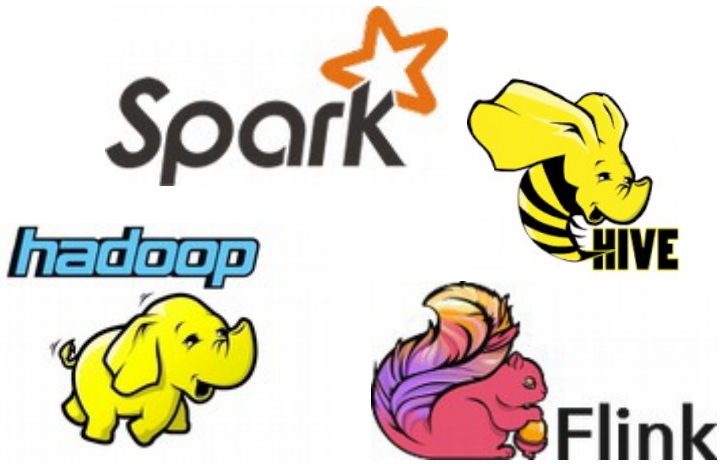
Systor2017
May 22 – 24 Haifa, Israel
10 The 10th ACM International Systems and Storage Conference

FlashNet: Flash/Network Stack Co-Design

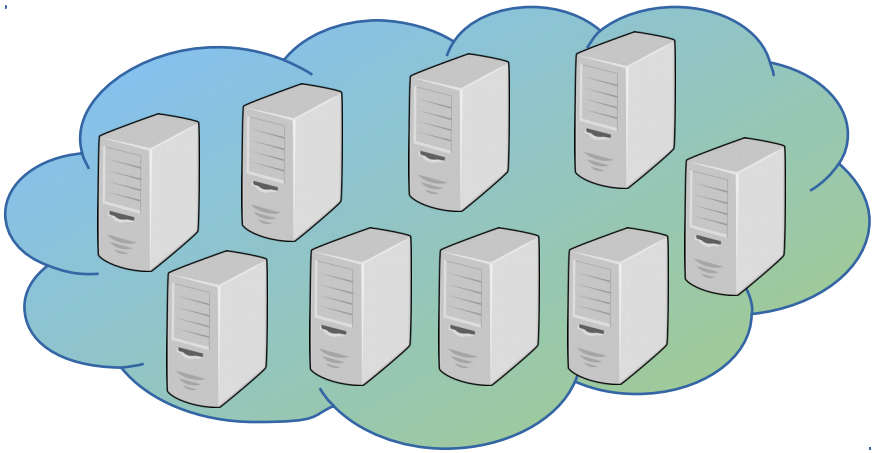
Animesh Trivedi, Nikolas Ioannou, Bernard Metzler, Patrick Stuedi,
Jonas Pfefferle, Ioannis Koltsidas, Kornilios Kourtis,
and
Thomas R. Gross

IBM Research and ETH Zurich, Switzerland

Modern Distributed Systems



- data intensive
- run on 100-1000s of servers
- performance depends upon
  both
  network and storage



Modern Distributed Systems

- performance depends upon
both
network and storage

Modern Distributed Systems

- StackMap: Low-Latency Networking with the OS Stack and Dedicated NICs, *USENIX'16*
- Network Stack Specialization for Performance, *SIGCOMM'14*
- mTCP: A Highly Scalable User-level TCP Stack for Multicore Systems, *NSDI'14*
- MegaPipe: A New Programming Interface for Scalable Network I/O, *OSDI'12*
- ...

- performance depends upon *both* network and storage

Modern Distributed Systems

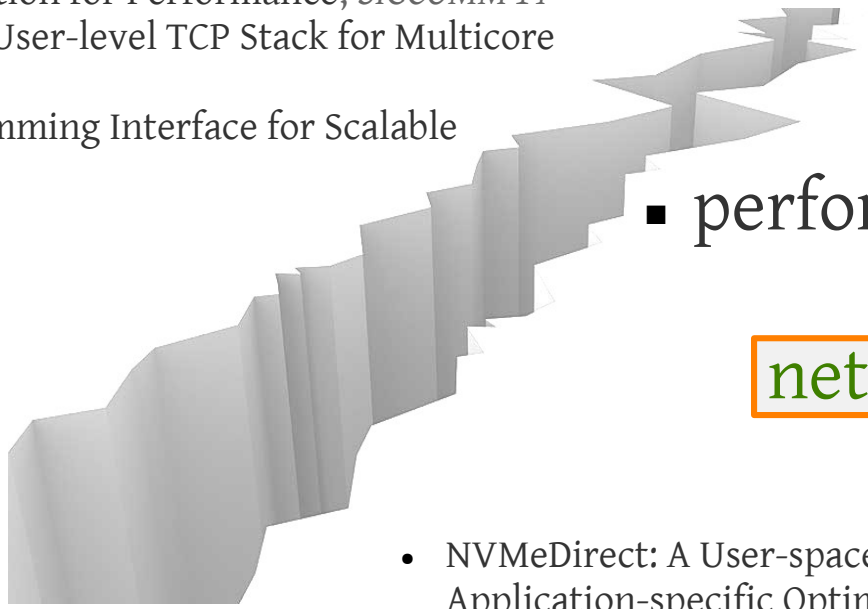
- StackMap: Low-Latency Networking with the OS Stack and Dedicated NICs, *USENIX'16*
- Network Stack Specialization for Performance, *SIGCOMM'14*
- mTCP: A Highly Scalable User-level TCP Stack for Multicore Systems, *NSDI'14*
- MegaPipe: A New Programming Interface for Scalable Network I/O, *OSDI'12*
- ...

- performance depends upon *both* network and storage

- NVMeDirect: A User-space I/O Framework for Application-specific Optimization on NVMe SSDs, *HotStorage'16*
- OS I/O Path Optimizations for Flash Solid-state Drives, *USENIX'14*
- Linux Block IO: Introducing Multi-queue SSD Access on Multi-core Systems, *SYSTOR'13*
- When Poll is Better Than Interrupt, *FAST'12*
- ...

Modern Distributed Systems

- StackMap: Low-Latency Networking with the OS Stack and Dedicated NICs, *USENIX'16*
- Network Stack Specialization for Performance, *SIGCOMM'14*
- mTCP: A Highly Scalable User-level TCP Stack for Multicore Systems, *NSDI'14*
- MegaPipe: A New Programming Interface for Scalable Network I/O, *OSDI'12*
- ...

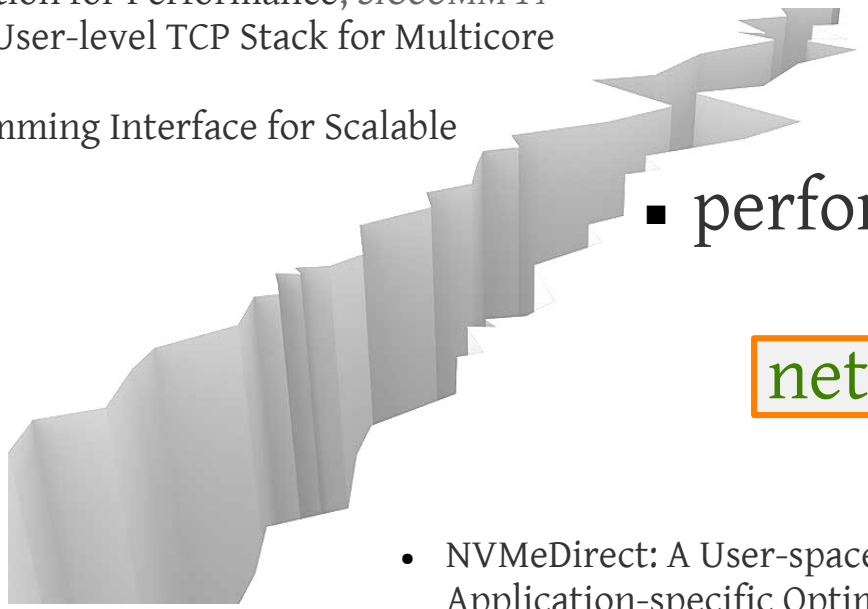


- performance depends upon *both* network and storage

- NVMeDirect: A User-space I/O Framework for Application-specific Optimization on NVMe SSDs, *HotStorage'16*
- OS I/O Path Optimizations for Flash Solid-state Drives, *USENIX'14*
- Linux Block IO: Introducing Multi-queue SSD Access on Multi-core Systems, *SYSTOR'13*
- When Poll is Better Than Interrupt, *FAST'12*
- ...

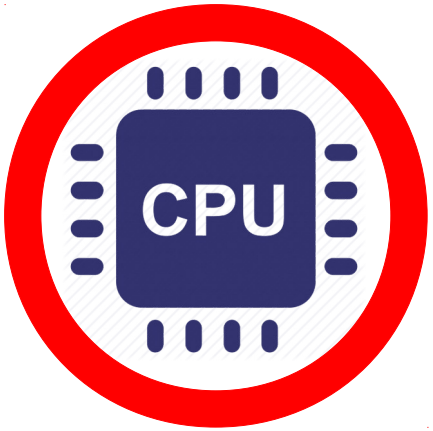
Modern Distributed Systems

- StackMap: Low-Latency Networking with the OS Stack and Dedicated NICs, *USENIX'16*
- Network Stack Specialization for Performance, *SIGCOMM'14*
- mTCP: A Highly Scalable User-level TCP Stack for Multicore Systems, *NSDI'14*
- MegaPipe: A New Programming Interface for Scalable Network I/O, *OSDI'12*
- ...



