Serverless Cloud Engineering and On-Device Computation for Complex Machine Learning Workloads: Fast and Furious for your Hardest Data Analytics Tasks

Somali Chaterji, Purdue University; https://schaterji.io
Assistant Professor, Purdue University;
CEO, KeyByte

Innovatory for Cells and Neural Machines (ICAN)

PURDUE UNIVERSITY

KeyByte

About Me

- CEO and co-founder of KeyByte, a blazing fast cloud computing company
- Assistant Professor, specializing in data engineering and applied ML, at Purdue
- Training in Computational Genomics (BME) and Computer Science
- Lead the Innovatory for Cells and Neural Machines (ICAN) at Purdue
  - ICAN innovates at the nexus of computer vision and mobile systems [Thrust 1], on one hand, and at the interface of machine learning and genomics [Thrust 2], on the other.
- Funding from NIH (R01), DOD (ARL), NSF (CISE), USDA, as well as private industries like Amazon, Microsoft, and Adobe Research.
- Won the NSF-CAREER award from CISE on streaming analytics for IoT and computer vision in January 2022, which is shaping up its cyber nook here: https://schaterji.io/projects/sirius.html [mobile computer vision, serverless, drone analytics]

Sirius (NSF-CAREER): I am hiring [Undergrads, Grads, Postdocs, Software Engineers, Interns]

March 2022

Taming the wild: Streaming — and streamlining — analytics from the Internet of Things

Affiliations

- *Assured Autonomy Innovation Institute (A2I2)*
- *KeyByte (keybyte.xyz)*
- *ICAN Data Engineering Fellow*
- *Purdue’s College of Engineering (Rank #4)*
- *Purdue’s ABE (Rank #1), CoE and CoA*
- *Purdue’s ECE (Rank #9), CoE*
- *WHIN Leadership (Lilly Endowment)*

Projects (https://schaterji.io/projects/sirius.html)

- *Computer vision*
- *Serverless*
- *Drones*
- *Computational genomics (Thrust 2)*

WiseFuse for Serverless

I will tell you about **WiseFuse** [*Sigmetrics 2022*], which performs end-to-end optimization of serverless DAG workflows, driven by our analysis of real serverless cloud computing workloads from Microsoft Azure. Concretely, our work introduces two optimizations: horizontal colocation or *bundling* of parallel invocations of a function and vertical *fusion* of in-series functions, while rightsizing the VMs hosting these functions.

WiseFuse: Workload Characterization and DAG Transformation for Serverless DAG Workflows

Sigmetrics 2022

Best Paper Award

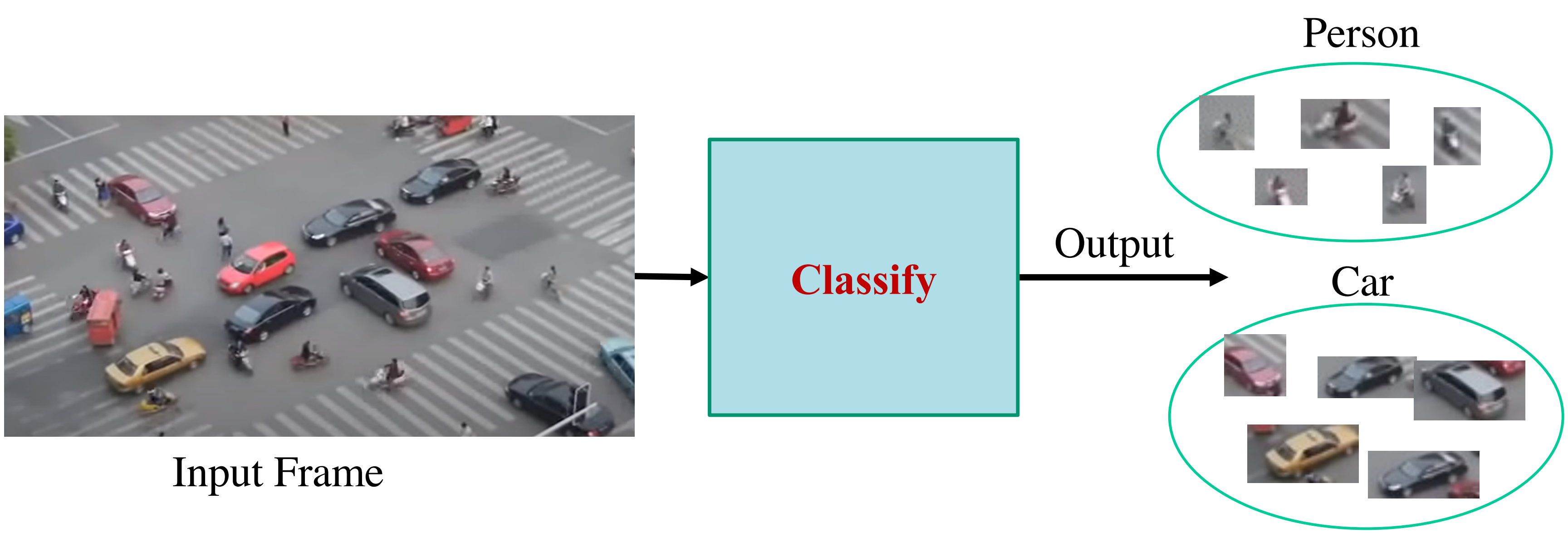
Purdue University

Carnegie Mellon University

Microsoft

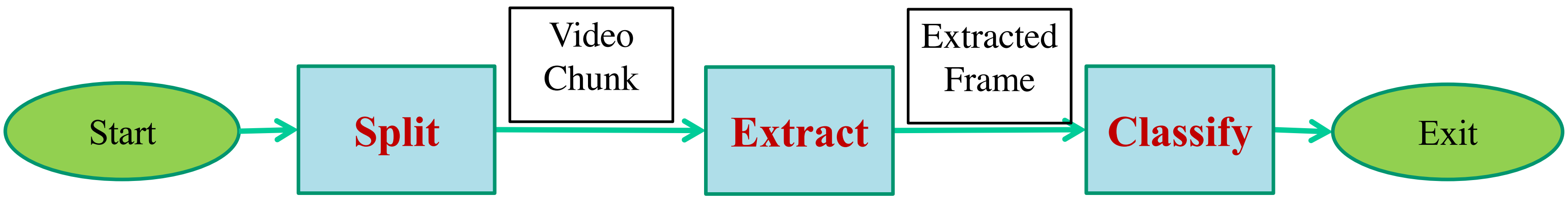
Introduction: Serverless Computing

- Attractive model:
  - Users write the code, and platform deploys and executes the function
  - *Pay-as-you-go* model



