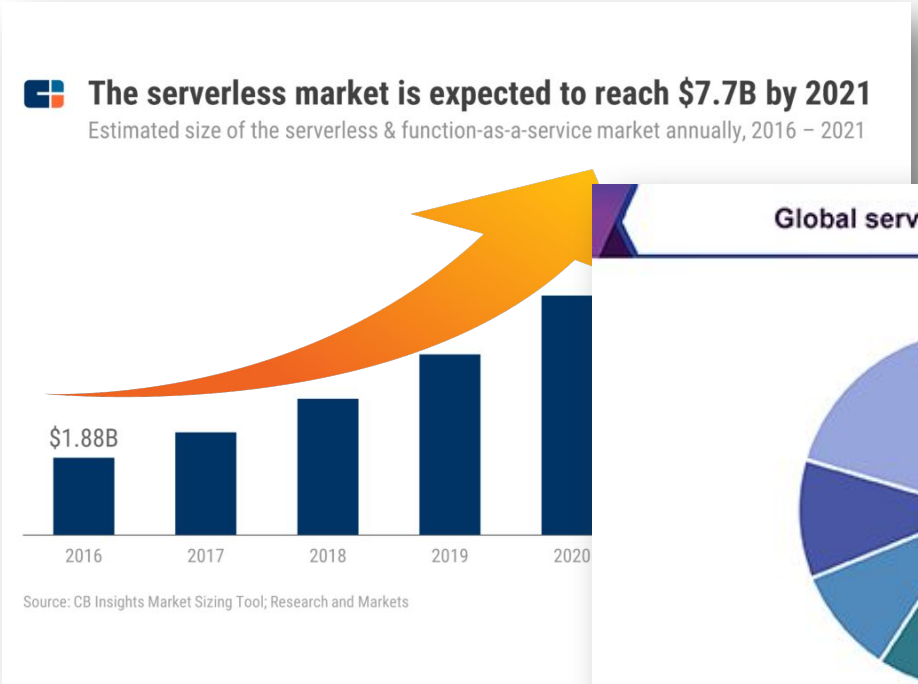
THE UNIVERSITY of EDINBURGH
informatics

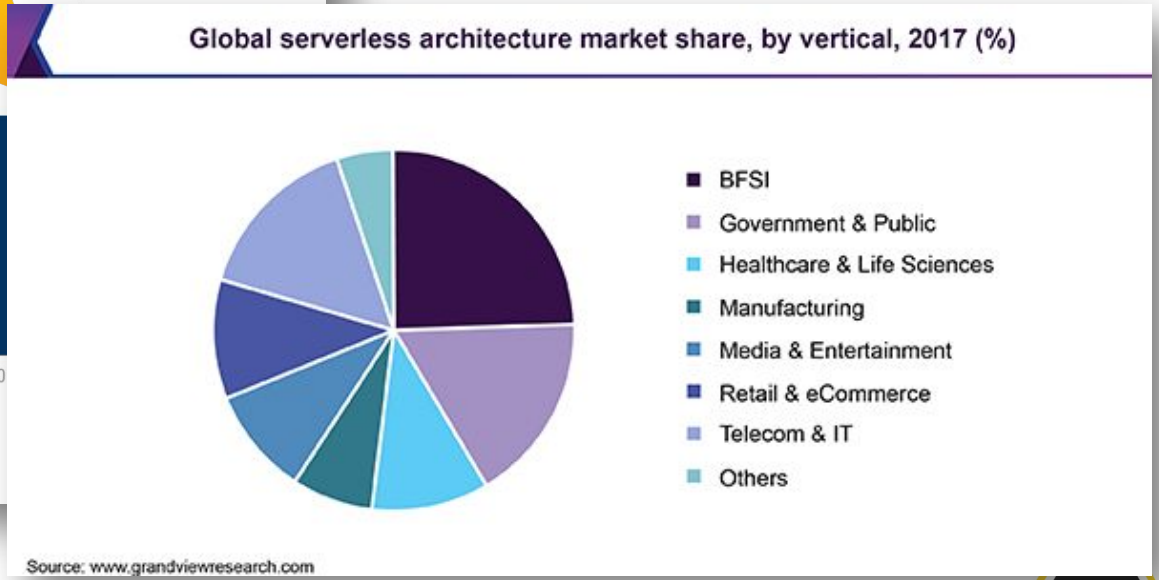
Turbocharging Serverless Research with vHive