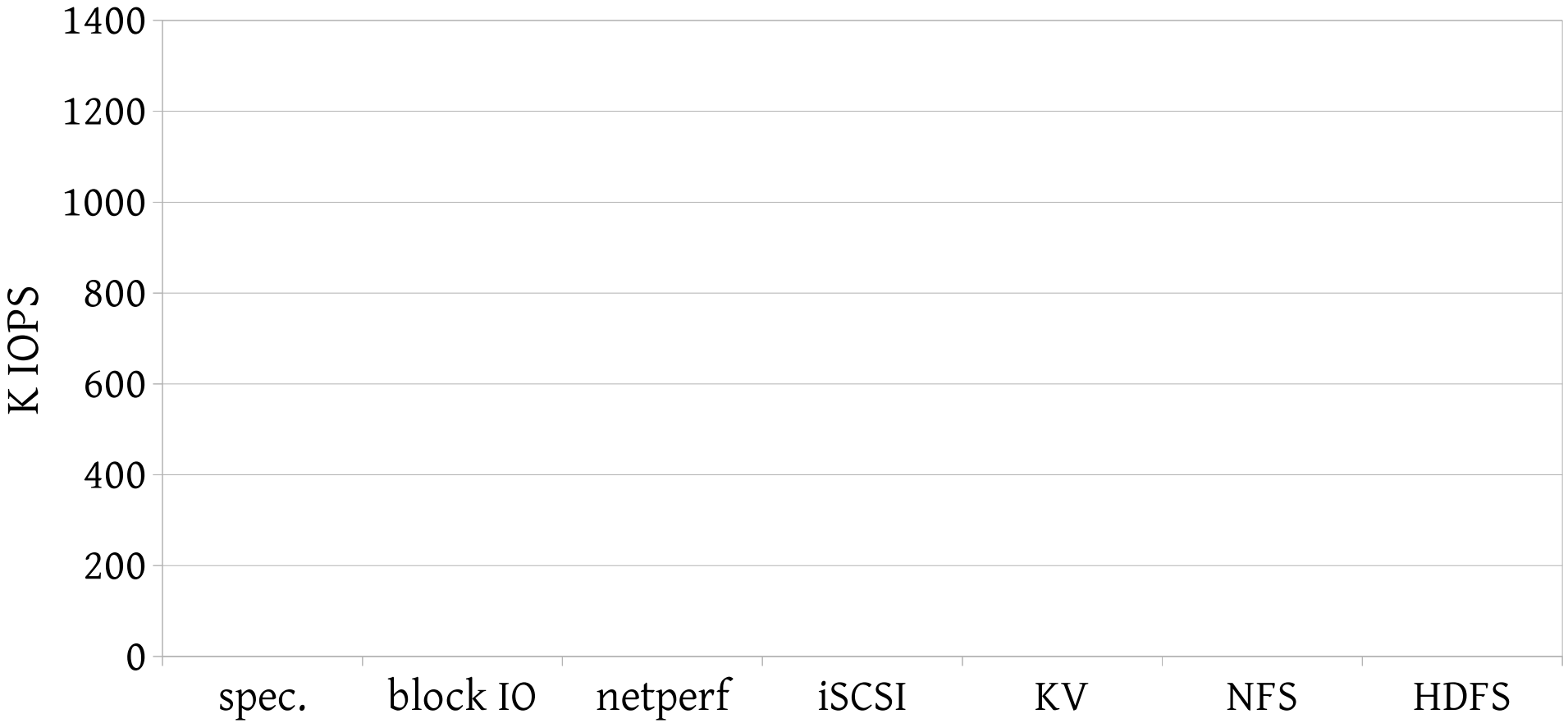
MIND THE GAP

- performance depends upon *both* network and storage

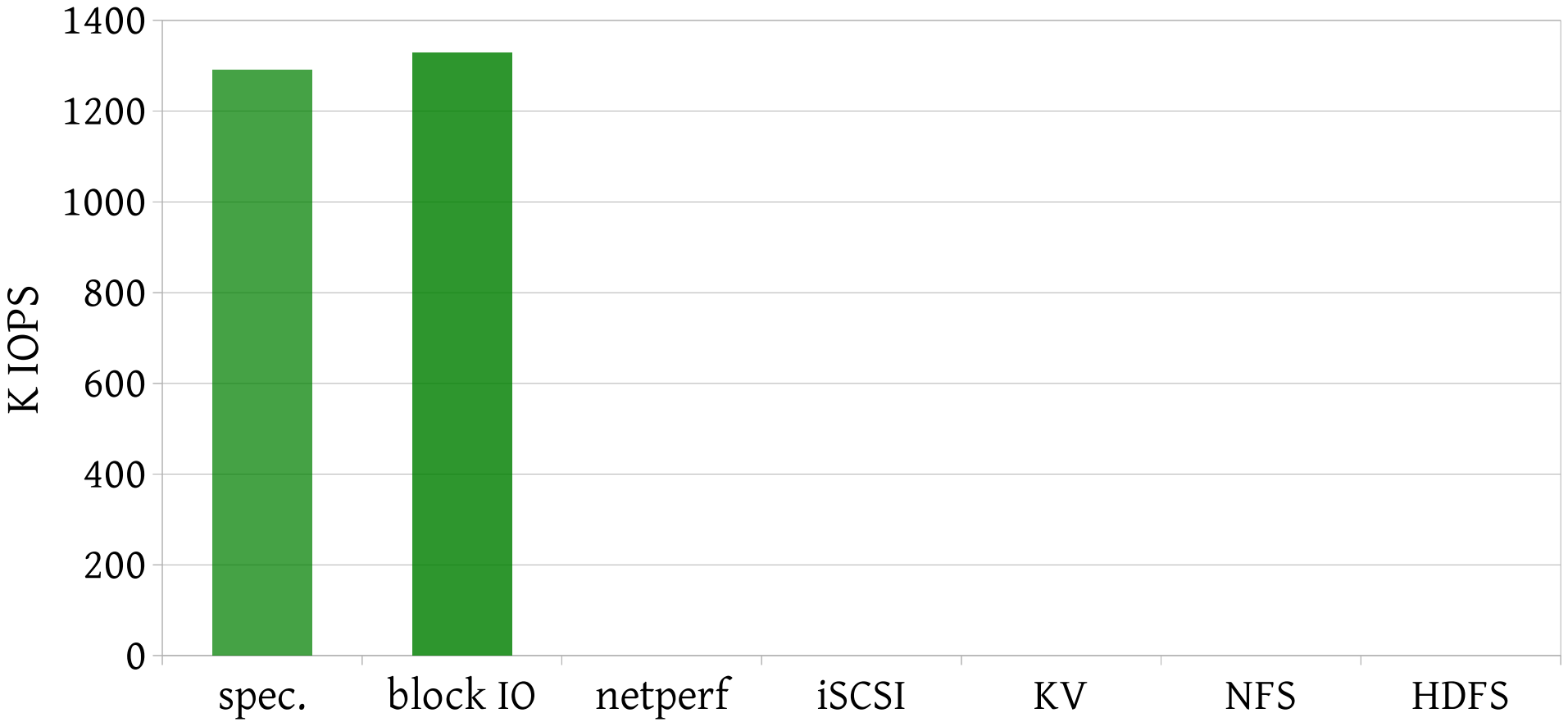


- NVMeDirect: A User-space I/O Framework for Application-specific Optimization on NVMe SSDs, *HotStorage'16*
- OS I/O Path Optimizations for Flash Solid-state Drives, *USENIX'14*
- Linux Block IO: Introducing Multi-queue SSD Access on Multi-core Systems, *SYSTOR'13*
- When Poll is Better Than Interrupt, *FAST'12*
- ...

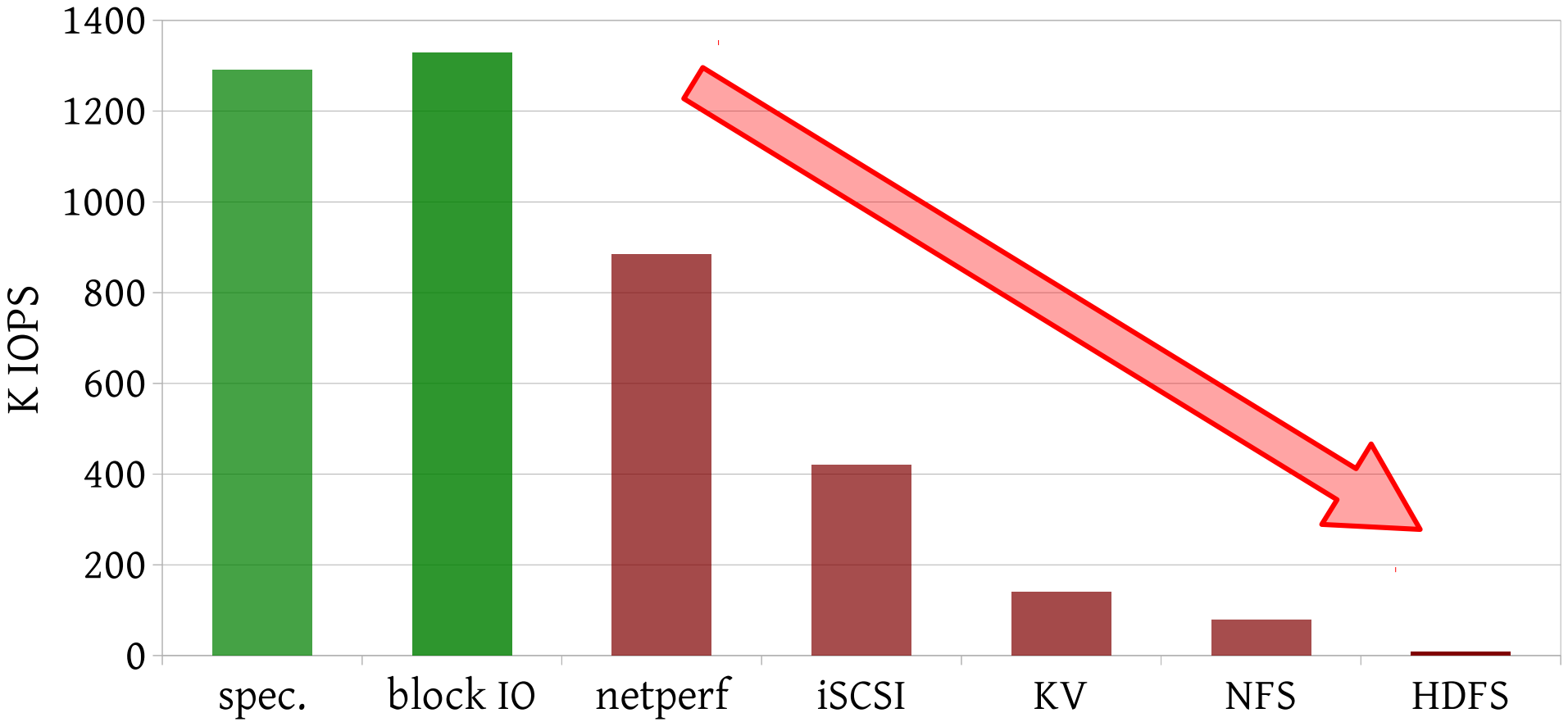
The Cost of the Gap



The Cost of the Gap

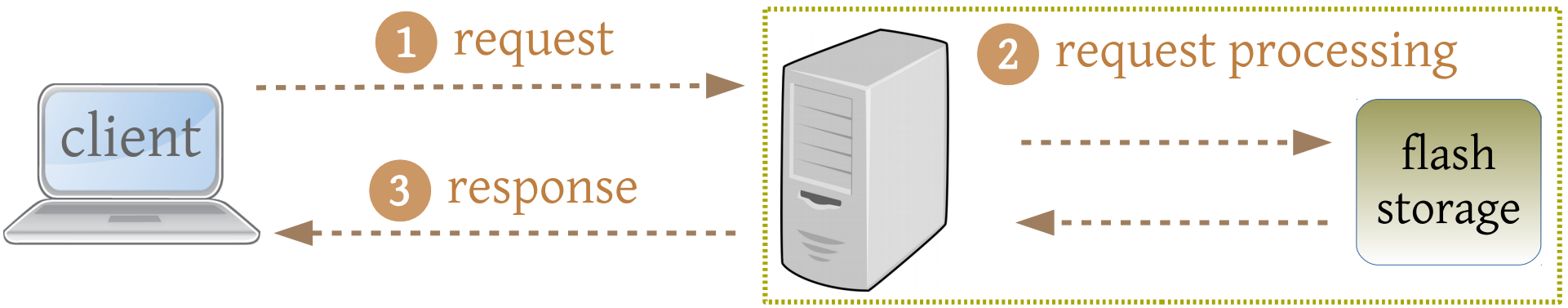


The Cost of the Gap

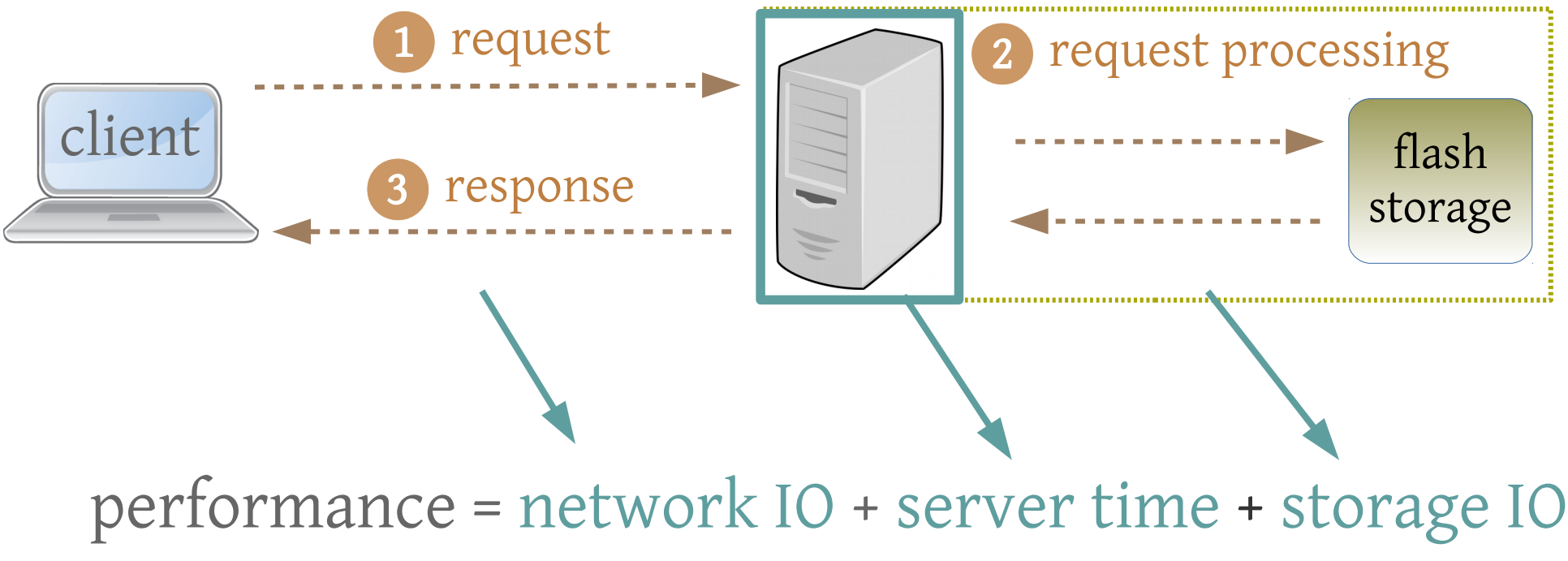


The Reason for the Gap

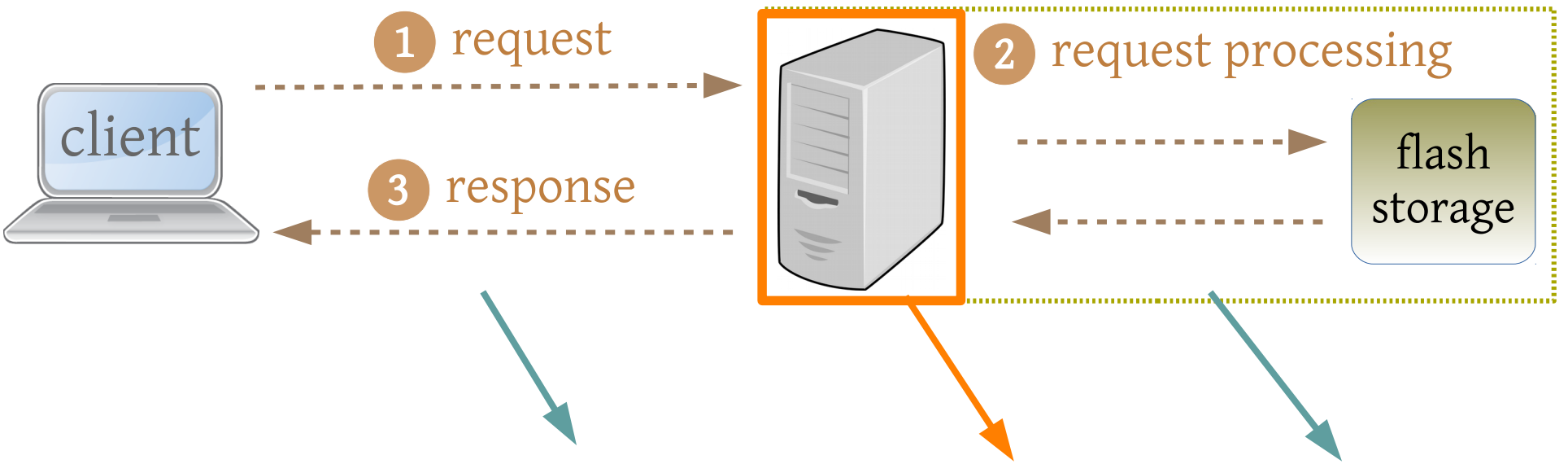
The Reason for the Gap



The Reason for the Gap



The Reason for the Gap



performance = network IO + server time + storage IO

application involvement
scheduling
fs lookups and overheads
...

