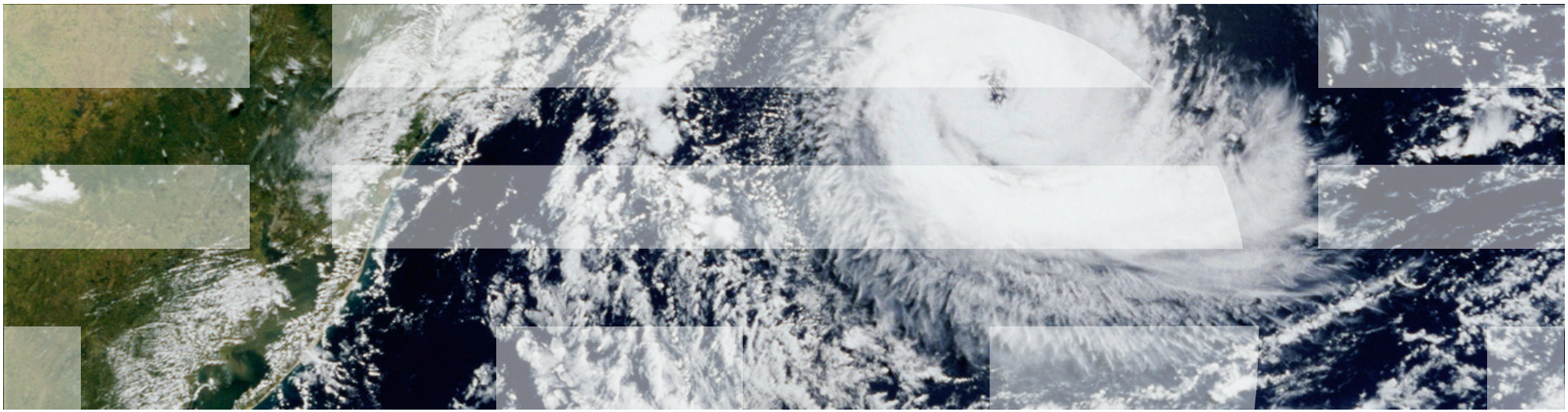
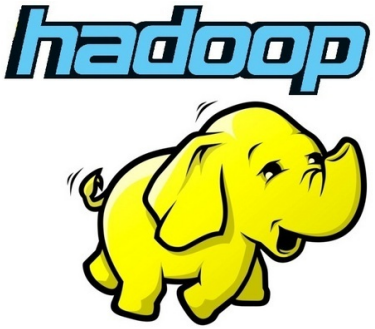
FlashNet: A Unified High-Performance IO Stack

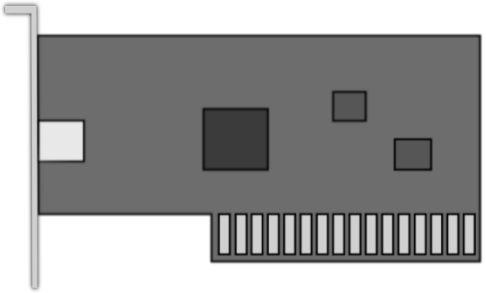
Animesh Trivedi, Nikolas Ioannou, Bernard Metzler, Patrick Stuedi,
Jonas Pfefferle, Ioannis Koltsidas



Why IO Matters?

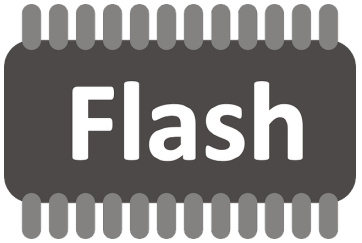


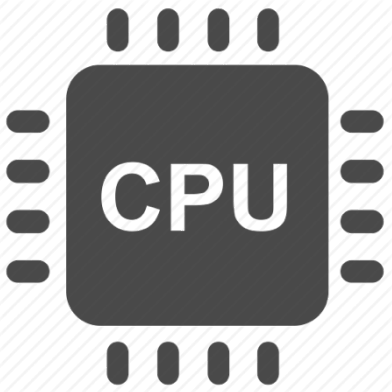
High-Performance IO



1 → 100 Gbit/sec, with ~1 usec link latencies

Rise of NVM devices, multi GBs/sec with ~10s usec device latencies





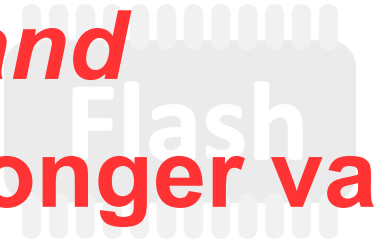
Marginal improvements

High-Performance IO

1 → 100 Gbit/sec, with ~1 usec link latencies

The notion of "*fast CPU and multiple slow IO devices*" is no longer valid

Rise of NVM devices, multi GBs/sec with ~10s usec device latencies



Marginal improvements

High-Performance IO

1 → 100 Gbit/sec, with ~1 usec link latencies

Traditional IO stacks built assuming slow IO and fail to deliver performance

Rise of NVM devices, multi GBs/sec with ~10s usec device latencies

Marginal improvements

Why Unify Network and Storage IO?

