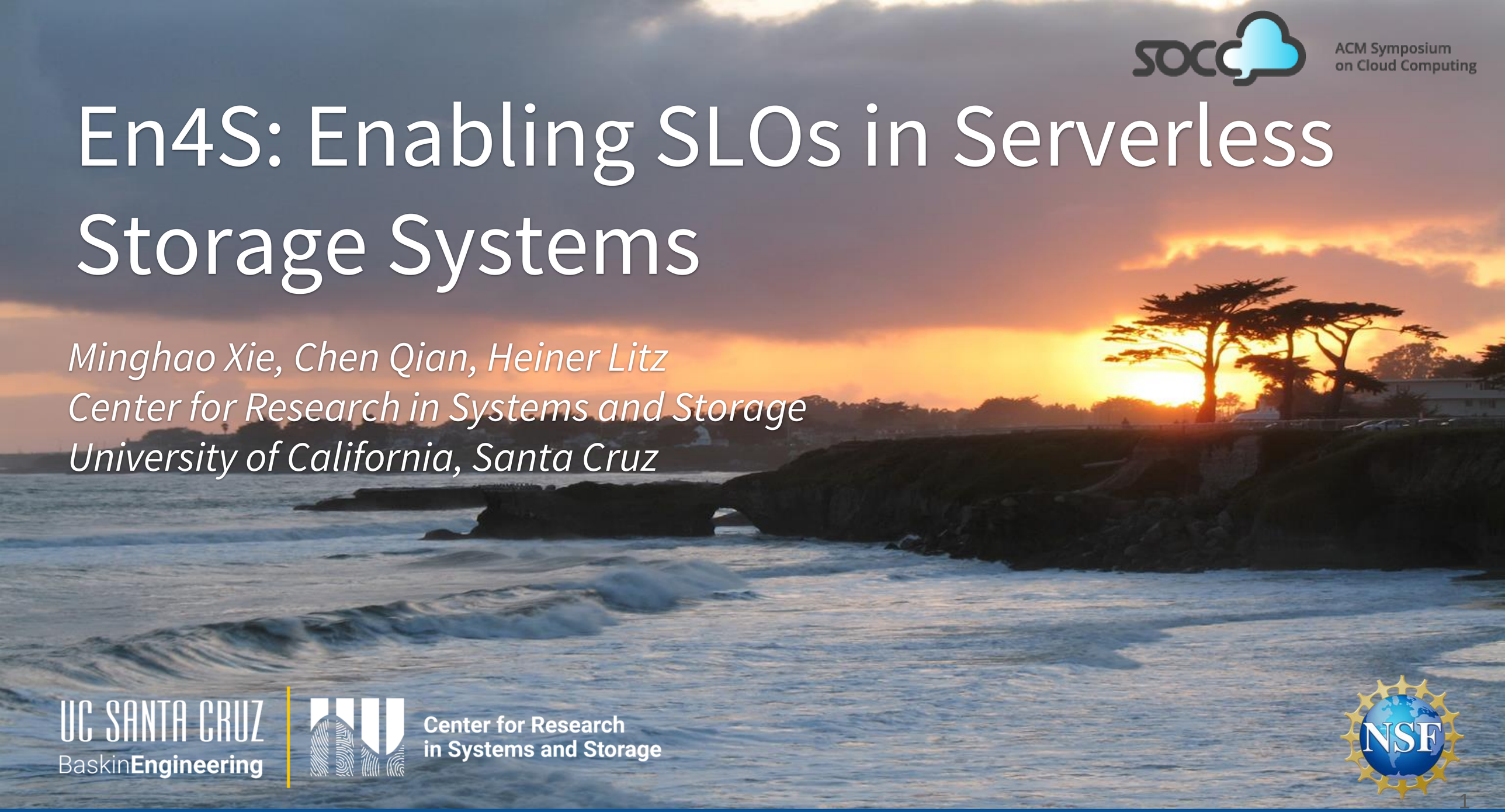


ACM Symposium on Cloud Computing

En4S: Enabling SLOs in Serverless Storage Systems

Minghao Xie, Chen Qian, Heiner Litz
Center for Research in Systems and Storage
University of California, Santa Cruz

UC SANTA CRUZ BaskinEngineering

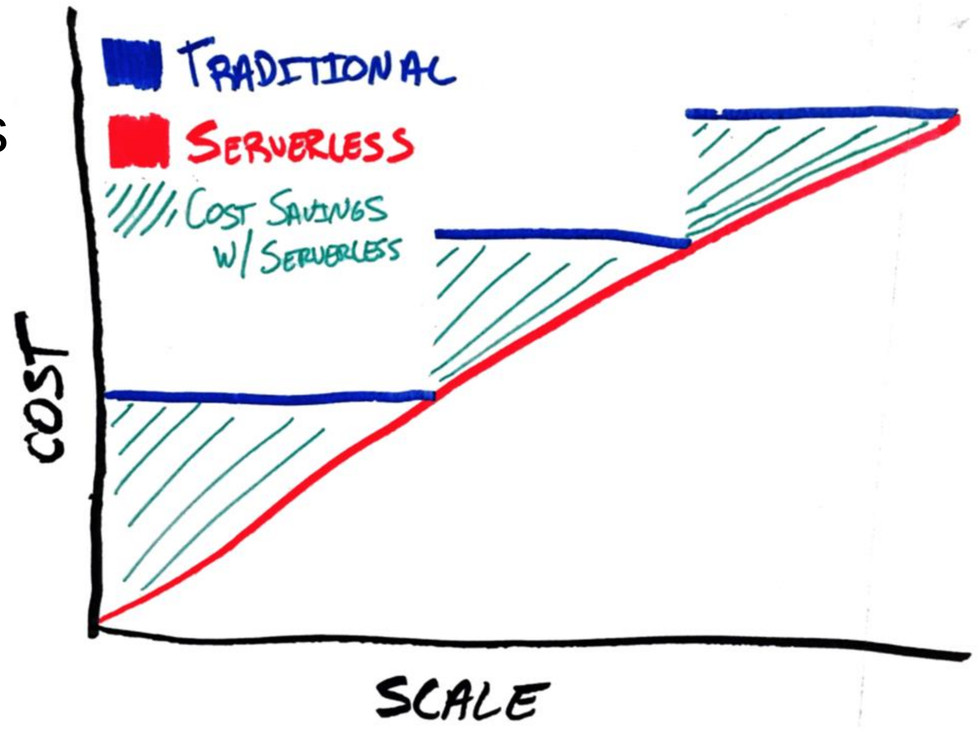
Center for Research in Systems and Storage

NSF

Serverless Computing

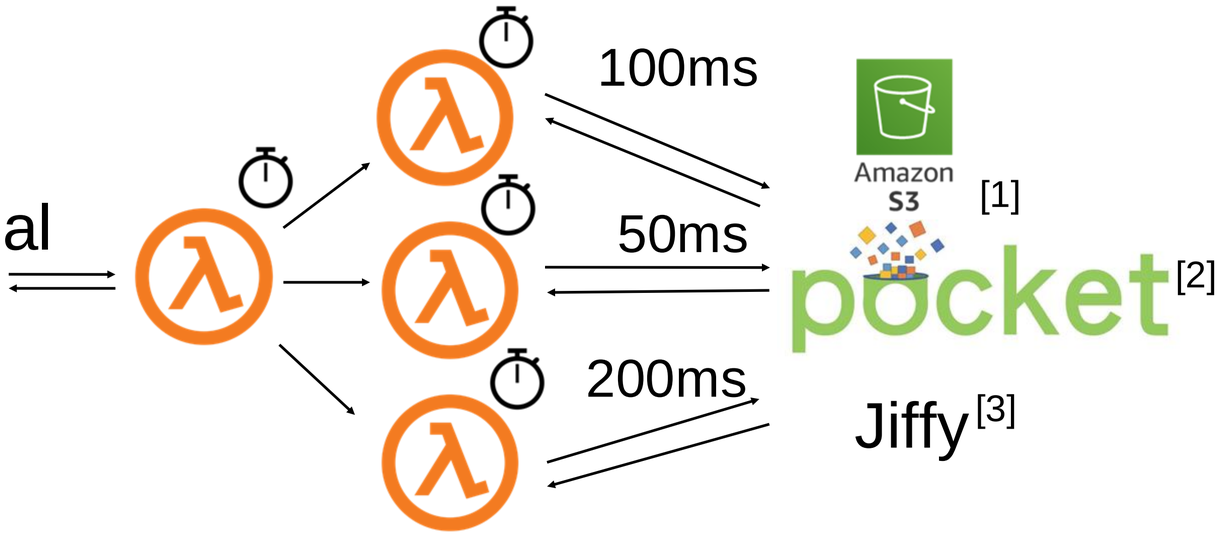
- Requires statelessness!
  - Hard to develop entirely stateless applications
  - Introduces performance overheads
  - **Serverless (Ephemeral) Storage**:

    Easy to allocate, cheap, low-latency, extremely short-lived storage



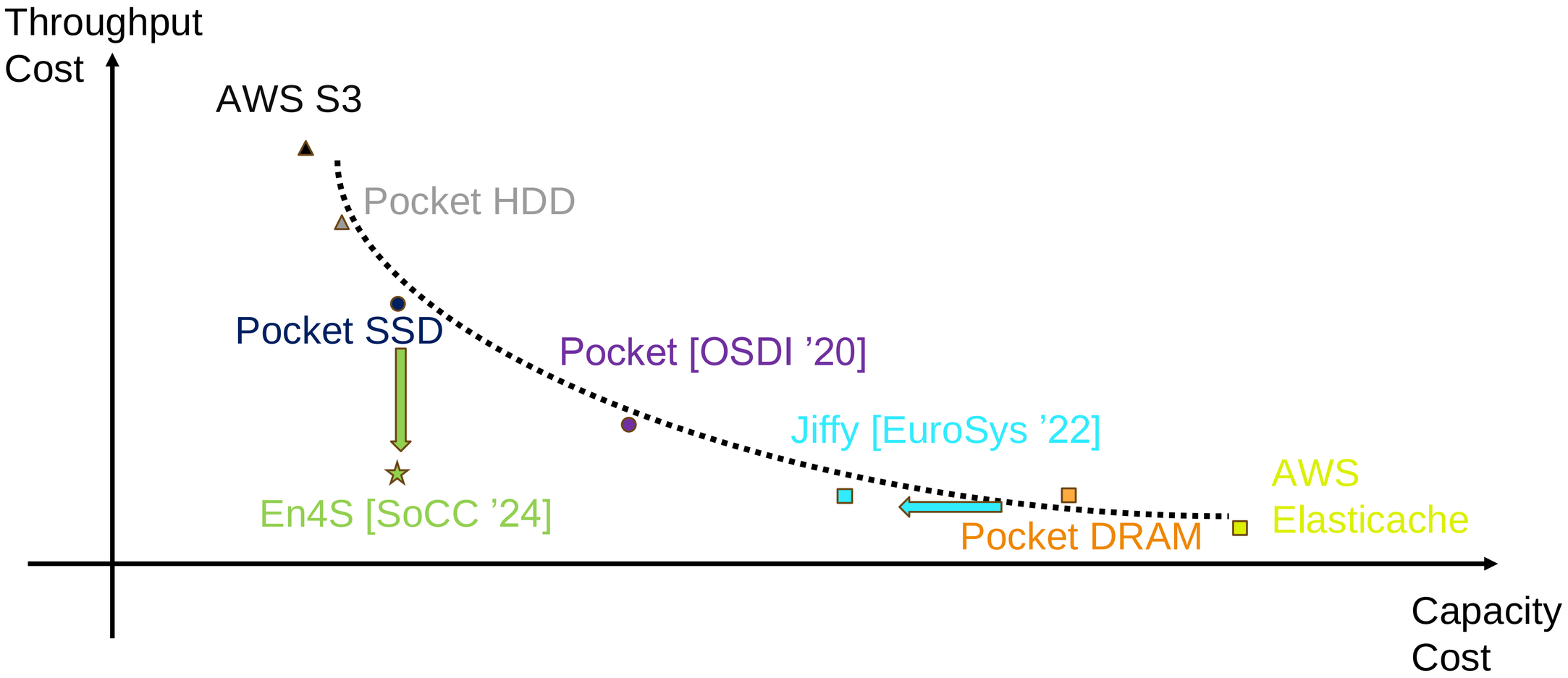
Challenges in Serverless Storage

- Serverless Environment
  - Thousands of very **short-lived** jobs
  - **Bursty** access pattern becomes normal
  - Performance **predictability** improves performance and saves costs



*Data Evenly Sharded

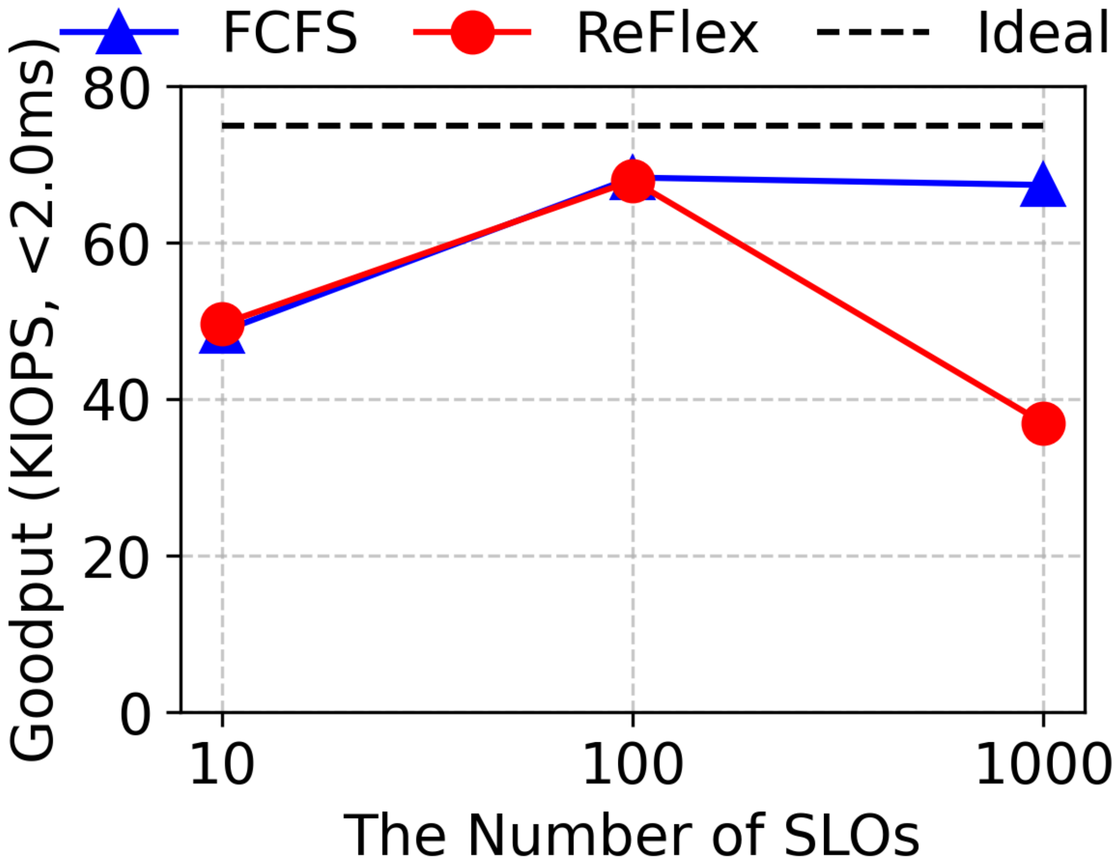
Why Flash-based Serverless Storage?

