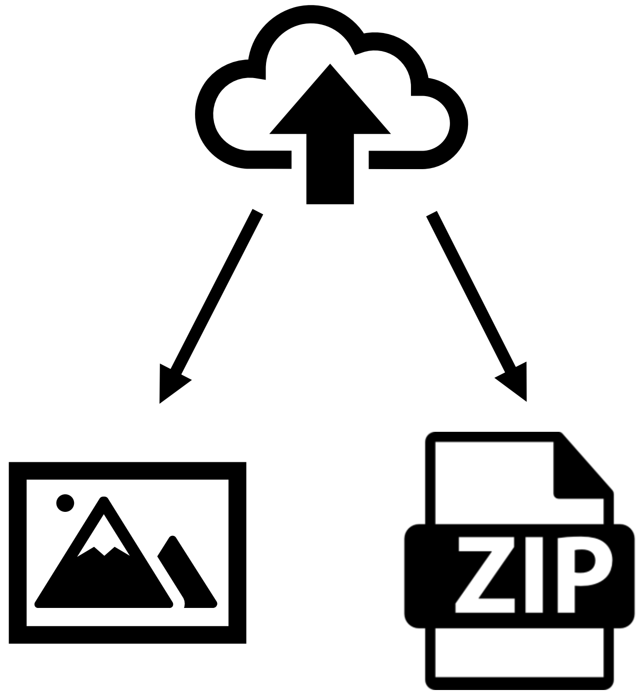




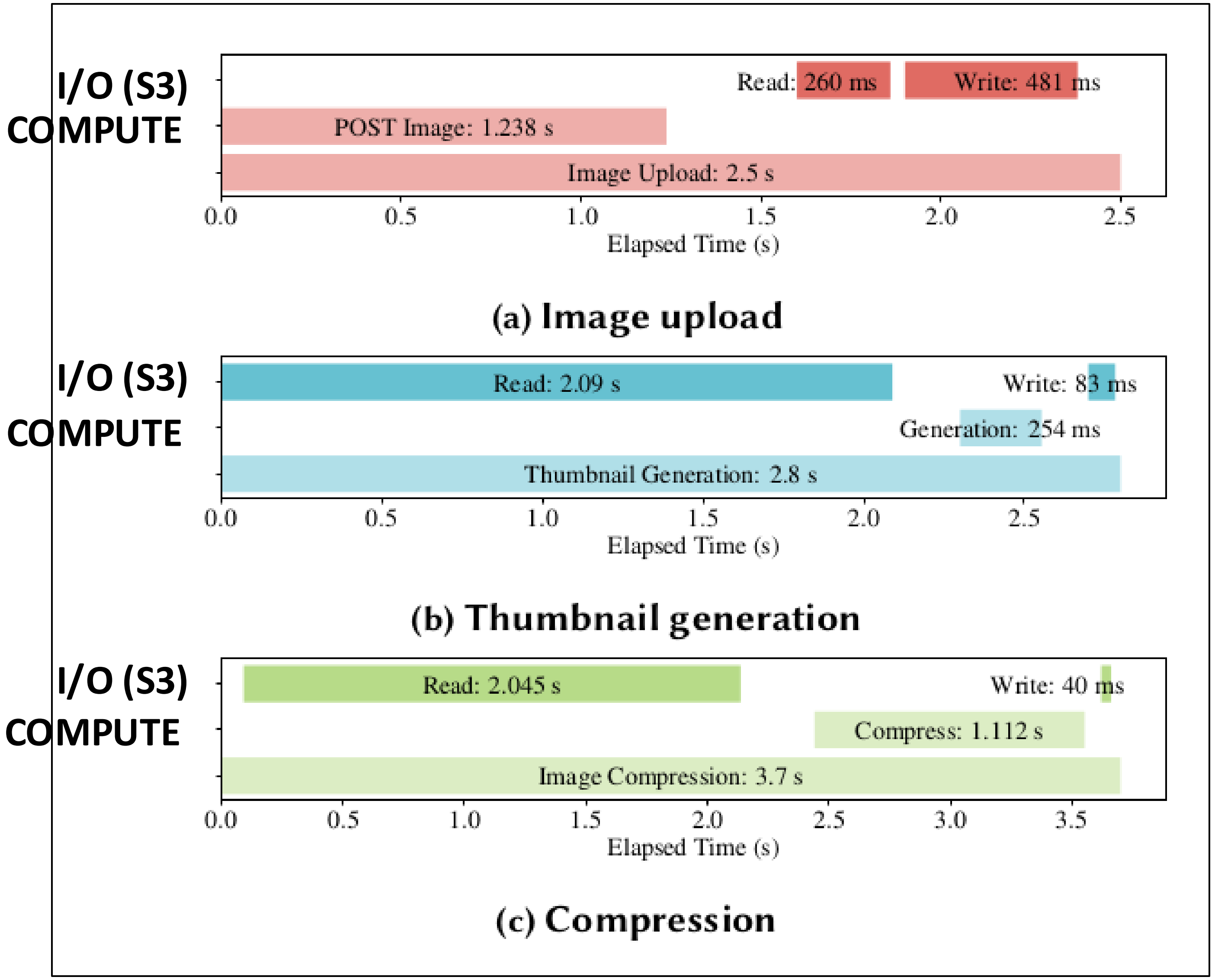
DFaaS: Function Fusion



On Azure DF, workflow spends 50% of the time blocked on I/O



DFaaS: Function Fusion

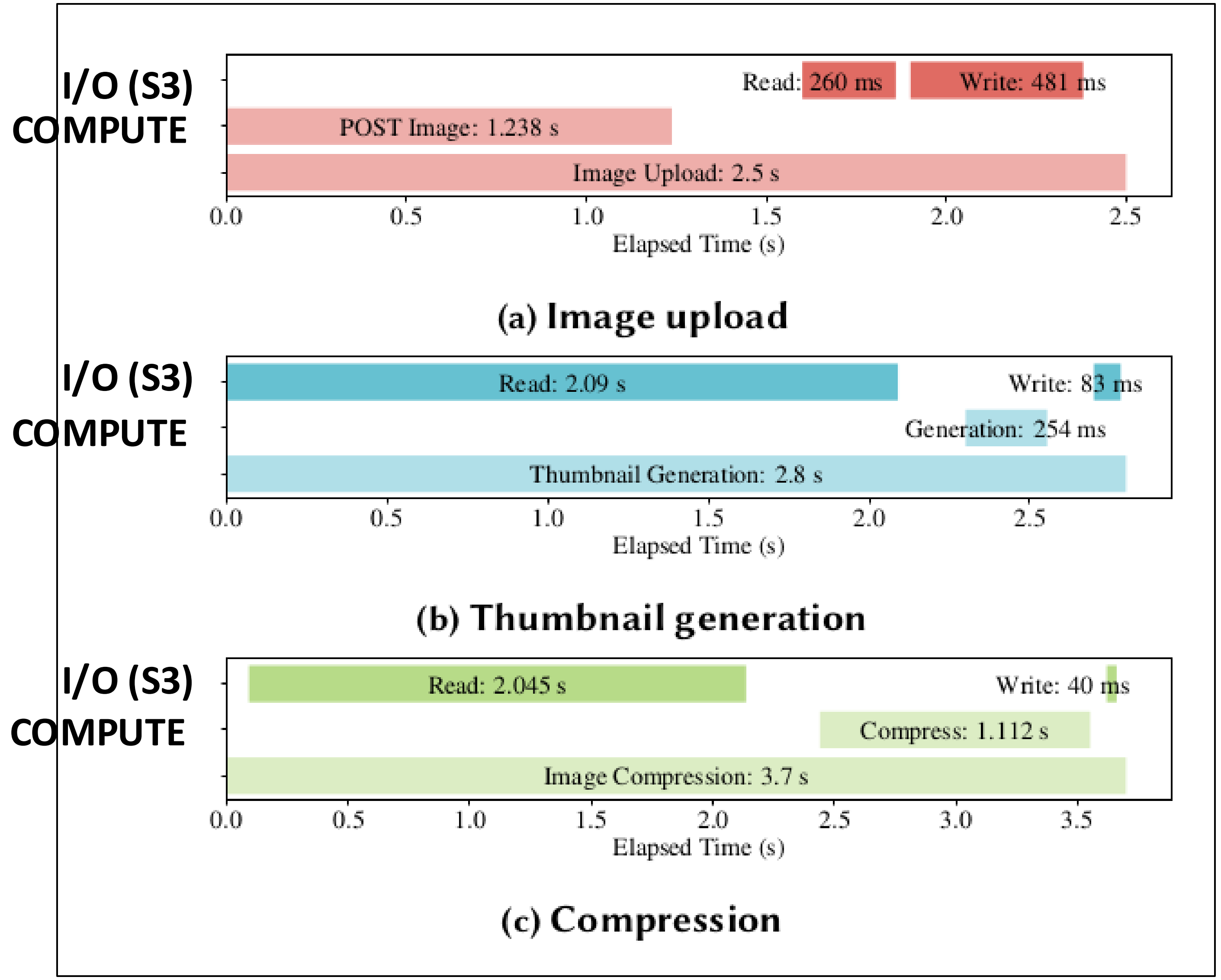


(a) Image upload

(b) Thumbnail generation

(c) Compression

DFaaS: Function Fusion

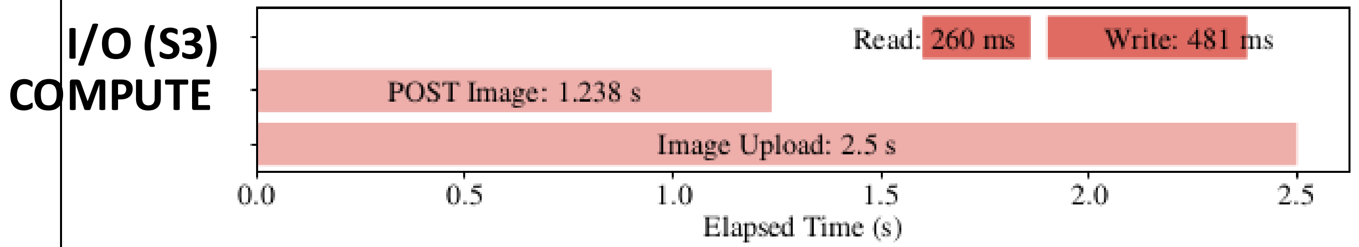


(a) Image upload

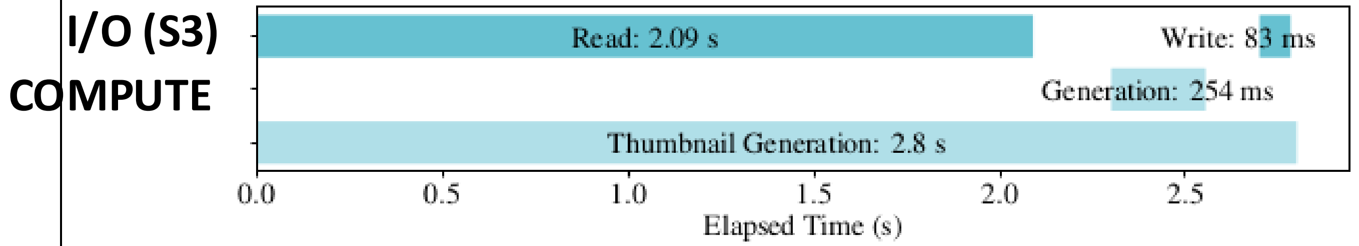
(b) Thumbnail generation

(c) Compression

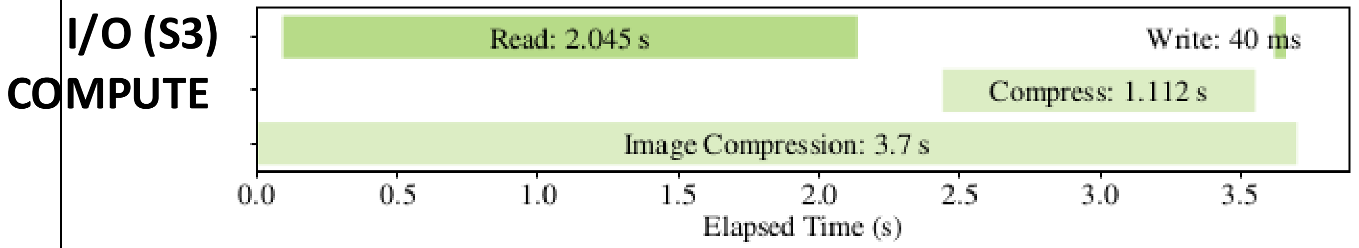
DFaaS: Function Fusion



(a) Image upload

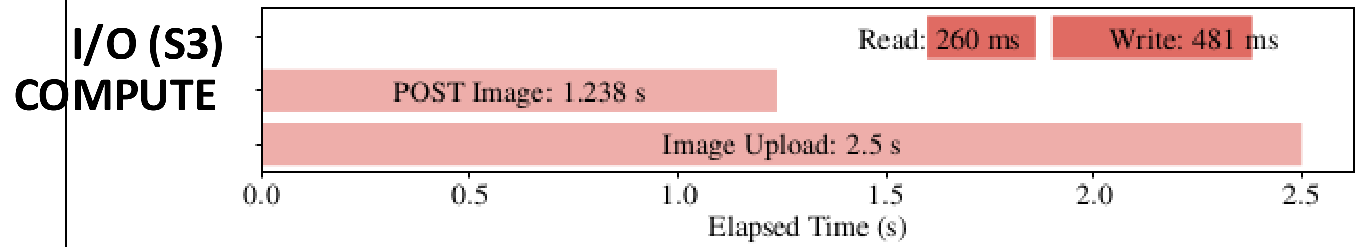