- Exposing high-speed networking performance to the user application:

  Polling, direct hardware access, OS-bypass,
  zero-copy data movement, RDMA, DPDK…

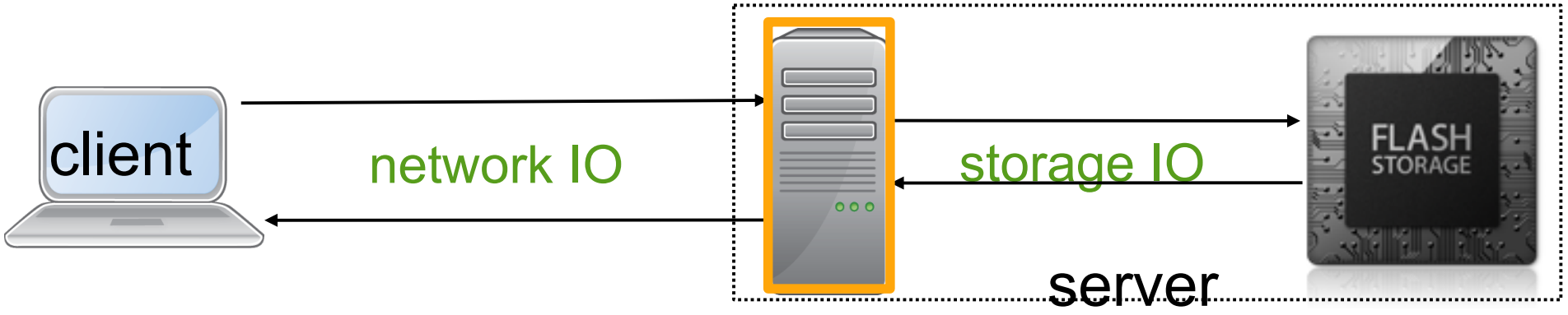
- Exposing NVM device performance to the user application:

  Polling, direct hardware access, OS-bypass,
  zero-copy data movement, NVMe, SPDK...

Proposal : Unify network and storage IO → **FlashNet !**

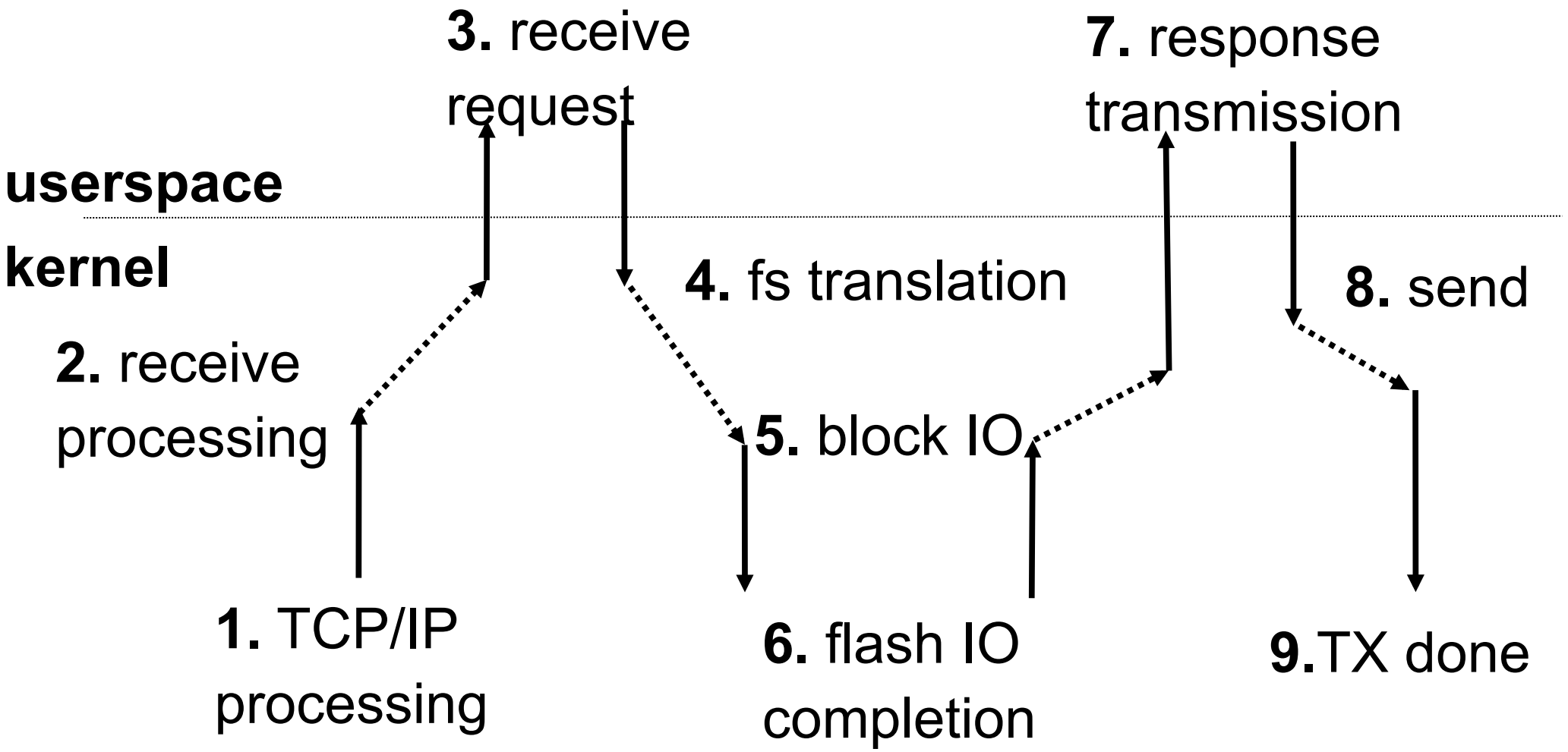
The Problem Scenario

Key-Value Stores, Distributed Overlay File Systems e.g., Hadoop.

