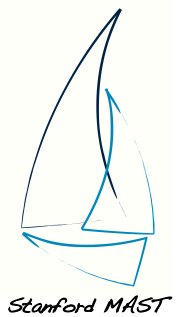
ReFlex: Remote Flash at the Performance of Local Flash

Ana Klimovic Heiner Litz
Christos Kozyrakis

Flash Memory Summit 2017
Santa Clara, CA



Flash Memory Summit

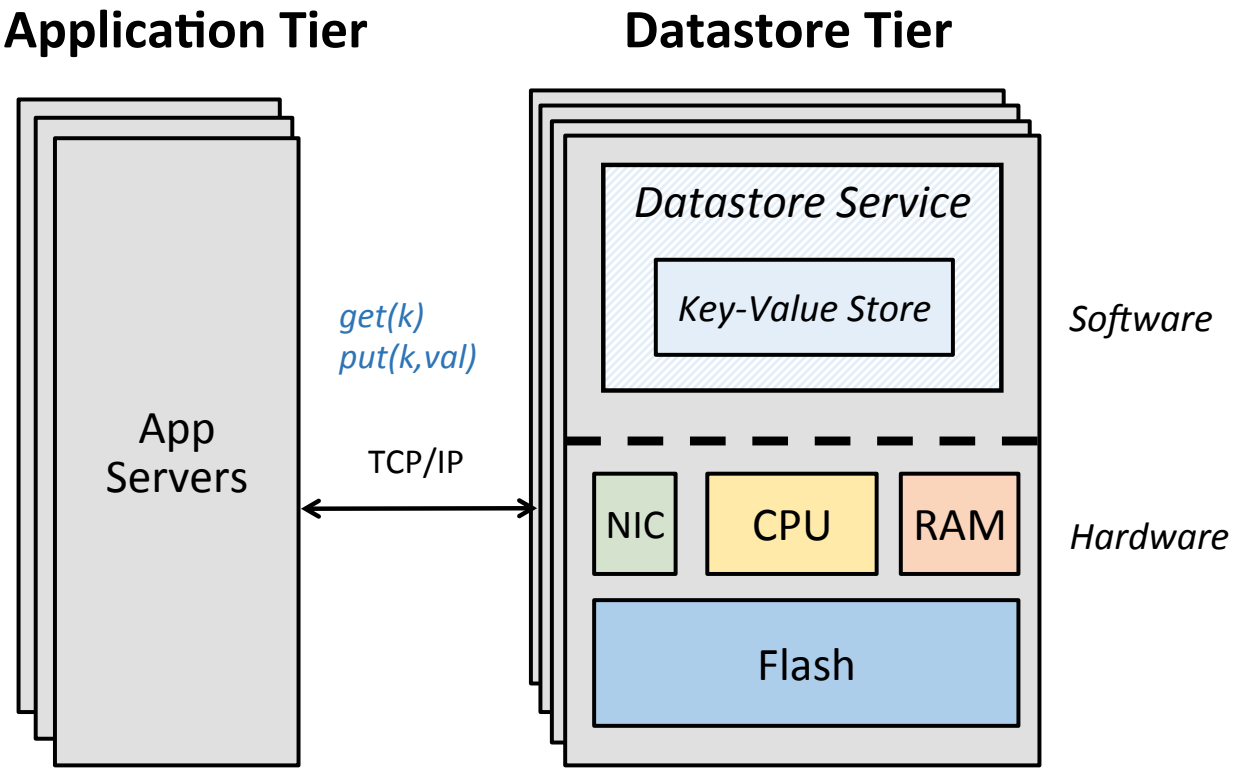
Flash in Datacenters

- NVMe Flash
  - 1,000x higher throughput than disk (1MIOPS)
  - 100x lower latency than disk (50-70usec)

- But Flash is often underutilized due to imbalanced resource requirements

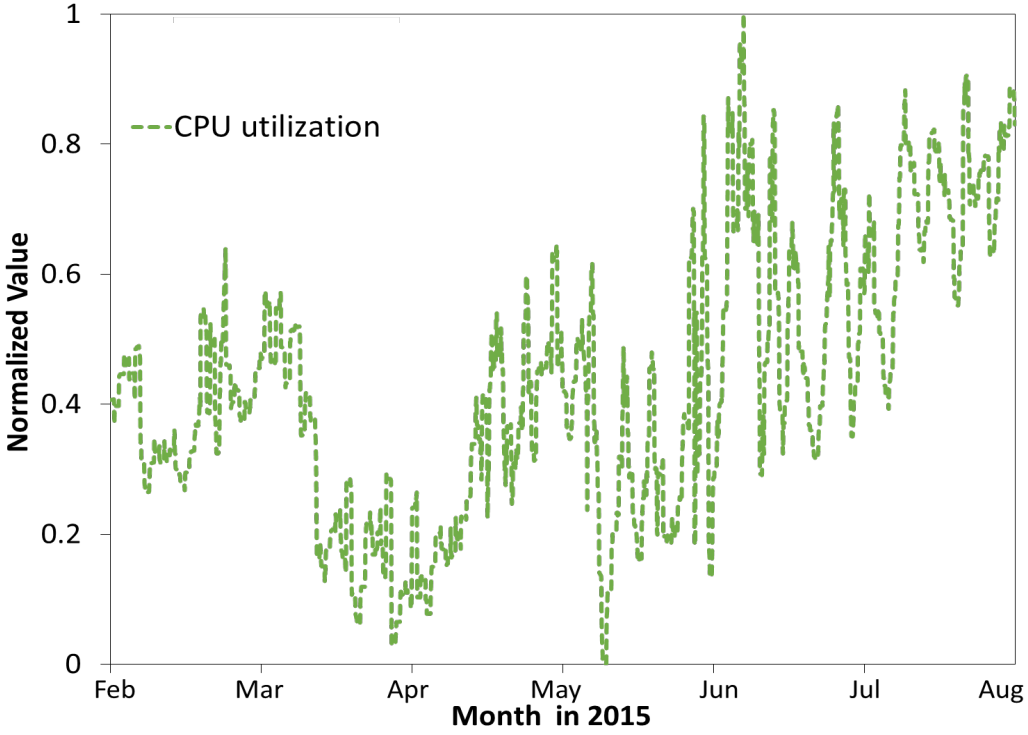
Flash Memory Summit

Example Datacenter Flash Use-Case



Imbalanced Resource Utilization

Sample utilization of Facebook servers hosting a Flash-based key-value store over 6 months

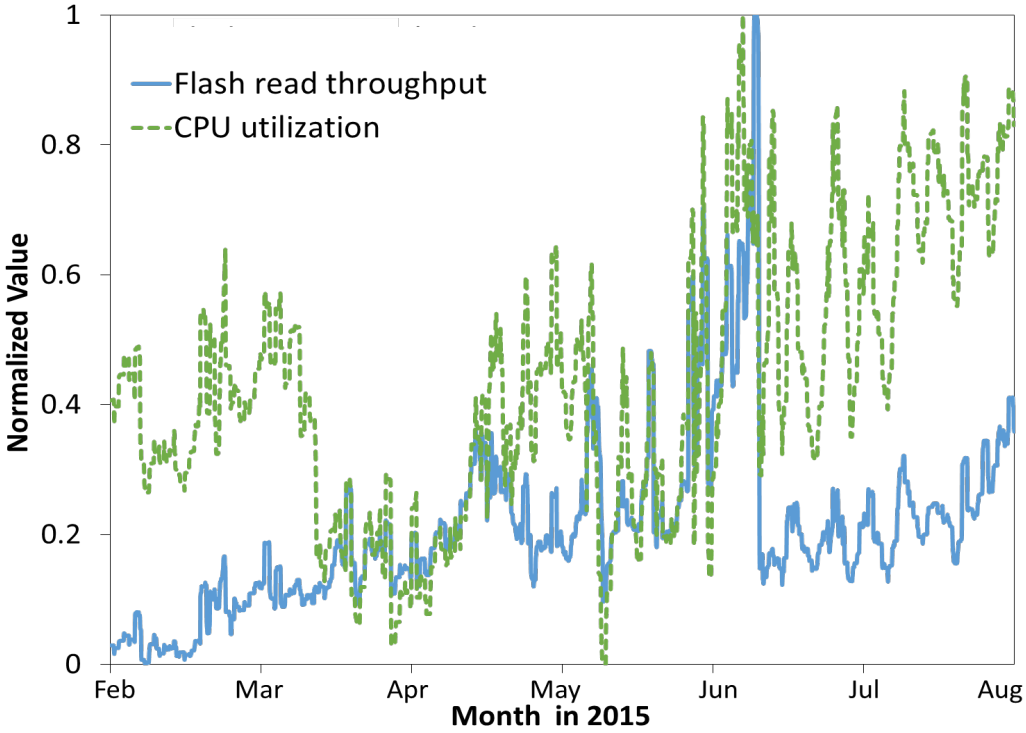


[EuroSys'16] *Flash storage disaggregation*. Ana Klimovic, Christos Kozyrakis, Eno Thereska, Binu John, Sanjeev Kumar.

Flash Memory Summit

Imbalanced Resource Utilization

Sample utilization of Facebook servers hosting a Flash-based key-value store over 6 months

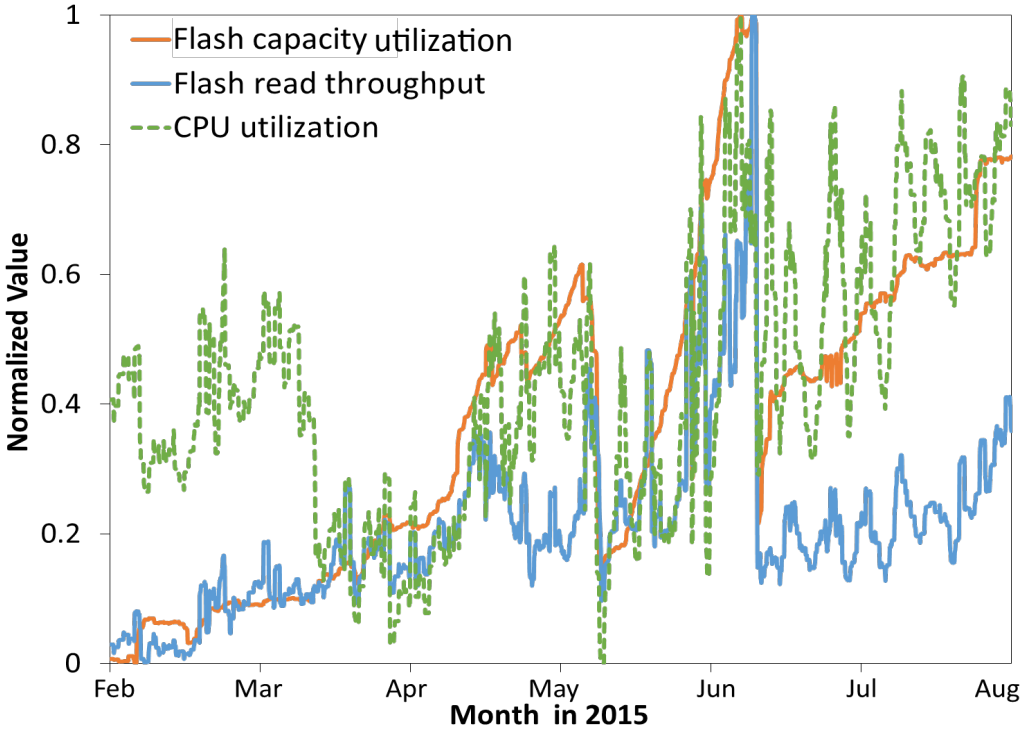


[EuroSys'16] *Flash storage disaggregation*. Ana Klimovic, Christos Kozyrakis, Eno Thereska, Binu John, Sanjeev Kumar.