Dmitrii Ustiugov

PhD Student at the University of Edinburgh

Serverless Rapid and Ubiquitous Adoption





Serverless has emerged as the next dominant cloud architecture

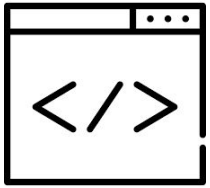
Why Service Developers Love Serverless

THE UNIVERSITY of EDINBURGH informatics

Happy serverless user

So, you will manage all infrastructure for me?

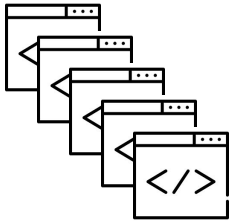
Functions



Serverless providers

No problem!

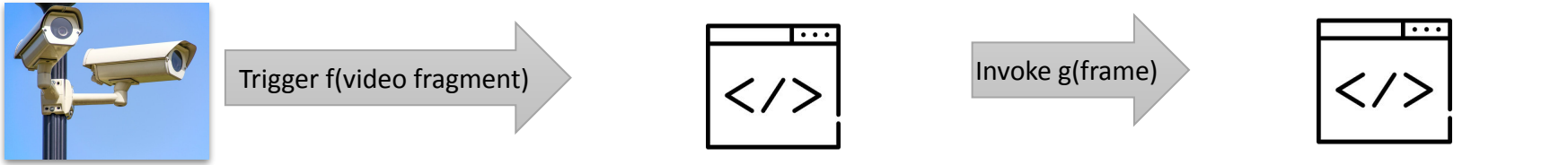
Manage



Labor division facilitates time to market

vHive

Service Developer's Perspective

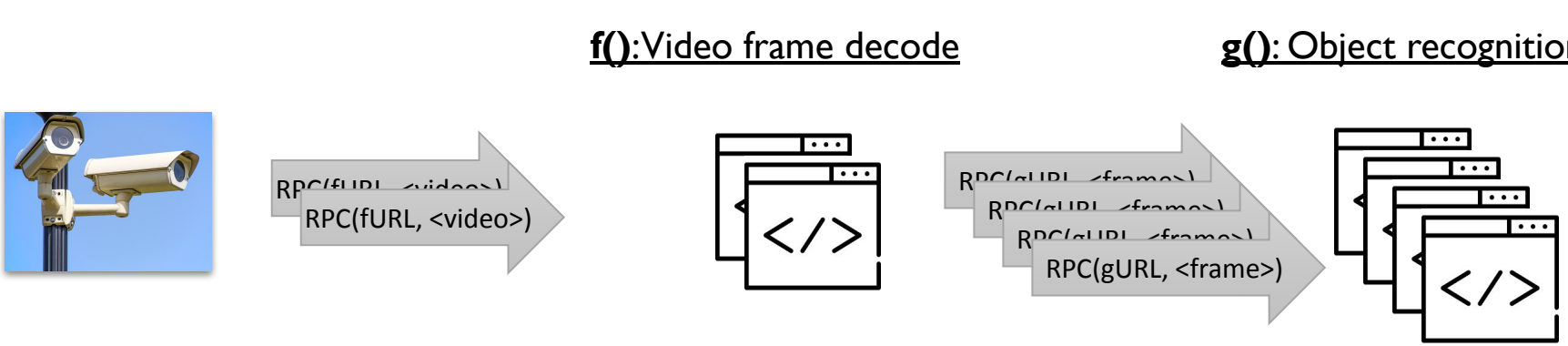


Write each function's business logic

Compose functions via event triggers and RPC calls

Serverless premise: "No need to think about servers"

Provider's Perspective



Function instances are *ephemeral*, spawned on demand

- 0 to ∞ instances of each function
- Provider to balance load and spawn / tear down instances

Serverless reality: Great for users, challenging for providers

THE UNIVERSITY of EDINBURGH
informatics

Time for Serverless Systems Research!