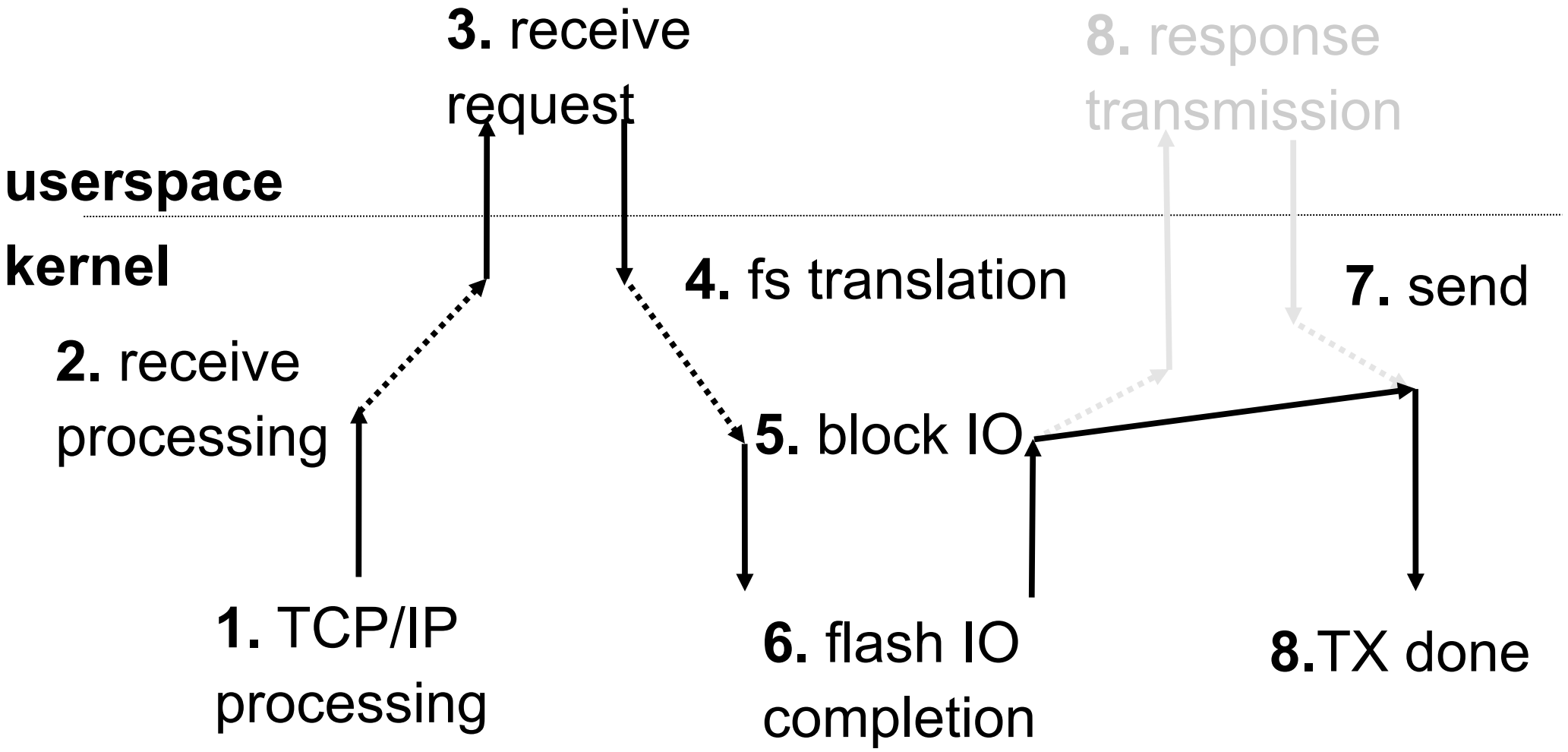


performance = network IO + storage IO + server time

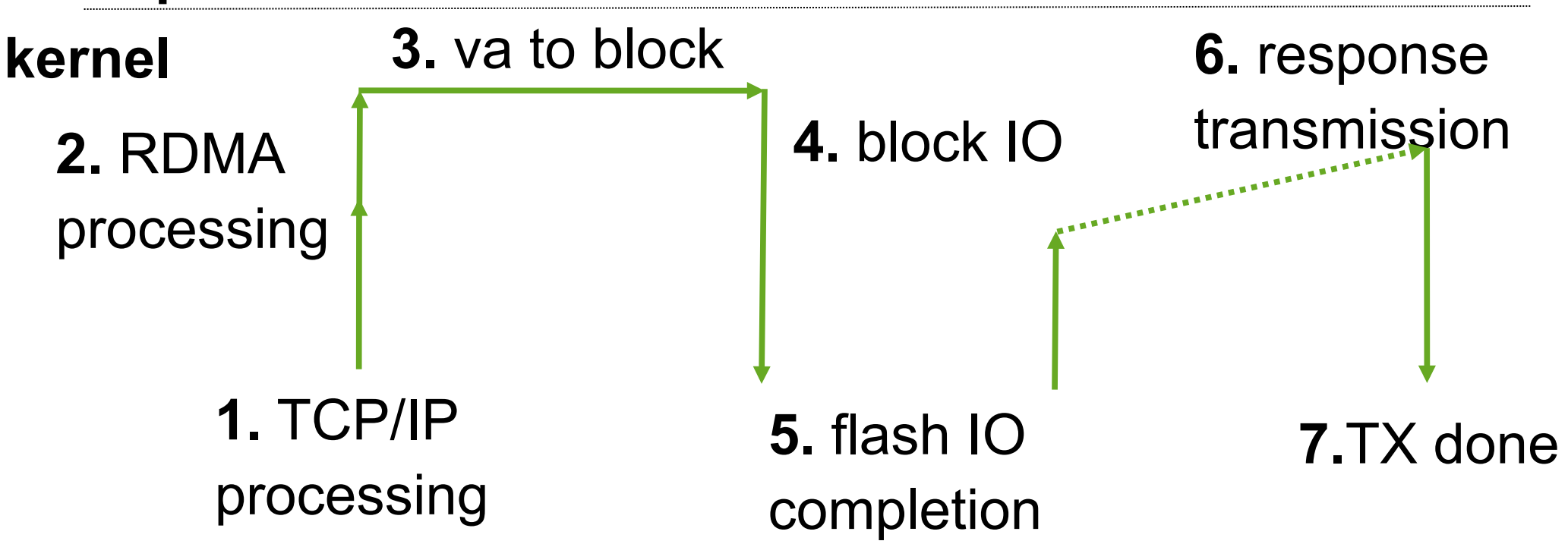
A Detailed Look: send



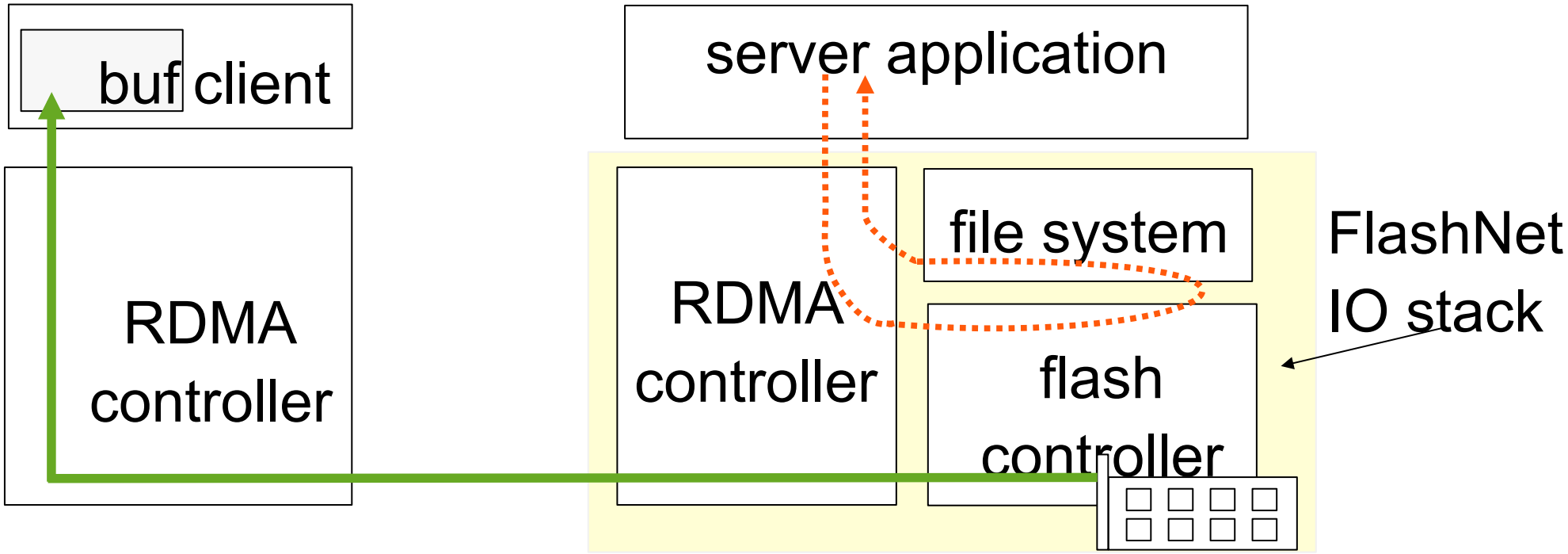