Flash Memory Summit

Imbalanced Resource Utilization

Sample utilization of Facebook servers hosting a Flash-based key-value store over 6 months

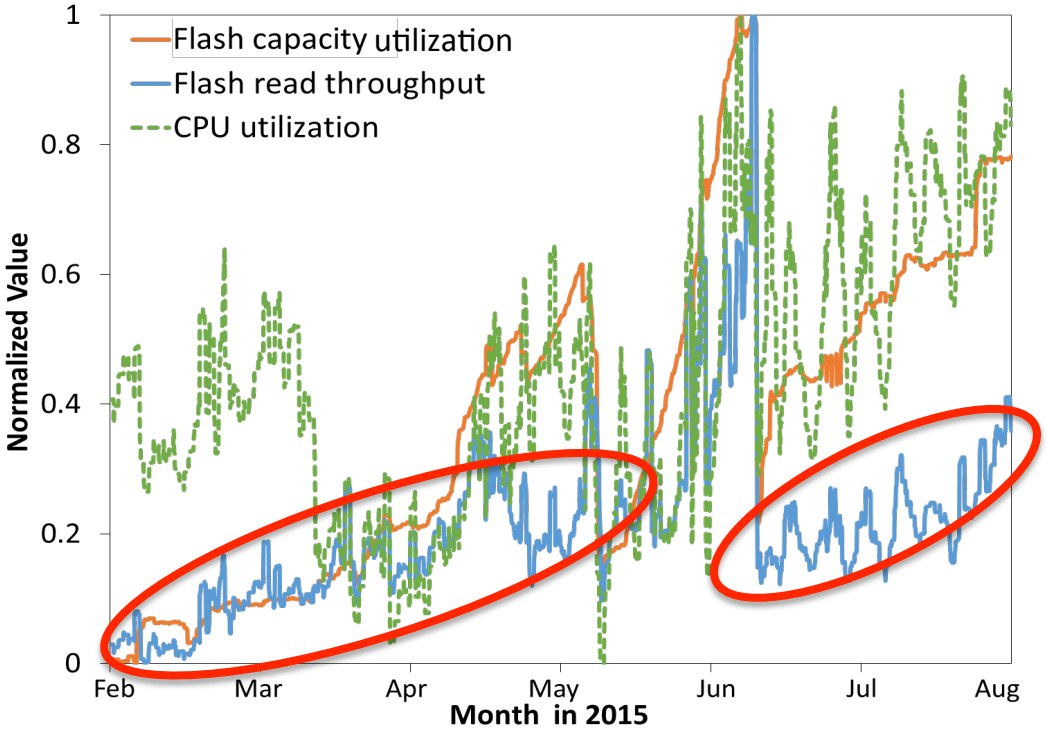


[EuroSys'16] *Flash storage disaggregation*. Ana Klimovic, Christos Kozyrakis, Eno Thereska, Binu John, Sanjeev Kumar.

Flash Memory Summit

Imbalanced Resource Utilization

Flash capacity and bandwidth are underutilized for long periods of time



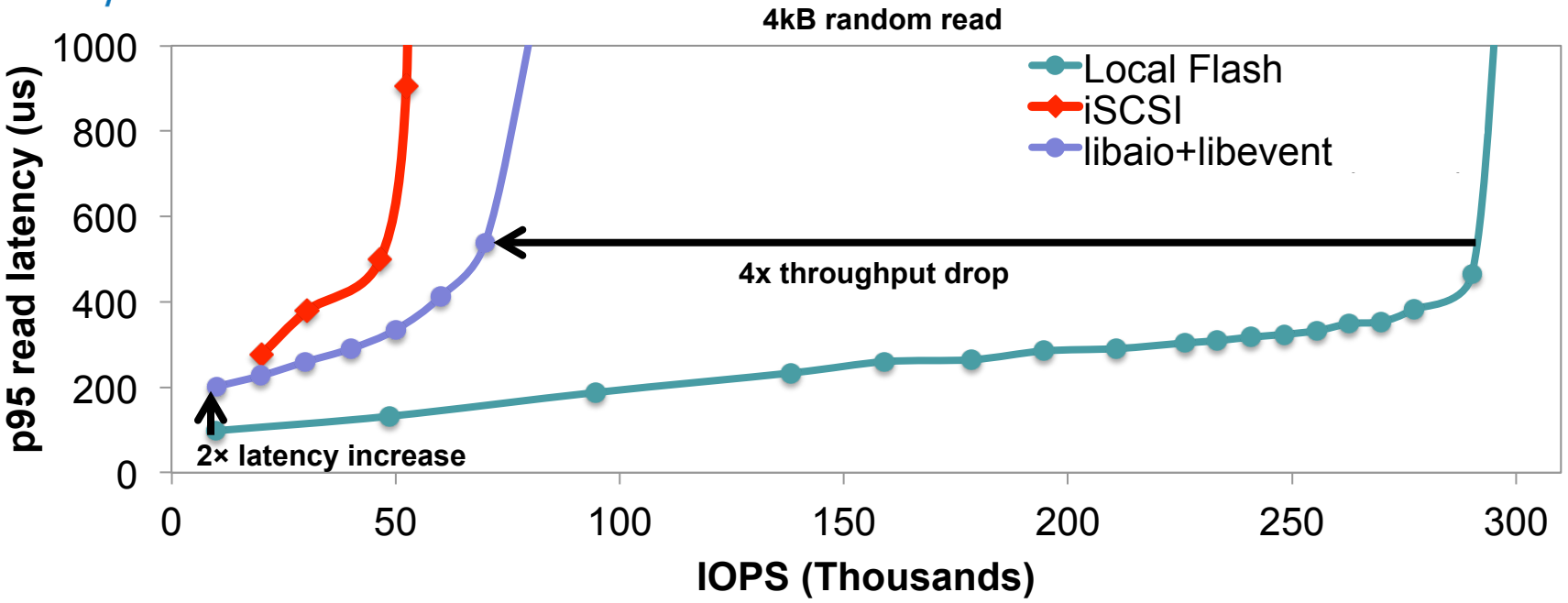
[EuroSys'16] *Flash storage disaggregation*. Ana Klimovic, Christos Kozyrakis, Eno Thereska, Binu John, Sanjeev Kumar

Solution: Remote Access to Storage

- Improve resource utilization by sharing NVMe Flash between remote tenants

- There are 3 main concerns:
  1. Performance overhead for remote access
  2. Interference on shared Flash
  3. Flexibility of Flash usage

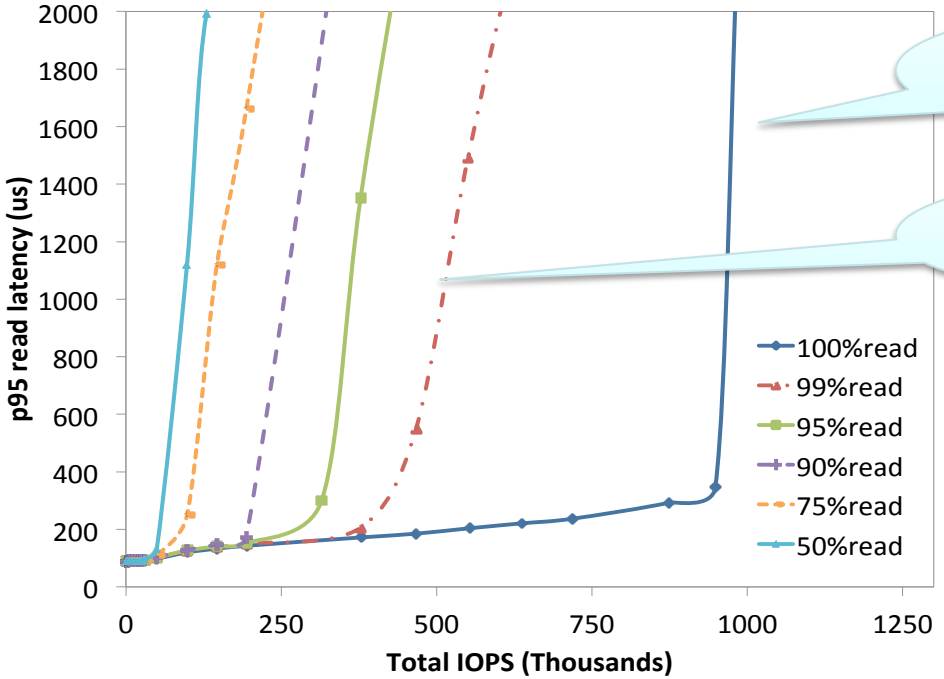
Issue 1: Performance Overhead



- Traditional network/storage protocols and Linux I/O libraries (e.g. libaio, libevent) have high overhead

Flash Memory Summit

Issue 2: Performance Interference



Latency depends on IOPS load

Writes impact read tail latency

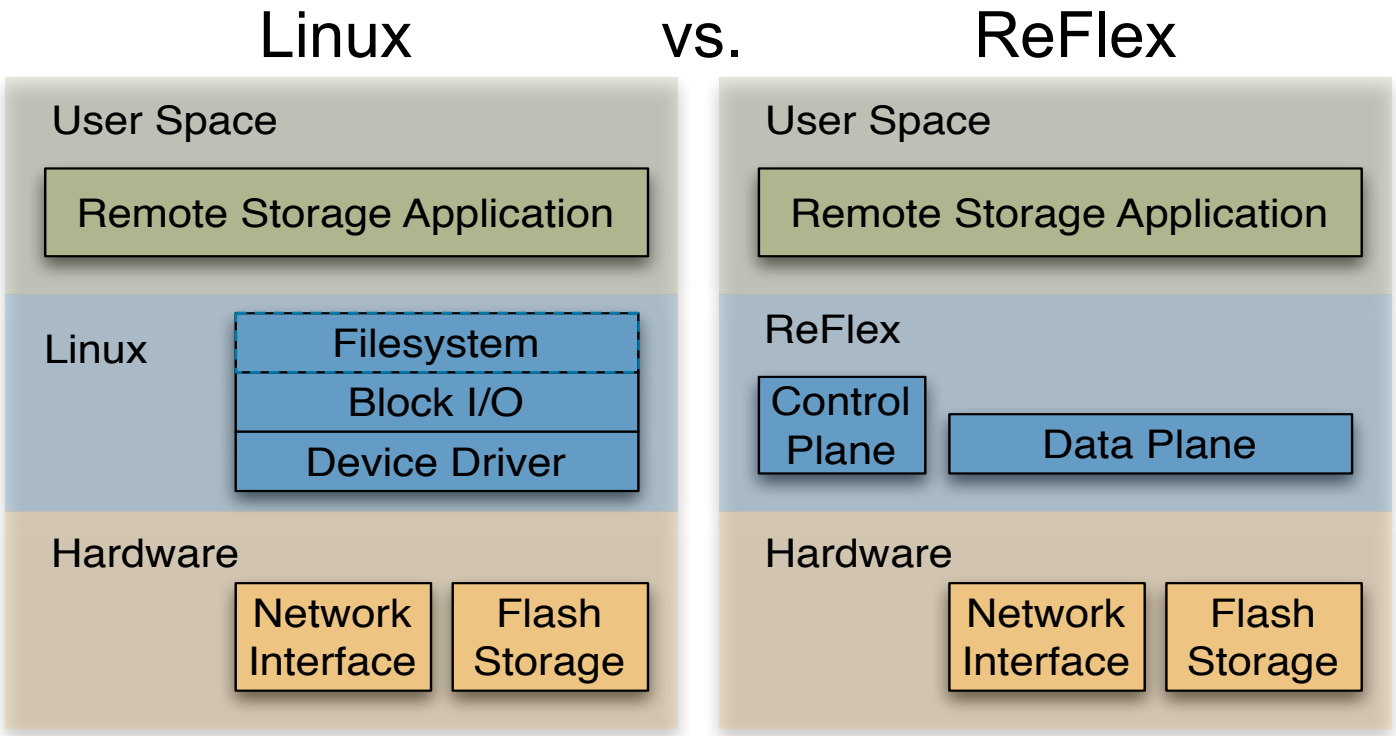
To share Flash, we need to enforce performance isolation

Issue 3: Flexibility

- Flexibility is key
- Want to allocate Flash capacity and bandwidth for applications:
  - On any machine in the datacenter
  - At any scale
  - Accessed using any network-storage protocol