How to Do Research in Serverless?

Study serverless cloud infrastructure

Innovate & prototype across deep software/hardware stack

Evaluate your prototype with real workloads


What is inside serverless clouds?


The vHive Ecosystem
Study clouds
Innovate & prototype
Evaluate

THE UNIVERSITY of EDINBURGH
informatics

Study Production Clouds with STeLLAR [IISWC'21]

Benchmarking a Serverless Cloud with STeLLAR

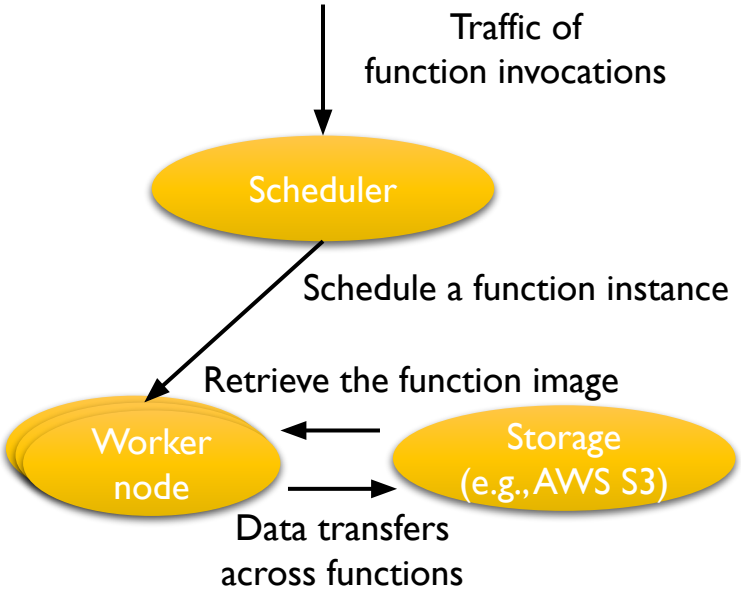
THE UNIVERSITY of EDINBURGH informatics

Key: Clouds are different, the architecture is the same

- Any serverless cloud has 3 components:
  the scheduler, a fleet of worker nodes, and storage

We introduce STeLLAR for **performance analysis**

- Configure function characteristics & traffic shape
- Benchmark each component separately with the STeLLAR client, loading & measuring response time
- Evaluated AWS, Azure, Google functions in California



vHive

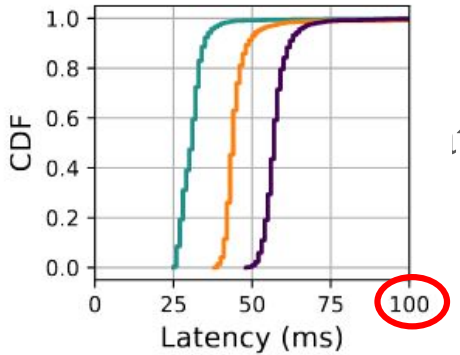
Worker: Warm & Cold Function Invocations