A Detailed Look: sendfile



FlashNet IO Operation



FlashNet: A Unified IO Stack



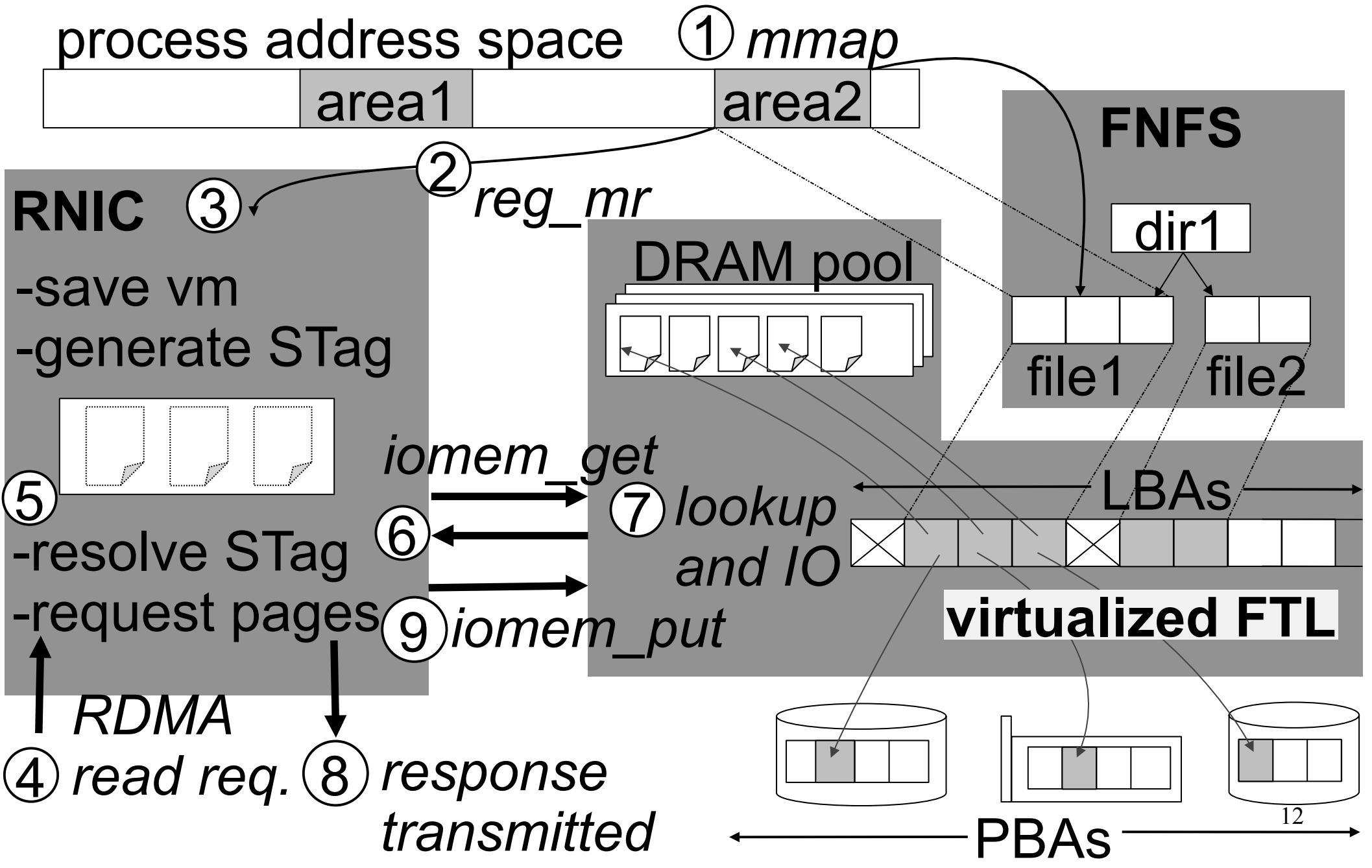
network control setup expanding to storage

data path from a flash device to a client buffer

[1] SoftiWARP: Software iWARP kernel driver and user library for Linux, Metzler et al, https://github.com/zrlio/softiwarp

[2] SALSA: A unified stack for SSDs and SMR disks, Koltsidas et al. http://ibm.biz/salsa-whitepaper

