HotStorage 2021
The 13th ACM Workshop on Hot Topics In Storage and File Systems
July 27-28 Virtual

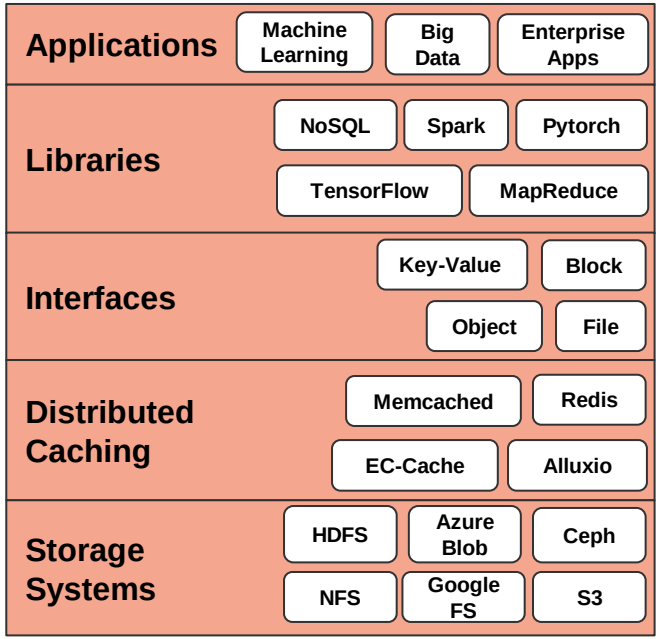
Unifying the Data Center Caching Layer - Feasible? Profitable?

Liana V. Rodriguez, Alexis Gonzalez, Pratik Poudel,
Raju Rangaswami, and Jason Liu

FIU FLORIDA INTERNATIONAL UNIVERSITY

Background

- Cloud Data Centers use **different storage types** (block, file, object , key-value)
- Caches typically target one storage type or one store instance or simply workloads running on a single host or VM
- New devices (3D-Xpoint, flash-based SSD technology) create new requirements for cache optimization
- High performance network fabrics are making remote data access more efficient (40 GigE, RoCE, InfiniBand, etc.)

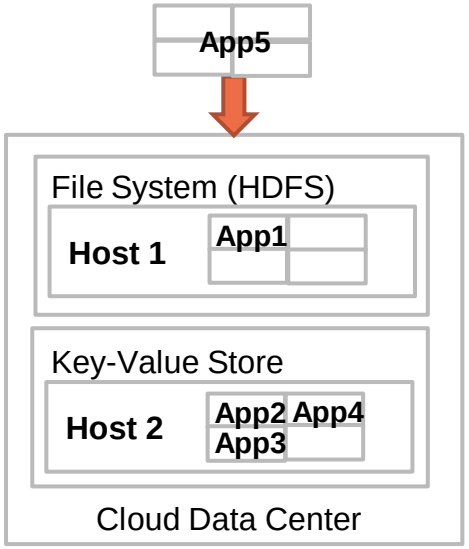


Outline

- Background
- **Motivation**
- Caching as a Service (CaaS)
- Preliminary Study
- Discussion and Future Work

Motivation

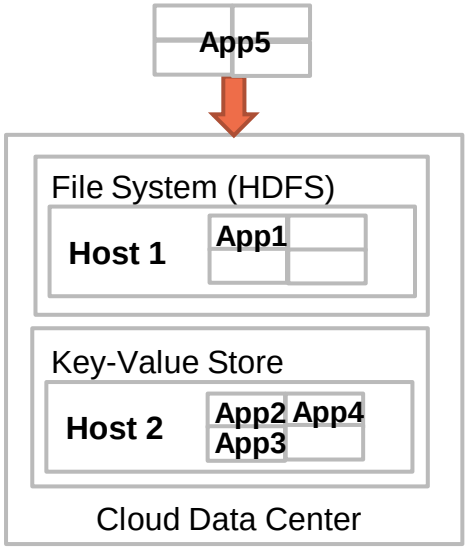
Cache Fragmentation



Cache resources should be shared across storage systems and applications to decrease **resource fragmentation** within the data center.

Motivation

Cache Fragmentation



Cache resources should be shared across storage systems and applications to decrease **resource fragmentation** within the data center.

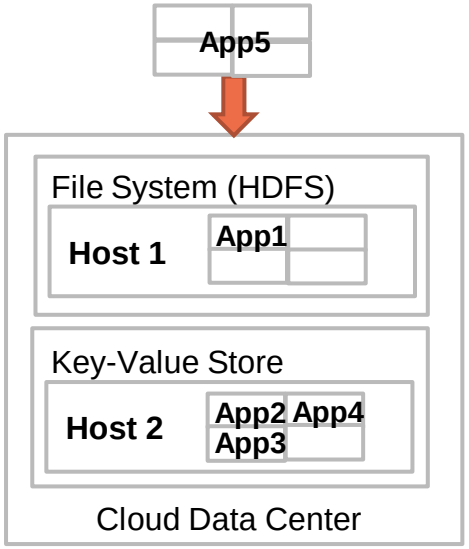
Cache Writes

Store Type	Writes
Block	71%
File	9.8%
Object	40.5%
Key-value	22.5%

Write-dominant workloads are common in today's cloud data centers.