Flash Memory Summit

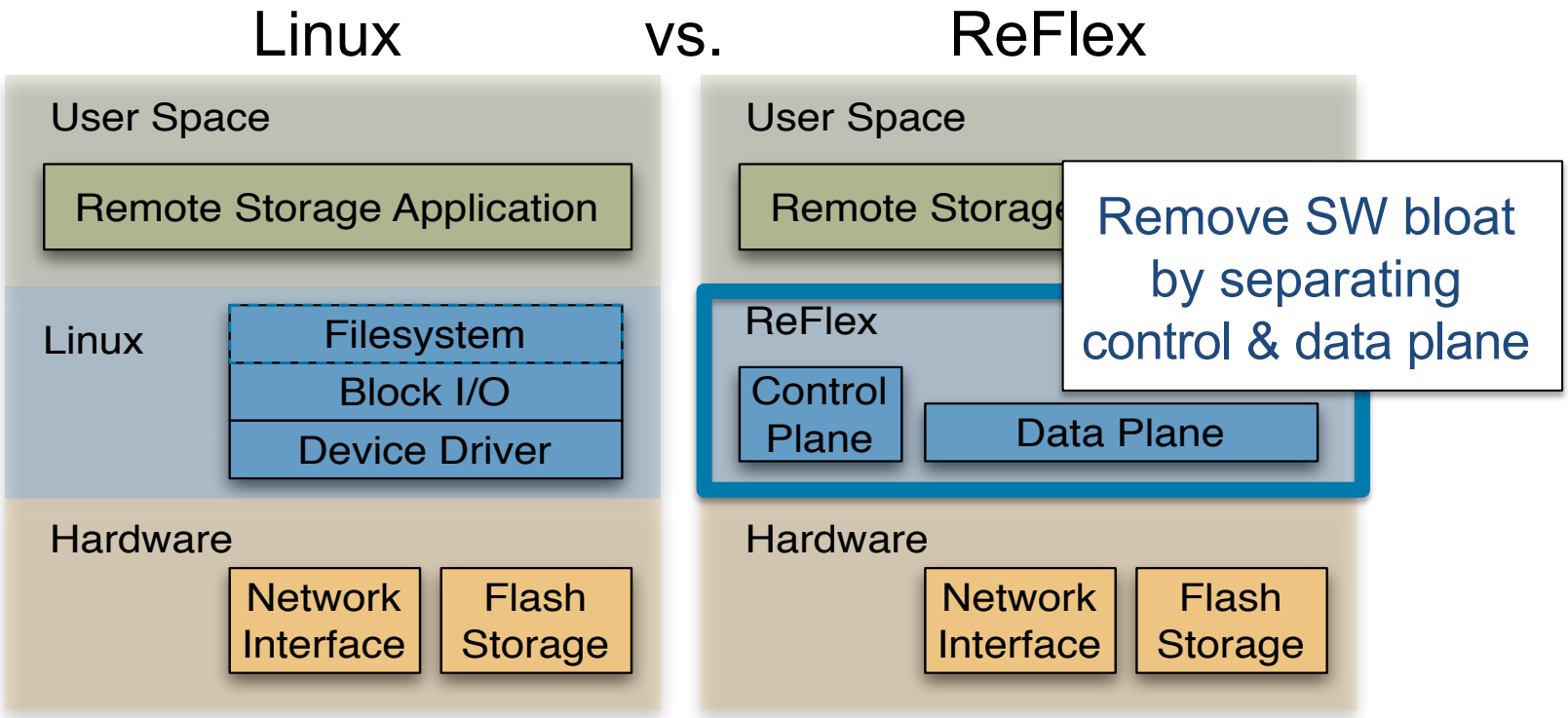
How does ReFlex achieve high performance?



[ASPLOS'17] *ReFlex: Remote Flash ≈ Local Flash*. Ana Klimovic, Heiner Litz, Christos Kozyrakis.

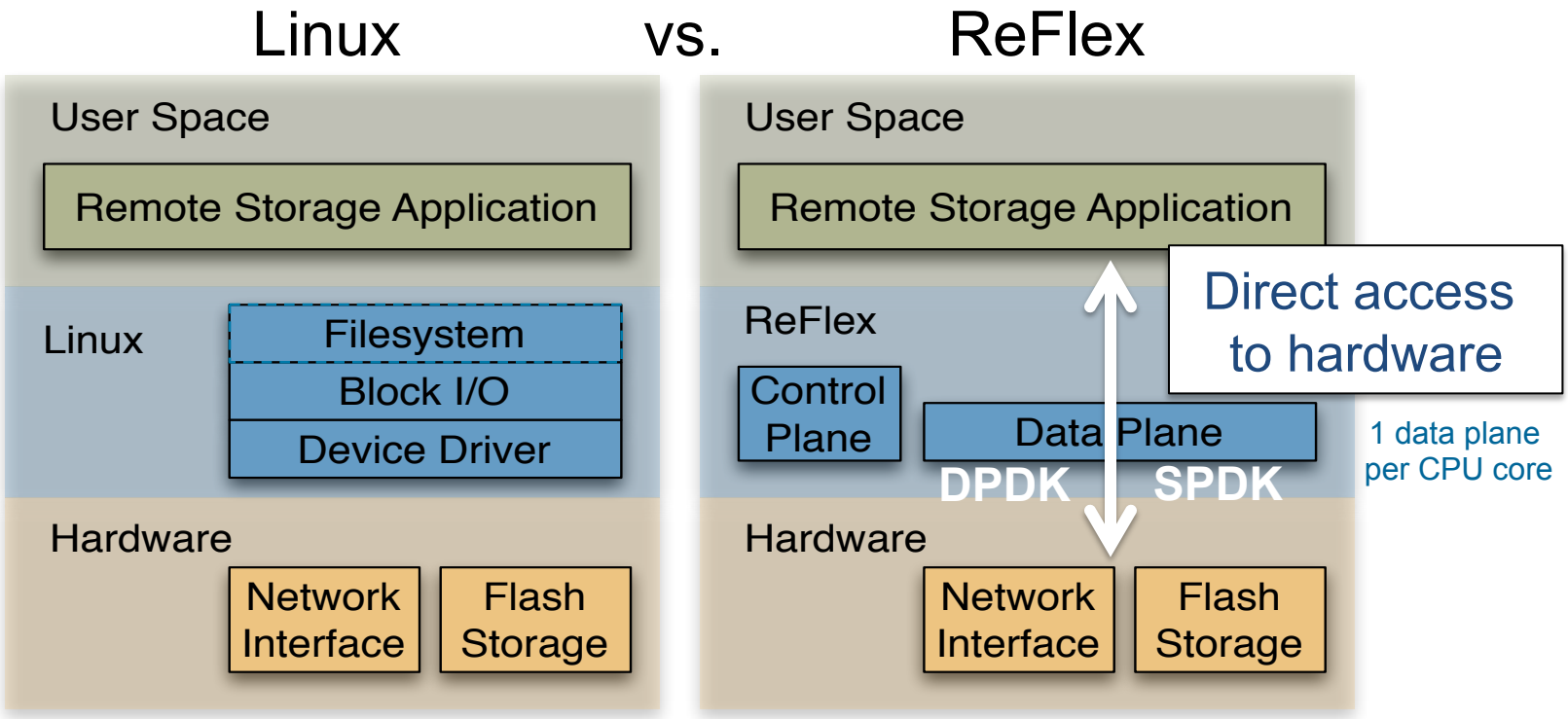
Flash Memory Summit

How does ReFlex achieve high performance?



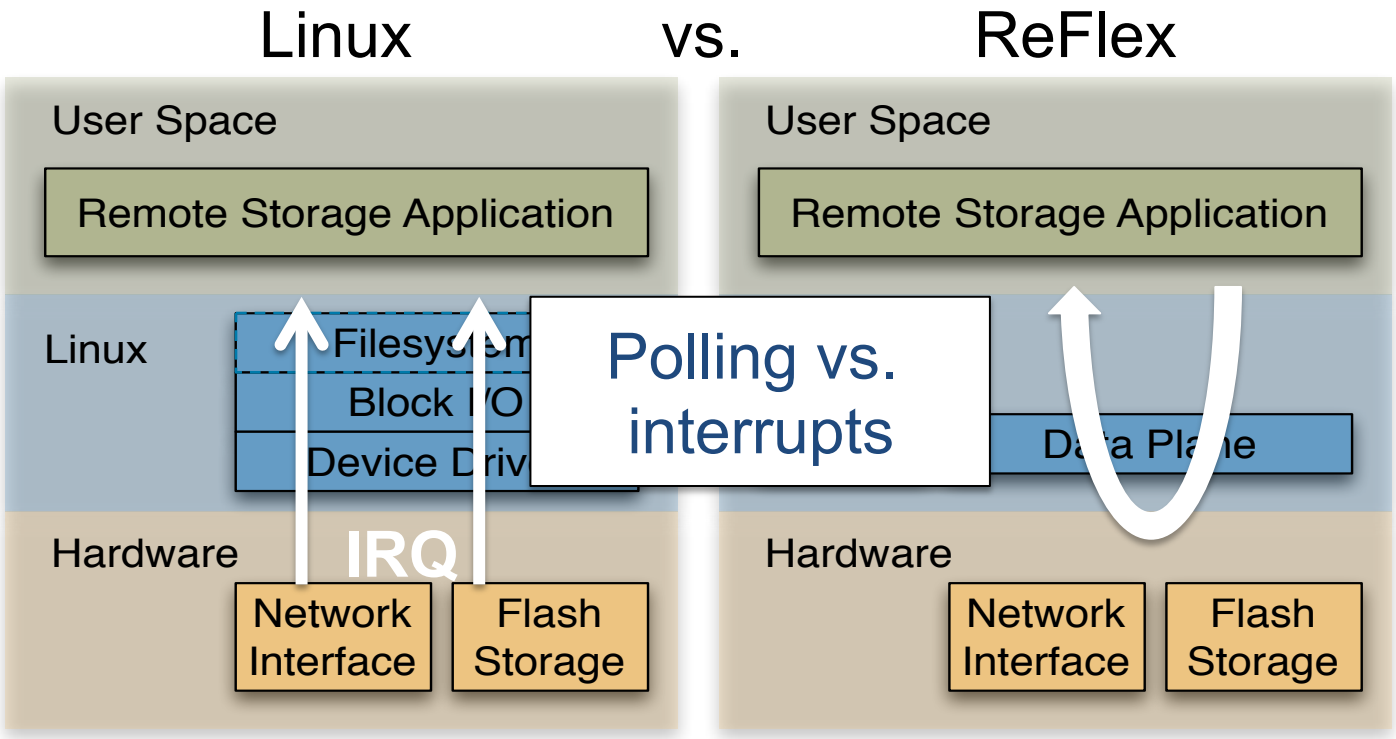
Flash Memory Summit

How does ReFlex achieve high performance?