Cold latency if launching a new instance of a function

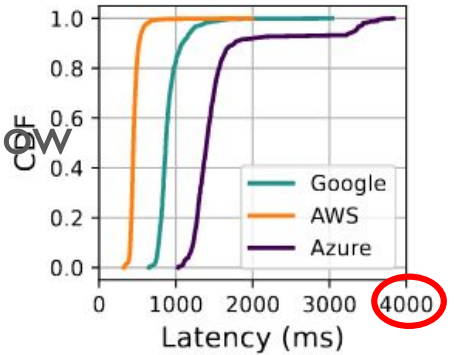
Warm la predictable

Cold lat tency variability is still low

Warm



Cold



Although cold-start latencies are much higher, latency is predictable

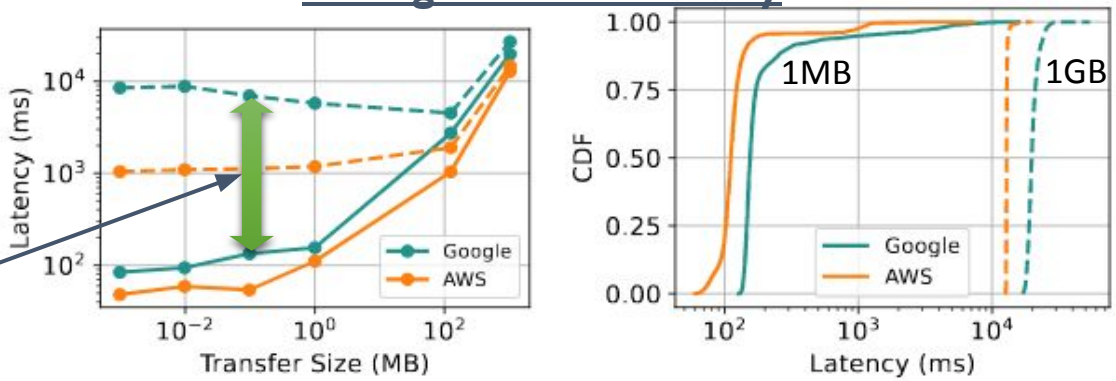
Storage: Data Transfers (Warm)

Functions can communicate only via storage (e.g., AWS S3)

Evaluation with 2 functions (producer and consumer)

Storage transfer latency

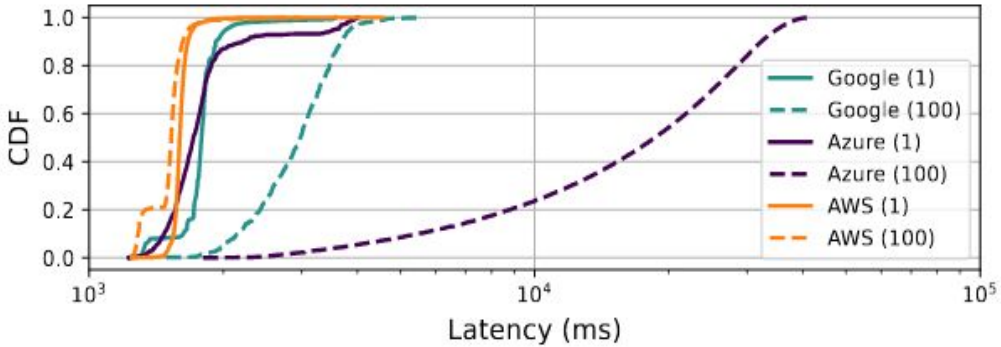
Tail is 40x vs. median latency



Storage-based transfers is the key tail latency source

Scheduler: Policy Implications (Warm)

Setup
- Warm invocations
- 1 and 100 invocations in a burst
- Function execution time: 1 second



Results
- Google: High latency but **moderate queuing**
- Azure: Very high latency, hence **abundant queueing** (~30% of invocations wait)
- AWS: No latency increase, hence no invocation queuing allowed

Scheduling policies can significantly impact median & tail latencies


The vHive Ecosystem
STeLLAR
Innovate & prototype
Evaluate

THE UNIVERSITY of EDINBURGH
informatics

Innovate & Prototype with vHive [ASPLOS'21]

Studying Serverless: State-of-the-Art Frameworks

Bleeding-edge but **proprietary** systems
- Complex distributed software stack

Incomplete or **non-representative**
- Single component, e.g., hypervisor
- Container isolation only (e.g., OpenWhisk, OpenLambda)
  - but >70% of the market (AWS, Azure, Google) rely on VMs