Motivation

Cache Fragmentation



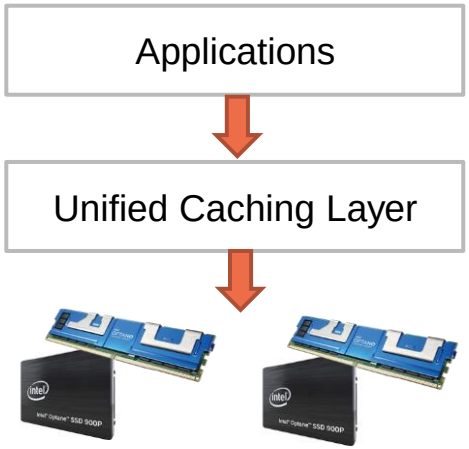
Cache resources should be shared across storage systems and applications to decrease **resource fragmentation** within the data center.

Cache Writes

Store Type	Writes
Block	71%
File	9.8%
Object	40.5%
Key-value	22.5%

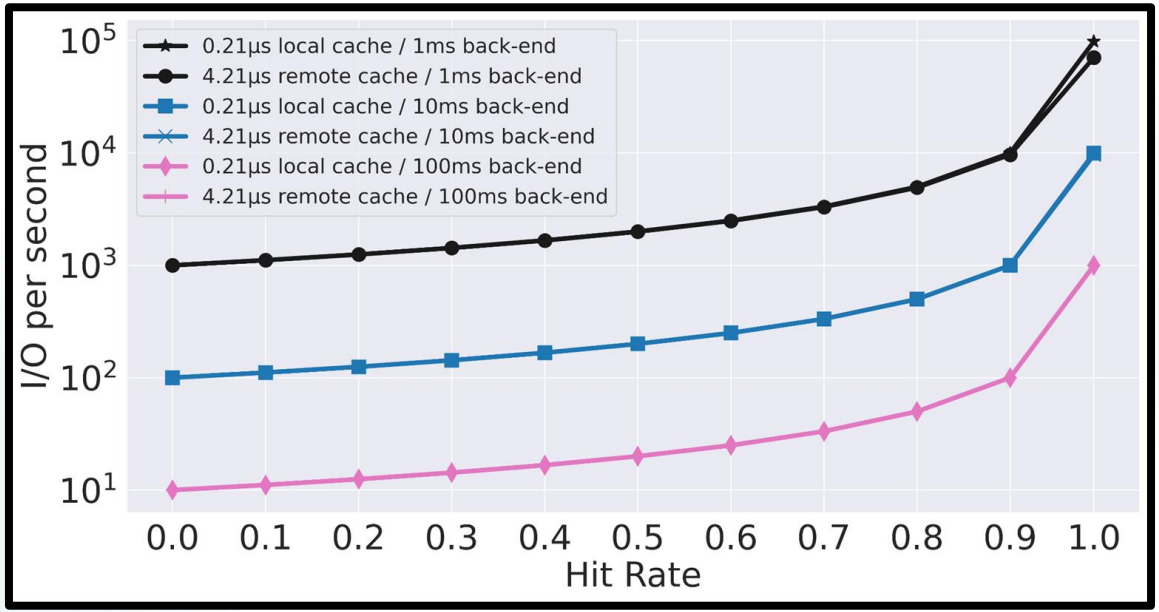
Write-dominant workloads are common in today's cloud data centers.