(b) Thumbnail generation

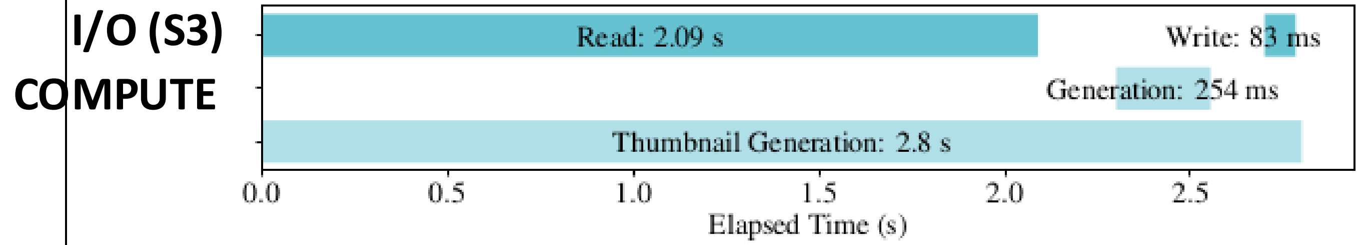


(c) Compression

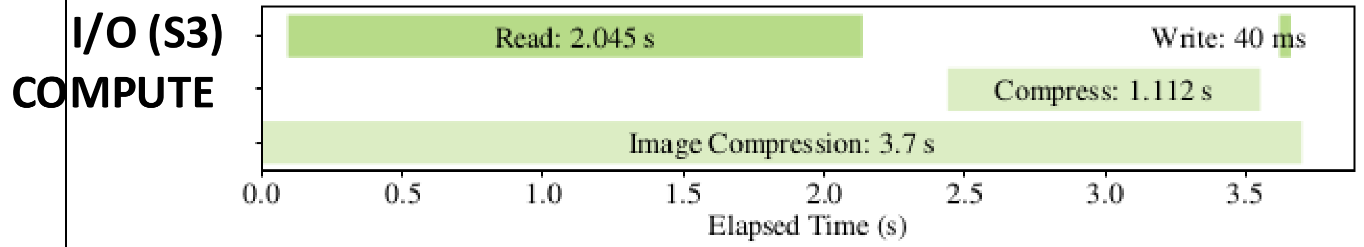
DFaaS: Function Fusion



(a) Image upload

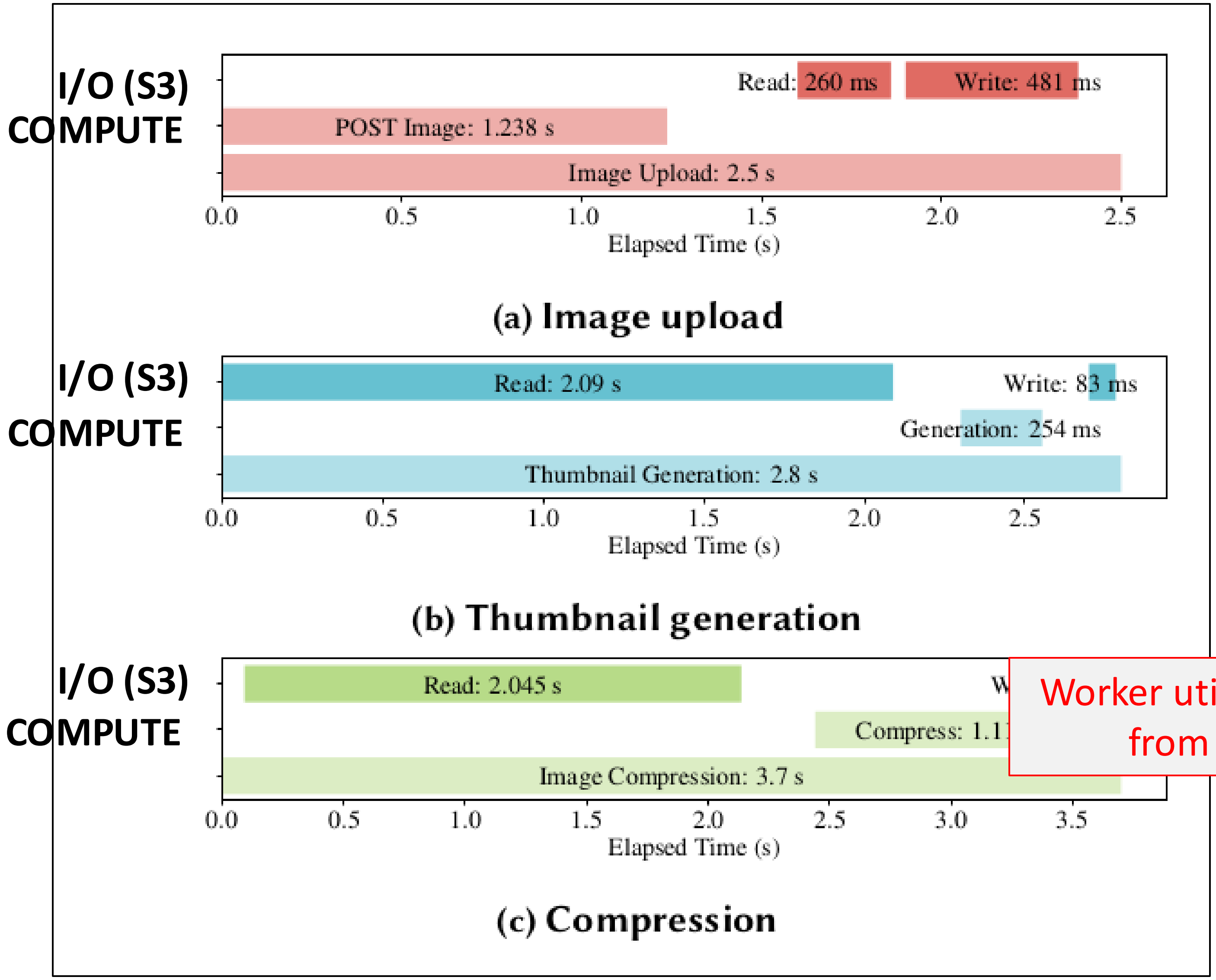


(b) Thumbnail generation



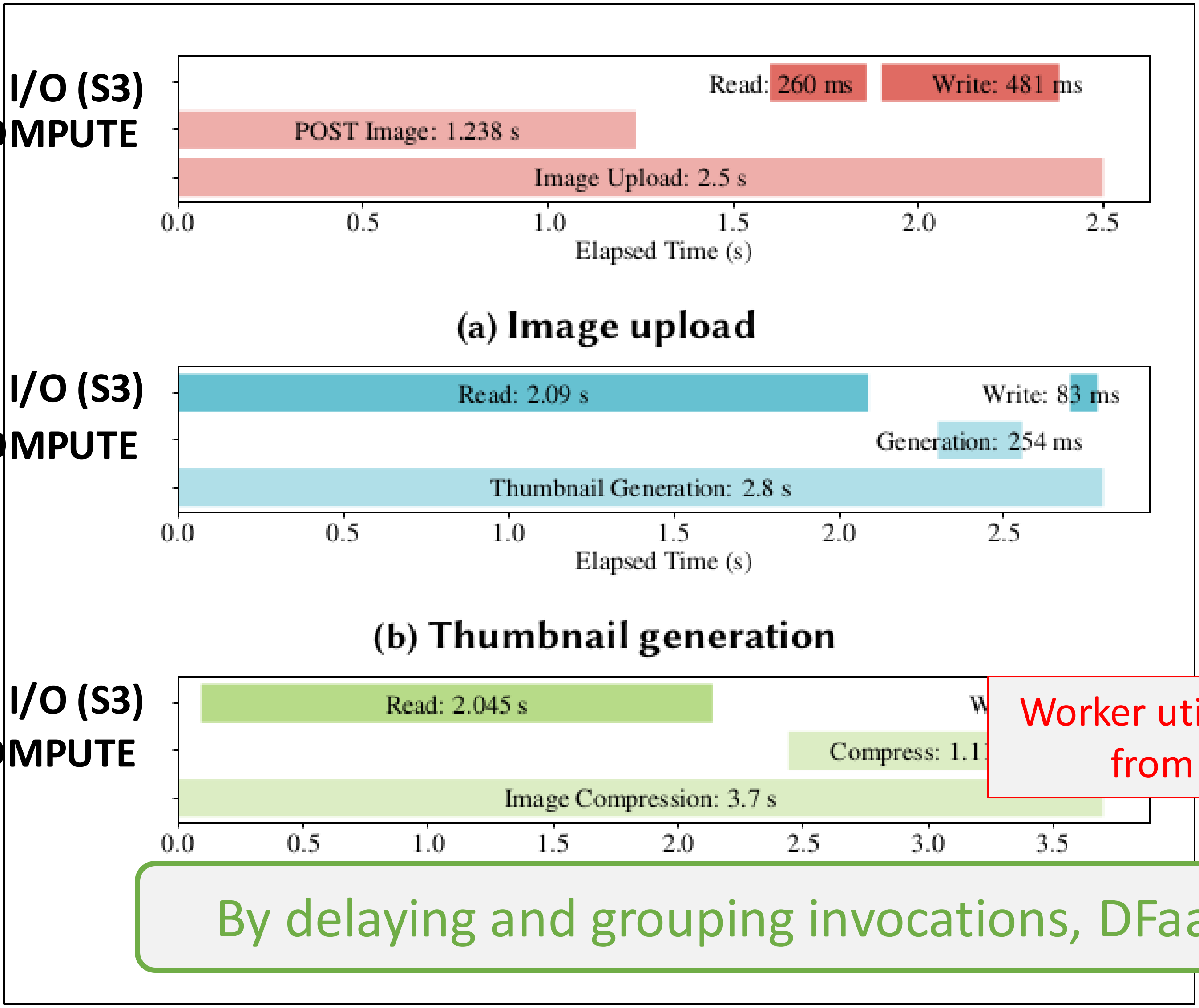
(c) Compression

DFaaS: Function Fusion



Worker utilization increases from 50% to 89%

DFaaS: Function Fusion



Worker utilization increases from 50% to 89%

By delaying and grouping invocations, DFaaS can improve **resource utilization**

Is It Time To Put Cold Starts In The Deep Freeze?