Lifetime of a Flashnet operation



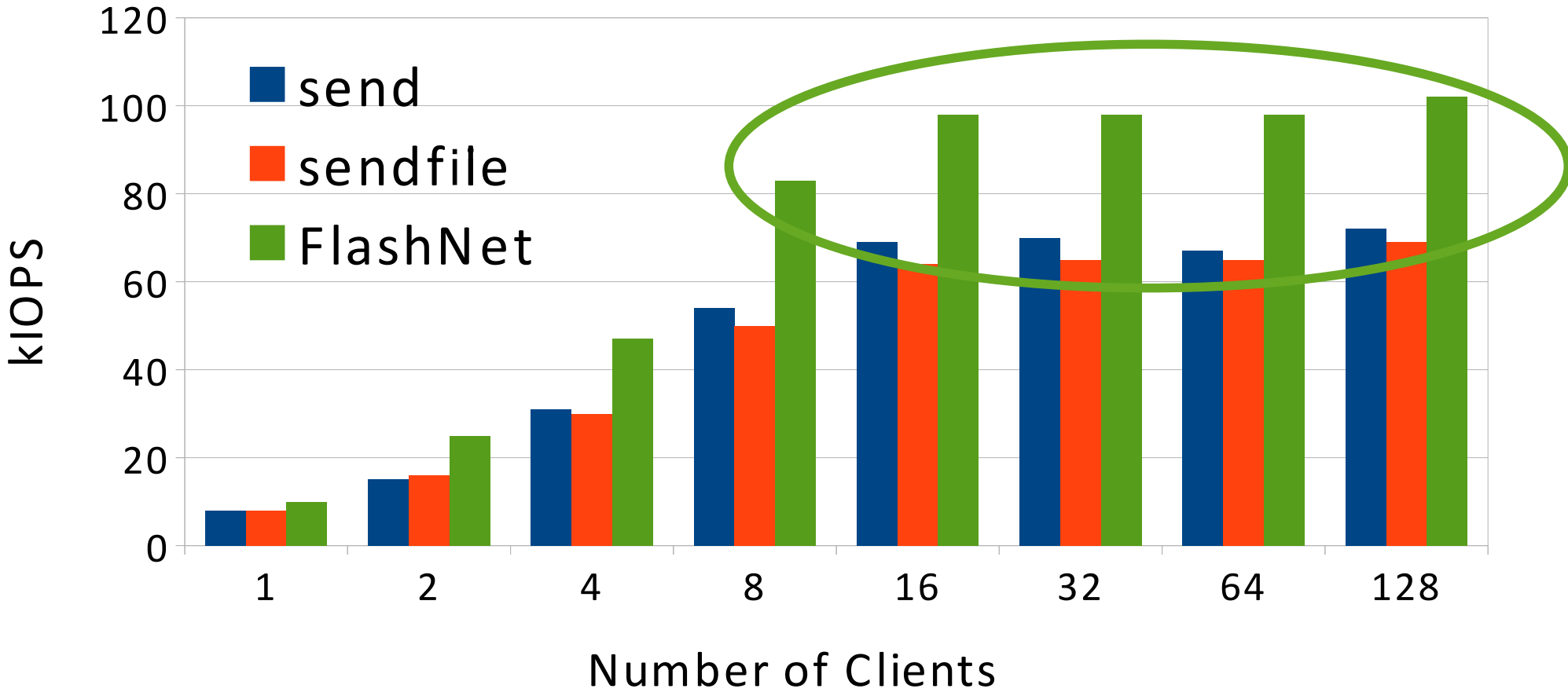
Performance Evaluation

How efficient is FlashNet's IO path?

9-machine cluster testbed

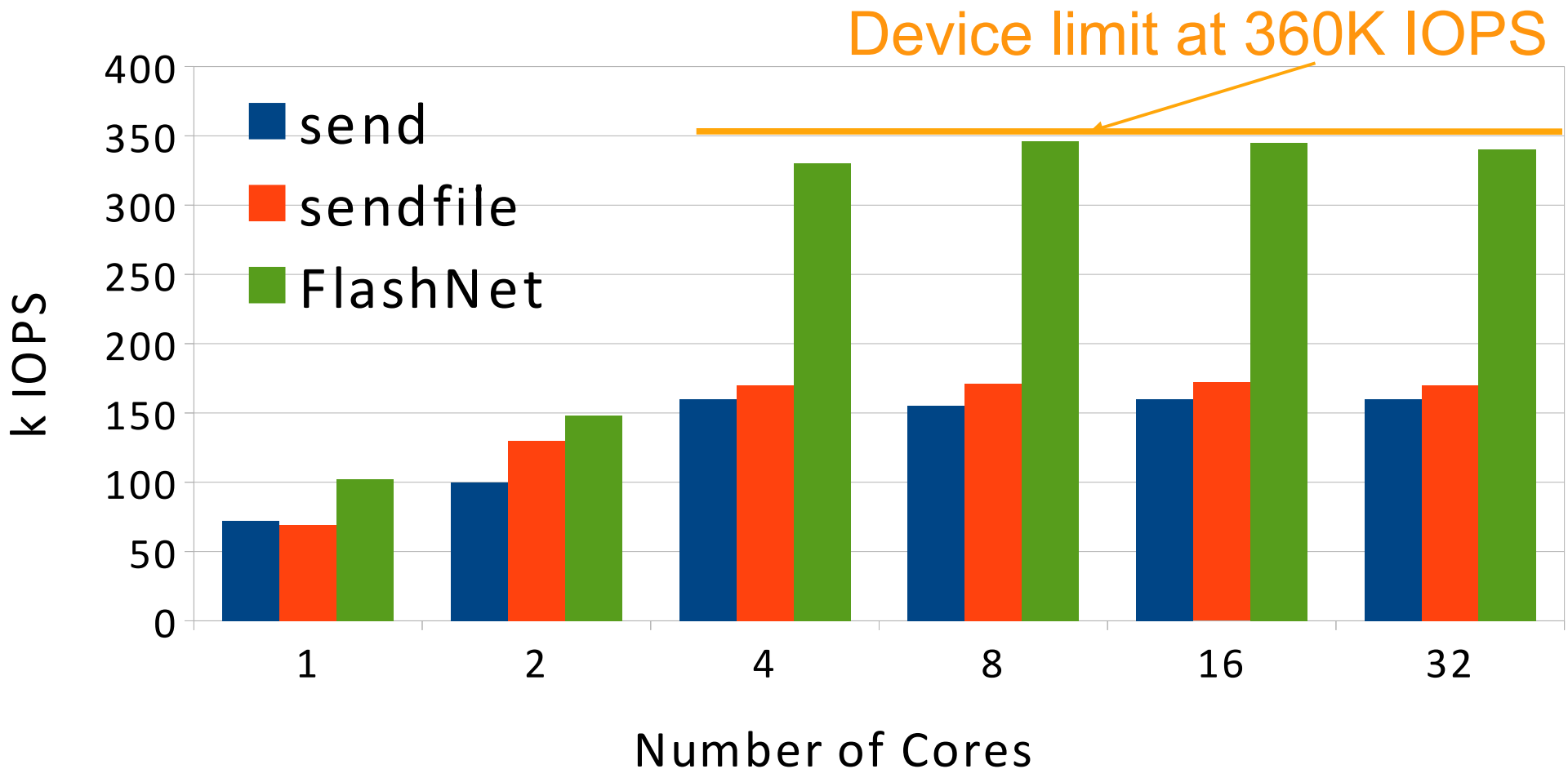
CPU : dual socket E5-2690, 2.9 GHz, 16 cores
DRAM : 256 GB, DDR3 1600 MHz
OS : Linux 3.19 kernel
NIC : 40Gbit/s Ethernet
Flash : 1.3 GB/sec (read), 680 MB/sec (write)
peak read IOPS: 360K, chip latency: 50µsec