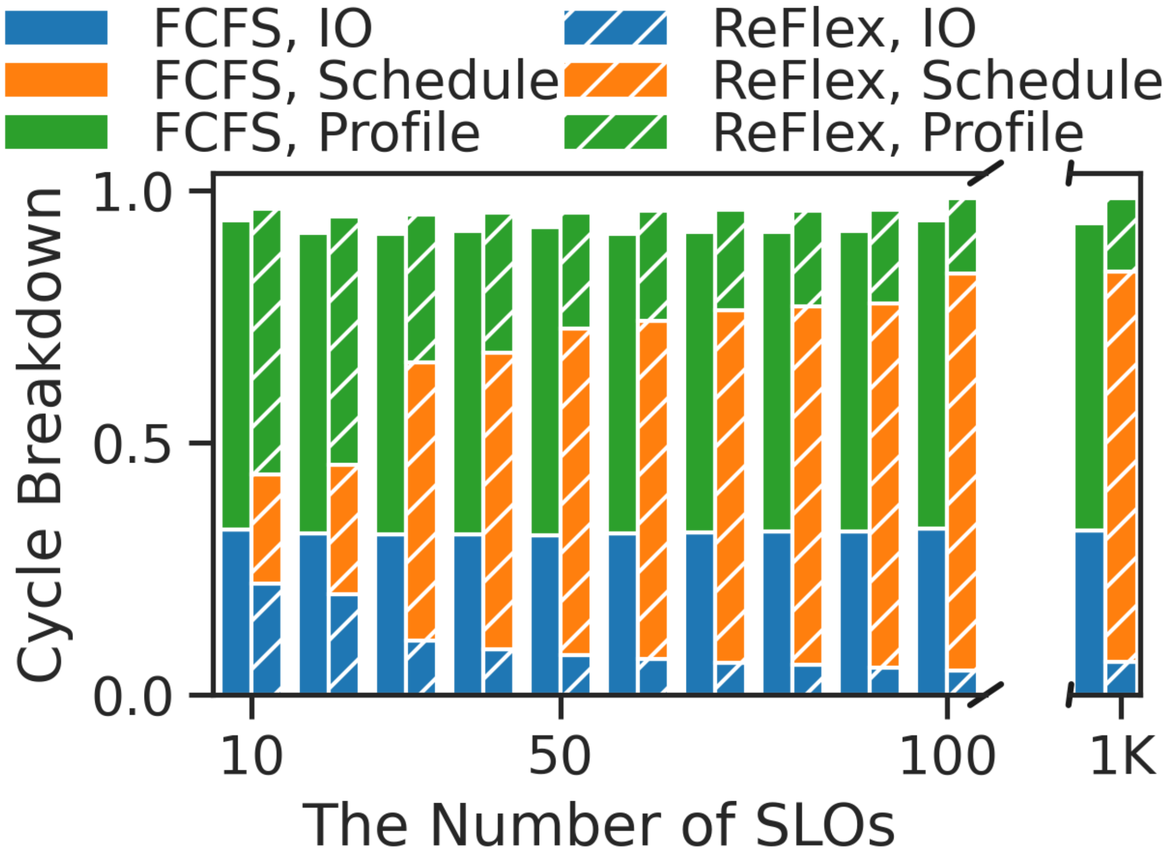


The Scaling Problems in Flash Storage Disaggregation in Serverless Environment

- Limited Scheduler Scalability
- Challenges in Managing Bursty Tenants
- Service Differentiation Failures at Scale