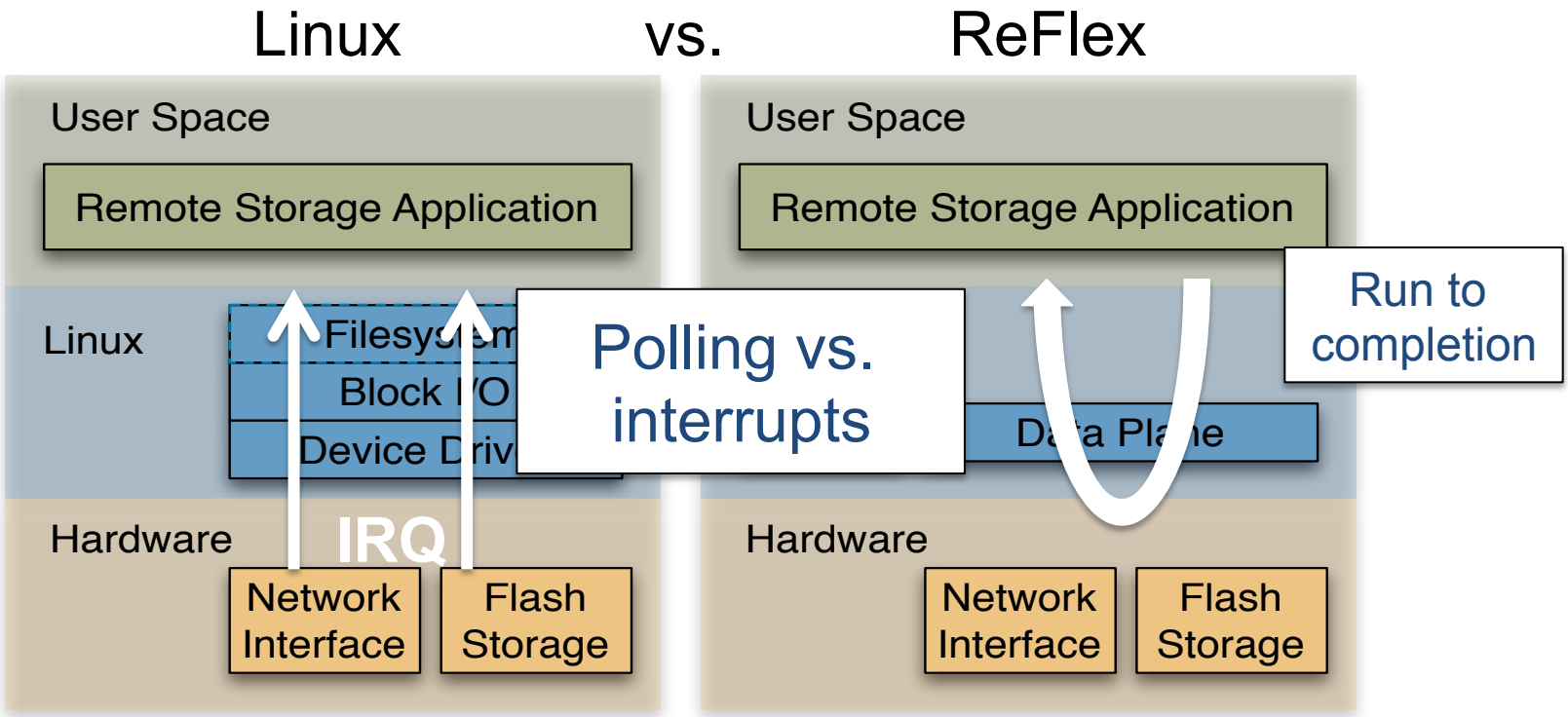
Flash Memory Summit

How does ReFlex achieve high performance?

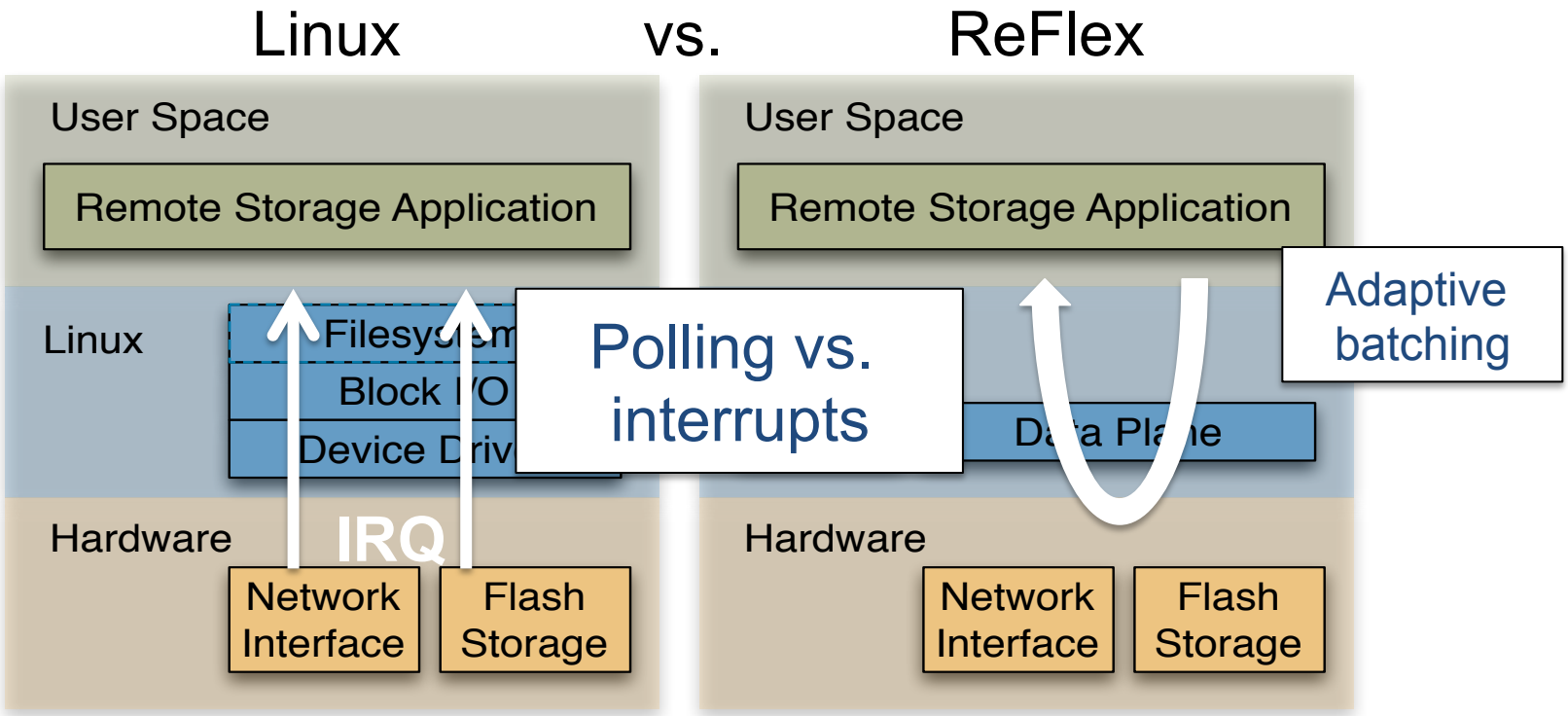


Flash Memory Summit

How does ReFlex achieve high performance?

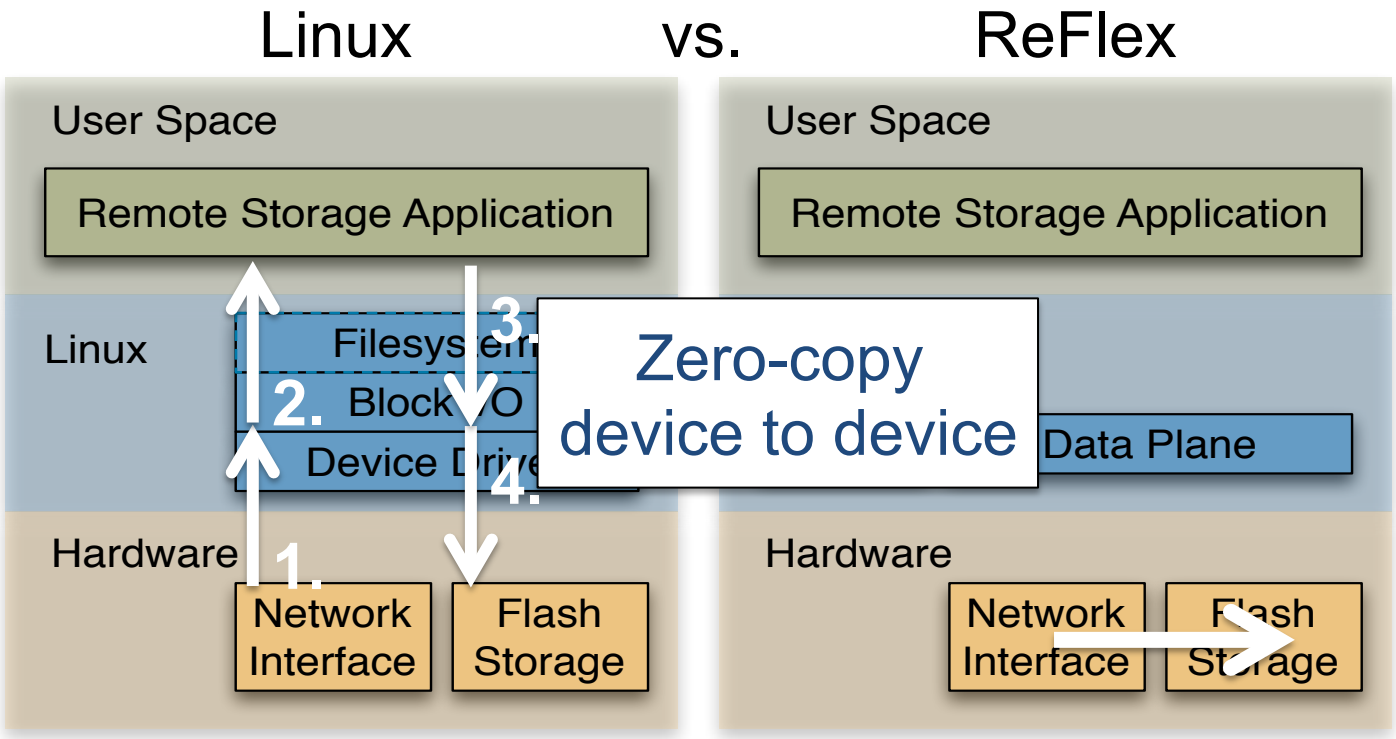


How does ReFlex achieve high performance?



Flash Memory Summit

How does ReFlex achieve high performance?



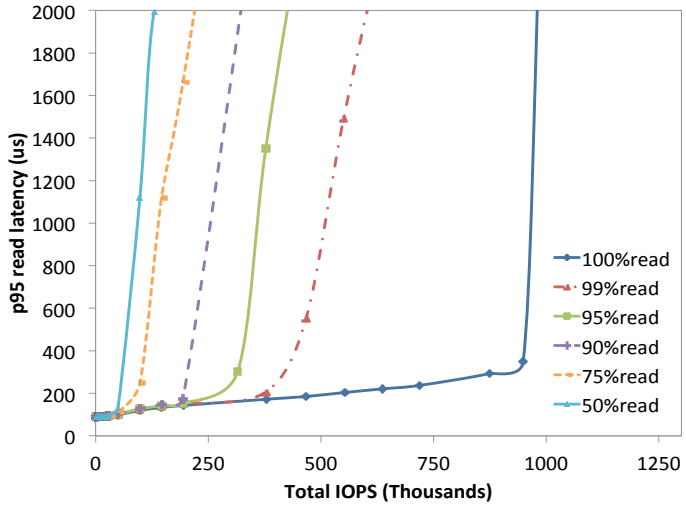
How does ReFlex enable performance isolation?