Performance - IOPS Efficiency



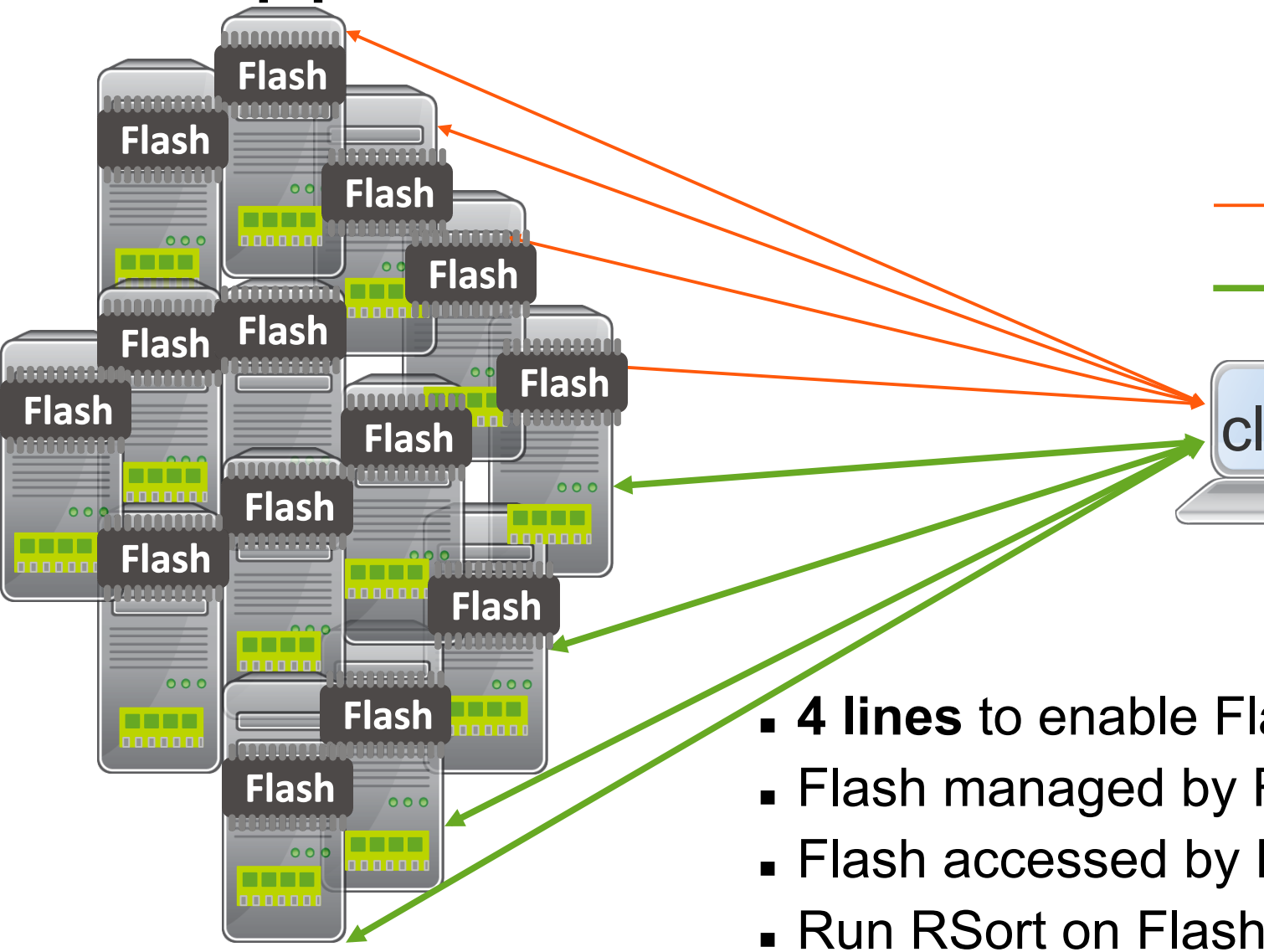
- FlashNet reads are almost 50% more efficient