Introduction: Serverless DAG

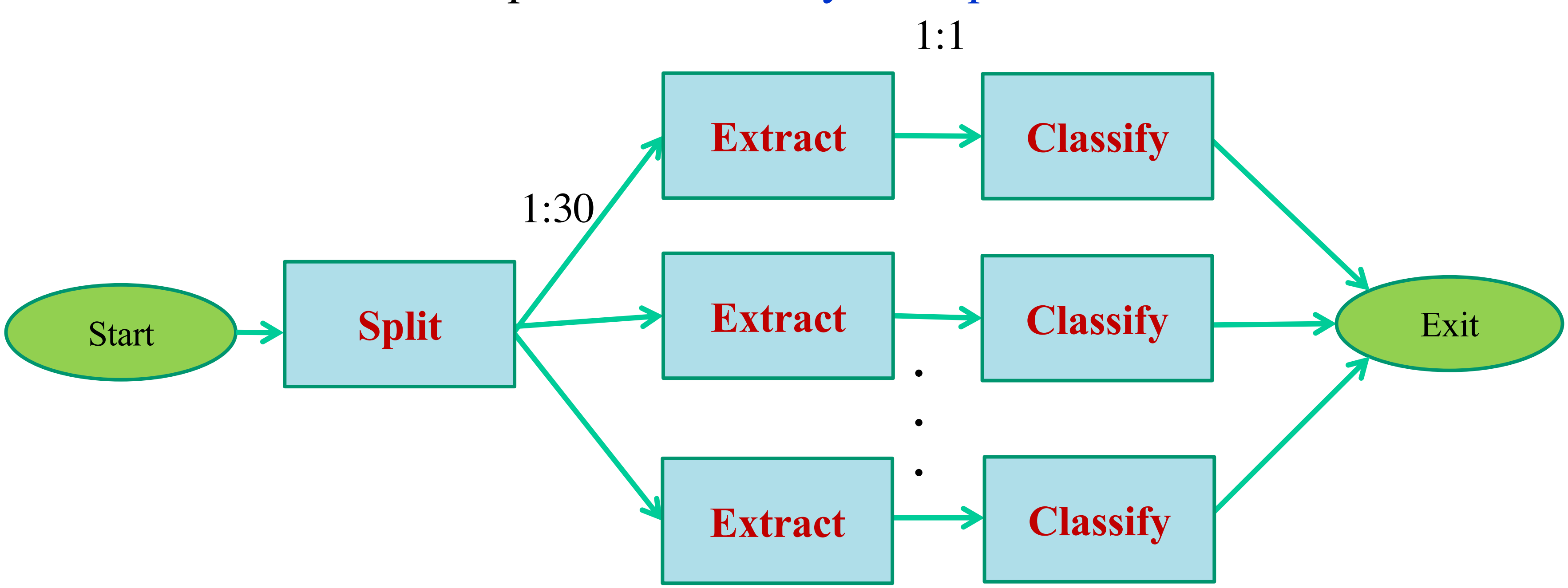
Serverless Chain Example: a sequence of functions executed as in-series functions



- One function's output becomes the input to next function

Introduction: Serverless DAG

- Serverless DAG Example: Video Analytics Pipeline

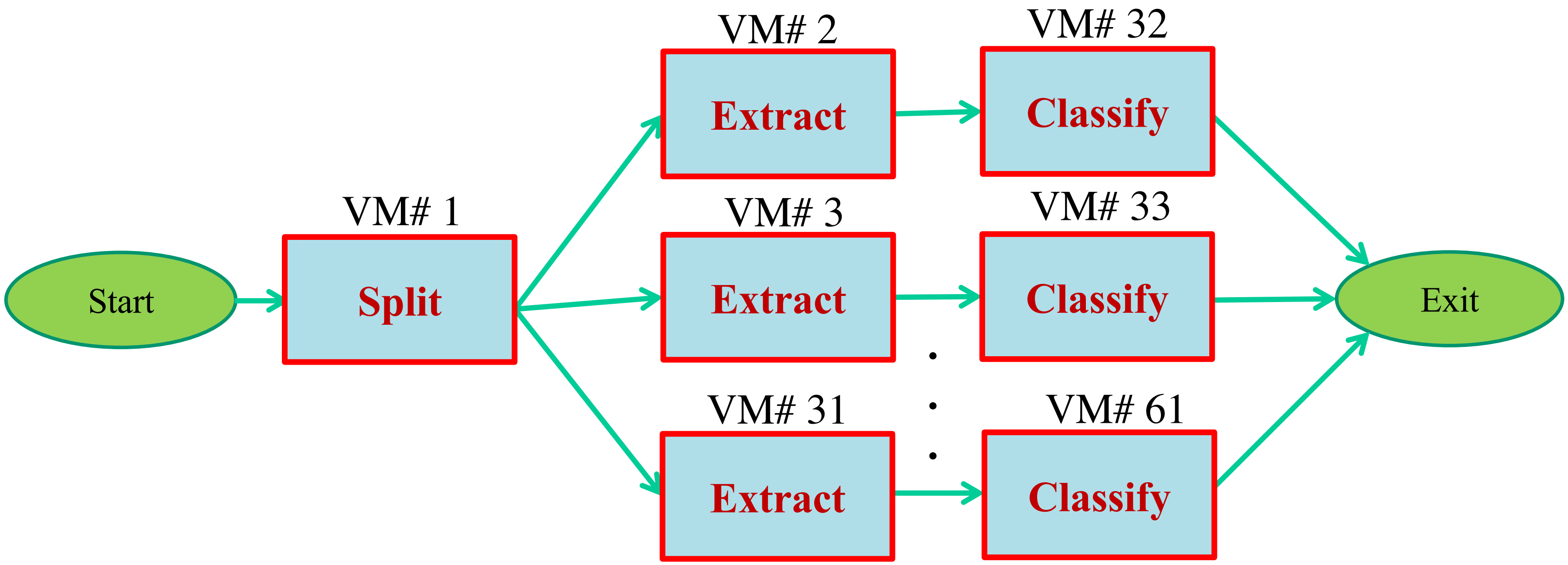


❑ DAG latency: elapsed time from start to end (after *all* functions finish execution)

PURDUE UNIVERSITY

Introduction: Serverless DAG Execution

Example: Video Analytics Pipeline



- ❑ Current FaaS platforms execute each function in a *separate* VM (*total of 61 VMs*)
- ❑ Users need to specify the VM size for each function