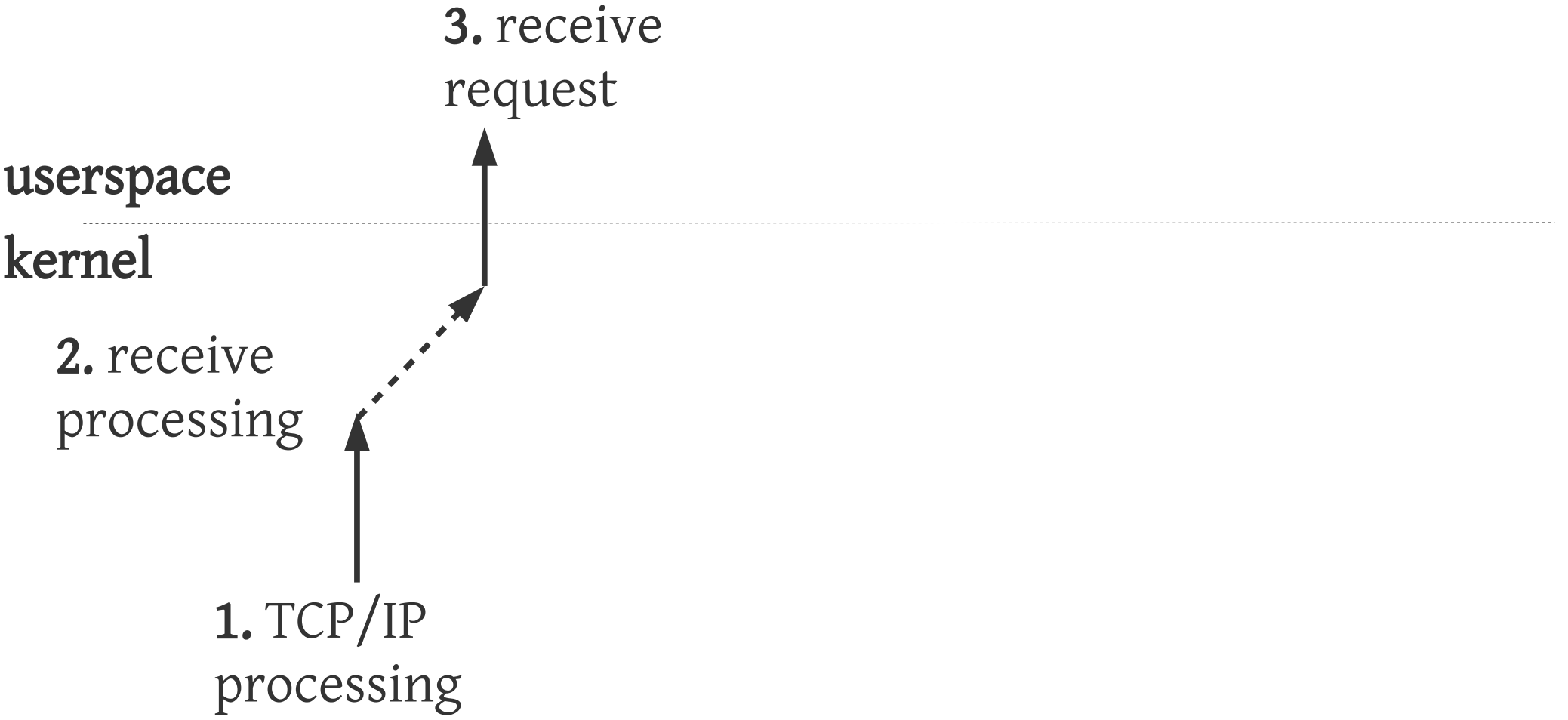
A Detailed Look: send

userspace

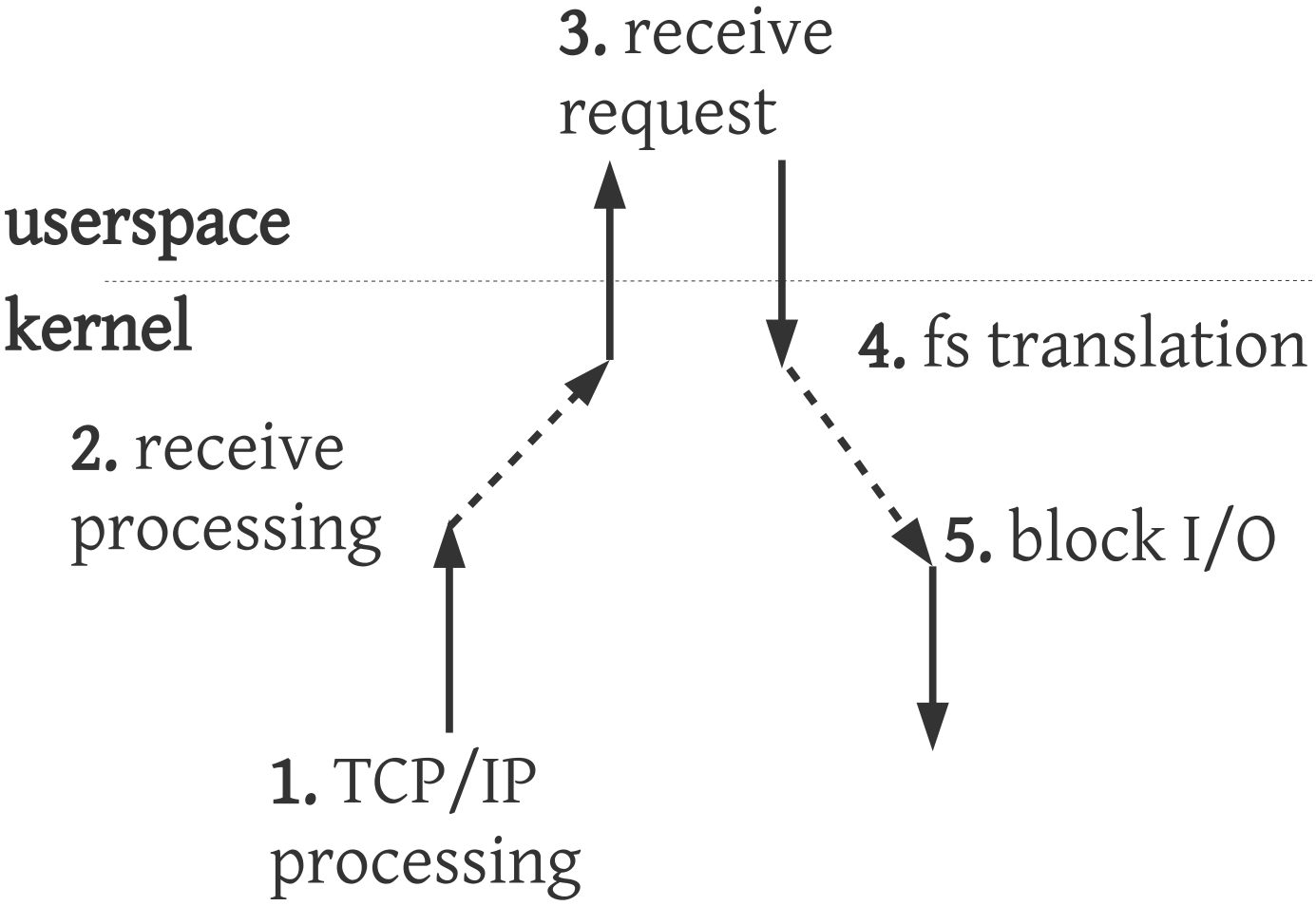
kernel

1. TCP/IP processing

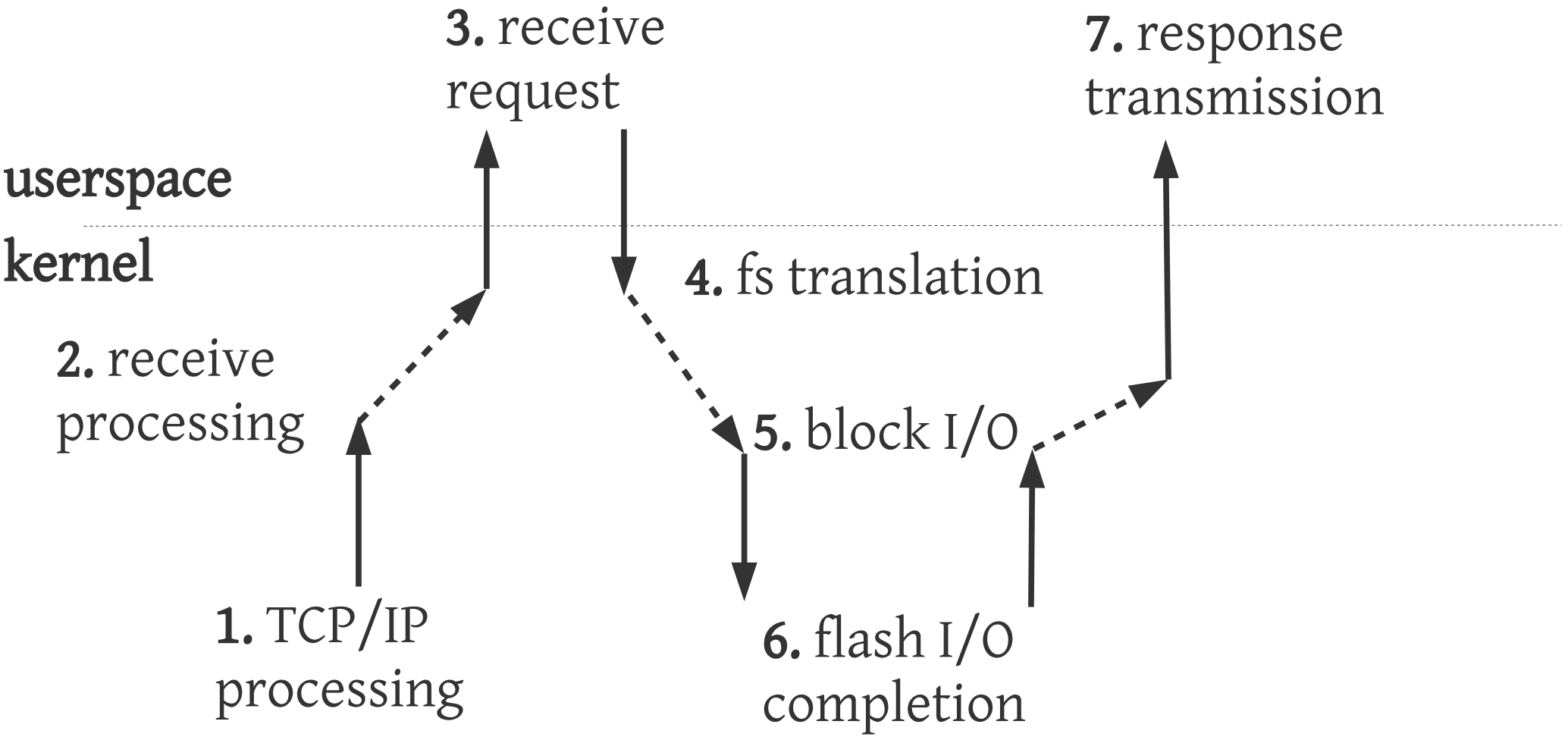
A Detailed Look: send



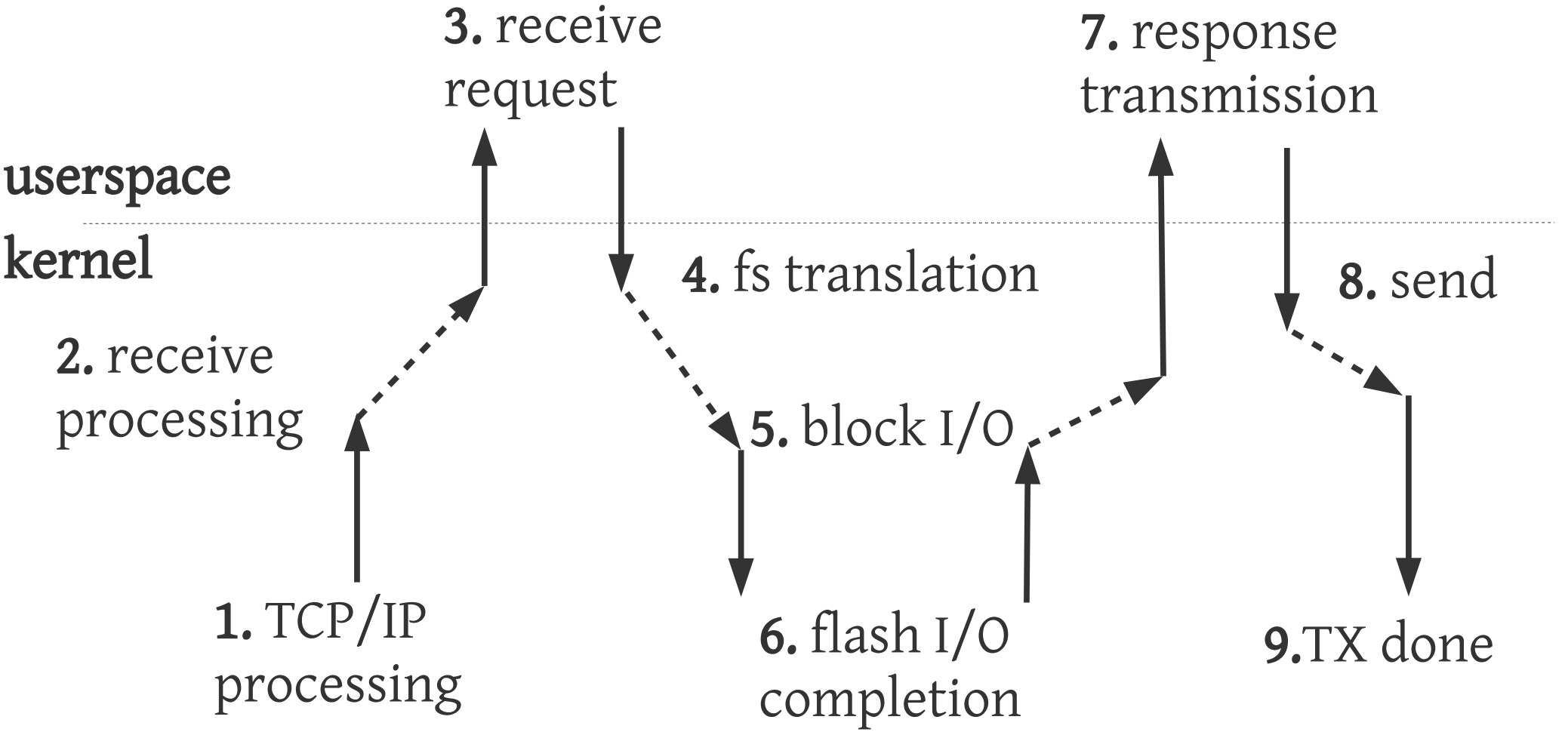
A Detailed Look: send



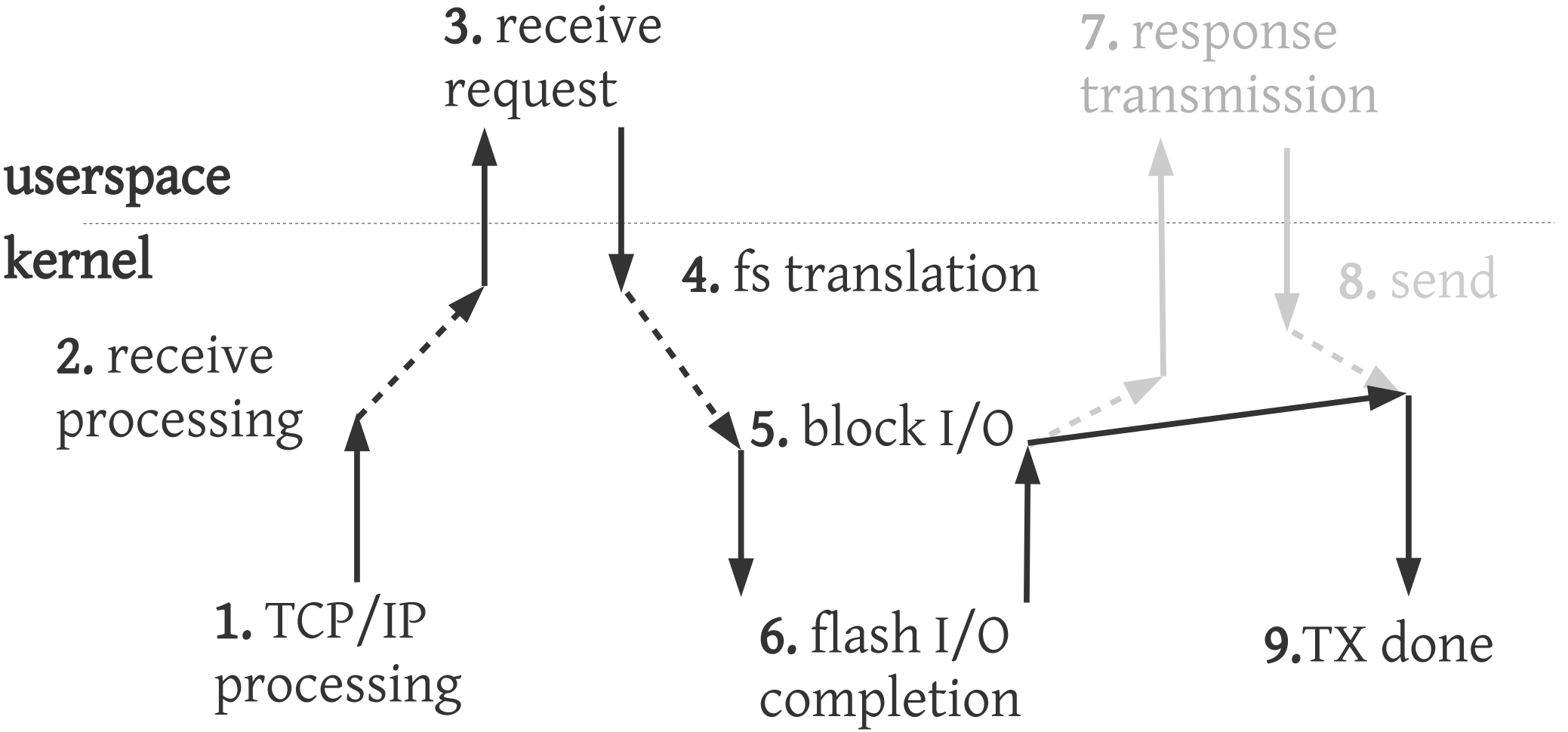