Cache Devices



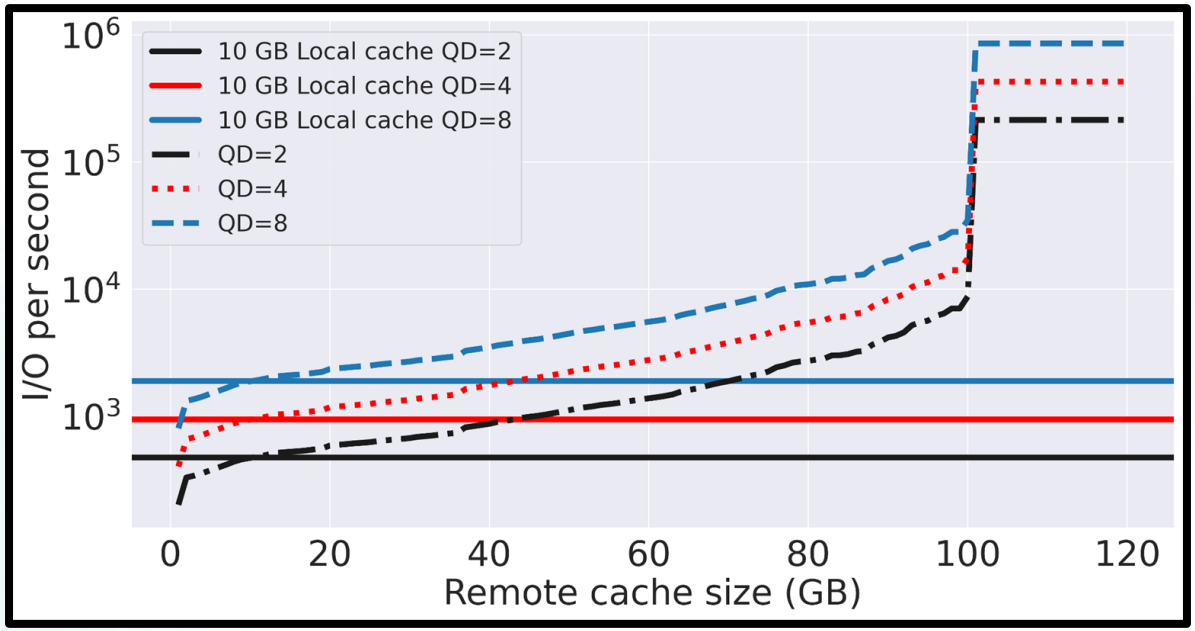
Different devices implies different optimizations for **cost-performance** in the caching layer.

Motivation



Throughput increases 10x in Local Cache and **7x** in Remote Cache when cache hit-rate increases from 90% to 99%, using storage back-end latency of 1 ms. Similar improvements are seen with larger (10ms and 100ms) storage latencies.

Motivation



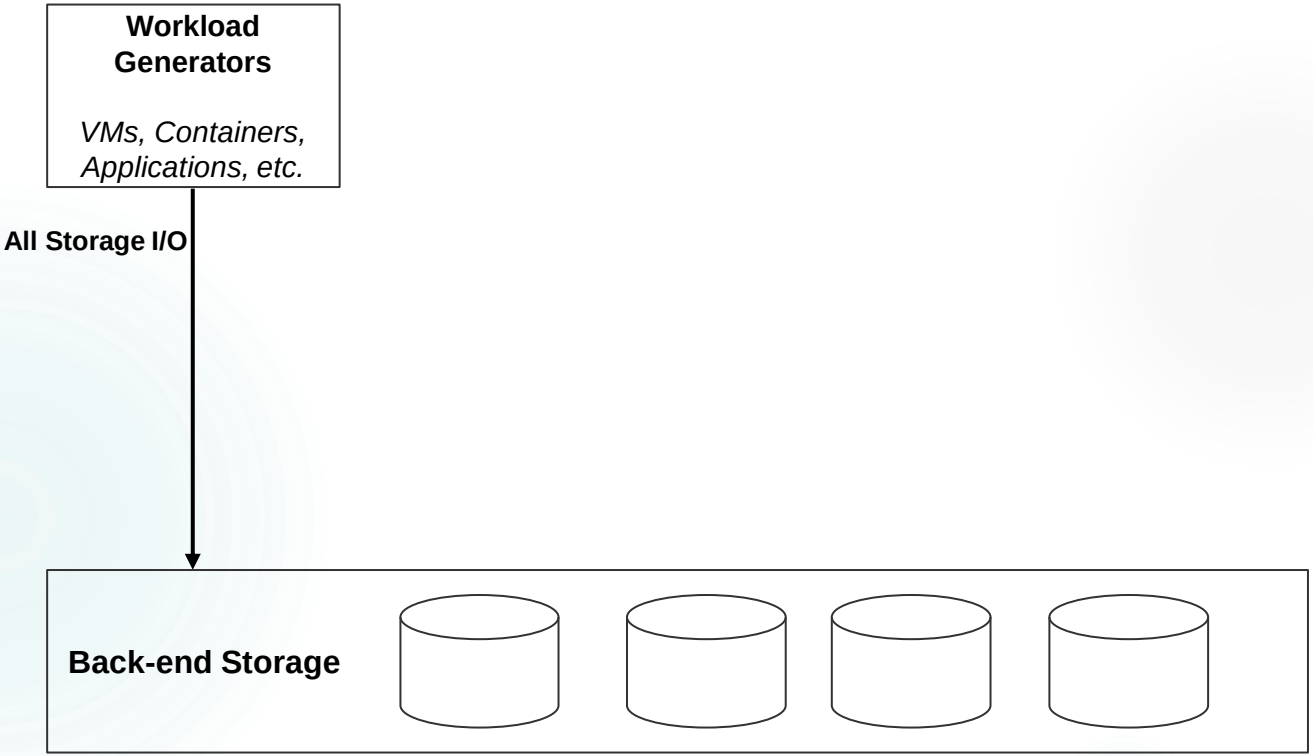
Throughput increases 1.24x with 20GB Remote cache with QD=8