Need for a complete open-source framework for serverless research

Serverless in the Age of Open Source

THE UNIVERSITY of EDINBURGH informatics

Kubernetes + Knative

Cluster scheduler & Function-as-a-Service API
(Google & CNCF)

MicroVM technologies

Firecracker gVisor

gRPC

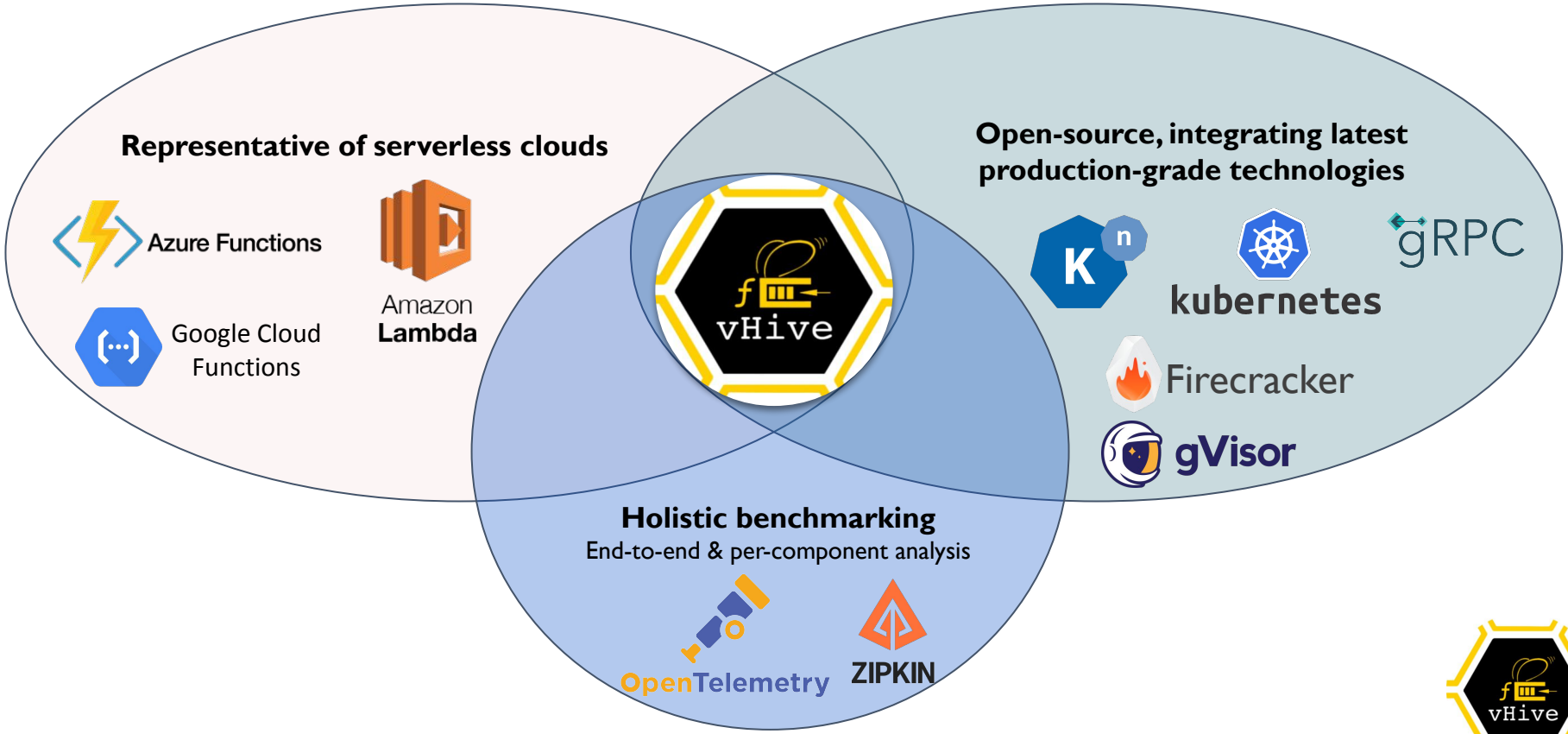
Communication fabric (Google)