PURDUE UNIVERSITY

Problems: Performance Bottlenecks

1. Communication latency between in-series functions

Extracted Frame

Extract → Classify

2. Computation skew among in-parallel invocations within the same stage

$Classify_1$ — Short Execution Time

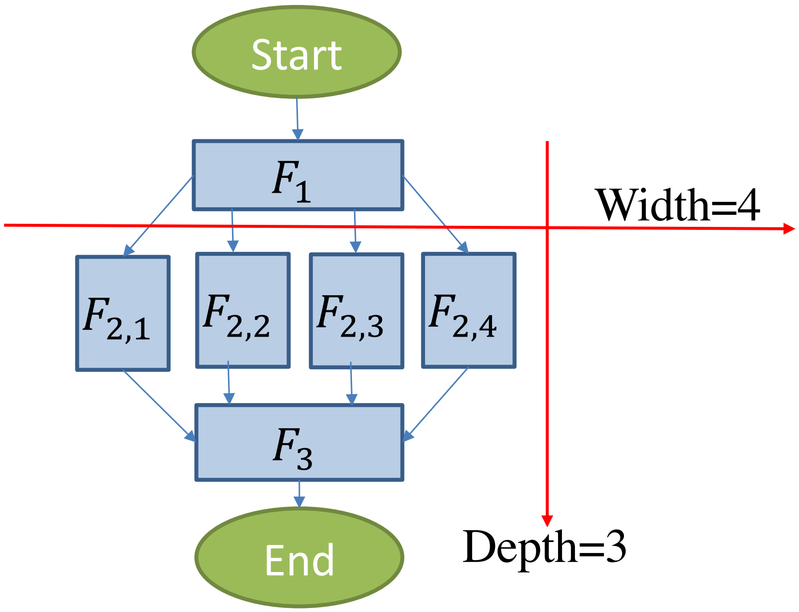
$Classify_2$ — Long Execution Time

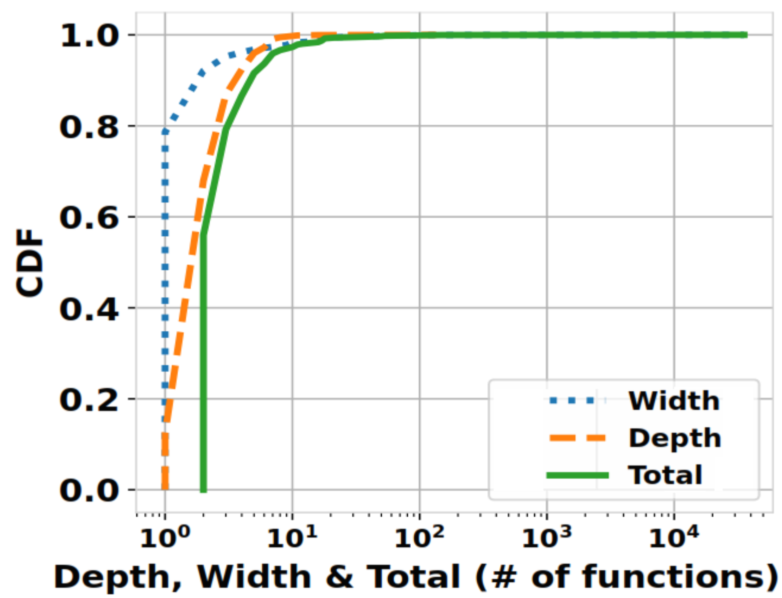
Next

Workload Characterization from Microsoft Azure

Workload Characterization (1/3): DAG Structure

Real workload traces from Azure Durable Functions

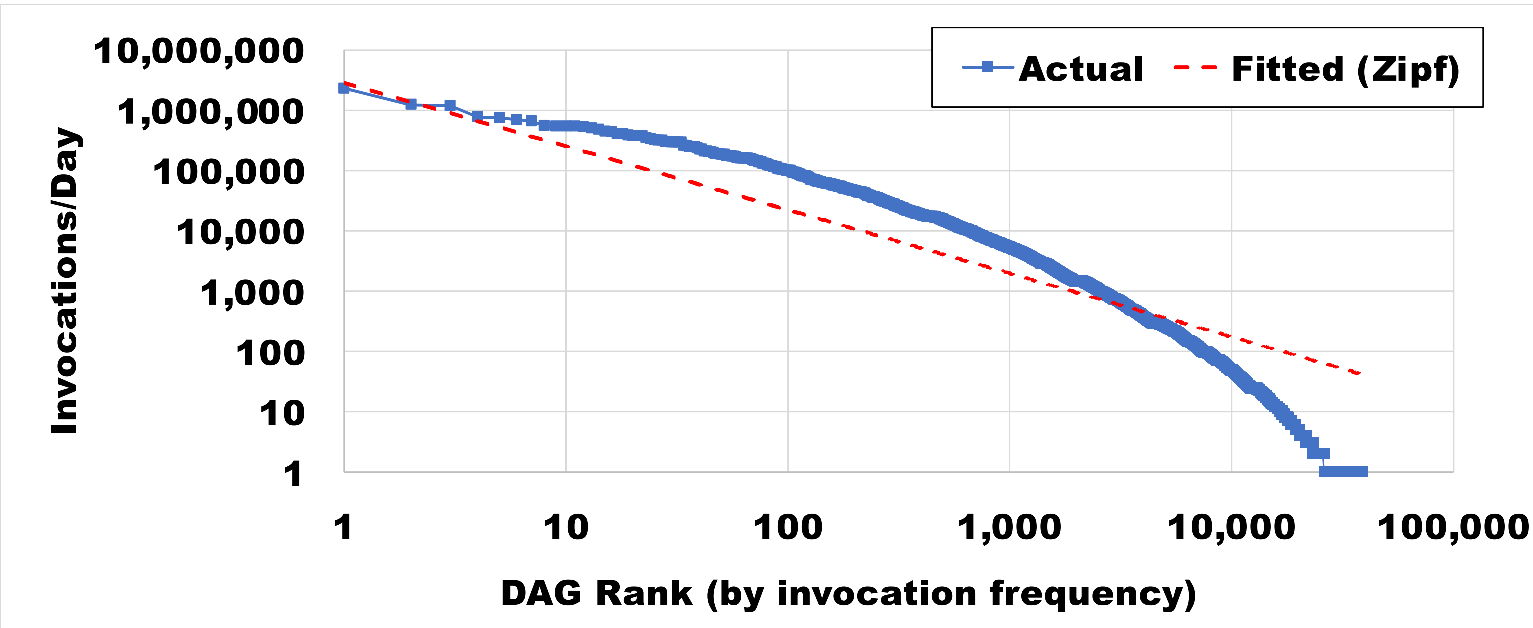




- DAG structure: *wide* and *shallow*
  - Max Width: 10.9K in-parallel invocations
  - Max Depth: 47 in-series stages