Throughput increases 18.7x with 100GB Remote cache with QD=8

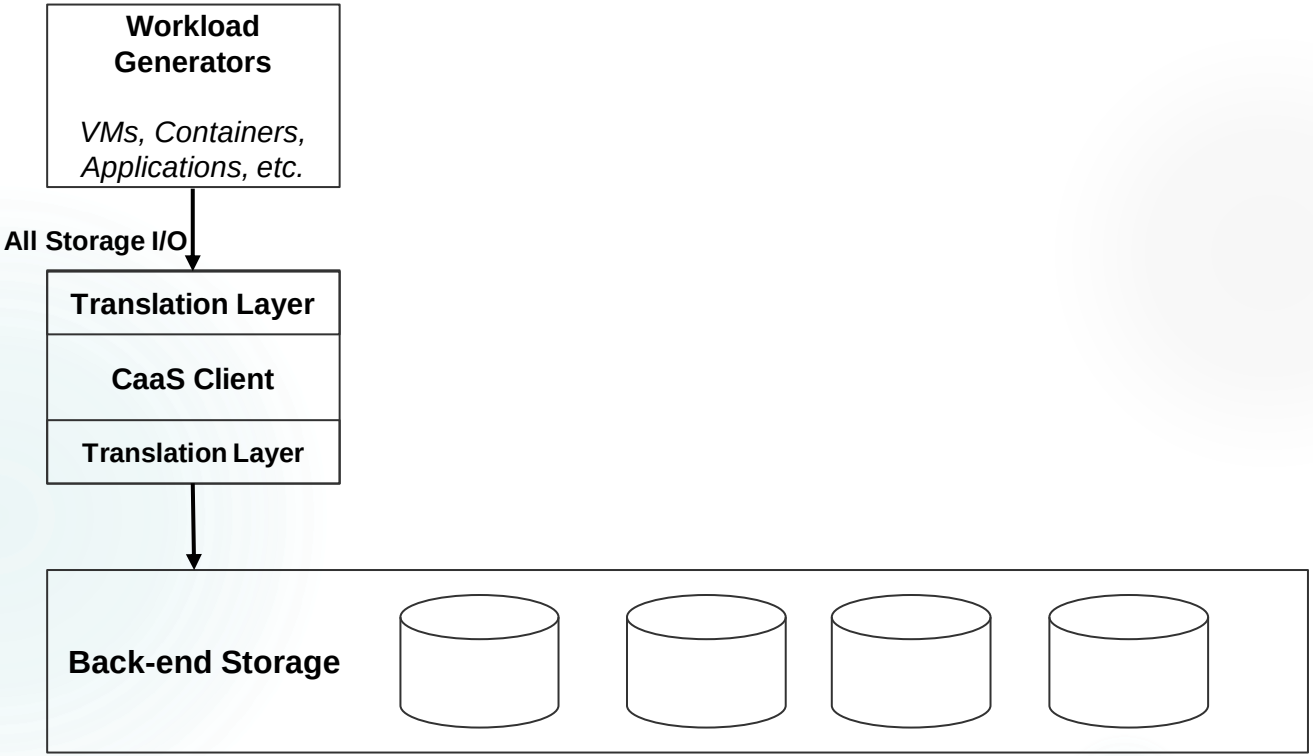
Caching as a Service (CaaS)

A general **distributed caching layer** that abstracts and exposes all the available cache resources in the cloud for all types of storage systems.