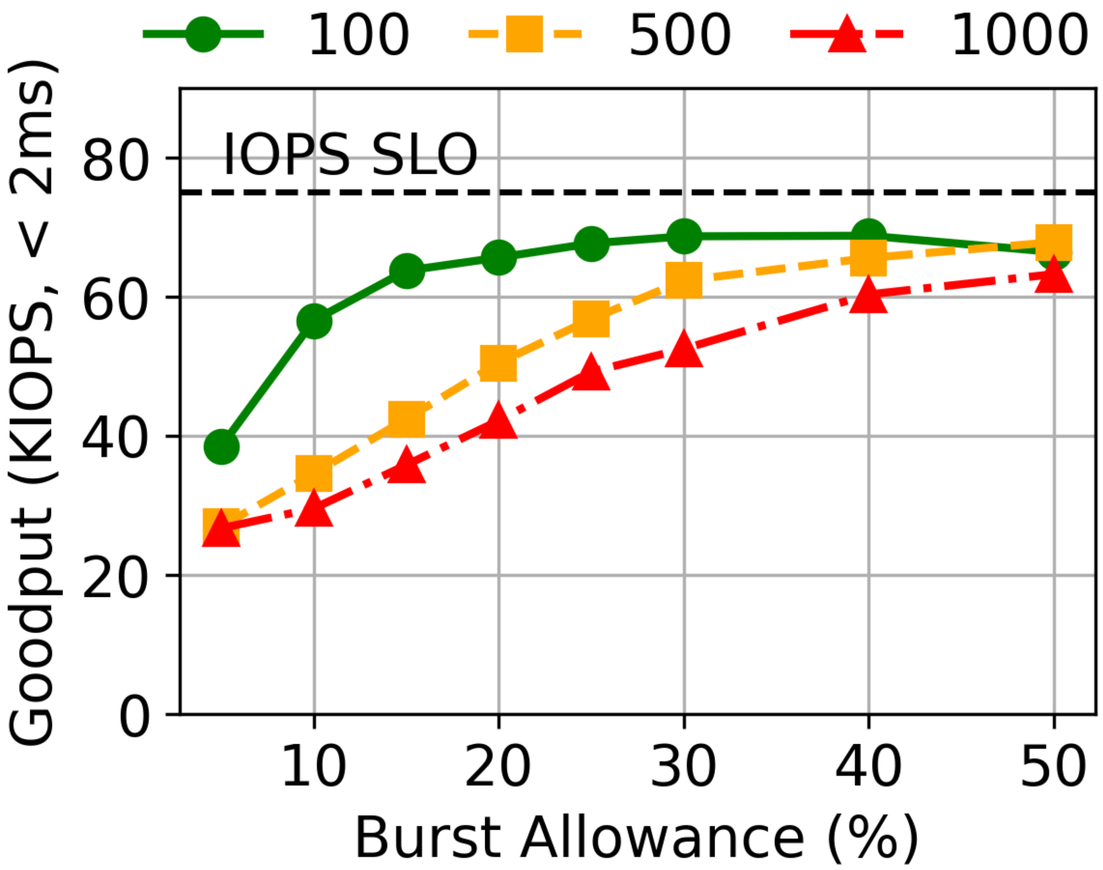
I - Limited Scheduler Scalability

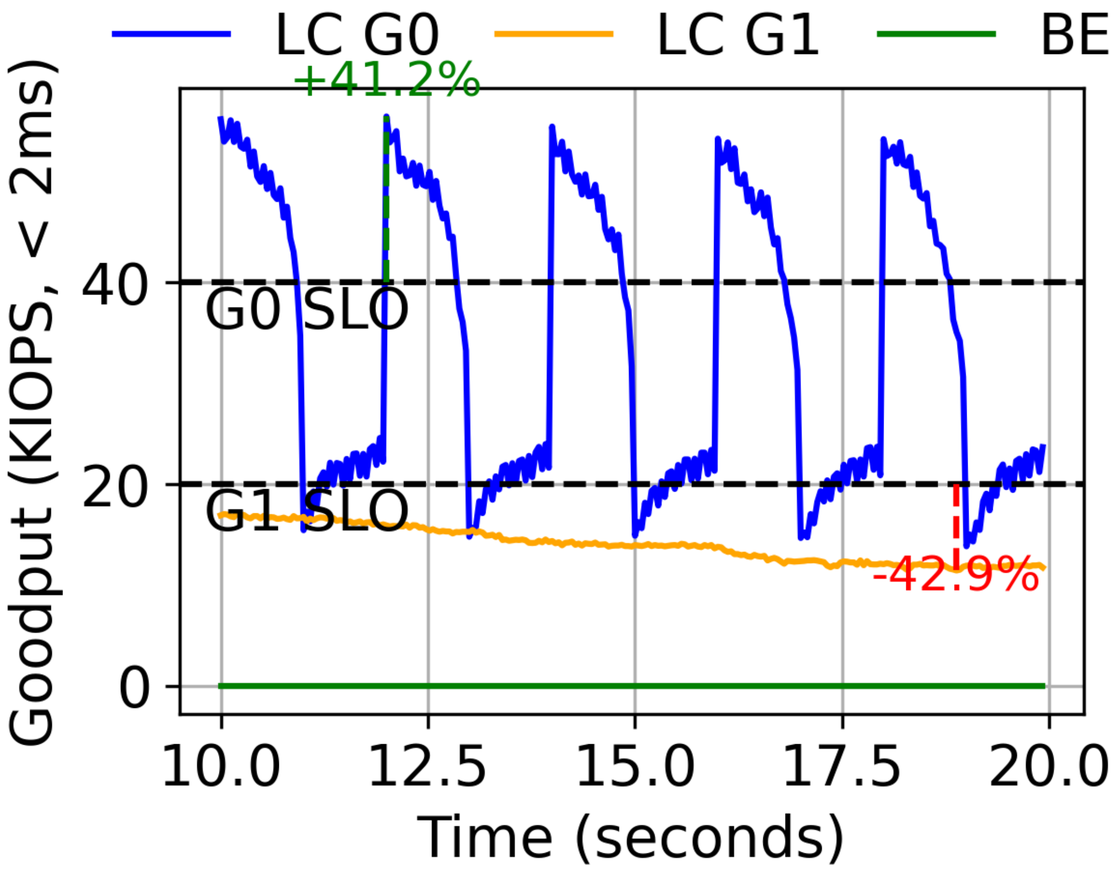




Existing QoS scheduler does not scale to thousands of SLOs

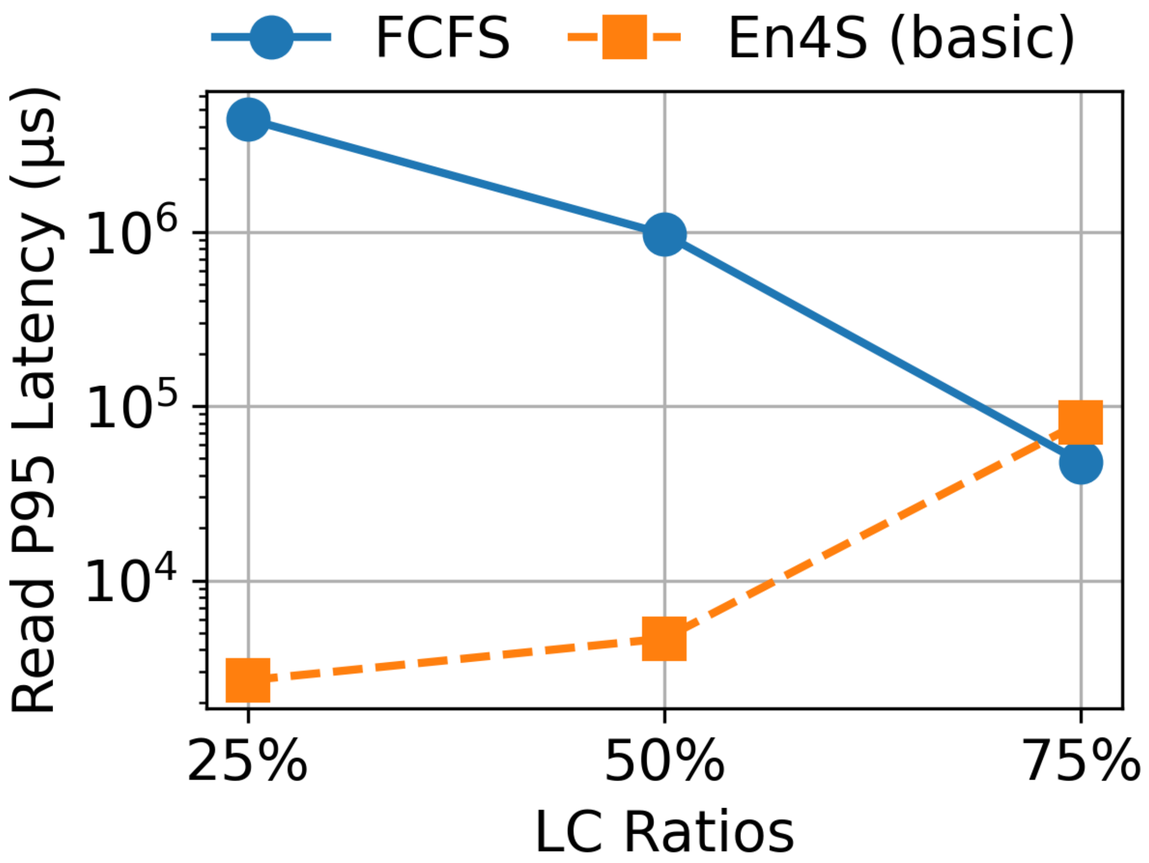
II - Challenges in Managing Bursty Tenants

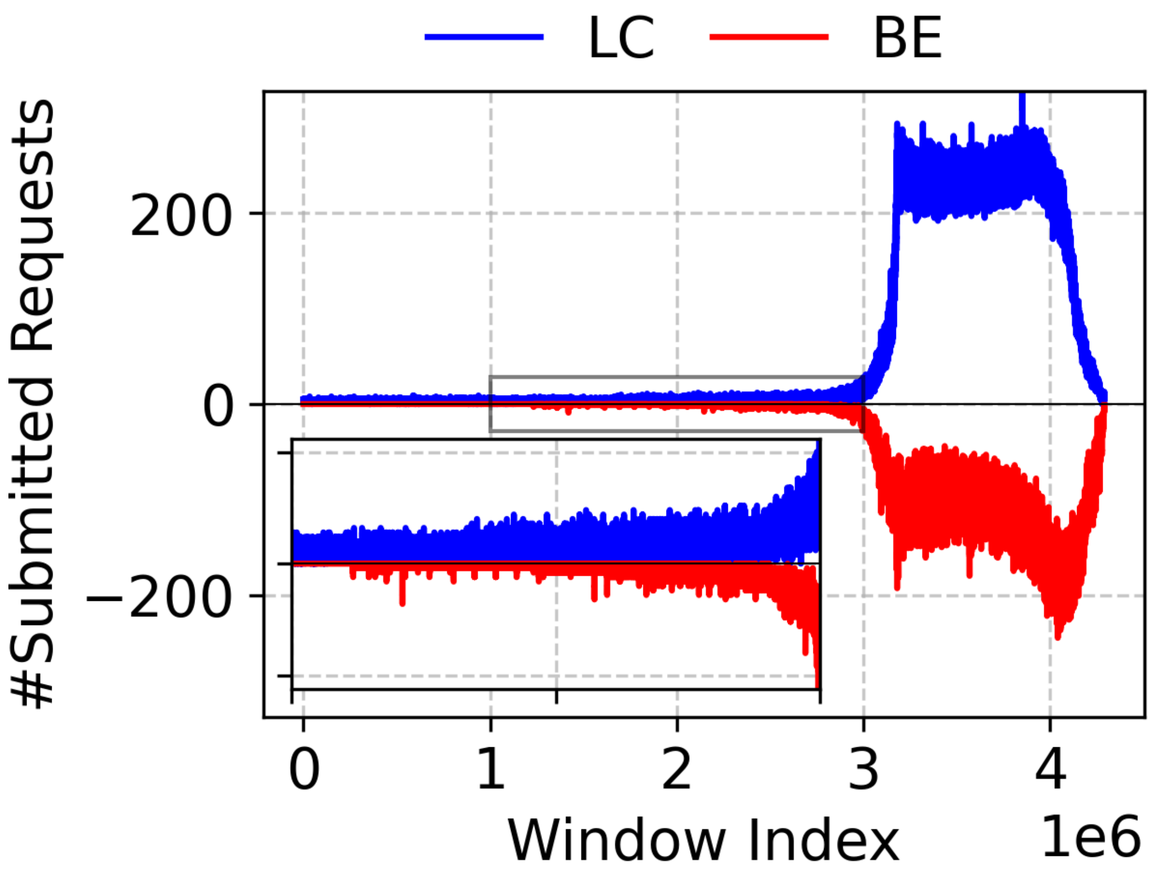




Low burst allowance is not work-conserving in the presence of bursty traffic while high burst allowance policy fails to enforce SLOs