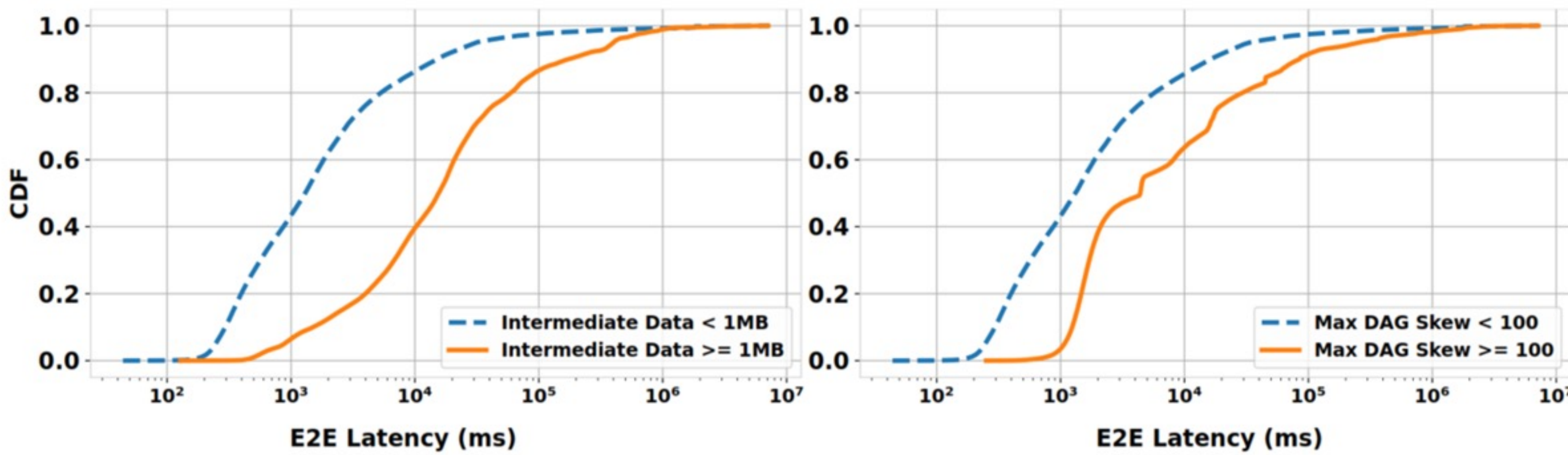
PURDUE UNIVERSITY

Workload Characterization (2/3): DAG frequency



- Top 5% most frequent DAGs:
  - Constitute 95% of all DAG invocations
  - Invocation rate ≥ 1.6K/day

Workload Characterization (3/3): Intermediate Data & Skew Impact

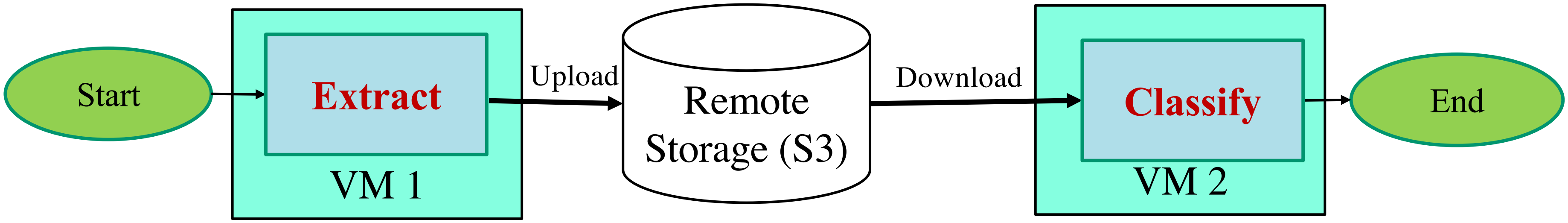


- DAGs with intermediate data size ≥ 1MB have 9.5× higher median latency than DAGs with size < 1MB.
- DAGs with skew ≥ 100 have 17× higher median latency than DAGs with skew < 100

PURDUE UNIVERSITY

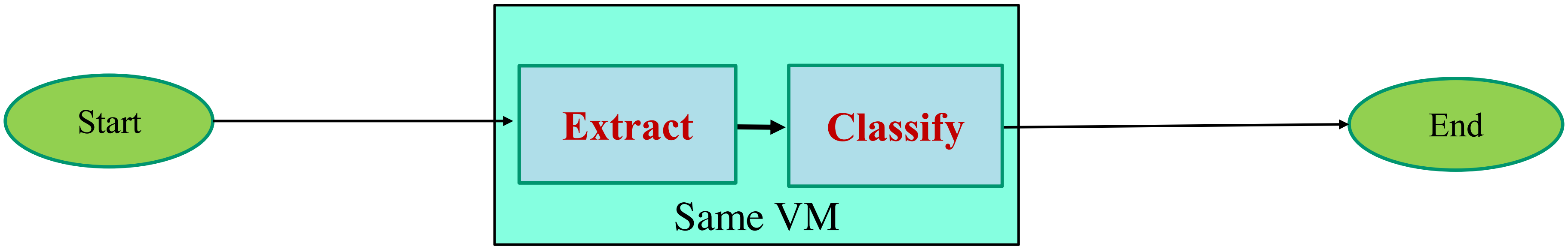
WISEFUSE *Design*

Fusion (1/2)



- ❑ Direct communication between serverless functions is infeasible
- ❑ Accordingly, asynchronous communication through remote storage is used
  - ❑ Can add significant delay to the DAG latency

Fusion (2/2)