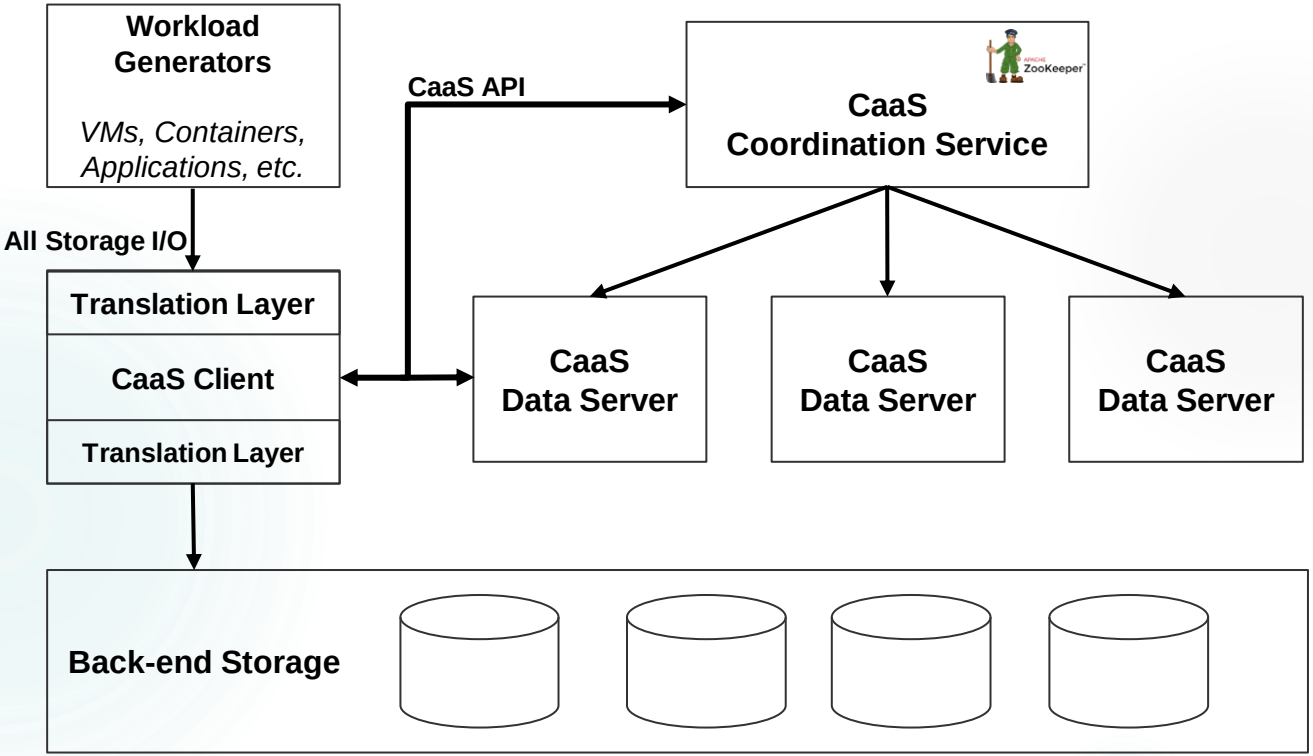
Architecture



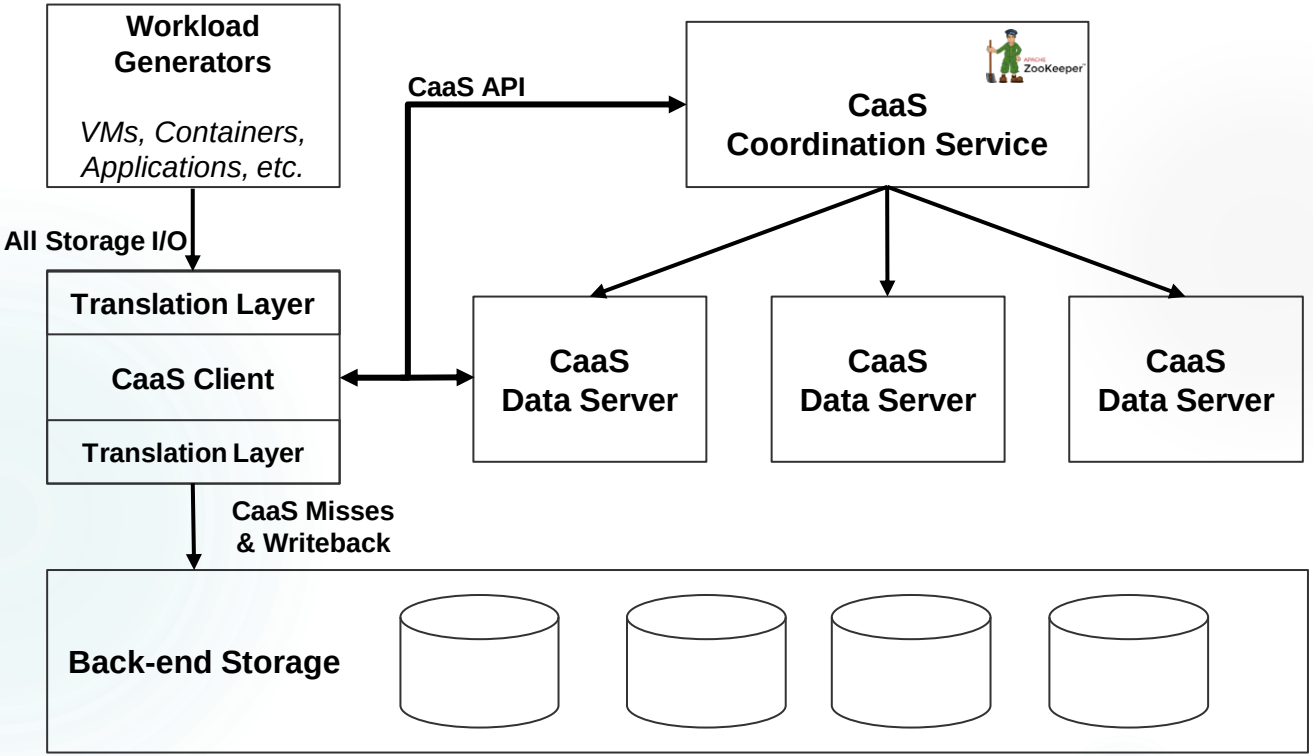
Architecture



Architecture



Architecture



CaaS API

CaaS Coordination/Data Server API

- CaaS API is designed around the concepts of ***Store*** and ***Clice*** (Cache-slice).
  - Stores: Unit of access-protected cache provisioning
  - Clices: Unit of CaaS data access
- CaaS Clients ***register*** with the Coordination Service
- CaaS Clients request clice locations using on-demand ***lookup*** calls
- CaaS Clients access data from CaaS Data Servers using ***Read***, ***writeClean/Dirty***