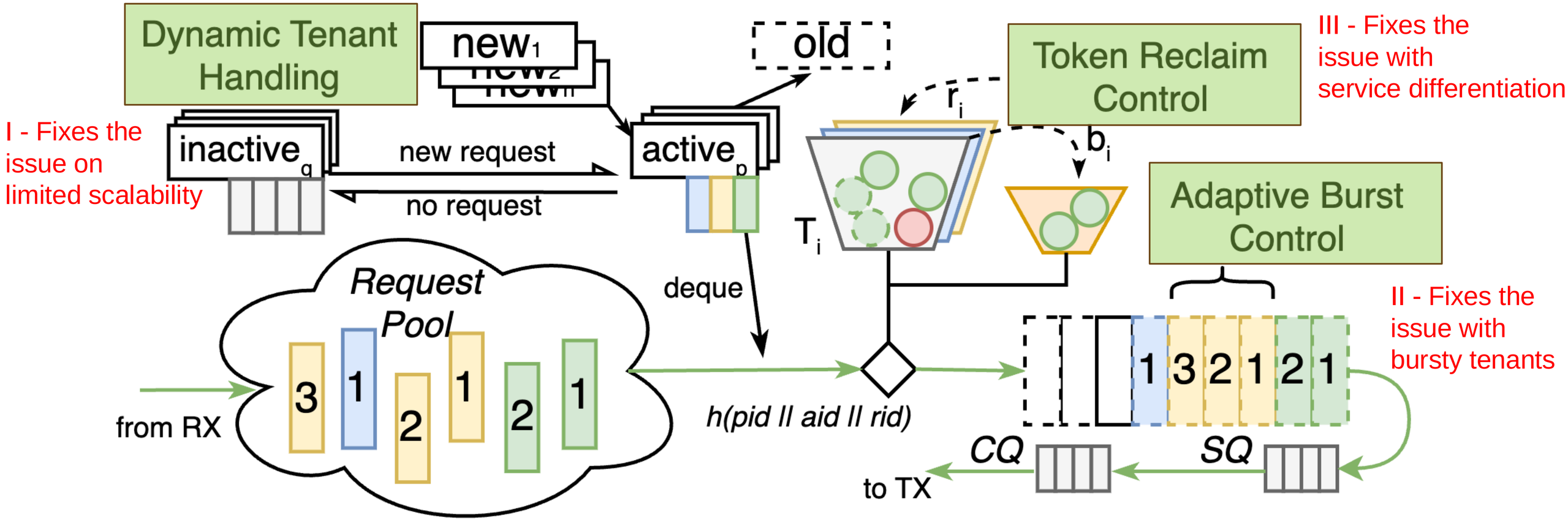
III: Service Differentiation Failures at Scale





Service differentiation between LC and BE tenants diminishes at scale

En4S: Redesigning QoS Scheduler for Serverless Storage Systems



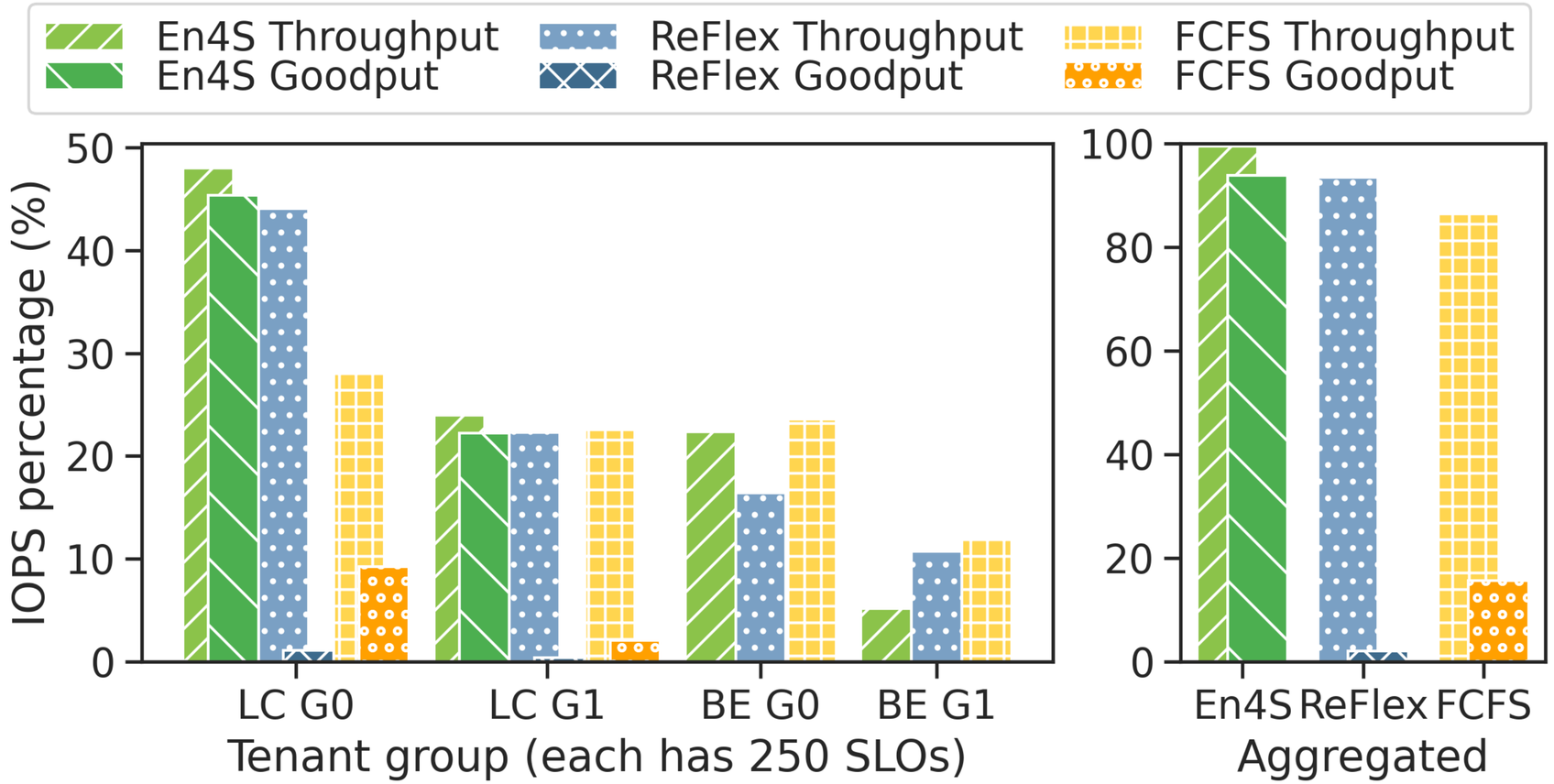
Evaluation Methodology

- Baselines:
  - ReFlex: Pocket's NVMe SSD Tier
  - Jiffy: DRAM-based ephemeral storage
  - S3: AWS cloud-native storage system
- Applications:
  - Serverless ETL Pipelines
  - Serverless Sort (Stream)
  - ML Analytics

Server Function	EC2 Instance	Cap (TB)	4K Thpt (KIOPS)	Cost ($) / IO at Full speed
En4S Storage Candidates	i3.l	.475	100	4.33×10^{-10}
	i3.xl	0.95	200	4.33×10^{-10}
	i3.2xl	1.90	236	7.34×10^{-10}
Controller[3]	m5.2xl	0	200*16	1.67×10^{-11}
Jiffy	m5.16xl	.256	600	1.42×10^{-9}

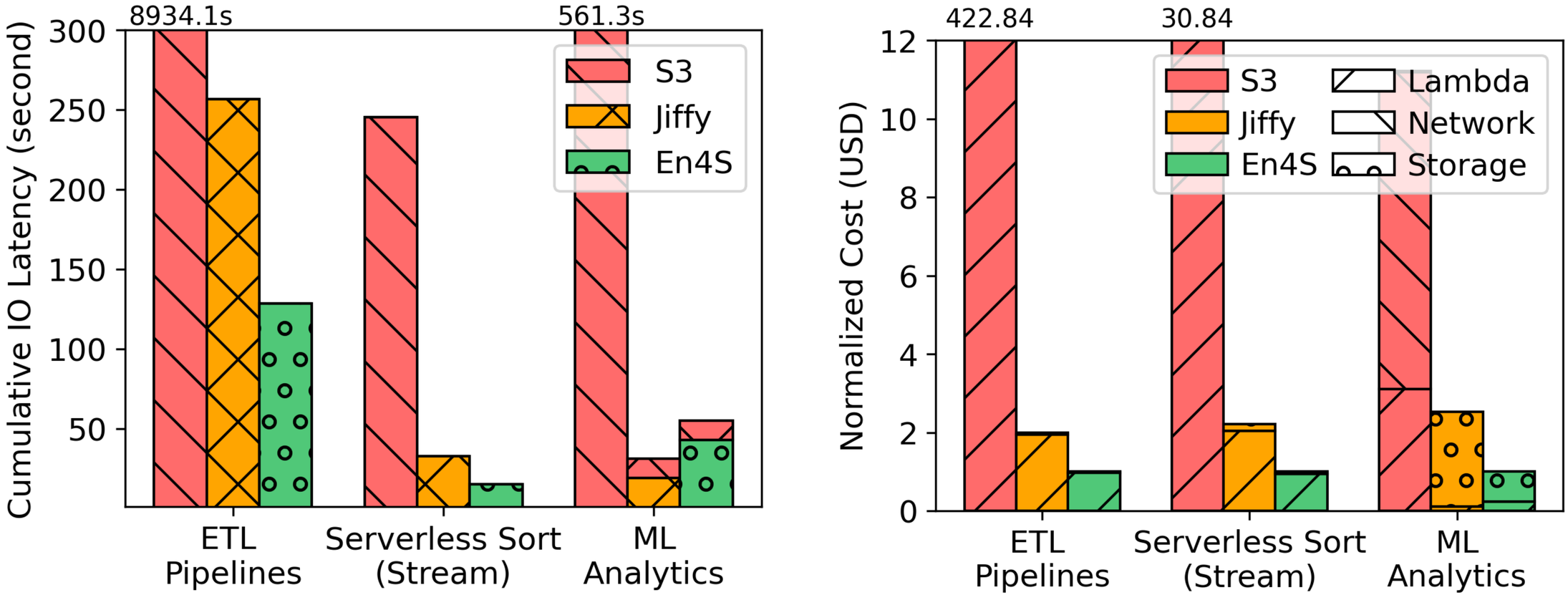
Table 2: Different AWS EC2 instances used for Jiffy and En4S clusters in US-West region