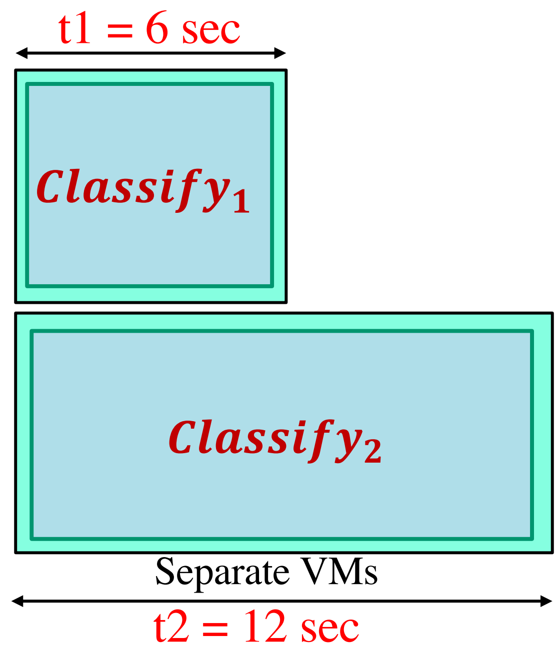


Our solution: **Fusion**

– We can execute the sending and receiving functions in one VM and leverage local data passing → reduce DAG latency

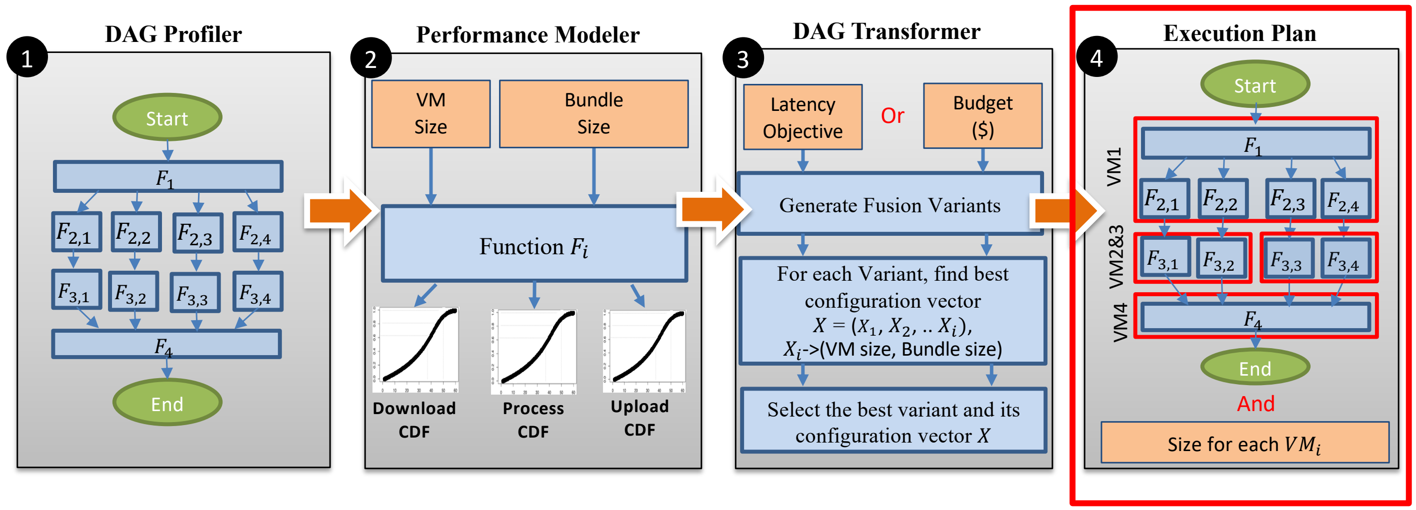
- Challenges:
  - Which functions to fuse?
  - Fusion increases cost if the functions have different resource requirements

Bundling



- ❑ Each invocation executes in a separate VM
- ❑ Straggler dominates the end-to-end latency

WISEFUSE'S *Overall Design*



- Optimizer uses fusion and bundling to generate the DAG execution plan
- *Execution Plan* describes:
  1. Which stages to be Fused together
  2. How many parallel invocations within a stage to be bundled together
  3. The VM size to allocate for each function or function bundle

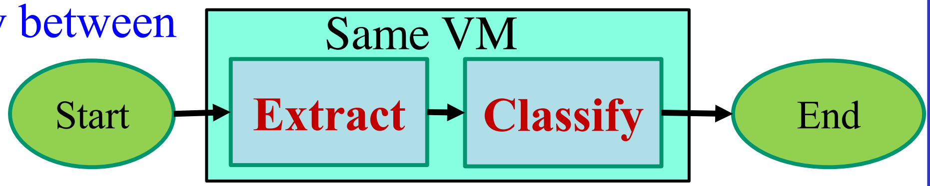
PURDUE UNIVERSITY

Summary of Main Insights

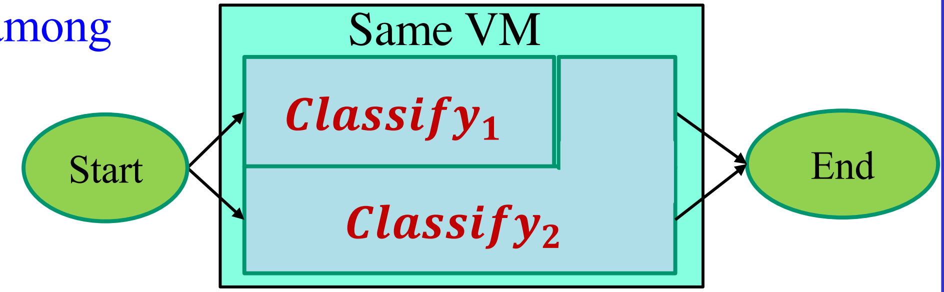
- Workload Characterization:
  - Top 5% most frequent DAGs constitute 95% of all DAG invocations
  - Serverless DAGs are short but wide

- Two important optimizations:

  (1) Fusion: Reduces communication latency between in-series functions



  (2) Bundling: Reduces computation skew among in-parallel invocations



PURDUE UNIVERSITY

Evaluation

Evaluation

- We evaluate WISEFUSE on three applications on AWS Lambda
  - ❑ Video Analytics
  - ❑ Approximate SVD
  - ❑ ML Pipeline

- Profiling is *fast* and *cheap* (using 300 profiling runs):
  - ❑ Error in P95 E2E latency: ≤ **13%**
  - ❑ Error in estimating the impact of Fusion or Bundling ≤ **7%**

- Recall that **95% of all invocations** are for **the top 5% most frequent DAGs**
  - ❑ Invocation rate ≥ 1.6 K per day

Evaluation: Comparison to Baselines and Related Works

We evaluate the following approaches on AWS Lambda:

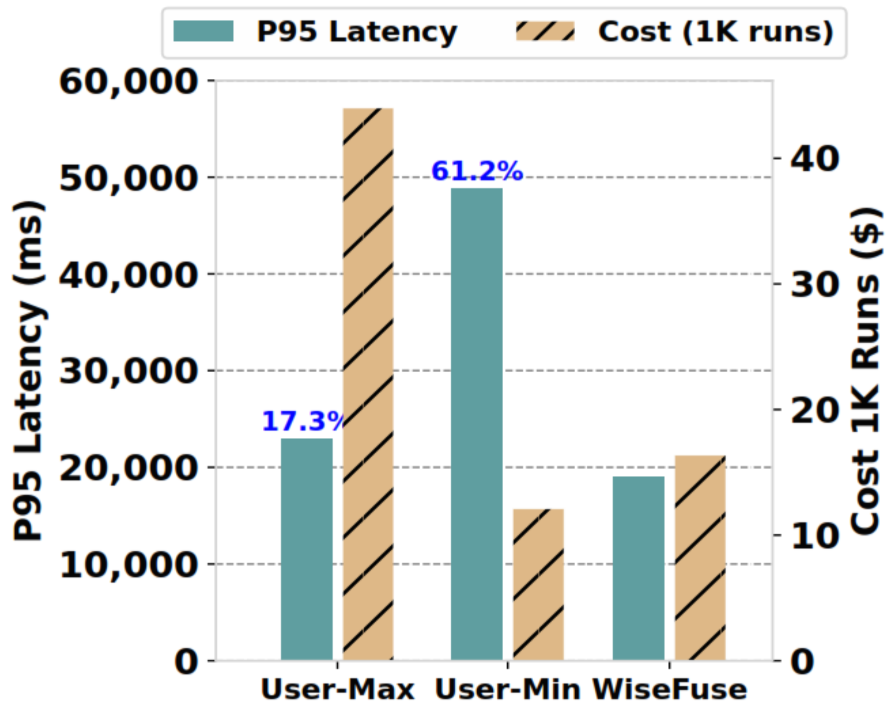
1. Baselines:
   1. User-Max: user-provided DAG using maximum VM sizes (lowest latency)
   2. User-Min: user-provided DAG using minimum VM sizes (lowest cost)

2. Related works: SONIC (ATC'21), Photons (SoCC'20), and FaastLane (ATC'21)

3. Three latency target settings for WISEFUSE (1.5X, 2.5X, 5X of the best theoretical latency)

PURDUE UNIVERSITY

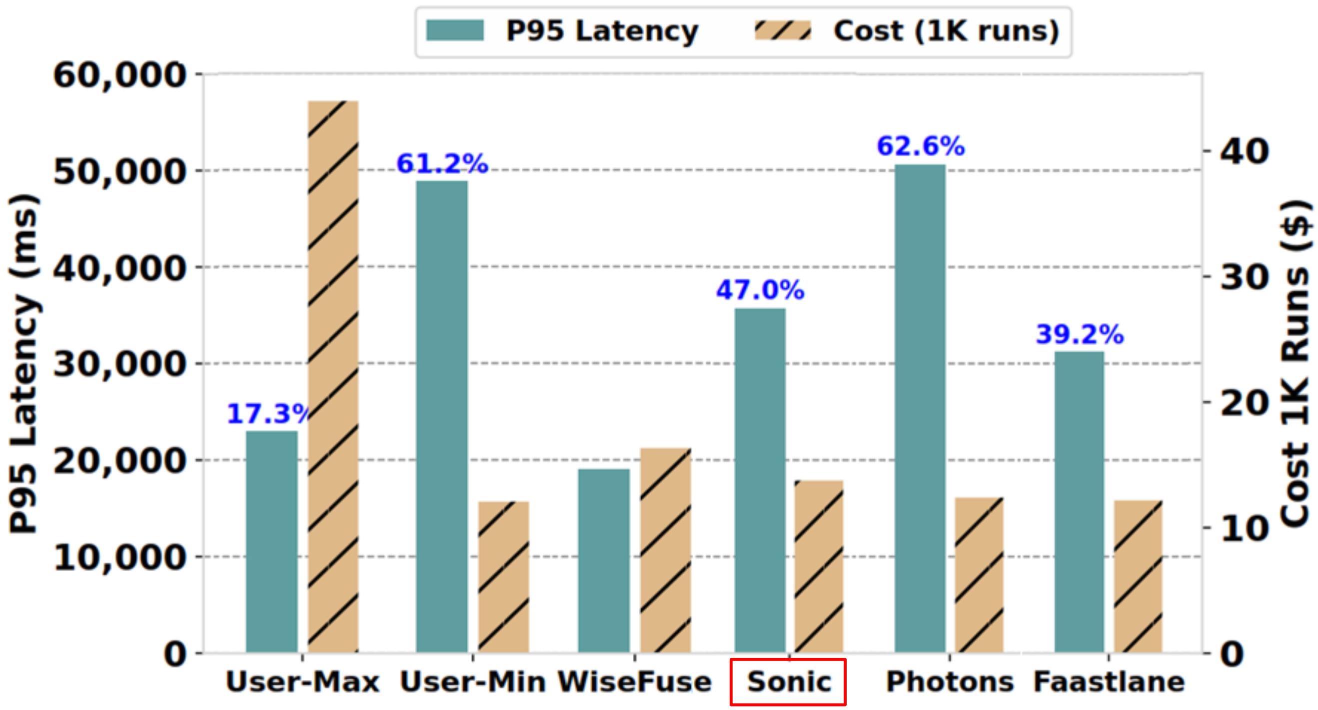
Evaluation with Video Analytics Application (1/5)



% over the bars show WISEFUSE's gains in E2E latency

WISEFUSE achieves 63% lower cost than User-Max and 61% lower P95 latency than User-Min