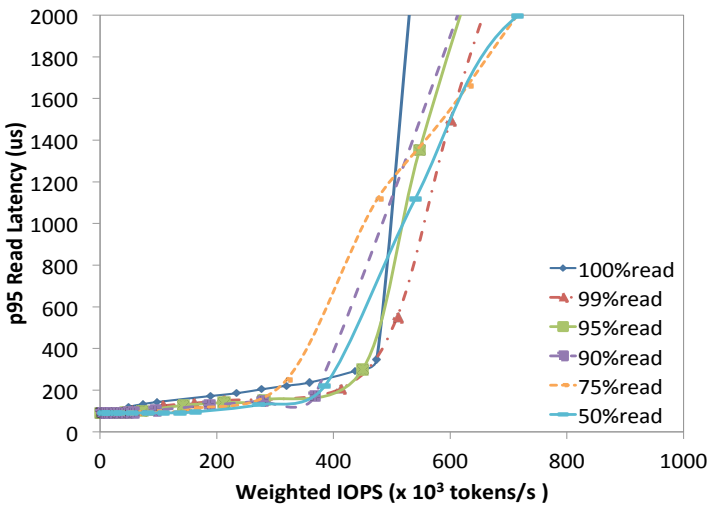
- Request cost based scheduling
- Determine the impact of tenant A on the **tail latency** and **IOPS** of tenant B
- Control plane assigns tenants with a **quota**
- Data plane enforces quotas through throttling

Request Cost Modeling

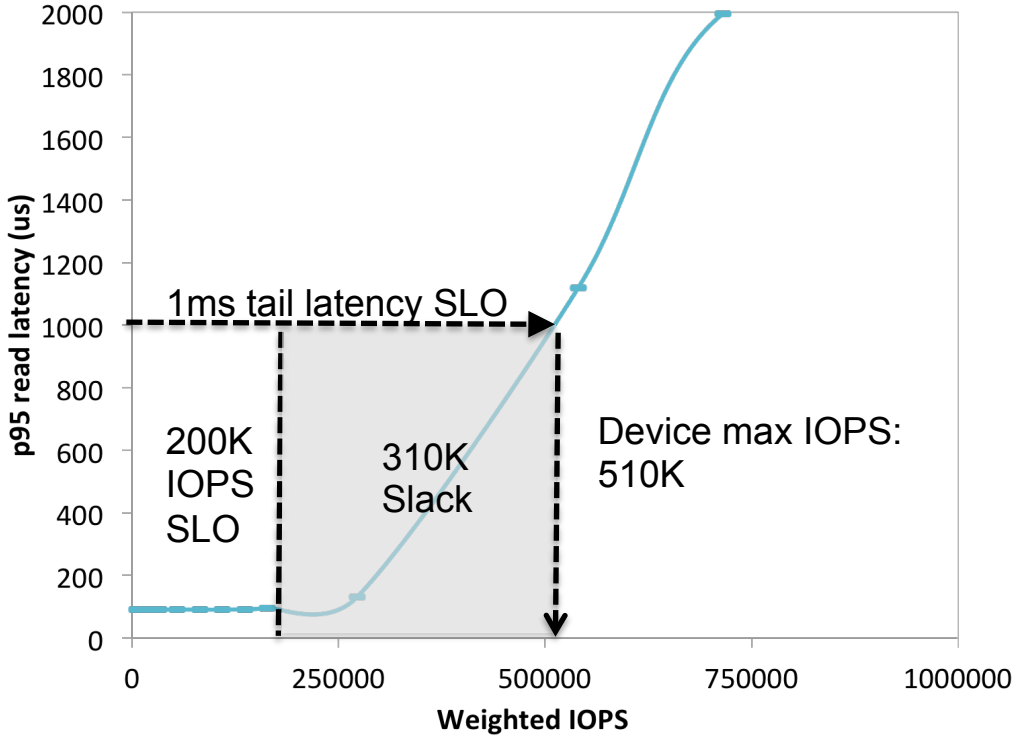
Compensate for read-write asymmetry

For this device:
Write == 10x Read





Request Cost Based Scheduling

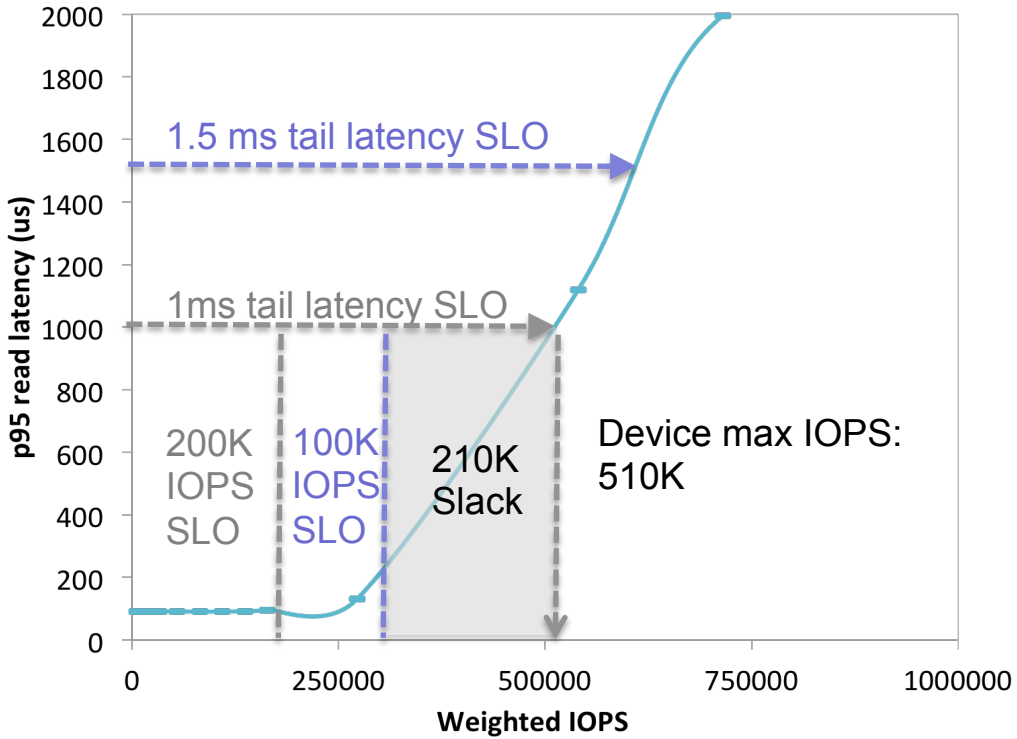


Tenant A SLO:

- 1ms tail latency
- 200K IOPS

Flash Memory Summit

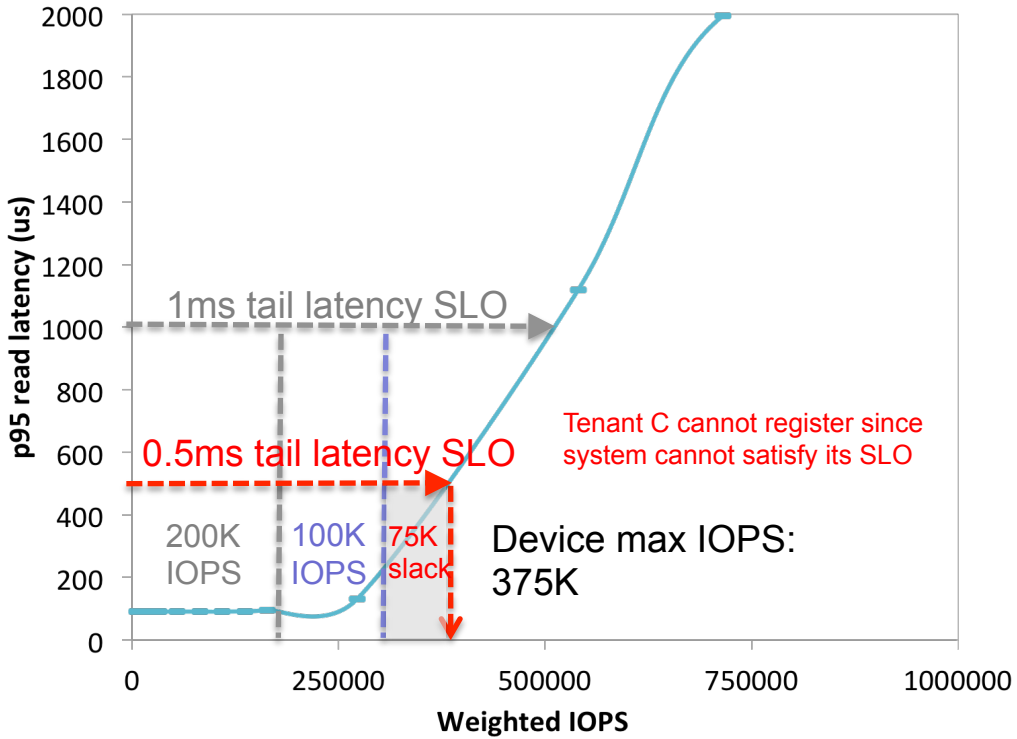
Request Cost Based Scheduling



Tenant A SLO:
- 1ms tail latency
- 200K IOPS

Tenant B SLO:
- 1.5ms tail latency
- 100K IOPS

Request Cost Based Scheduling

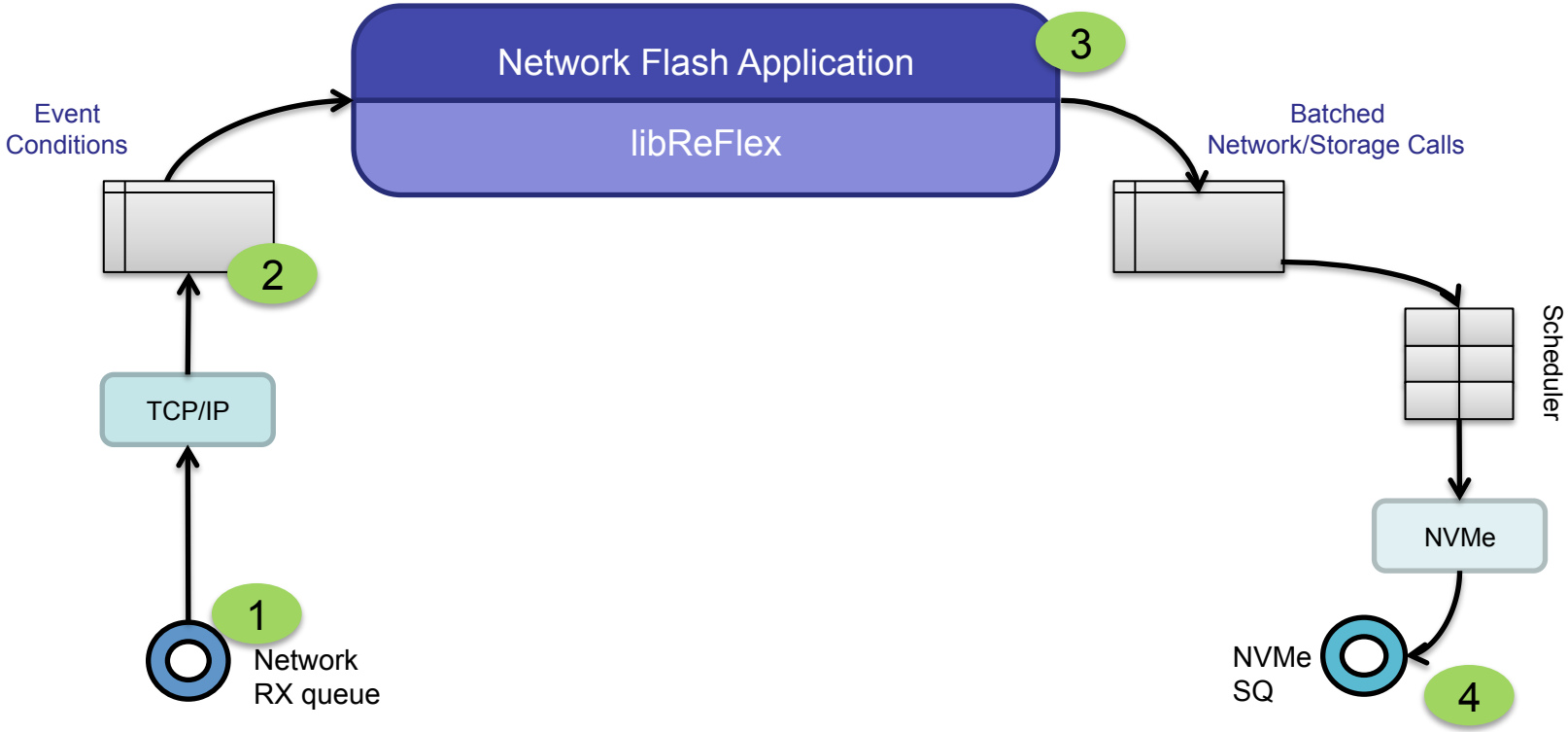


Tenant A SLO:
- 1ms tail latency
- 200K IOPS

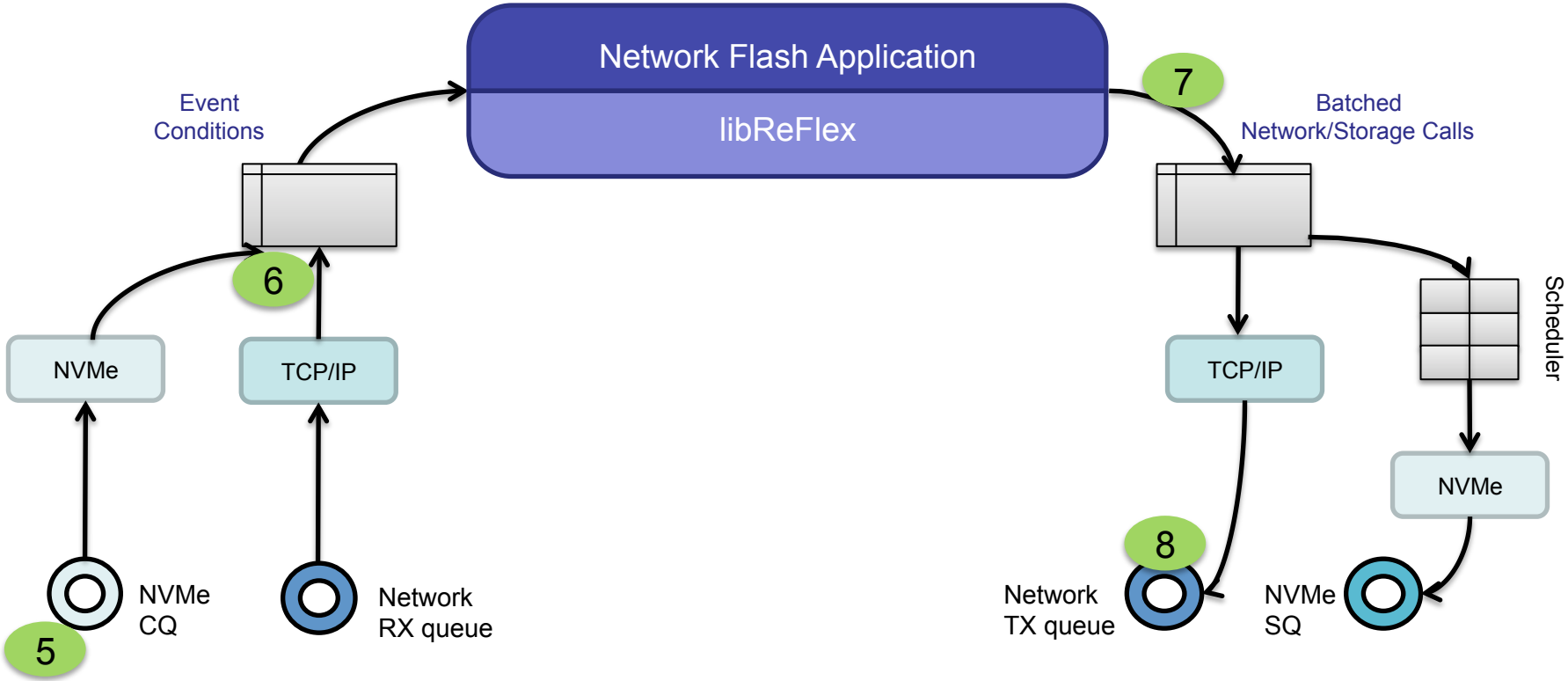
Tenant B SLO:
- 1.5ms tail latency
- 100K IOPS