A Detailed Look: send



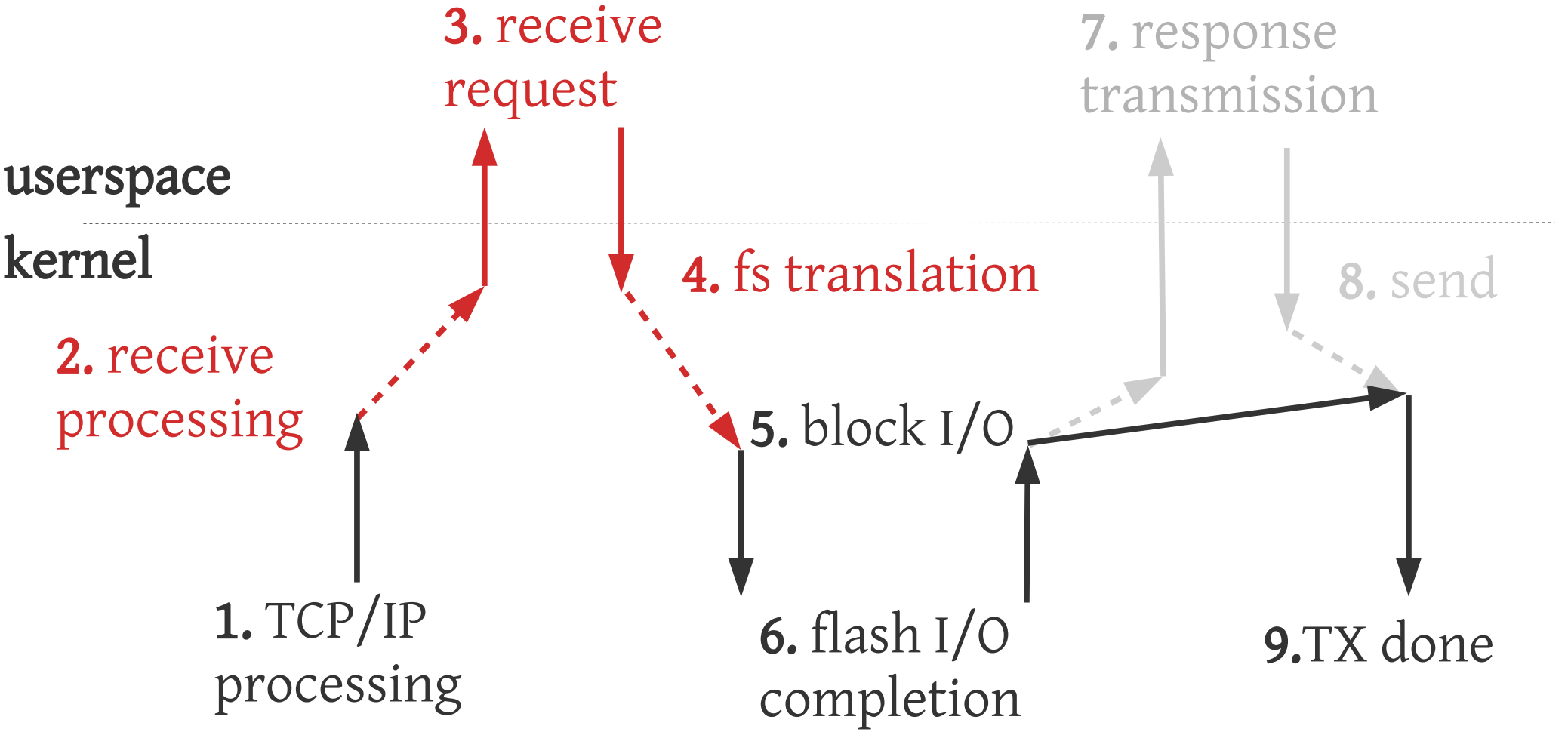
A Detailed Look: send



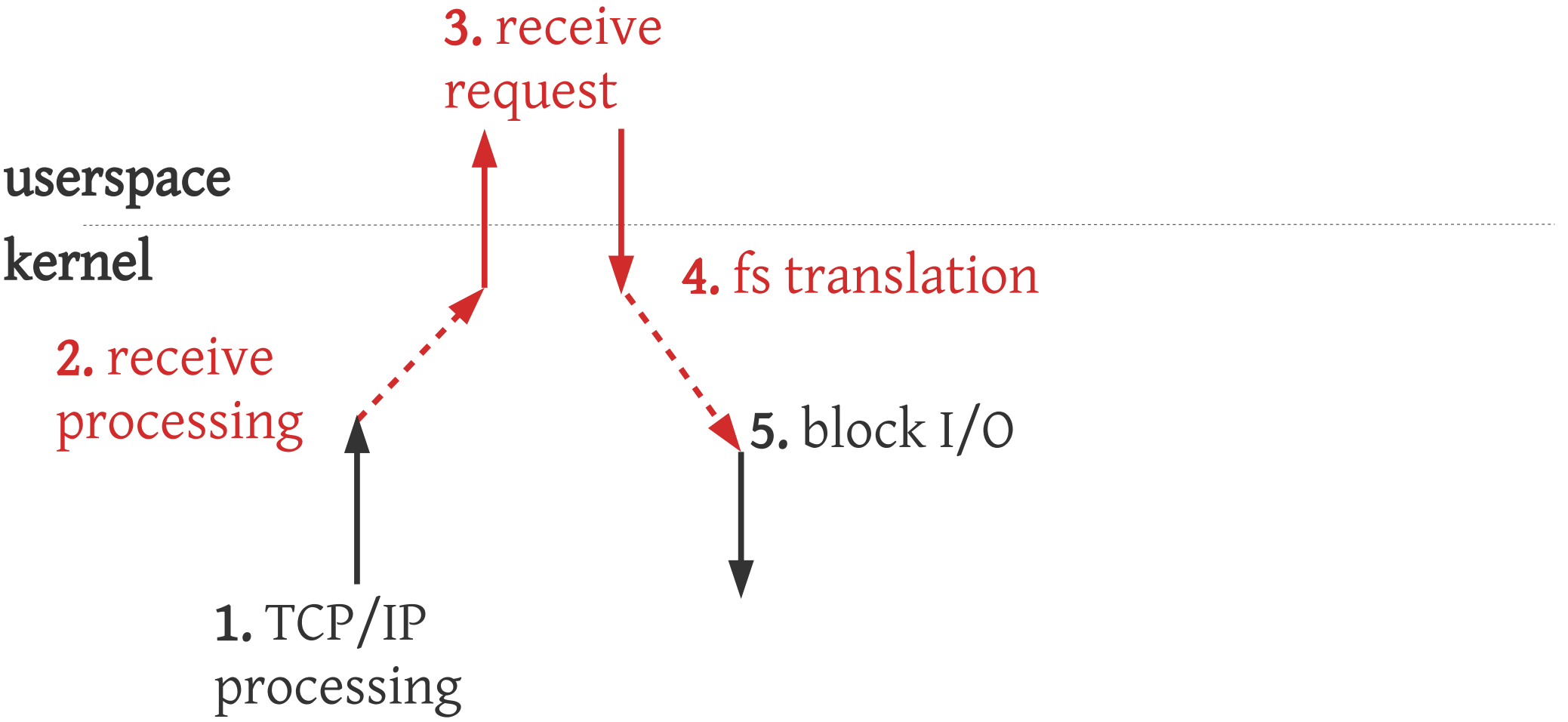
A Detailed Look: sendfile



A Detailed Look: sendfile

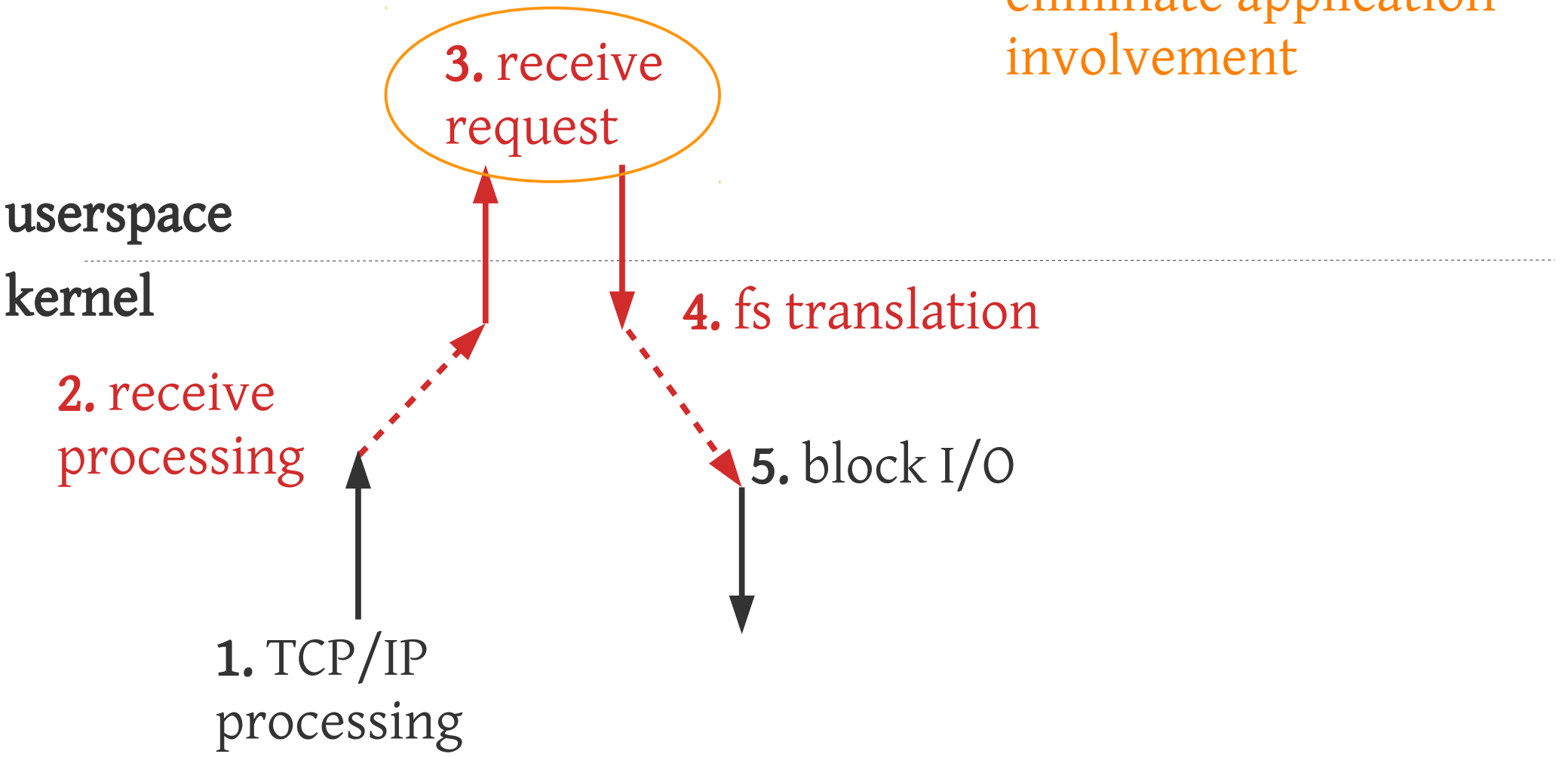


The FlashNet Approach

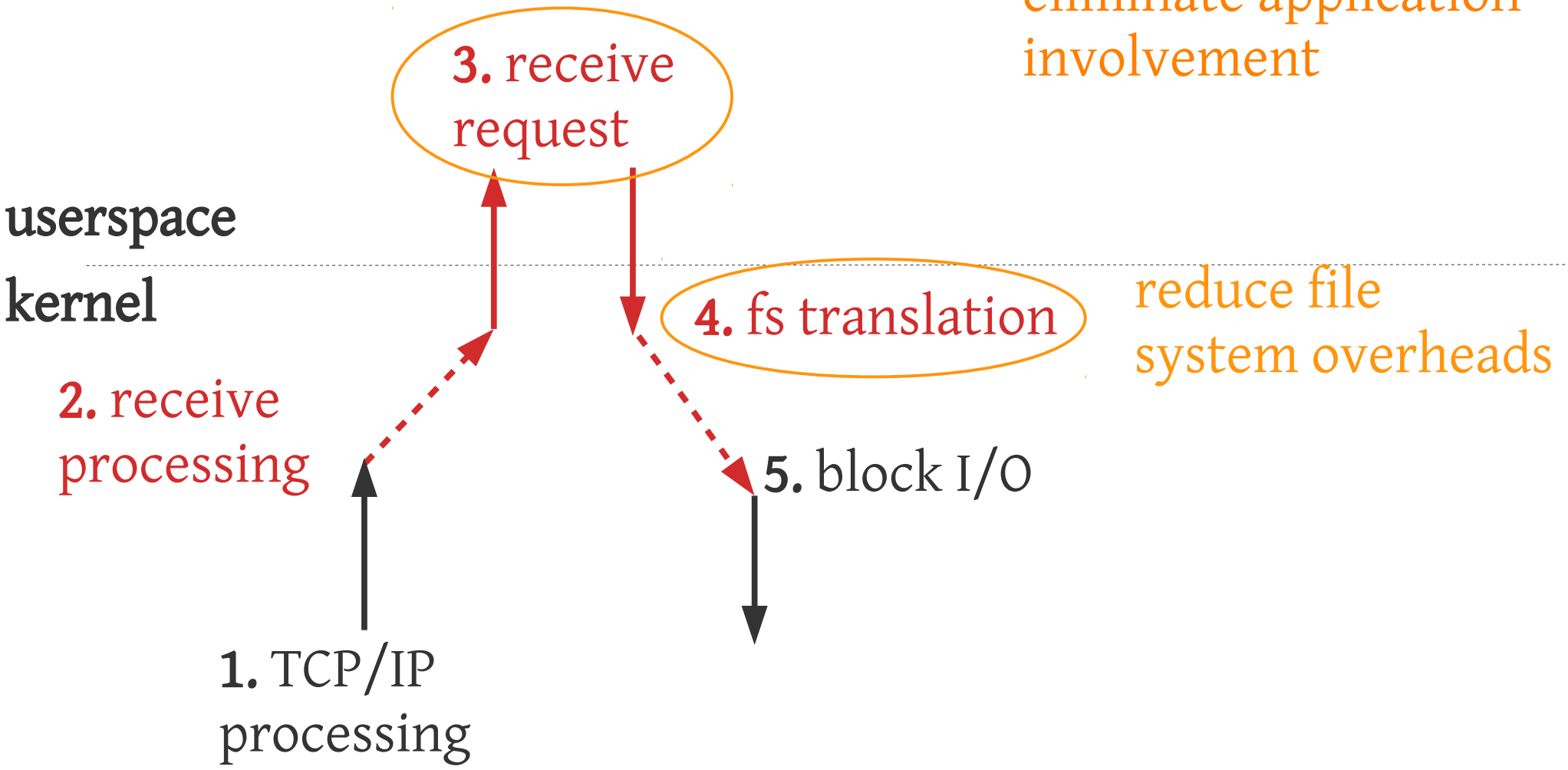


The FlashNet Approach