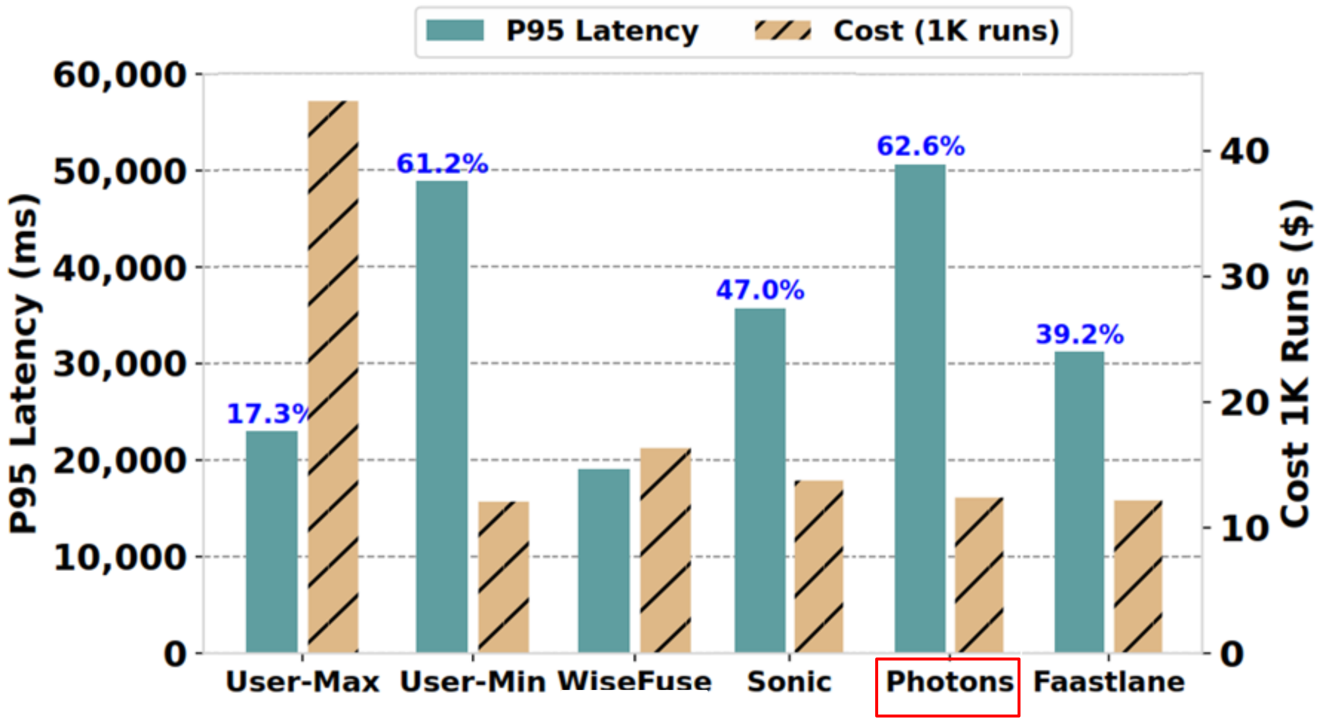
Evaluation with Video Analytics Application (2/5)



SONIC (ATC'21): considers fusing in-series functions only to leverage data locality. It does not perform bundling and does not consider the latency distribution

PURDUE UNIVERSITY

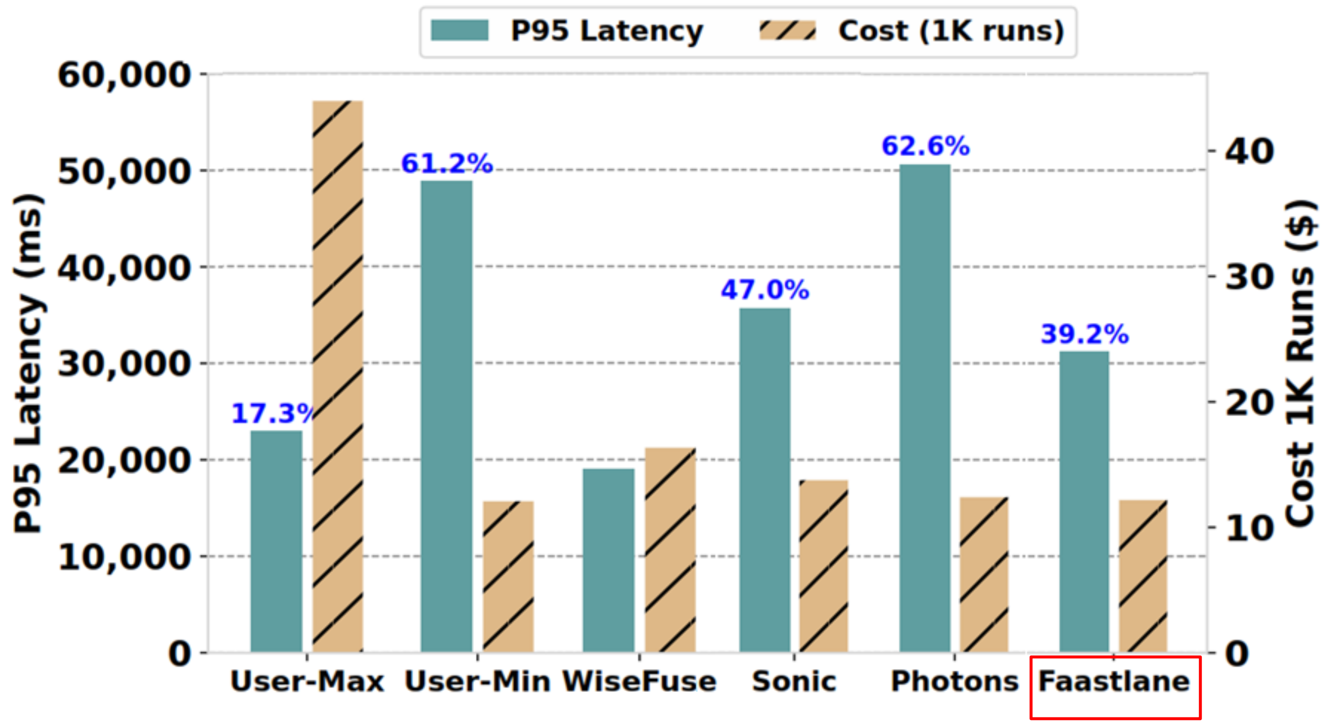
Evaluation with Video Analytics Application (3/5)



Photons (SoCC'20): performs Bundling mainly to improve memory utilization. It bundles as many parallel invocations as possible based on the functions' memory footprint.