Scheduler Scalability



Tenants categorized into four groups based on their SLO criticality and average read ratio, LC Tenant Group 1 (LC G0) at 100%, LC Tenant Group 1 (LC G1) at 80%, BE Tenant Group 0 (BE G0) at 95%, and BE Tenant (BE G1) at 25%..

Performance and Cost Improvement



Cumulative IO Time of End-to-End Application Performance and Corresponding Normalized Costs for Running Equivalent Workloads.

Summary

- We introduce En4S, a high-performance, flash-based storage system designed for data-intensive serverless applications:
  - Flash disaggregation with cloud-native interfaces;
  - Performance isolation for LC/BE and both compliant and malicious tenants;
  - Scalable SLO guarantees for thousands of tenants;

- Source code available at https://github.com/mhxie/En4S

Status and Future Work

- Status:
  - En4S published at SoCC’24
  - Benchmark: New findings on AWS EC2 Instance Storage
  - Control Plane Extension: From SLO-Enforced Storage to Performance Gains in DAG-Based Serverless Applications
- Next Steps:
  - Submit the Control Plane and Benchmark Papers to the target conferences

References

1. *AWS Simple Object Store, https://aws.amazon.com/s3/*
2. *Klimovic, A., Wang, Y., Stuedi, P., Trivedi, A., Pfefferle, J., & Kozyrakis, C. (2018). Pocket: Elastic ephemeral storage for serverless analytics. In 13th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 18) (pp. 427-444).*
3. Khandelwal, A., Tang, Y., Agarwal, R., Akella, A., & Stoica, I. (2022, March). Jiffy: Elastic far-memory for stateful serverless analytics. In *Proceedings of the Seventeenth European Conference on Computer Systems* (pp. 697-713).
4. *Klimovic, A., Litz, H., & Kozyrakis, C. (2017). Reflex: Remote flash≈ local flash. ACM SIGARCH Computer Architecture News, 45(1), 345-359.*
5. *Dean, J., & Barroso, L. A. (2013). The tail at scale. Communications of the ACM, 56(2), 74-80.*
6. Luciano Lenzini, Enzo Mingozzi, and Giovanni Stea. 2002. Aliquem: a novel DRR implementation to achieve better latency and fairness at O (1) complexity. In IEEE 2002 Tenth IEEE International Workshop on Quality of Service (Cat. No. 02EX564). IEEE, 77–86
7. *Klimovic, Ana, Yawen Wang, Christos Kozyrakis, Patrick Stuedi, Jonas Pfefferle, and Animesh Trivedi. "Understanding ephemeral storage for serverless analytics." In 2018 {USENIX} Annual Technical Conference ({USENIX}{ATC} 18), pp. 789-794. 2018.*

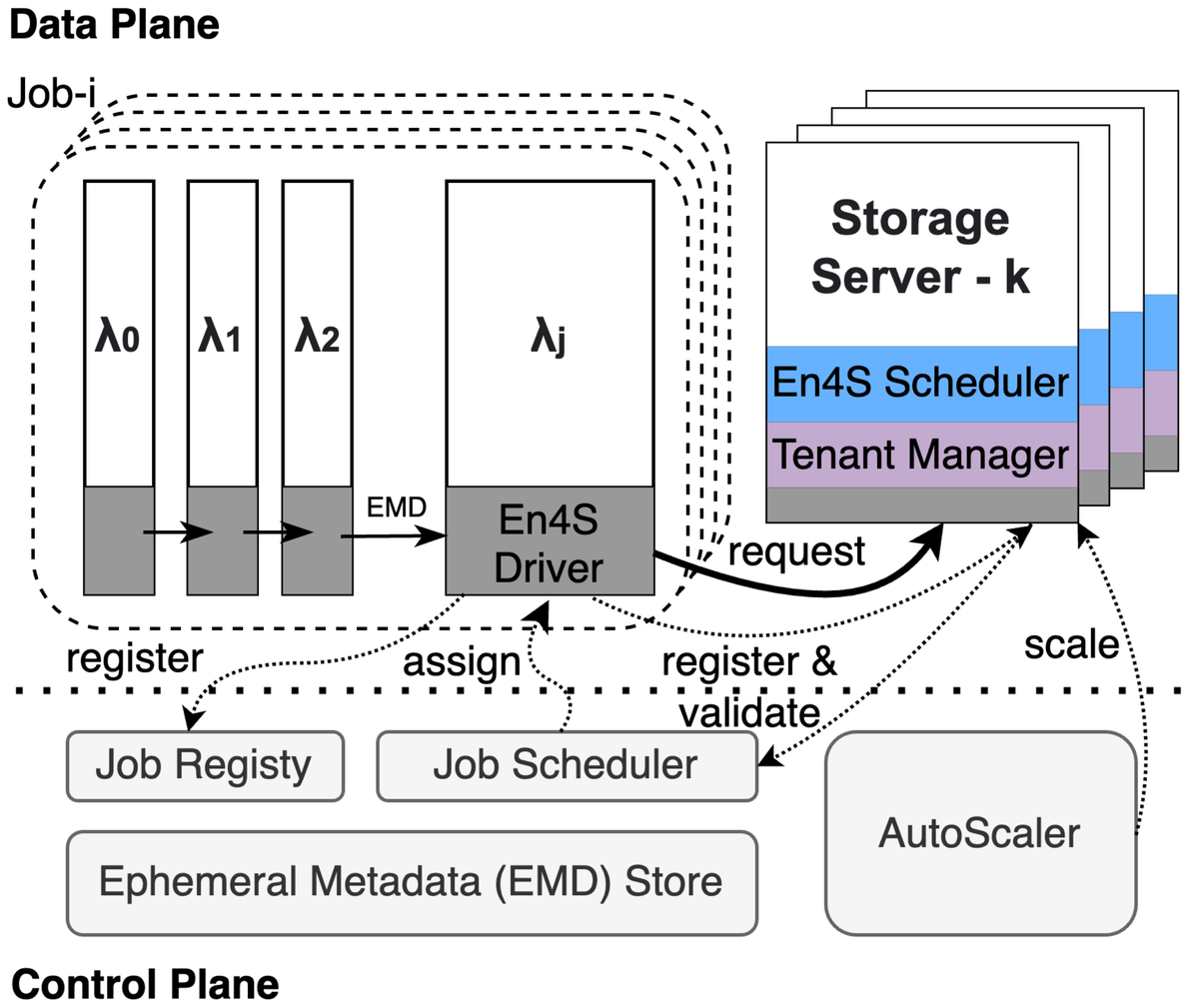
Thanks!