Tenant C SLO:
- 0.5ms tail latency
- 200K IOPS

Summary: Life of a Packet in ReFlex

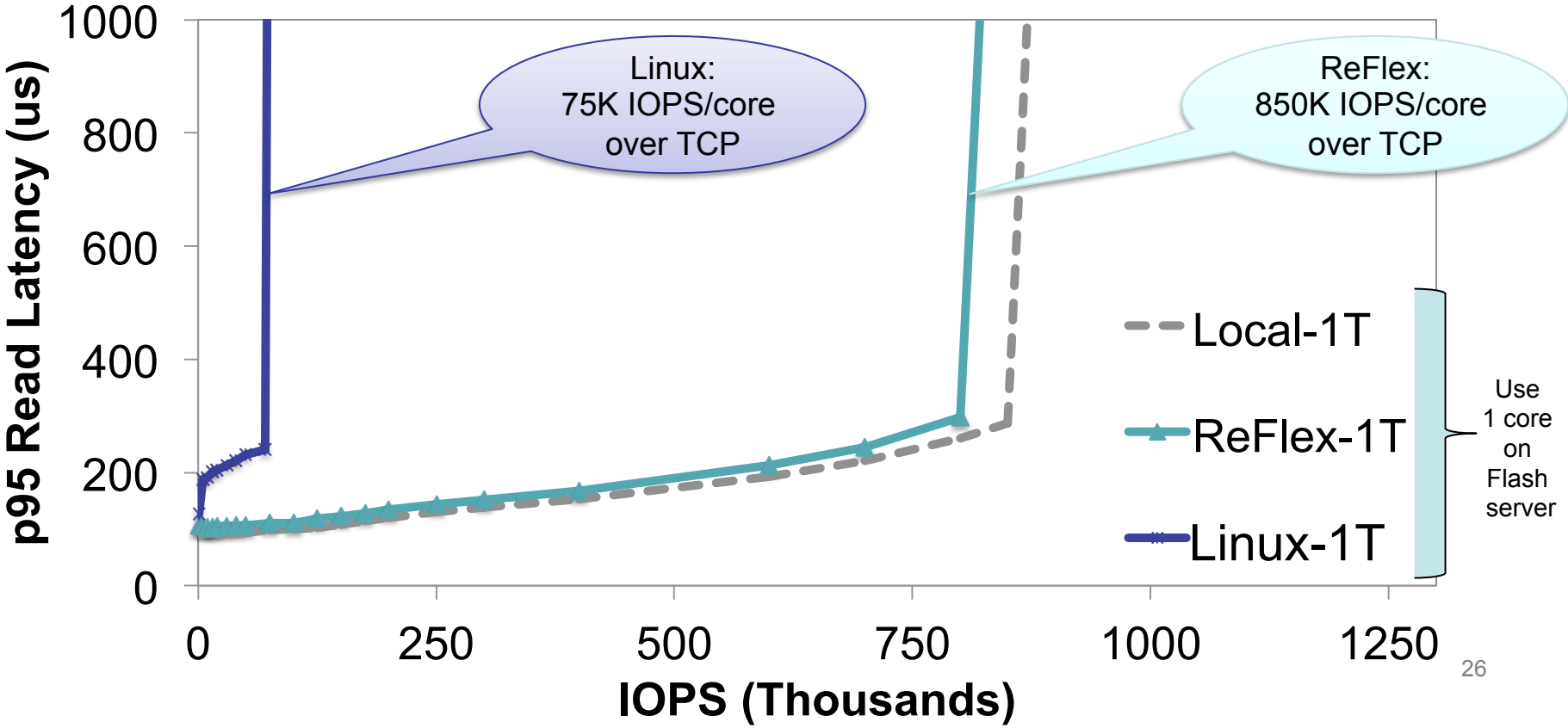


Summary: Life of a Packet in ReFlex



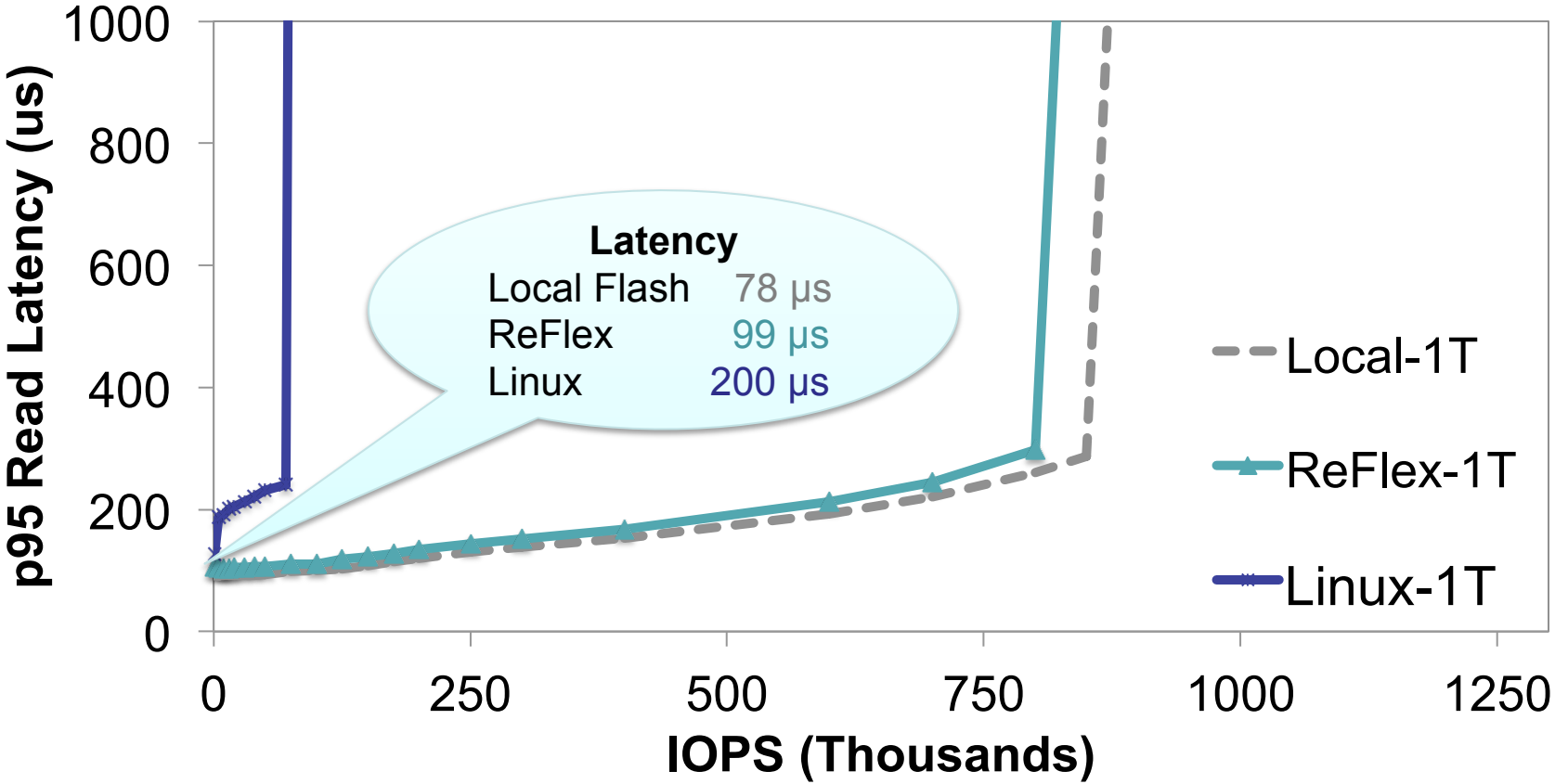
Flash Memory Summit

Results: Local ≈ Remote Latency



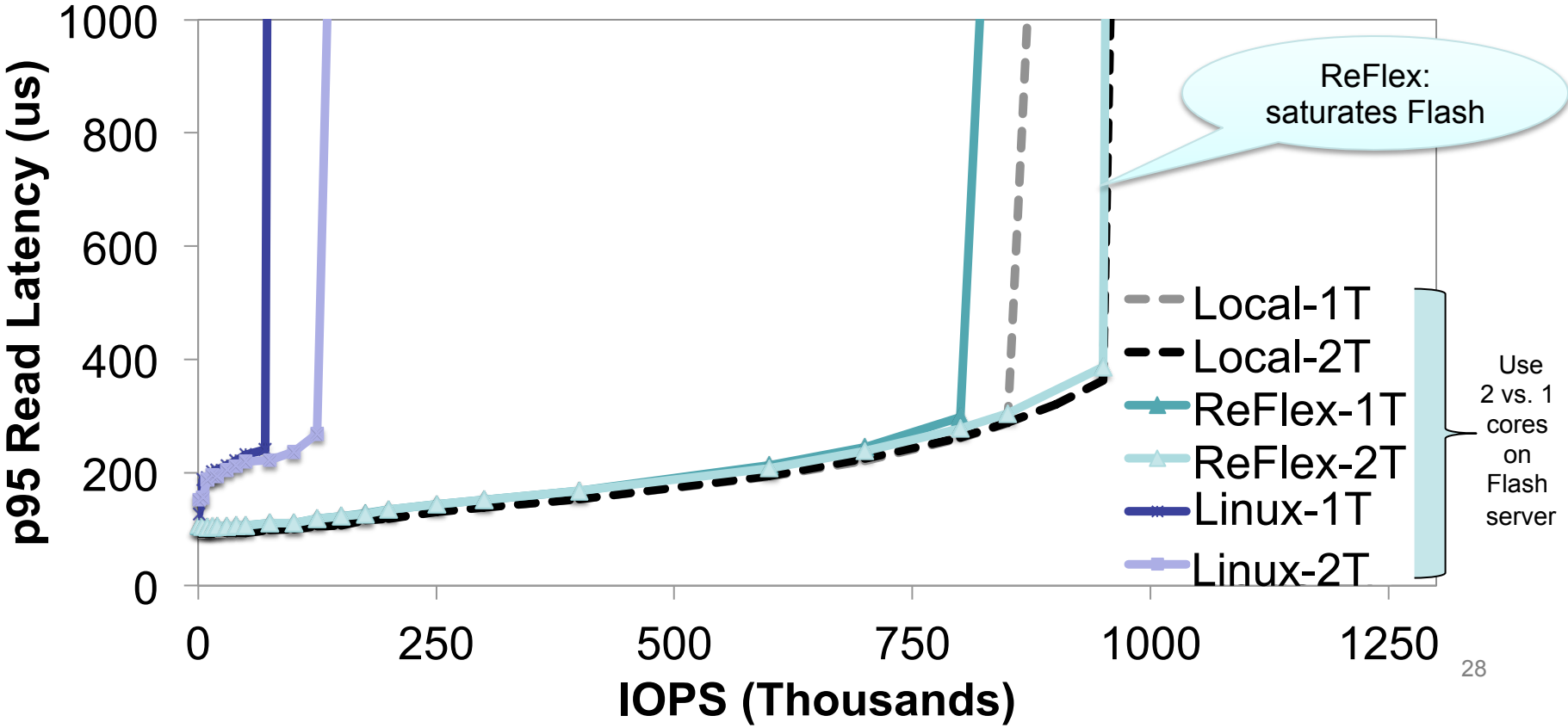
Flash Memory Summit

Results: Local ≈ Remote Latency

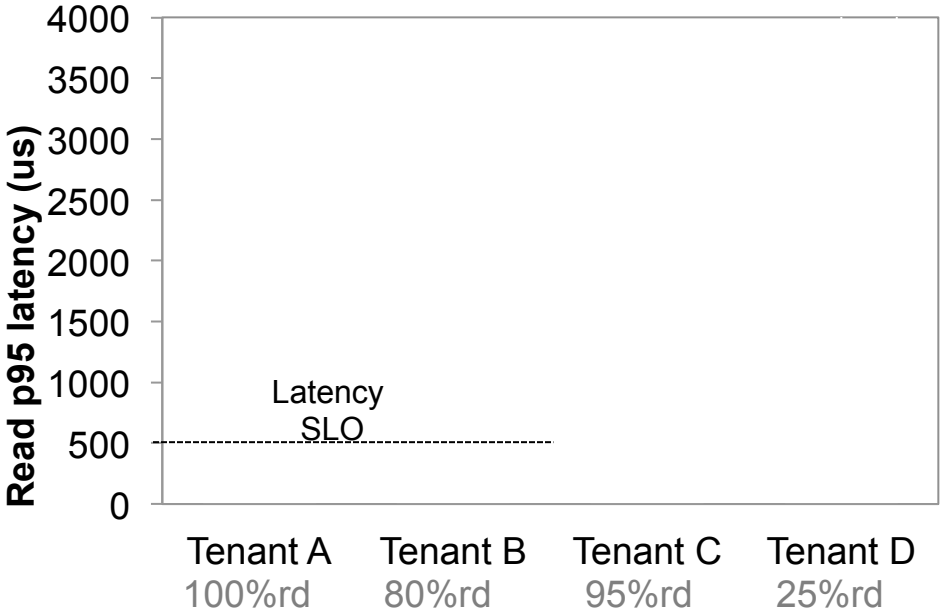


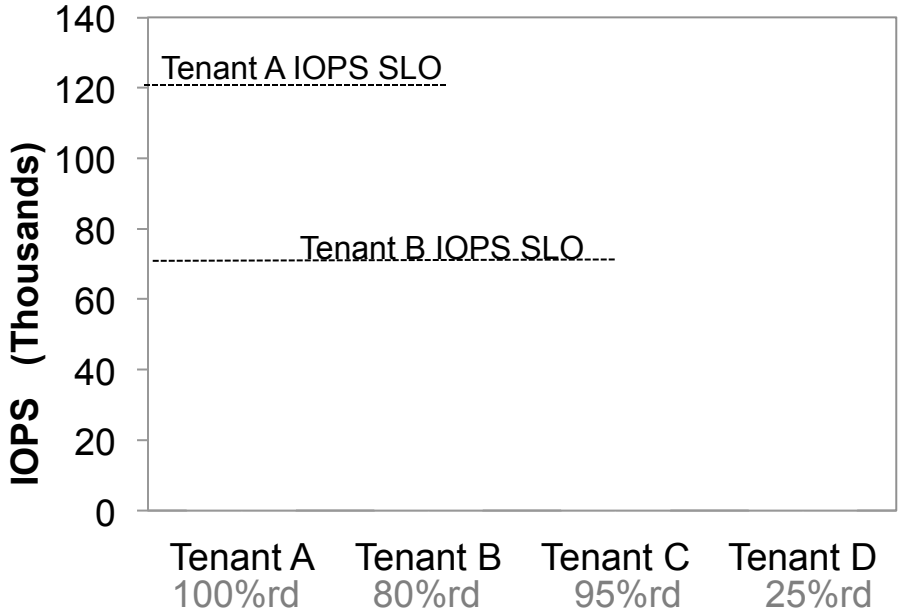
Flash Memory Summit

Results: Local ≈ Remote Latency



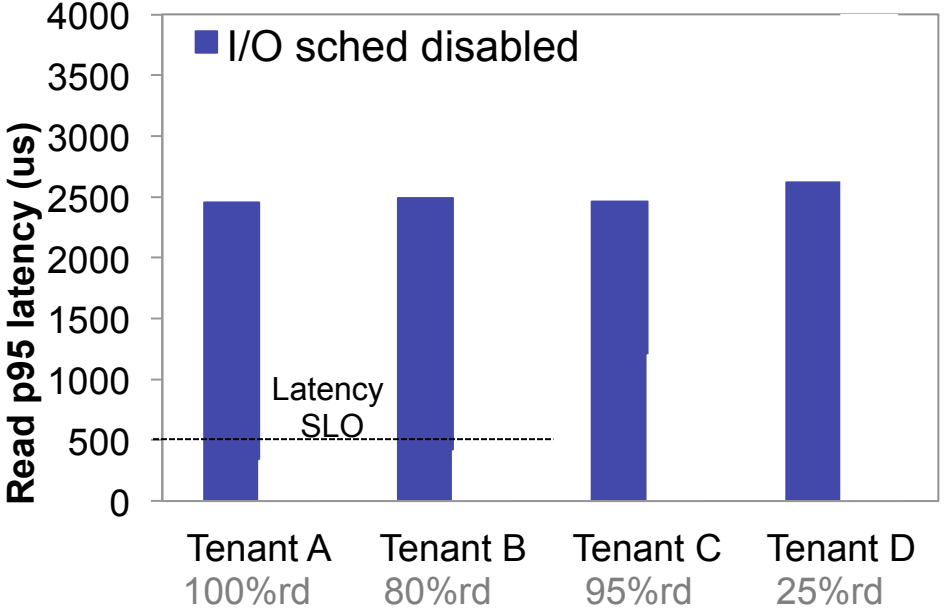
Results: Performance Isolation

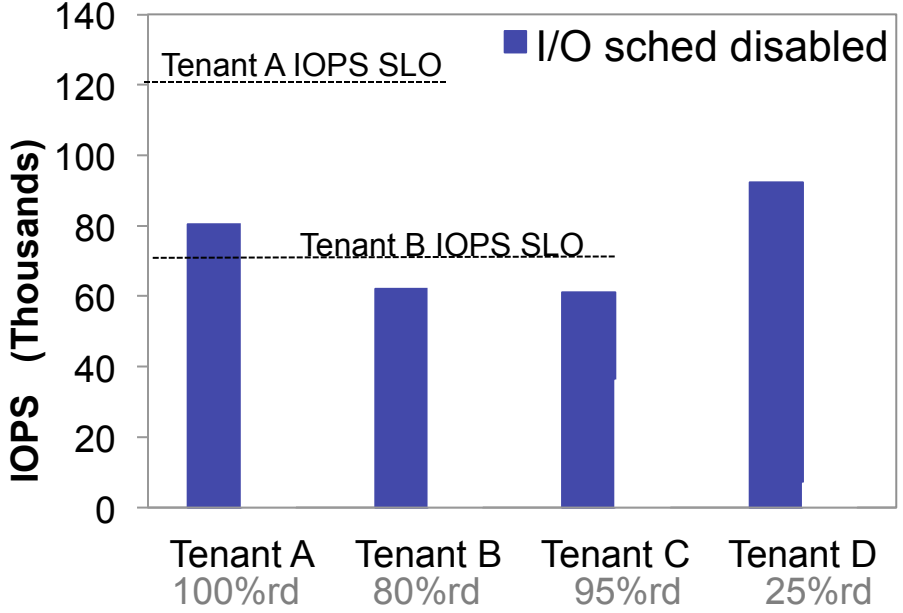




- Tenants A & B: latency-critical; Tenant C + D: best effort

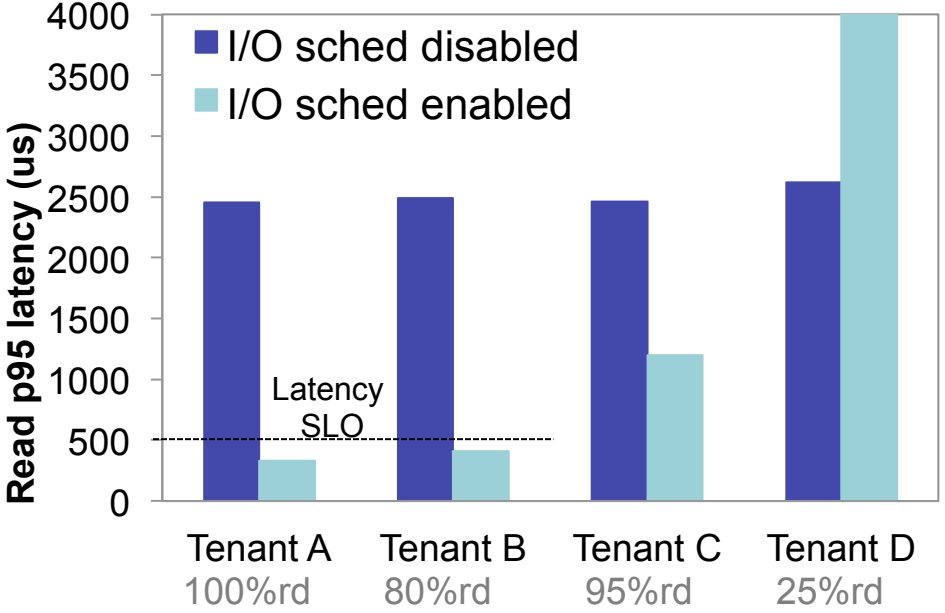
Results: Performance Isolation

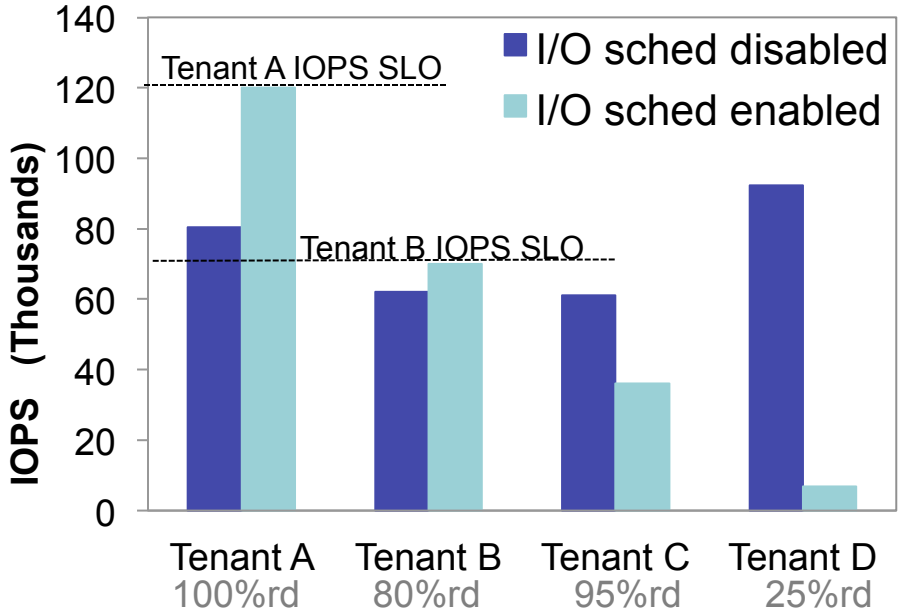




- Tenants A & B: latency-critical; Tenant C + D: best effort
- Without scheduler: latency and bandwidth QoS for A/B are violated