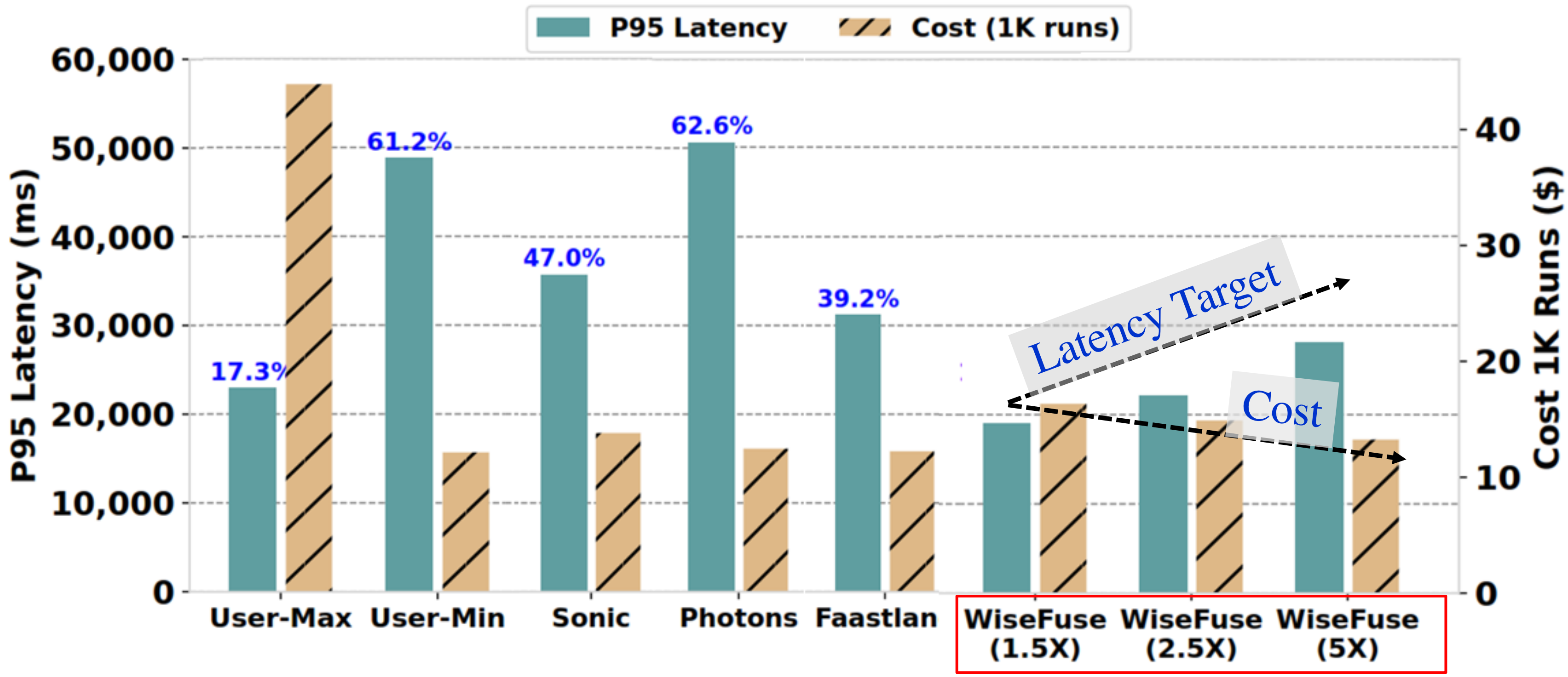
PURDUE UNIVERSITY

Evaluation with Video Analytics Application (4/5)



- Faastlane (ATC'21): uses a fixed bundle size of 6 workers (to match the 6 vCPUs that are provided by AWS Lambda's Max VM size).

Evaluation with Video Analytics Application (5/5)



WISEFUSE adjusts the execution plan based on the user specified latency target

- ❑ Higher latency target → Lower cost

Contributions

- Workload characterization for real-world serverless DAGs in Azure Durable Functions

- Two important optimizations:
  - Fusion: Communication latency between in-series functions
  - Bundling: Computation skew among in-parallel function invocations

- WISEFUSE performs Fusion and Bundling to derive an optimized execution plan that meets a user-defined latency SLO with low cost

- Experimental evaluation on AWS Lambda
  - Our performance model answers What-If questions (*e.g.*, impact of Fusion or Bundling, or impact of changing the Bundle size).
  - WISEFUSE generates different execution plans to meet different tail latency targets

PURDUE UNIVERSITY

LiteReconfig for Mobile Computer Vision

LiteReconfig [*EuroSys 2022*] performs principled approximation in streaming video analytics so that it can run on mobile or embedded devices and keep up with the video rate. It performs the approximation in a cost-benefit and video content-aware manner. We have also created a frontend called **ApproxLive** that makes our innovation available to end users.

LiteReconfig: Cost and Content Aware Reconfiguration of Video Object Detection Systems for Mobile GPUs

EuroSys 2022

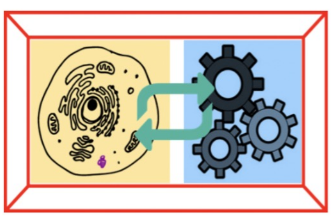
Overall Takeaways

- Serverless for complex data analytics
  - Can reduce latencies for latency-sensitive applications.
  - Increases the applicability of serverless to non-traditional workloads such as heavyweight ML and recommendation systems.
  - We drive our optimizations using real workloads for Microsoft Azure.
  - We also show how to use our tools on the side of the cloud provider and on the side of the clients, say, IoT company or in e-commerce.
- Approximate computing for streaming video analytics
  - Can be used in a data-driven manner to drive latency-sensitive, energy-aware computing on mobile devices.
  - This paradigm is extensible to a wide range of computer vision backends and can be used as plug and play tools for different AR/VR applications.

Acknowledgment

- National Science Foundation under Grant Numbers CNS-2038566, and CNS-2146449 (NSF-CPS Medium; NSF CAREER award on streaming video analytics for computer vision on the edge)
- Army Research Lab awards
- Amazon AI award
- Adobe Research Gift
- Microsoft Research

https://schaterji.io/projects/sirius.html

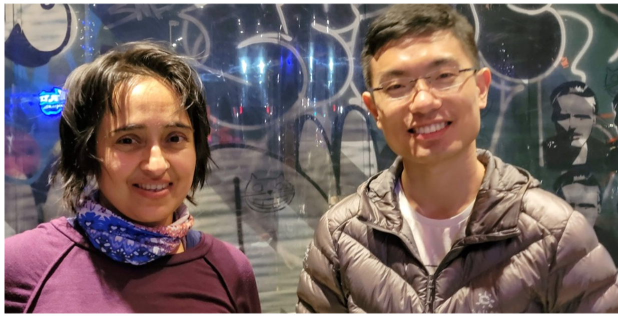


ICAN

Innovatory for Cells and Neural Machines

Read More

PURDUE UNIVERSITY



Innovatory for Cells and Neural Machines (ICAN), celebrating EuroSys, CVPR, OSDI, Sigmetrics, KDD 2022.