Minghao Xie
Contact: mhxie@ucsc.edu

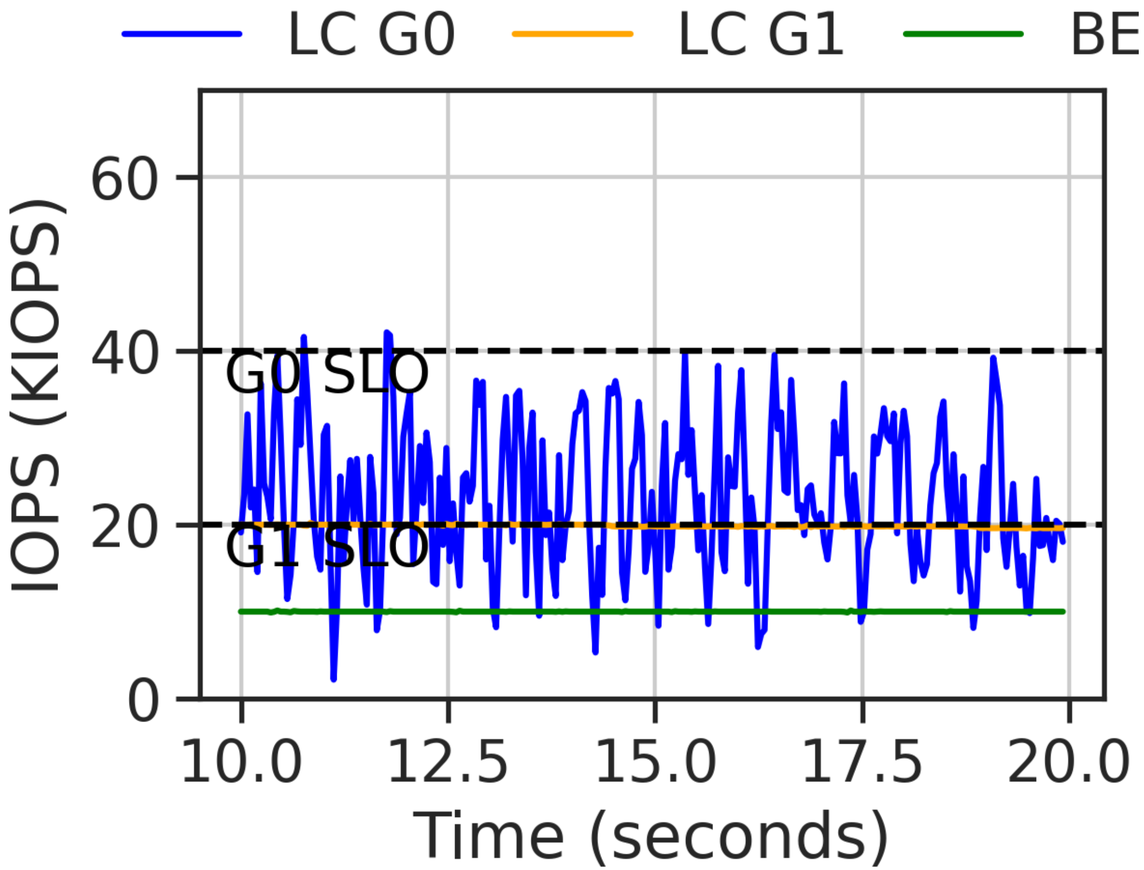
Backup Slides

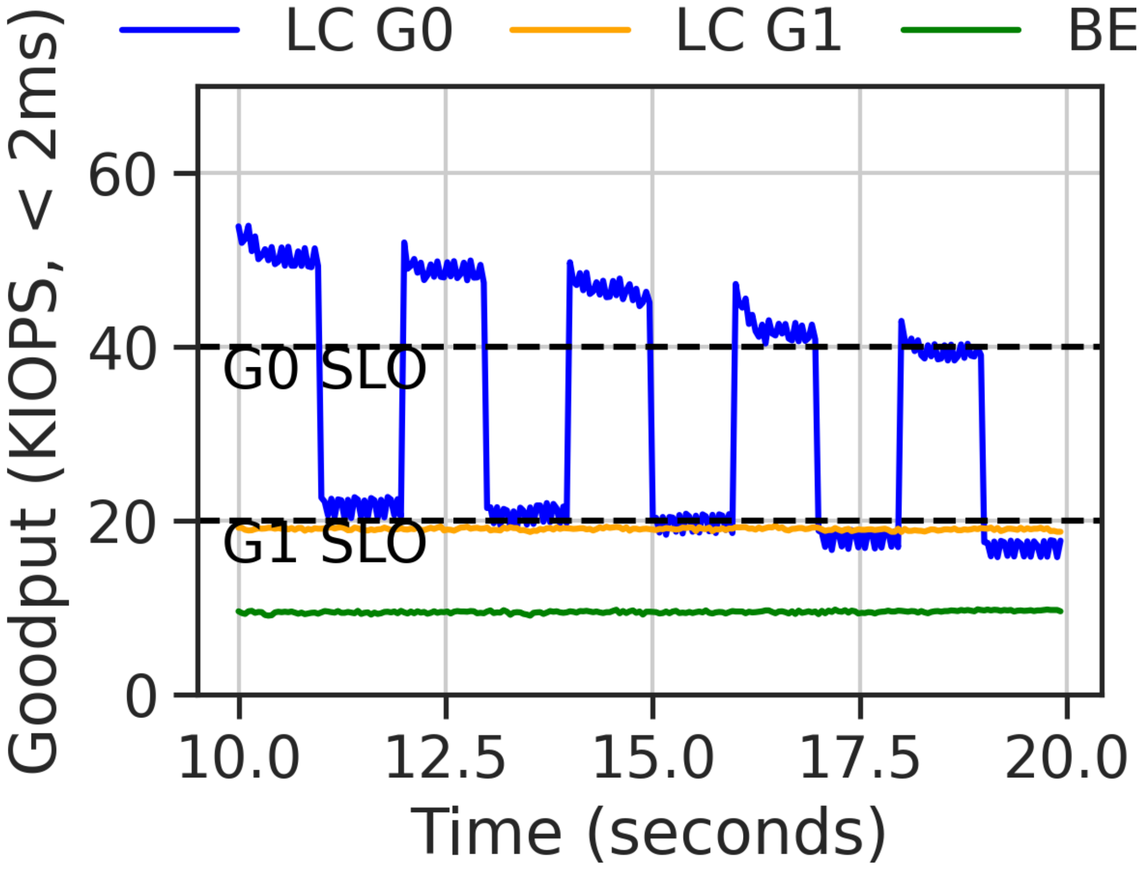
En4S in a glance

- Design Principles:
  - Performance
  - Predictability
  - Cost Efficiency
- Key Designs
  - Ephemeral Metadata Store
  - Distributed Control Plane
  - SLO-enforced Data Plane