CaaS API

CaaS Coordination/Data Server API

- CaaS API is designed around the concepts of ***Store*** and ***Clice*** (Cache-slice).
  - Stores: Unit of access-protected cache provisioning
  - Clices: Unit of CaaS data access
- CaaS Clients ***register*** with the Coordination Service
- CaaS Clients request clice locations using on-demand ***lookup*** calls
- CaaS Clients access data from CaaS Data Servers using ***Read***, ***writeClean/Dirty***

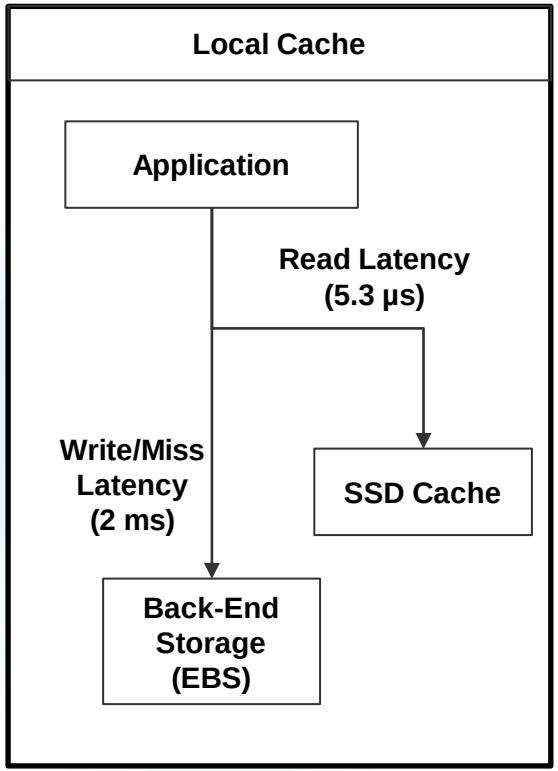
CaaS Client API

- ***Write-back*** calls atomically write inter-dependent dirty data to storage back-end