Cold start optimizations have dominated serverless research for the past decade

Carlos Segarra (Imperial College London)

carlossegarra.com <cs1620@ic.ac.uk>

Is It Time To Put Cold Starts In The Deep Freeze?

Cold start optimizations have dominated serverless research for the past decade

Much serverless usage today is for **non-interactive, latency insensitive** workloads

Carlos Segarra (Imperial College London)

carlossegarra.com <cs1620@ic.ac.uk>

Is It Time To Put Cold Starts In The Deep Freeze?

Cold start optimizations have dominated serverless research for the past decade

Much serverless usage today is for **non-interactive, latency insensitive** workloads

We believe these workloads **are a good fit for a serverless model**, and we should optimize for them

Carlos Segarra (Imperial College London)

carlossegarra.com <cs1620@ic.ac.uk>

Is It Time To Put Cold Starts In The Deep Freeze?

Cold start optimizations have dominated serverless research for the past decade

Much serverless usage today is for **non-interactive, latency insensitive** workloads

We believe these workloads **are a good fit for a serverless model**, and we should optimize for them

DFaaS

delay tolerance

Carlos Segarra (Imperial College London)

carlossegarra.com <cs1620@ic.ac.uk>

Is It Time To Put Cold Starts In The Deep Freeze?

Cold start optimizations have dominated serverless research for the past decade

Much serverless usage today is for **non-interactive, latency insensitive** workloads

We believe these workloads **are a good fit for a serverless model**, and we should optimize for them

DFaaS

delay tolerance

Improve **resource utilization** and reduce cost

Carlos Segarra (Imperial College London)

carlossegarra.com <cs1620@ic.ac.uk>

Is It Time To Put Cold Starts In The Deep Freeze?

Cold start optimizations have dominated serverless research for the past decade

Much serverless usage today is for **non-interactive, latency insensitive** workloads

We believe these workloads **are a good fit for a serverless model**, and we should optimize for them

DFaaS

delay tolerance

Improve **resource utilization** and reduce cost

Future: improve **throughput** with transparent access to accelerators

Carlos Segarra (Imperial College London)

carlossegarra.com <cs1620@ic.ac.uk>