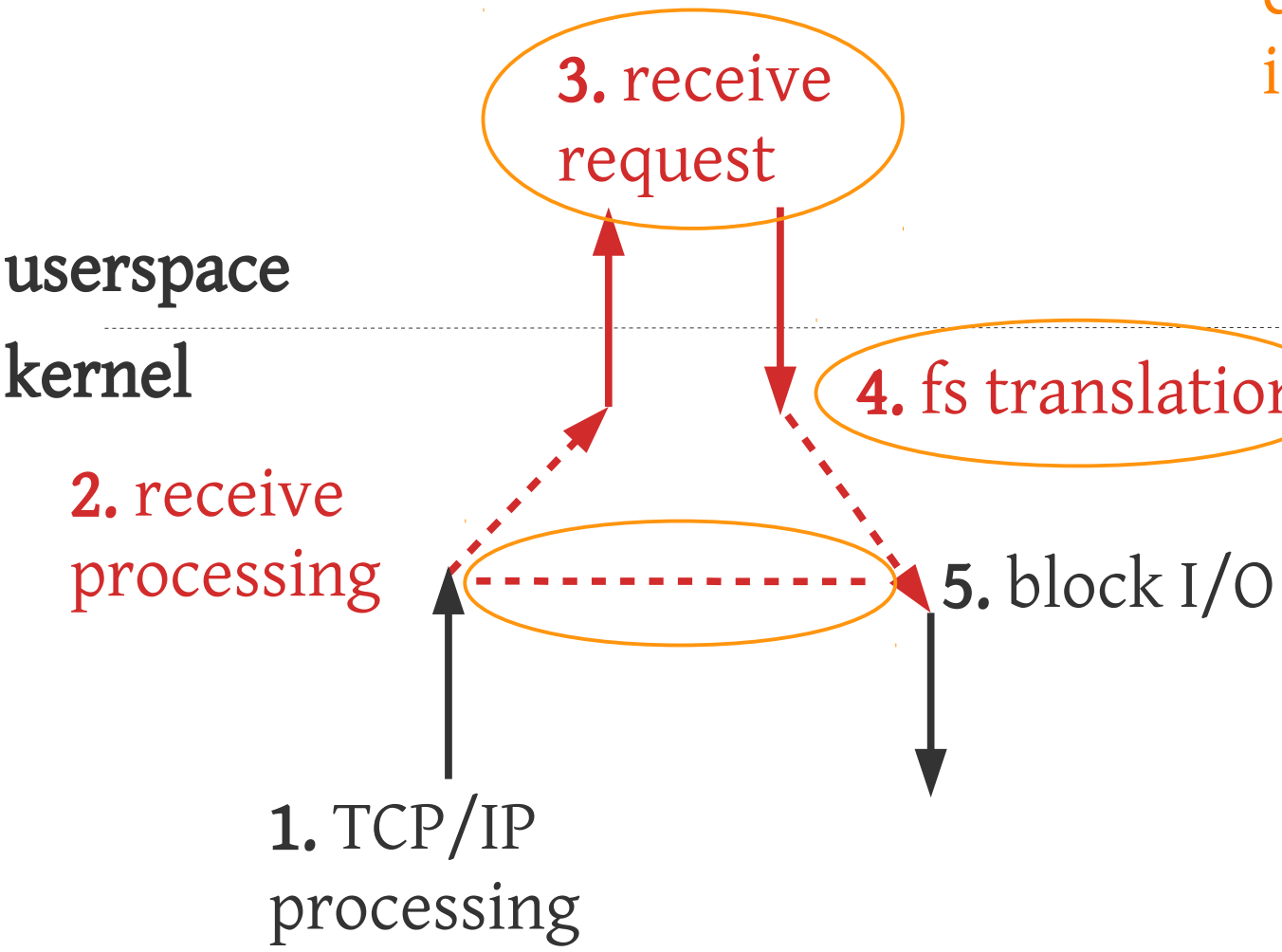
eliminate application involvement



The FlashNet Approach



The FlashNet Approach

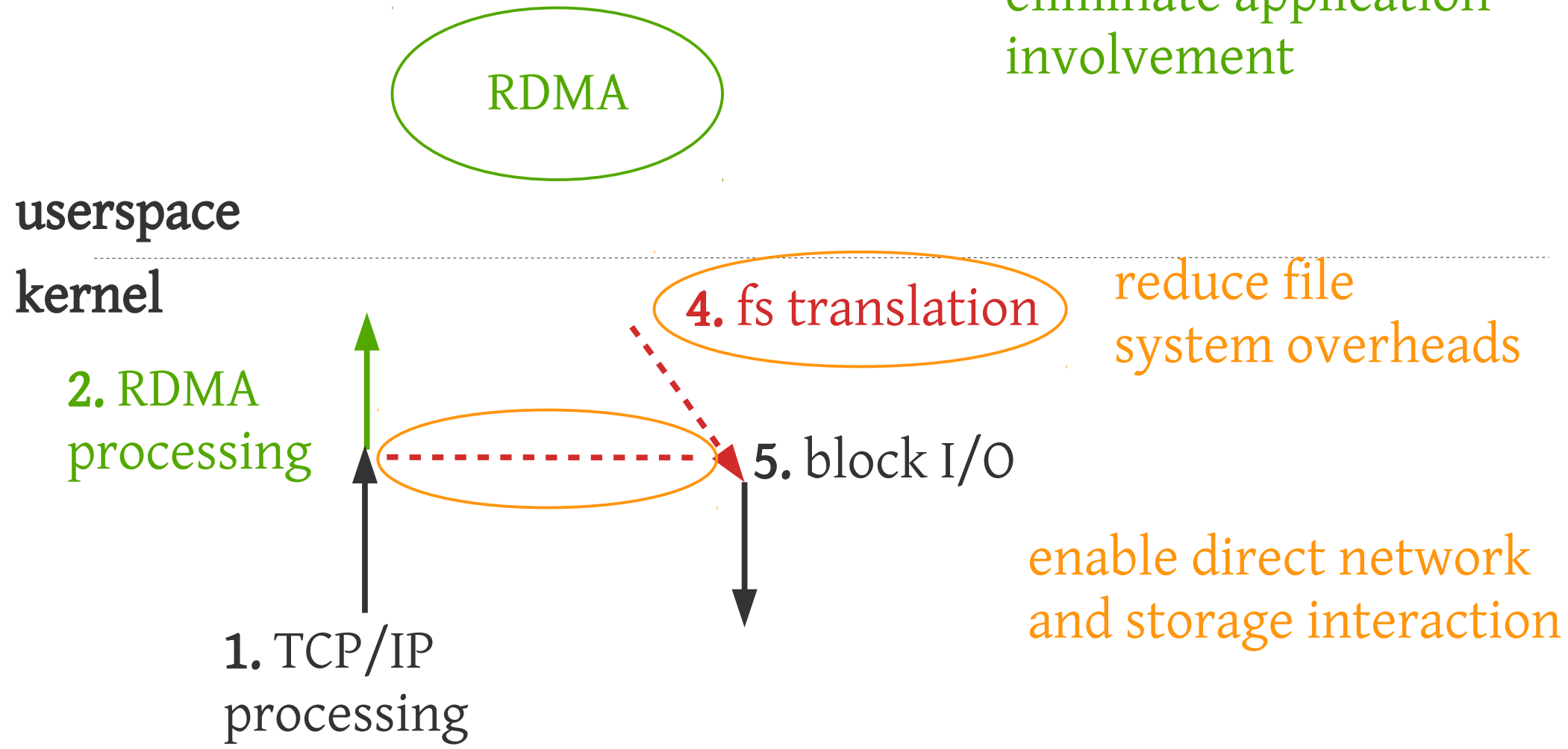


eliminate application involvement

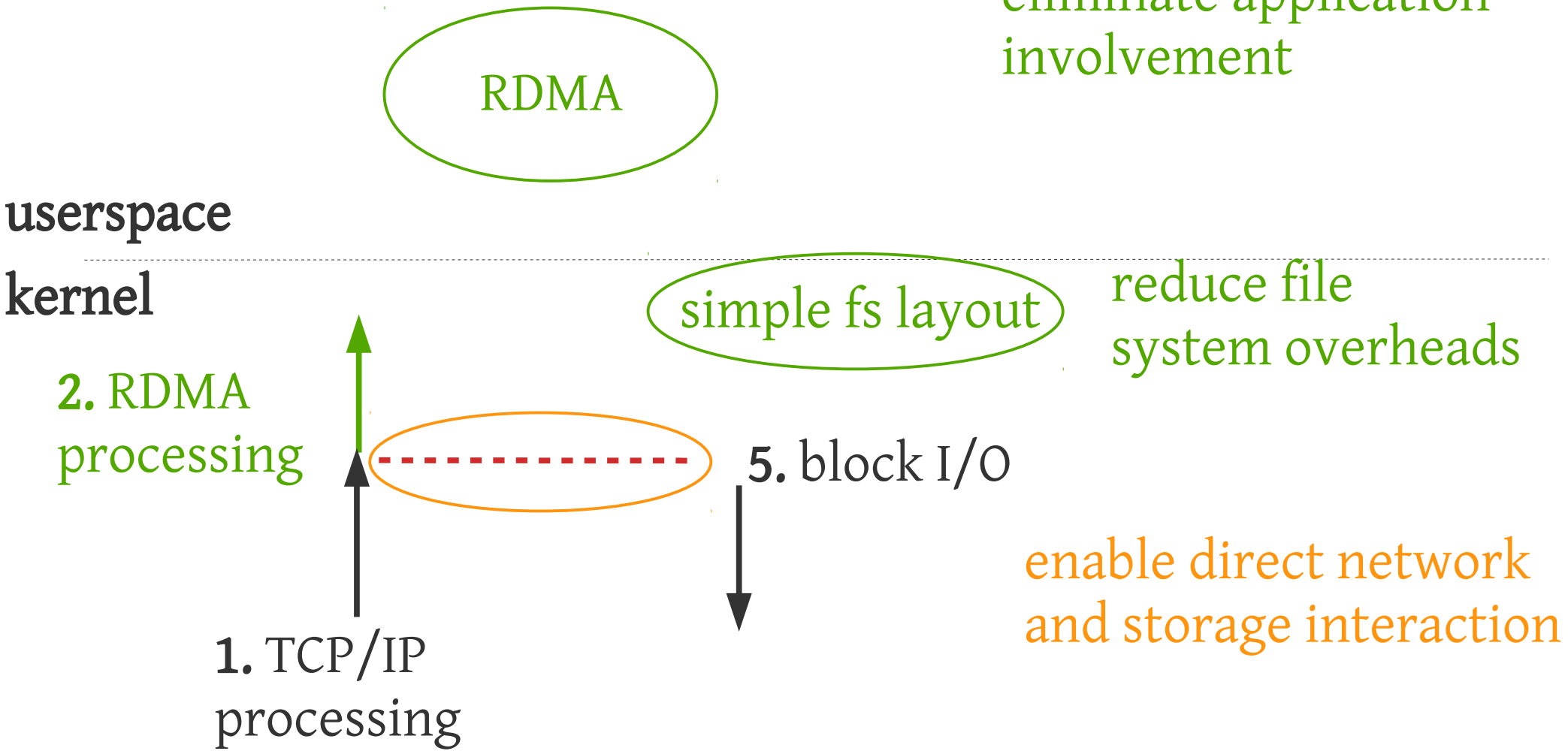
reduce file system overheads

enable direct network and storage interaction