Results: Performance Isolation



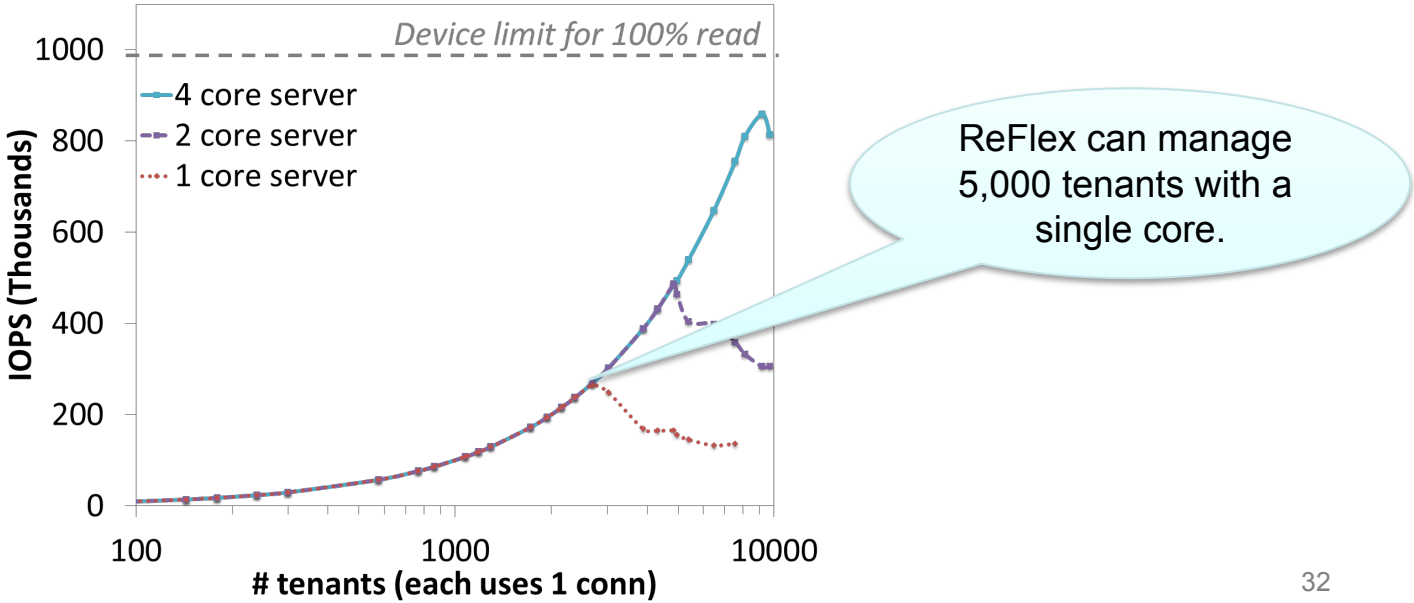


- Tenants A & B: latency-critical; Tenant C + D: best effort
- Without scheduler: latency and bandwidth QoS for A/B are violated
- Scheduler rate limits best-effort tenants to enforce SLOs

Results: Scalability

- ReFlex scales to multiple cores (we tested with up to 12 cores)
- ReFlex scales to thousands of tenants and connections

Each tenant issues
100 1kB reads/sec



How can I use ReFlex?

- ReFlex is open source:
  www.github.com/stanford-mast/reflex

- Two versions of ReFlex available:
  1. Kernel implementation:
     - Provides a protected network-storage data plane (user application not trusted)
     - Builds on hardware support for virtualization
     - Requires Dune kernel module for direct access to hardware and memory management in Linux
  2. User space implementation:
     - Network-storage data plane runs as user space application; kernel-bypass
     - Portable implementation, tested on Amazon Web Services i3 instance

ReFlex Clients

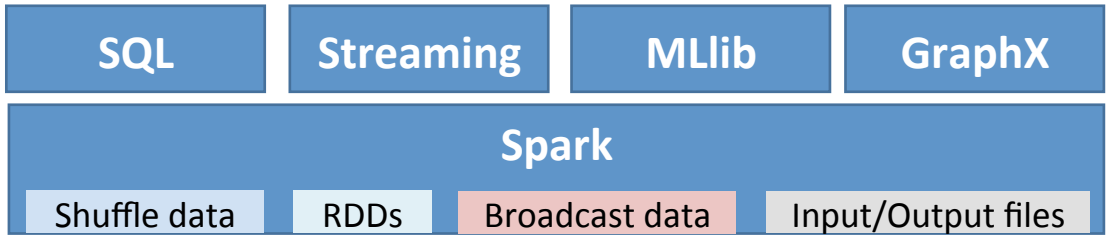
- ReFlex exposes a logical block interface
- ReFlex client implementations available:
  1. Block I/O client
     → For issuing raw network block I/O requests to ReFlex
  2. Remote block device driver
     → For legacy Linux applications

Control Plane for ReFlex

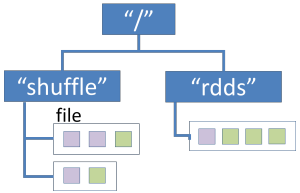
- ReFlex provides an efficient data plane for remote access
- Bring your own control plane or distributed storage system:
  - Global control plane allocates Flash capacity & bandwidth in cluster
  - Control plane can be a cluster manager or a distributed storage system (e.g. distributed file system or database)
- Example: ReFlex storage tier for Crail distributed file system
  - Spark applications can use ReFlex as a remote Flash storage tier, managed by the Crail distributed data store

Example Use-Case

- Big data analytics frameworks such as Spark store a variety of data types in memory, Flash and disk



- Crail is a distributed storage system that manages and provides high-performance access to data across multiple storage tiers

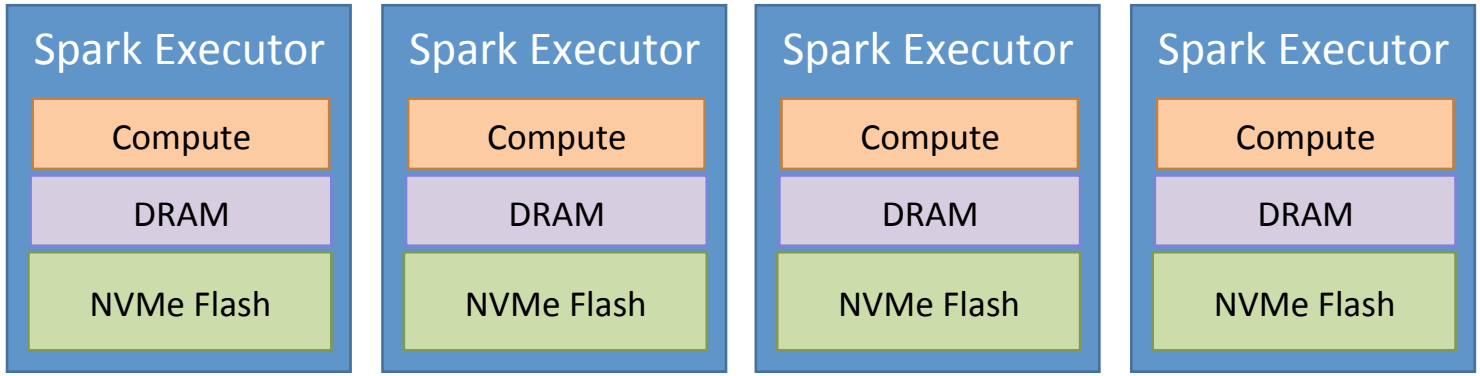


crail

www.crail.io

Example Use-Case

- Spark executors often use (local) Flash to store temporary data (e.g. shuffle)

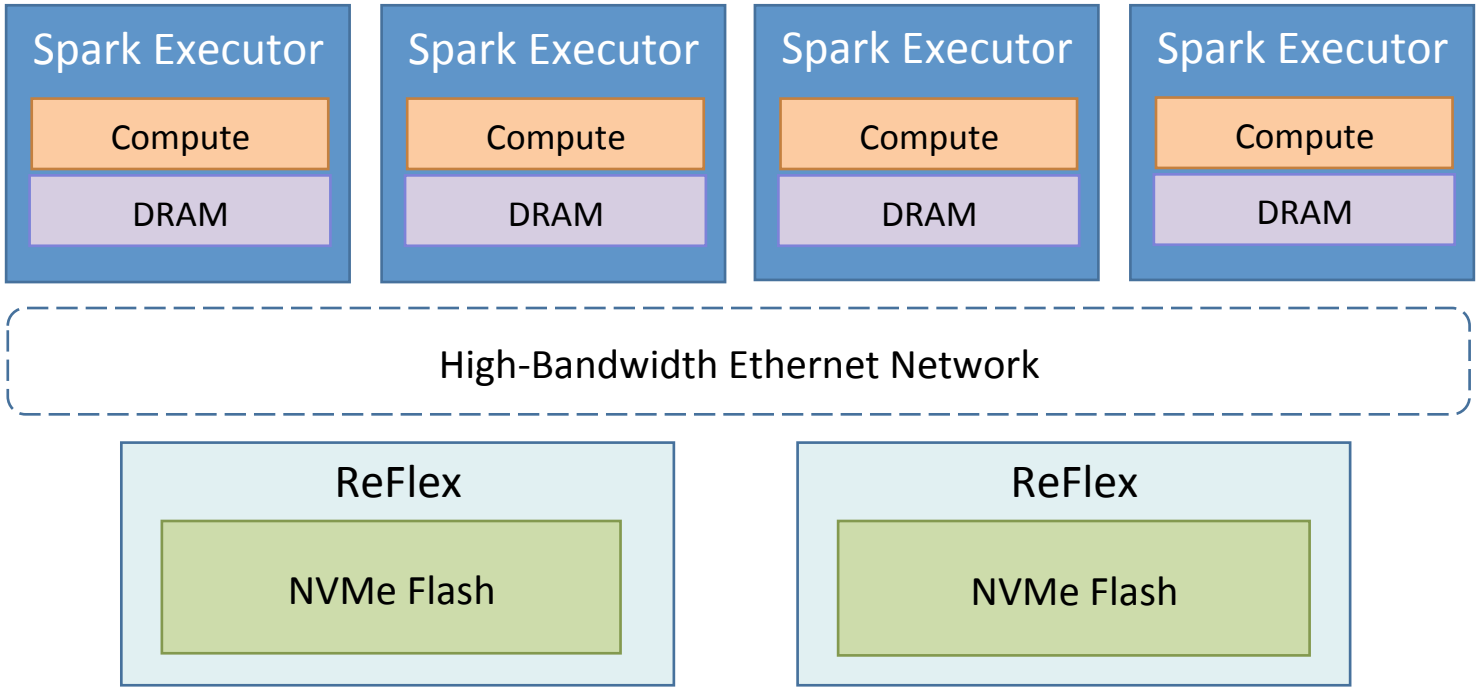


- Spark applications often saturate CPU resources before they can saturate NVMe Flash IOPS

  → Local Flash bandwidth is underutilized

Example Use-Case

- ReFlex provides a shared remote Flash tier for big data analytics
  - Improves resource utilization while offering local Flash performance



Summary

- ReFlex enables Flash disaggregation
- Performance: *remote* ≈ *local*
- Commodity networking, low CPU overhead
- QoS on shared Flash with I/O scheduler

Thank You!

ReFlex is available at:

https://github.com/stanford-mast/reflex