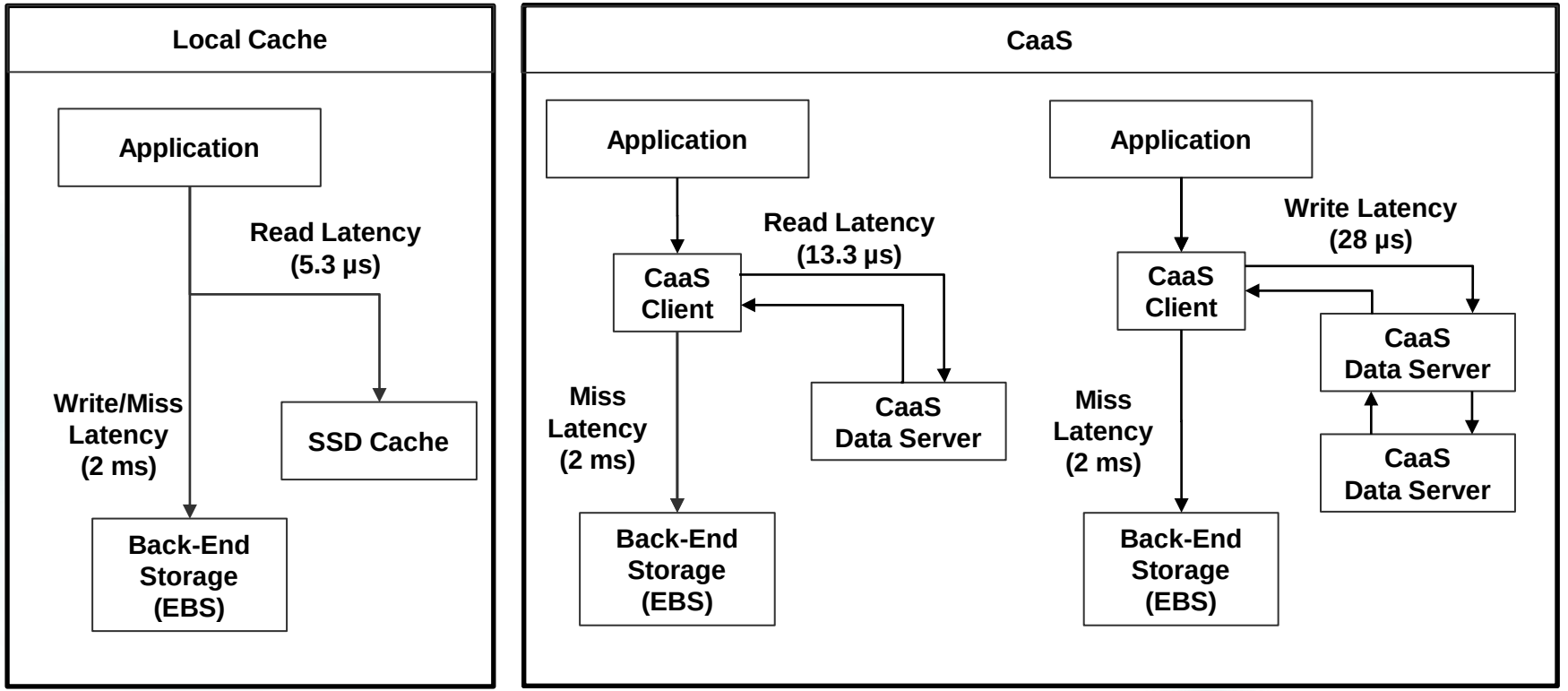
Experimental Setup

- 363 Block Storage Workloads.
  - User home/project directories; Web-based servers; Webpage; Online course management system; Hardware monitoring; Source control; Web staging; Terminal; Web/SQL; Media; Test web; Firewall/web proxy; VMware VMs.
- 40 Gbit Ethernet network latency equal to 4µs. Back-end storage latency (AWS EBS) equal to 2ms
- ARC replacement algorithm in SSD-based Local Cache and CaaS
- Cache simulation used to obtain the number of Read Hits
- Writes in local cache are simulated misses while writes in CaaS are simulated hits
- Writeback fault-tolerance in CaaS simulated using Primary-Backup replication with one leader and one follower