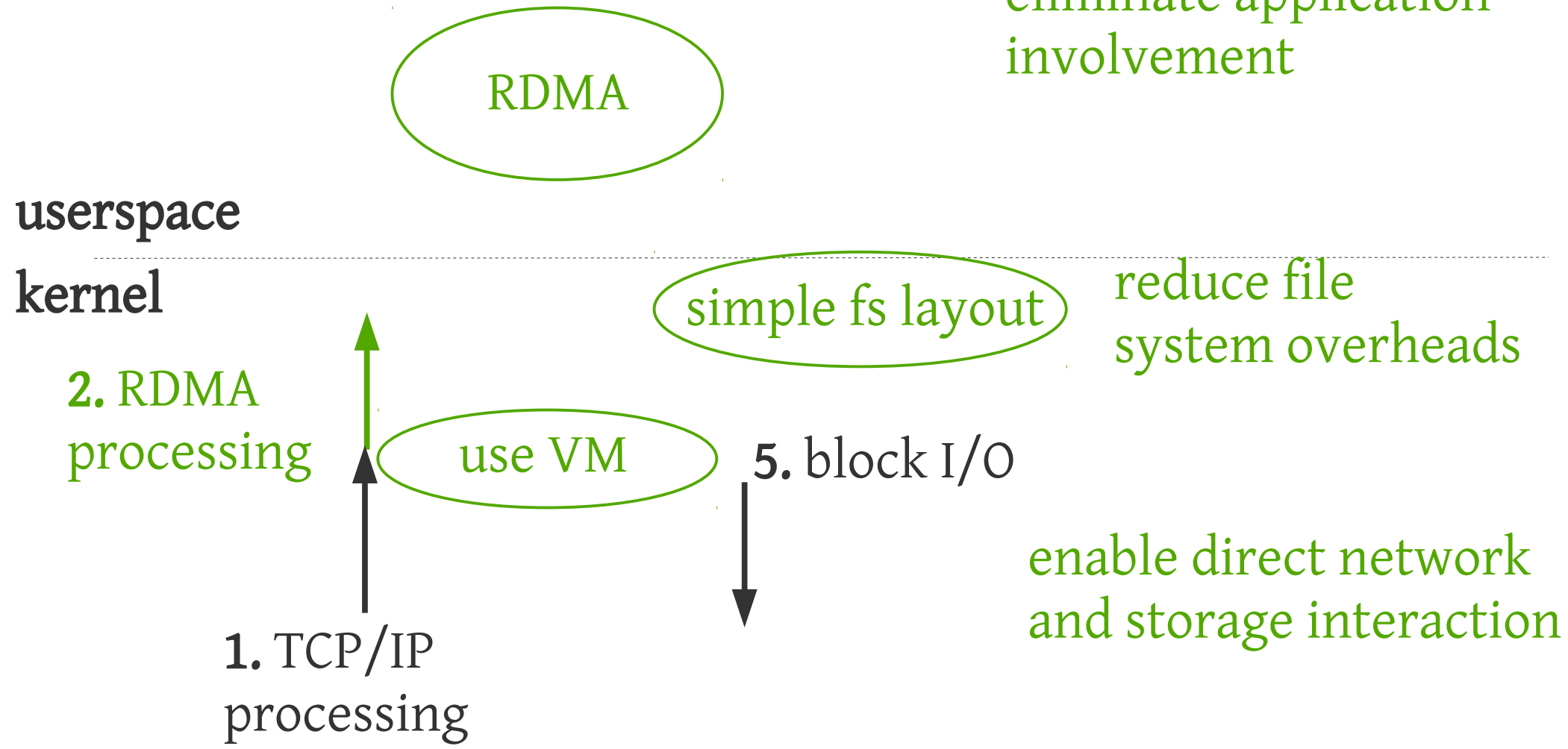
The FlashNet Approach



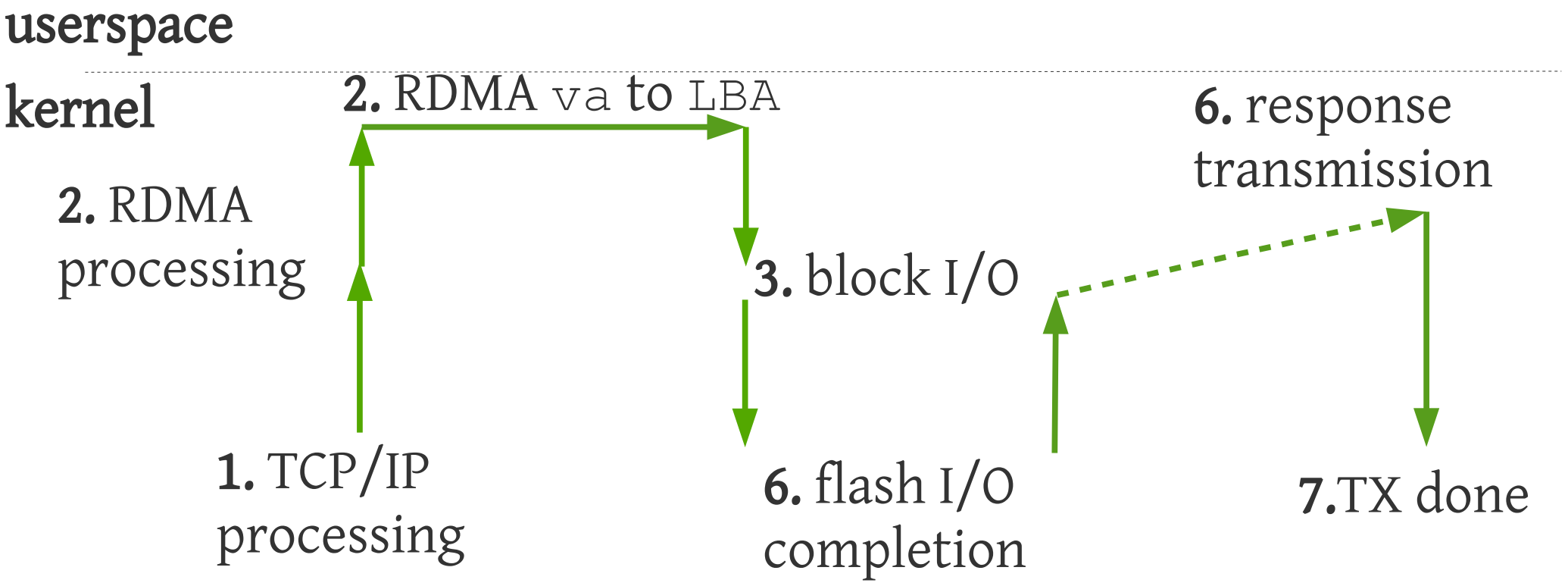
The FlashNet Approach



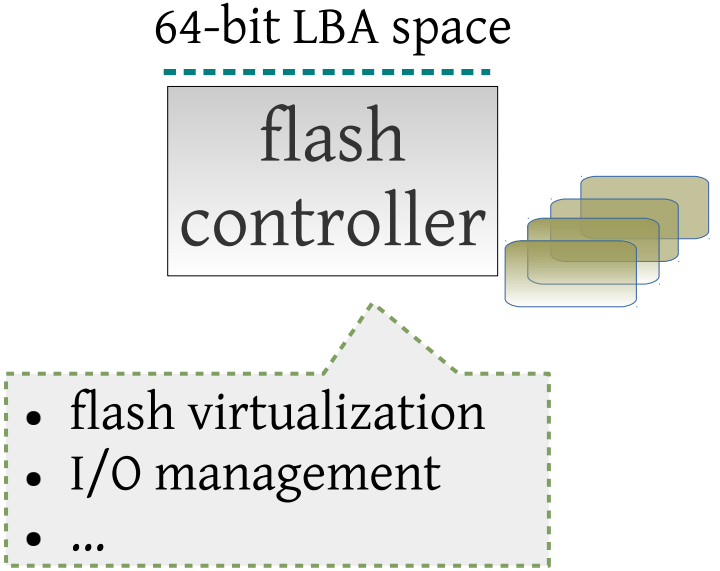
The FlashNet Approach



The FlashNet Approach

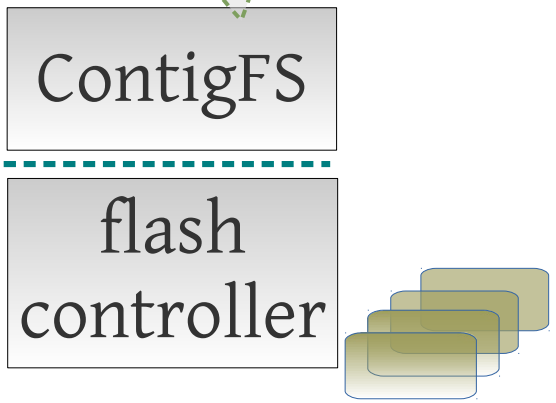


FlashNet: A Co-Designed Network and Storage Stack