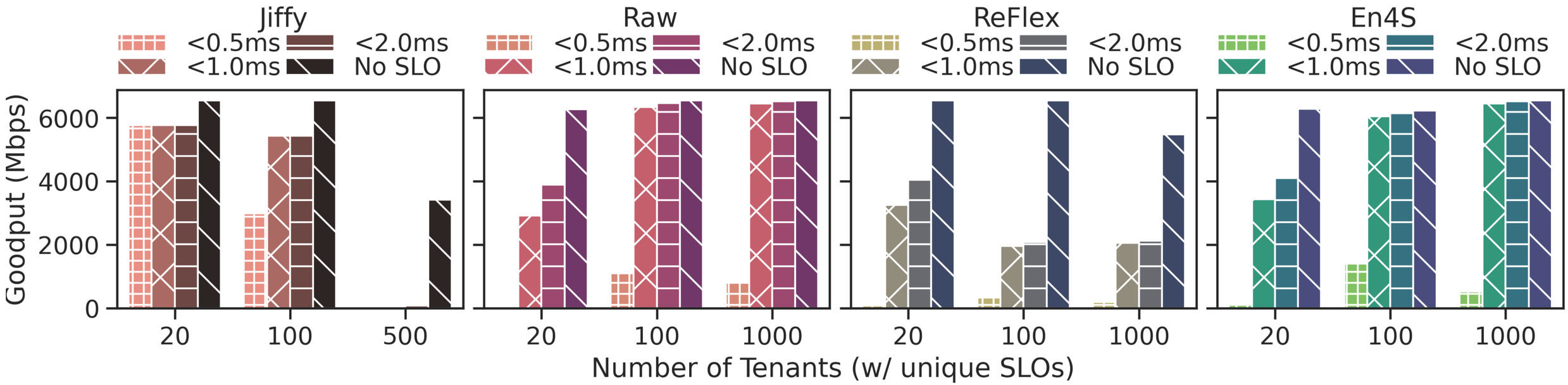


Performance Isolation

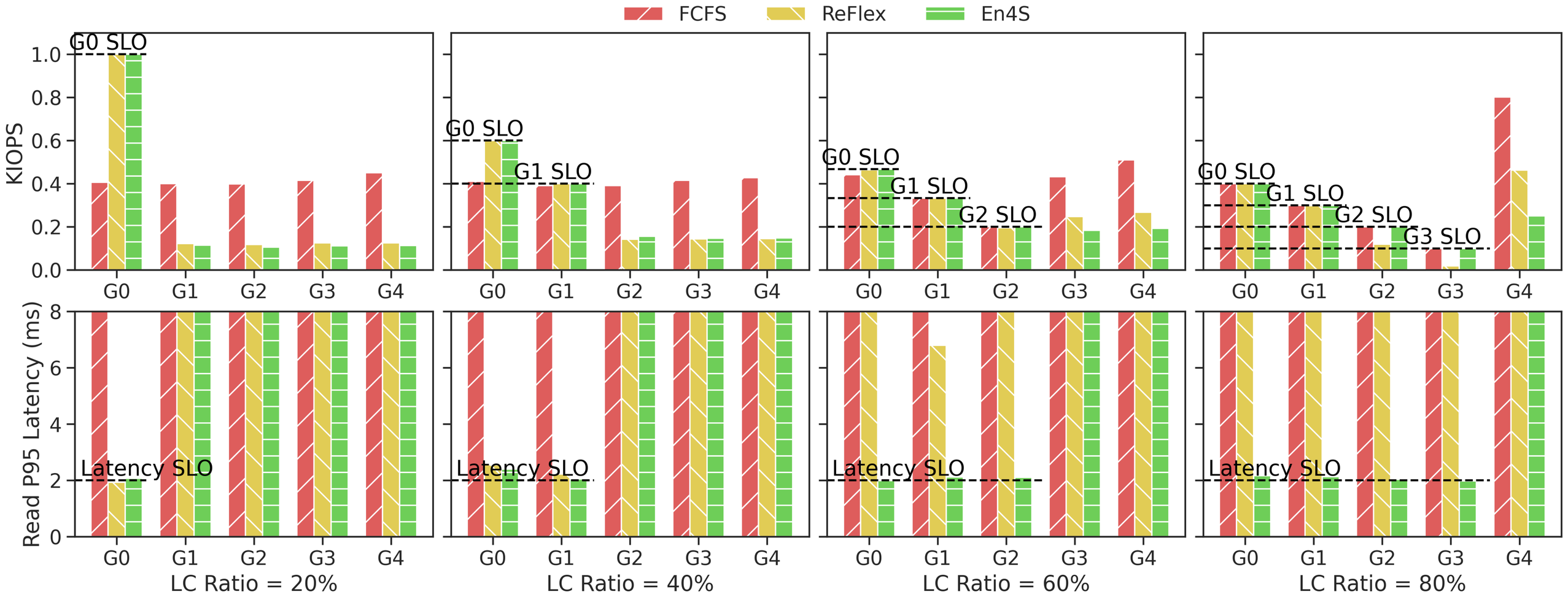




Enforcement at Scale - I



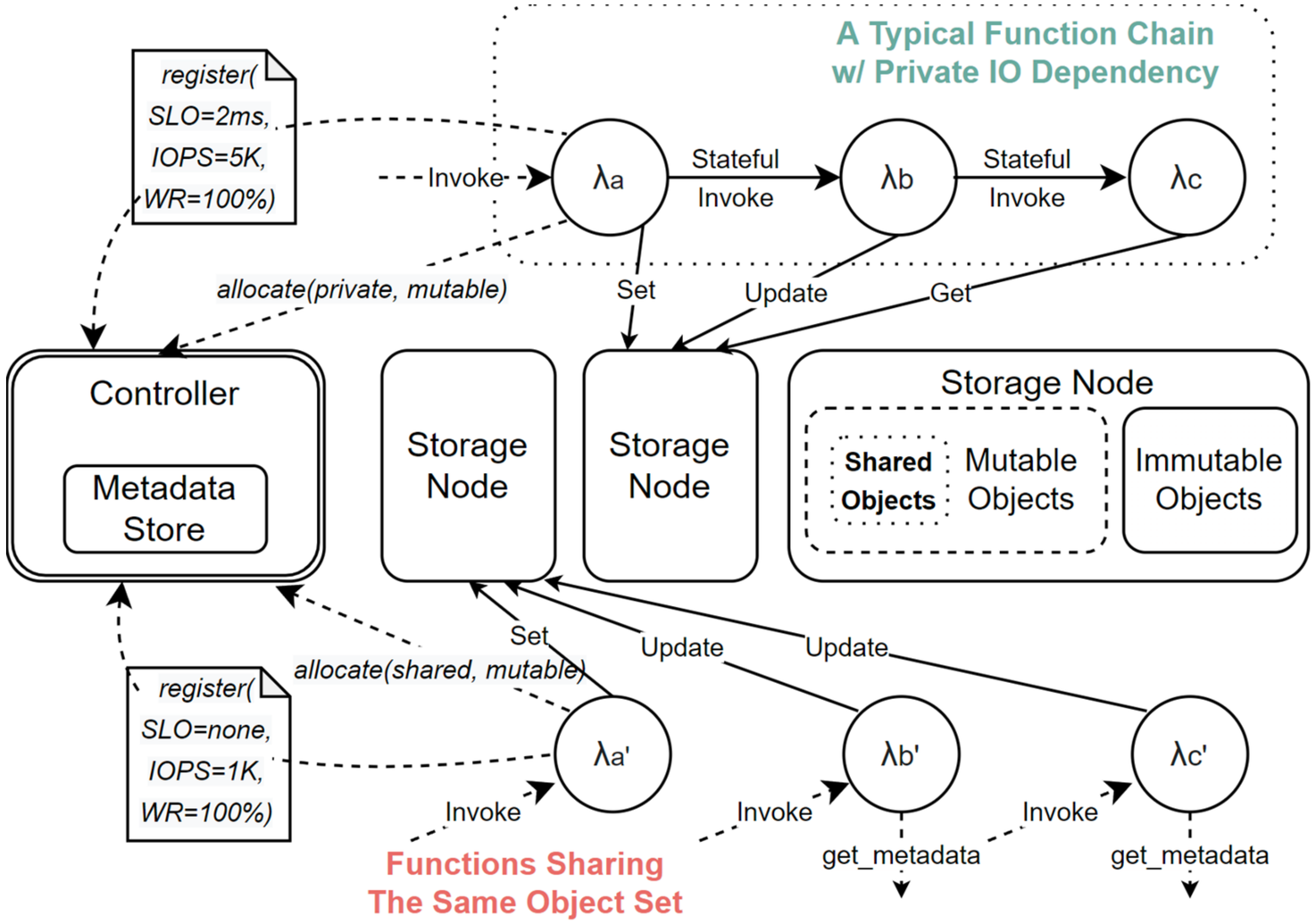
Enforcement at Scale - II



OSS Sub-repositories

- Flash Storage Disaggregation as Ephemeral Storage: mhxie/ESReFlex
  - High-performance TCP-based networked storage stack
- Asynchronous ReFlex Client Library: mhxie/asyncreflex
  - Optimized the storage driver on AWS lambda python runtime w/ Cython
- Ephemeral Storage Cluster: mhxie/LESS-cluster
  - SLO Registration & Flow Control Implementation
  - Efficient Control Channel RPC
  - Scheduler & Metadata Module & AutoScaler
- Ephemeral Storage Benchmark: mhxie/esbench
  - Includes synthetic workloads & real-world workloads

Ephemeral Metadata



Available APIs

Client APIs	Descriptions
__init__(controller, context)	Initiates or loads a context with EMD, returning a handler.
allocate / free(handler, job_ctx)	Explicitly allocates or frees a job with capacity, with job-level SLO hints. allocates storage resources, must called for each new job.
__enter__(handler, flow_hints)	Handler optionally registers or updates a flow with latency/IOPS/(rw_ratio) SLO. The controller will verify job limits and then return the connection to storage nodes.
__exit__(handler)	Handler closes connections to nodes and deregisters the flow for the connection in that job. Flushes all the EMD for all shared mutable objects to the store in the controller.
put / get / update(handler, id, data)	Puts, gets, or updates an object to assigned nodes, returning an object future.
invoke(handler, func_to_contexts)	Invokes stateful functions with encoded contexts and EMD in the payload.

Table 1: Available APIs in En4S Client Library