containerd

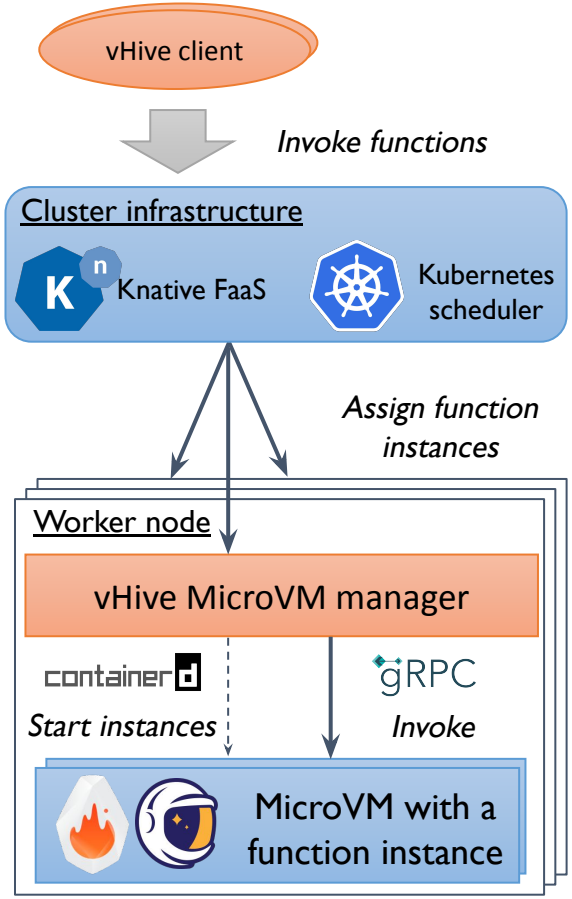
Host management (CNCF)

vHive

vHive: Framework for Serverless Experimentation



vHive Architecture



vHive clients: Load and measure latency of invocations

Cluster infrastructure

- Kubernetes cluster scheduler
- Knative Function-as-a-Service programming interface
  - Arbitrary **Docker-images** deployment
  - **Autoscaling** function instances on demand

Worker nodes

- MicroVM manager that drives MicroVMs lifecycle
- **Control plane**: Containerd
- **Data plane**: gRPC

First to support snapshots (Firecracker) at scale

The vHive Ecosystem

STeLLAR

Evaluate

vSwarm: The Representative Benchmark Suite

Teamed up with ETH, Stanford, and the broad systems community

vSwarm benchmarks include

- 30 individual (leaf) functions in 4 language runtimes
- 8 multi-function applications (video analytics, ML training, distributed compilation, …)
- Integration with AWS S3, AWS Elasticache Redis, Apache Kafka, KubeEdge (in progress)

Workloads come with distributed tracing & microarchitectural analysis tools

Future work: Gem5 simulator images (stay tuned!)

What Kind of Research Can vHive Help?

THE UNIVERSITY of EDINBURGH informatics

Operating systems

- Record-and-Prefetch snapshots for accelerating cold starts [ASPLOS'21]

Communication & distributed systems

- Fast & autoscaling communication fabric for serverless [under submission]

Processor microarchitecture

- Microarchitectural state prefetching for serverless [under submission]

vHive

vHive Open-Source Community Today

THE UNIVERSITY of EDINBURGH informatics

ASPLOS'21: Distinguished Artifact Award

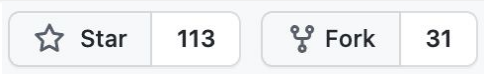
Academic contributors:

The University of Edinburgh, EPFL, ETH zürich, Technische Universität München, Stanford University

Industrial collaborators:

aws, Microsoft Research, arm, HUAWEI

- Used at 16+ universities (research & course)
- 5 external contributing organizations
- 100 unique cloners/day (GitHub)



vHive

The vHive Core Team

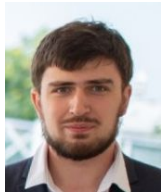
THE UNIVERSITY of EDINBURGH informatics

Supervisors


PhD student & leader

Students & Interns & Alumni







More contributors at https://github.com/ease-lab/vhive


What is inside serverless clouds?