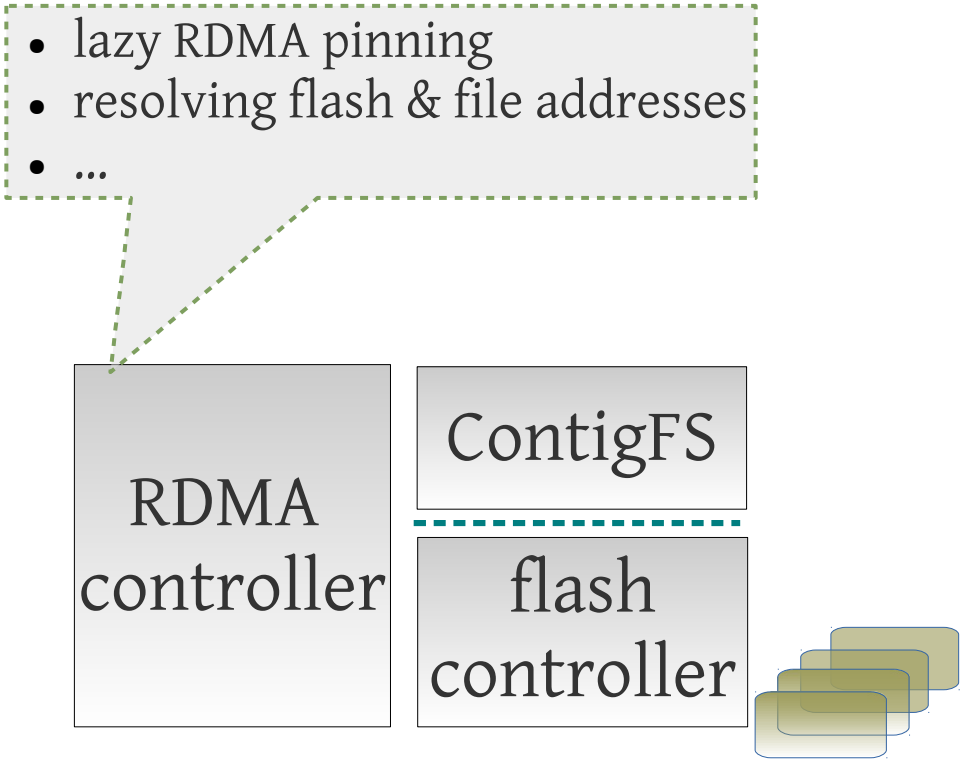


FlashNet: A Co-Designed Network and Storage Stack

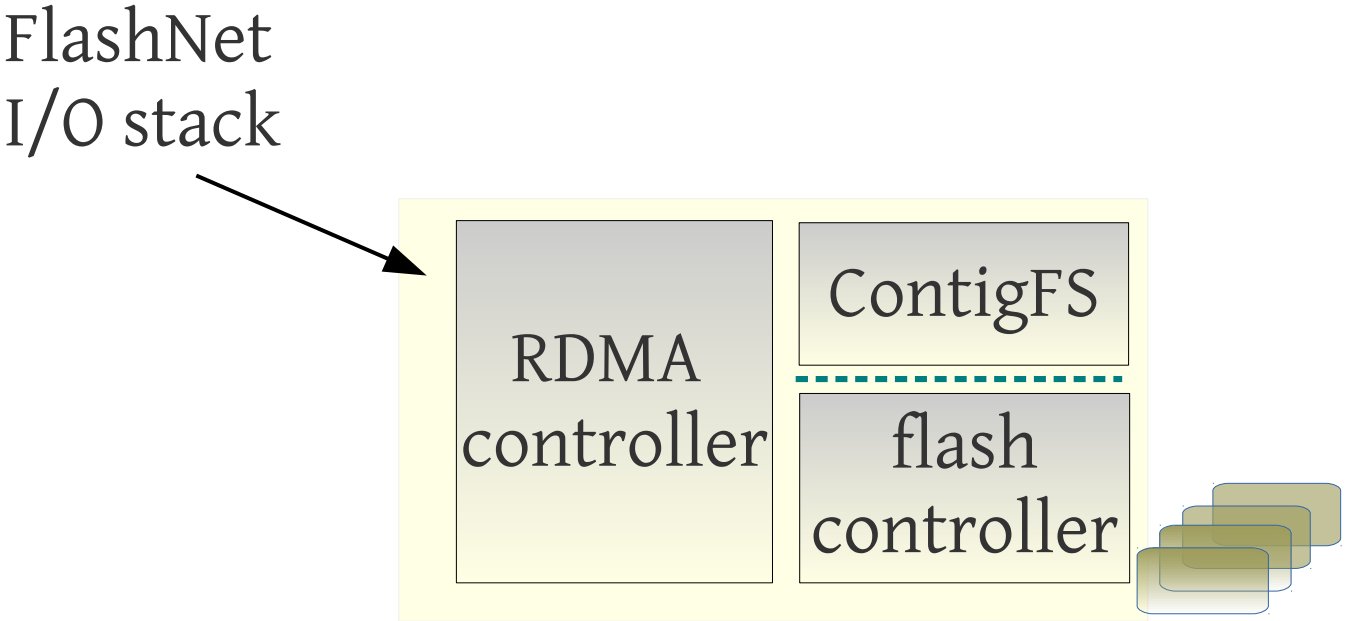
- contigious file allocation
- supporting `mmap` & local file I/O
- ...



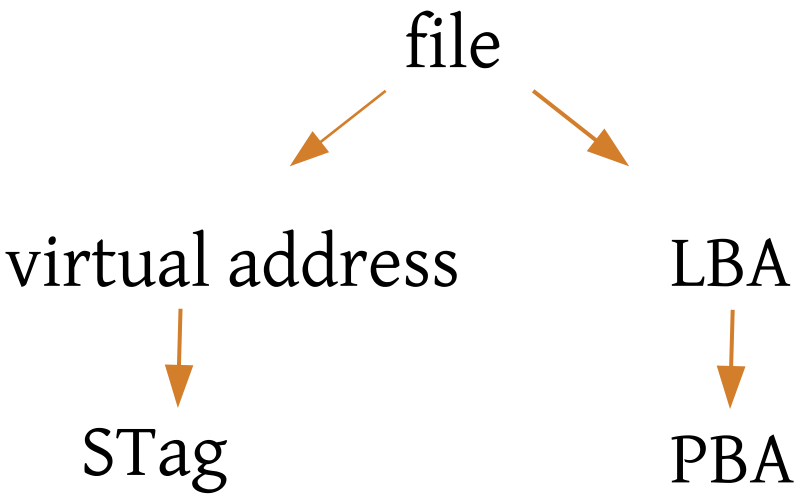
FlashNet: A Co-Designed Network and Storage Stack