Performance - Core Scaling



- FlashNet IO operations scale better

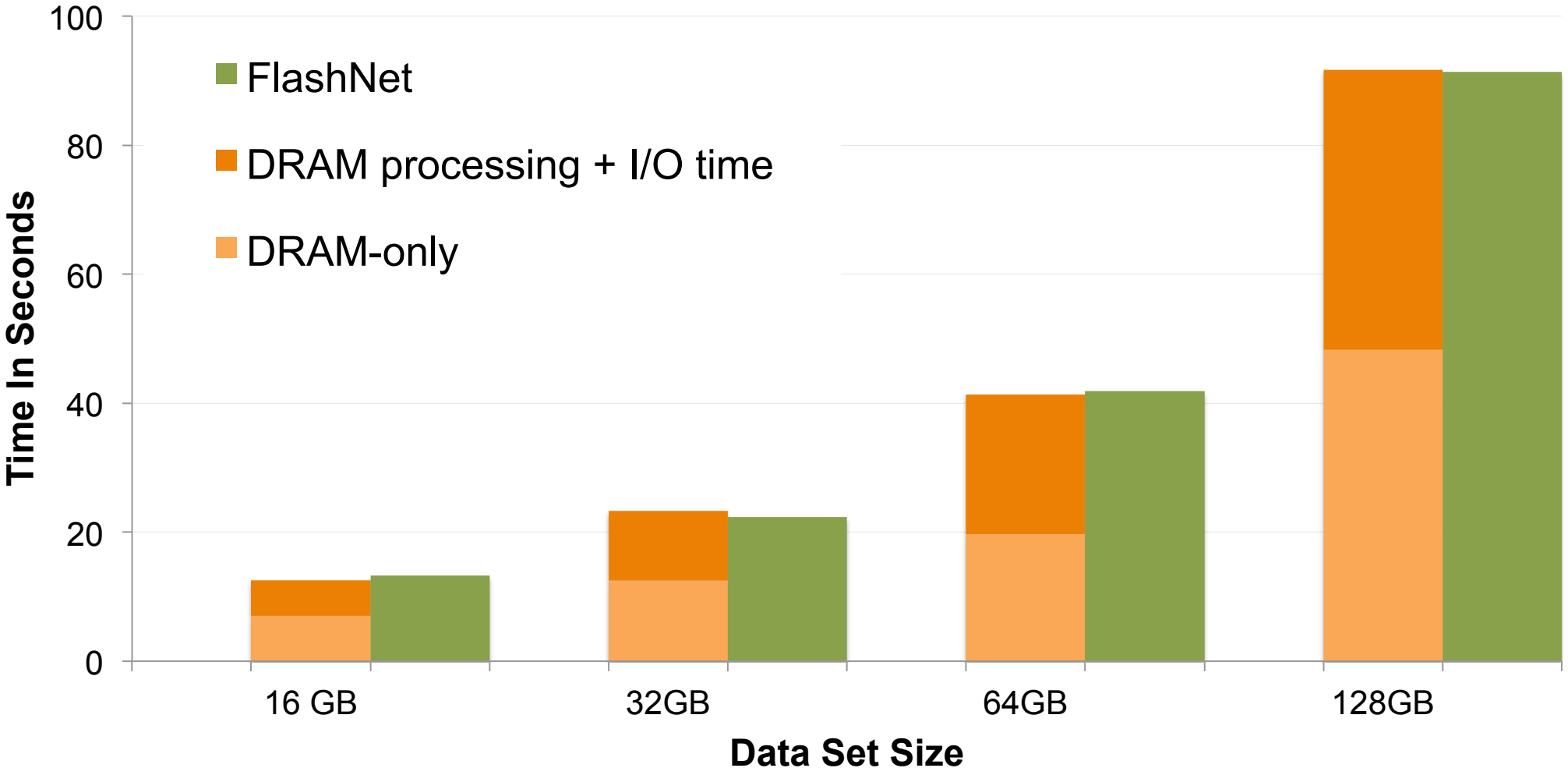
Application: RStore on FlashNet



- **4 lines** to enable FlashNet
- Flash managed by RStore's control API
- Flash accessed by RStore's data API
- Run RSort on FlashNet/Rstore
- 4 servers with direct-attached SSDs

[1] RStore: A Direct-Access DRAM-based Data Store, Trivedi et al., ICDCD'15

RStore on FlashNet: TeraSort



- FlashNet adds minimum overheads to RDMA-ready applications

Conclusions

- Unified RDMA operations and semantics
- A hierarchical data management with a file system
- Transparent to an RDMA client
- Defined network and storage API, implemented software prototype that benefits from unified semantics
- Exploring hardware implementation
- “leaner, directly-accessible, application-managed IO resources”
  - Arrakis, IX, NVMe, NVMeOverFabrics, DiDAFS, DPDK, PeerDirect-RDMA...

Thank you!