With vHive, the clouds are clear.
Study clouds
Innovate & prototype
Evaluate

THE UNIVERSITY of EDINBURGH
informatics

Join the vHive Open-Source Community

https://github.com/ease-lab/STeLLAR
https://github.com/ease-lab/vhive
https://github.com/ease-lab/vSwarm

firecracker-microvm.slack.com - *channel:* #firecracker-vhive-research

vHive

THE UNIVERSITY of EDINBURGH
informatics

Backup

Tools for In-Depth Performance Analysis

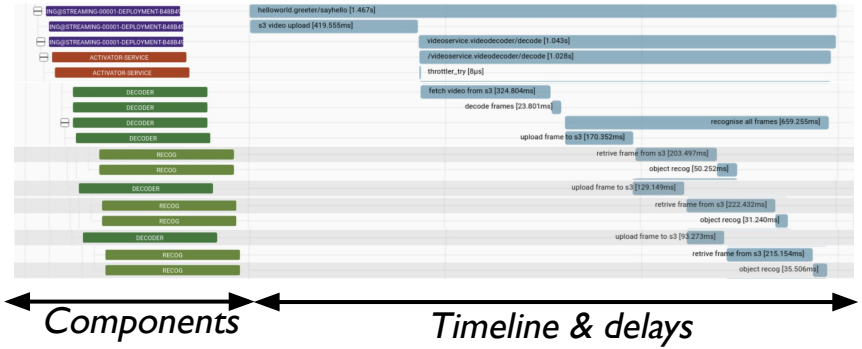
THE UNIVERSITY of EDINBURGH informatics

Distributed Tracing

Serverless systems are complex & distributed

- Diverse provider and user components

vHive is first to combine **all in one trace**!

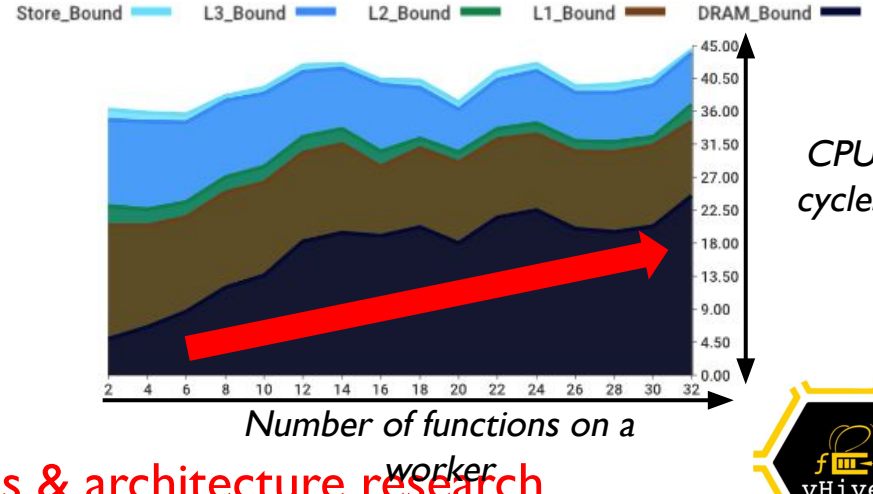


CPU Microarchitectural Profiling

Worker nodes run up to 1000s of functions

- All CPU resources are shared and/or multiplexed

vHive natively supports **Intel TopDown** [Yasin'14]



vHive is tailored for systems & architecture research