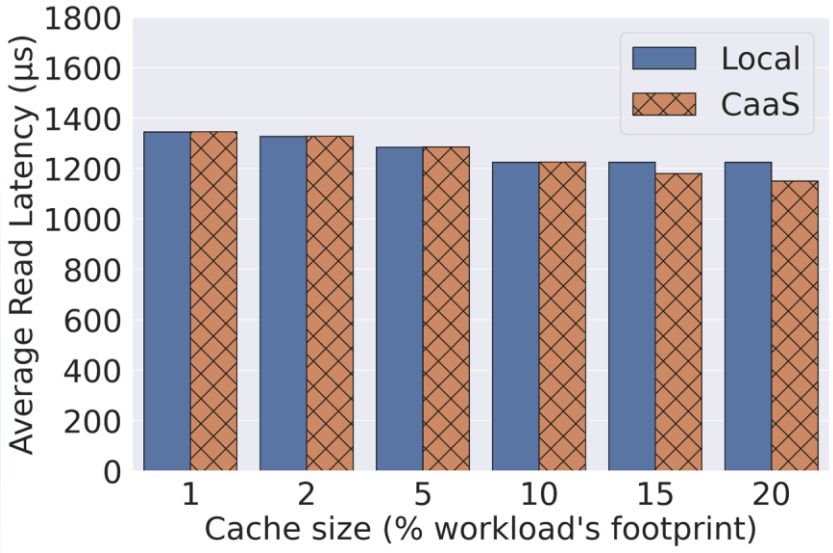
Preliminary Study Model

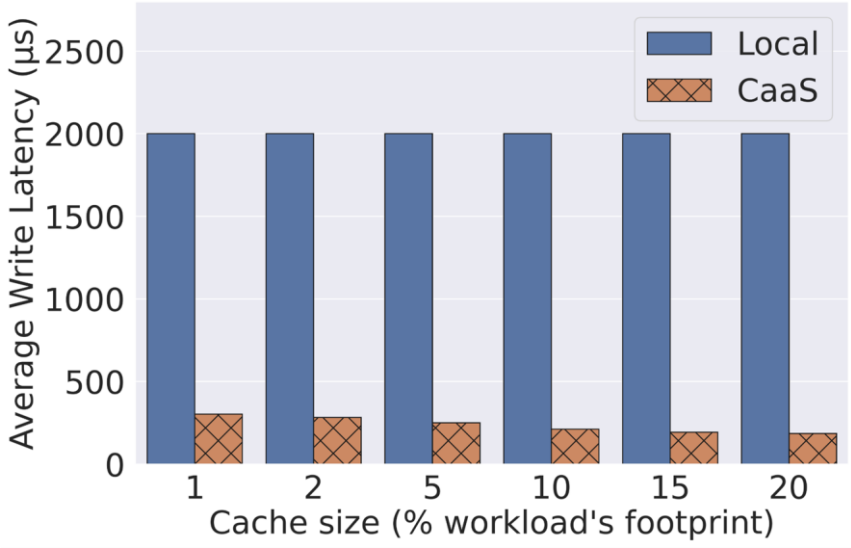


Preliminary Study Model



Preliminary Study Results

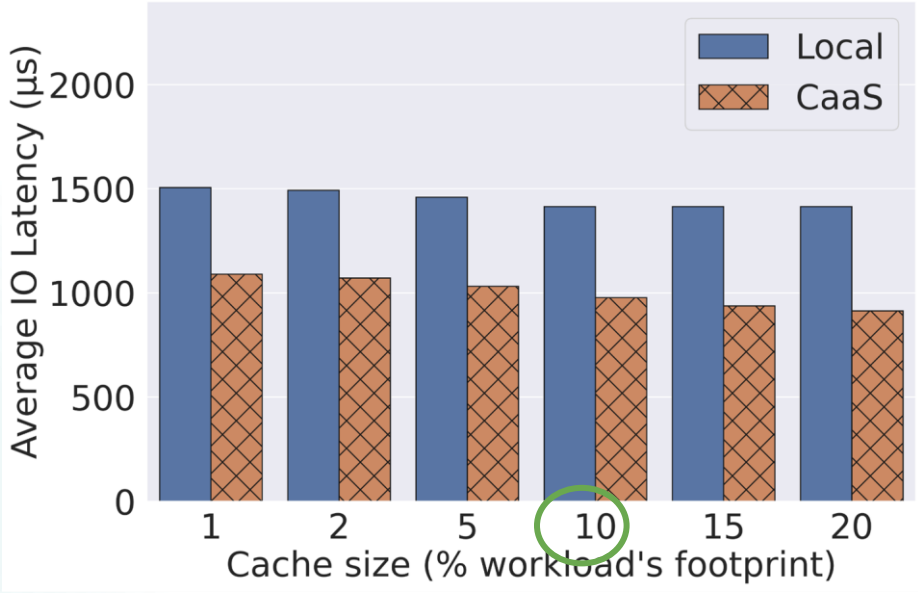




Read Latency up to 6%

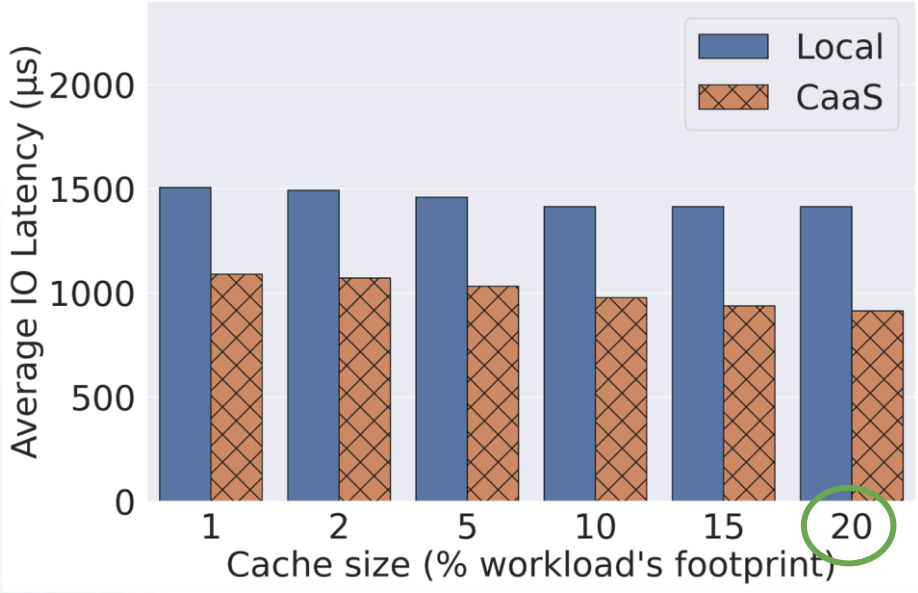
Write Latency up to 90%

Preliminary Study Results



I/O Latency 31% improvement

Preliminary Study Results



I/O Latency 36% improvement

Future Work

- **Writeable caching:** Design a caching layer that provides the fault-tolerance and consistency requirements.
- **Cache unit size:** Select the *clice size(s)* in CaaS to optimize for both cache fragmentation and metadata overhead.
- **Cache management:** Design an allocation scheme and the corresponding eviction strategies.
- **Data placement and Load balancing:** Policies for data placement and load balancing decisions across different CaaS data servers.
- **Service Level Agreements (SLAs):** A distributed cache resource allocation algorithm that guarantees a user-defined level of Quality of Service (QoS).

Thanks!

- lvald108@fiu.edu
- http://sylab-srv.cs.fiu.edu

FIU | FLORIDA INTERNATIONAL UNIVERSITY