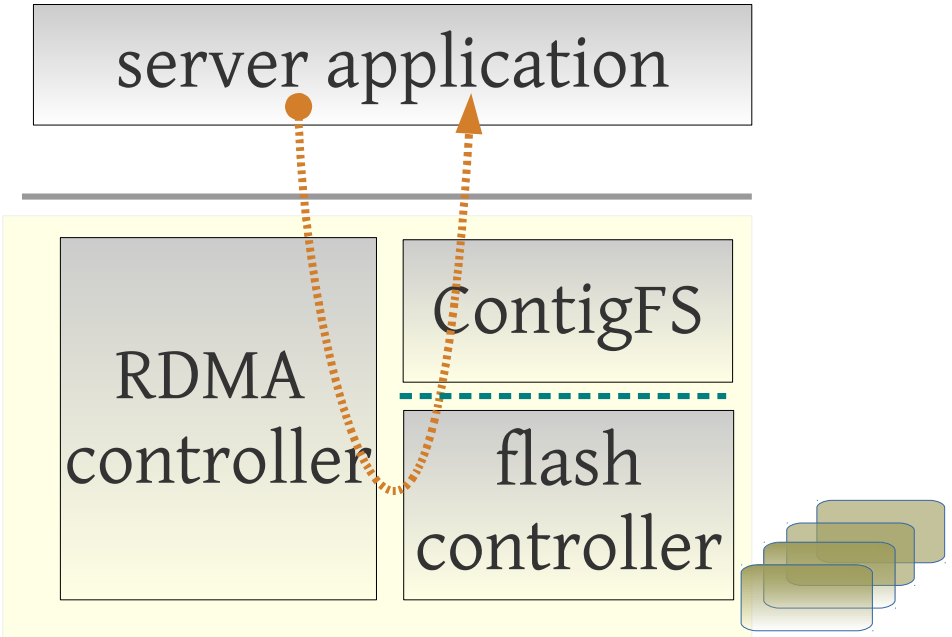


FlashNet: A Co-Designed Network and Storage Stack



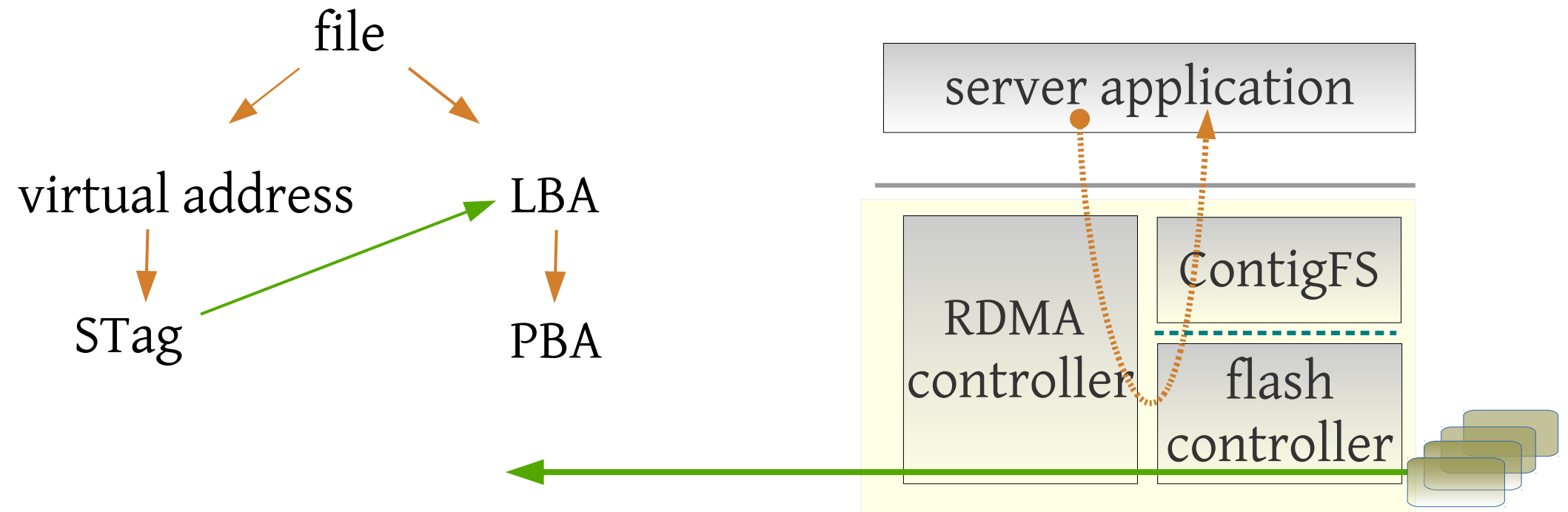
FlashNet: A Co-Designed Network and Storage Stack





network control setup expanding to storage

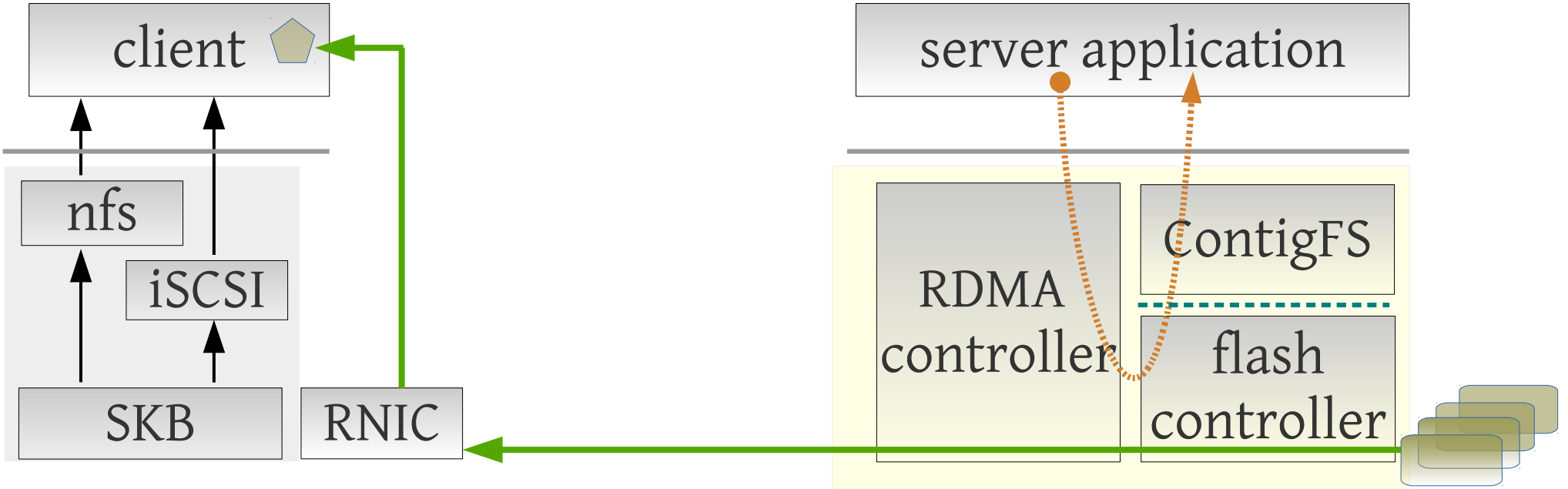
FlashNet: A Co-Designed Network and Storage Stack



network control setup expanding to storage

data path from a flash device to a client buffer

FlashNet: A Co-Designed Network and Storage Stack



network control setup expanding to storage

data path from a flash device to a client buffer

Performance Evaluation

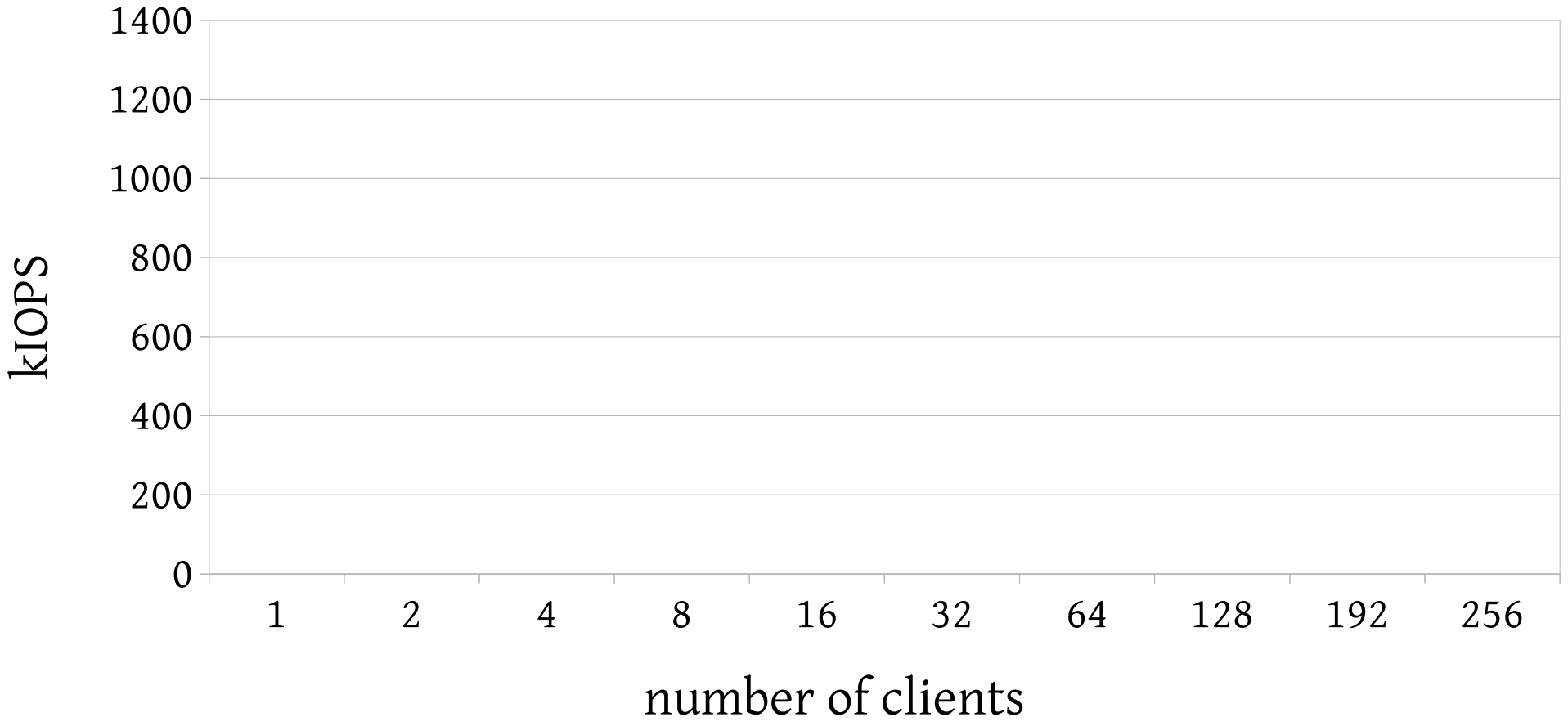
How efficient is FlashNet's IO path?
Does it help with applications?

...more in the paper

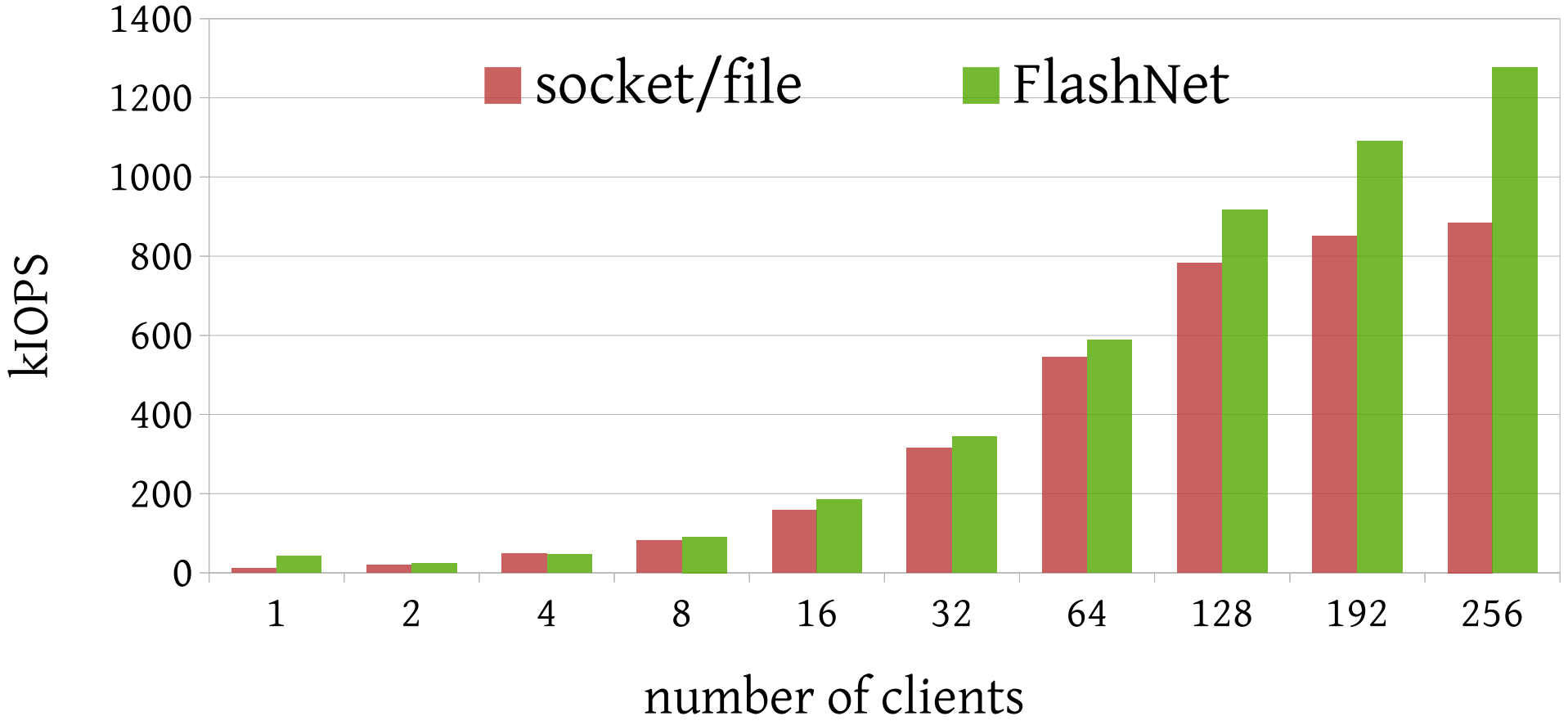
9-machine cluster testbed

CPU	: dual socket E5-2690, 2.9 GHz, 16 cores
DRAM	: 256 GB, DDR3 1600 MHz
NIC	: 40Gbit/s Ethernet
3xNVMe Flash	: 6.6 GB/sec (read), 2.7 GB/sec (write) peak 4kB read IOPS: 1.3 M

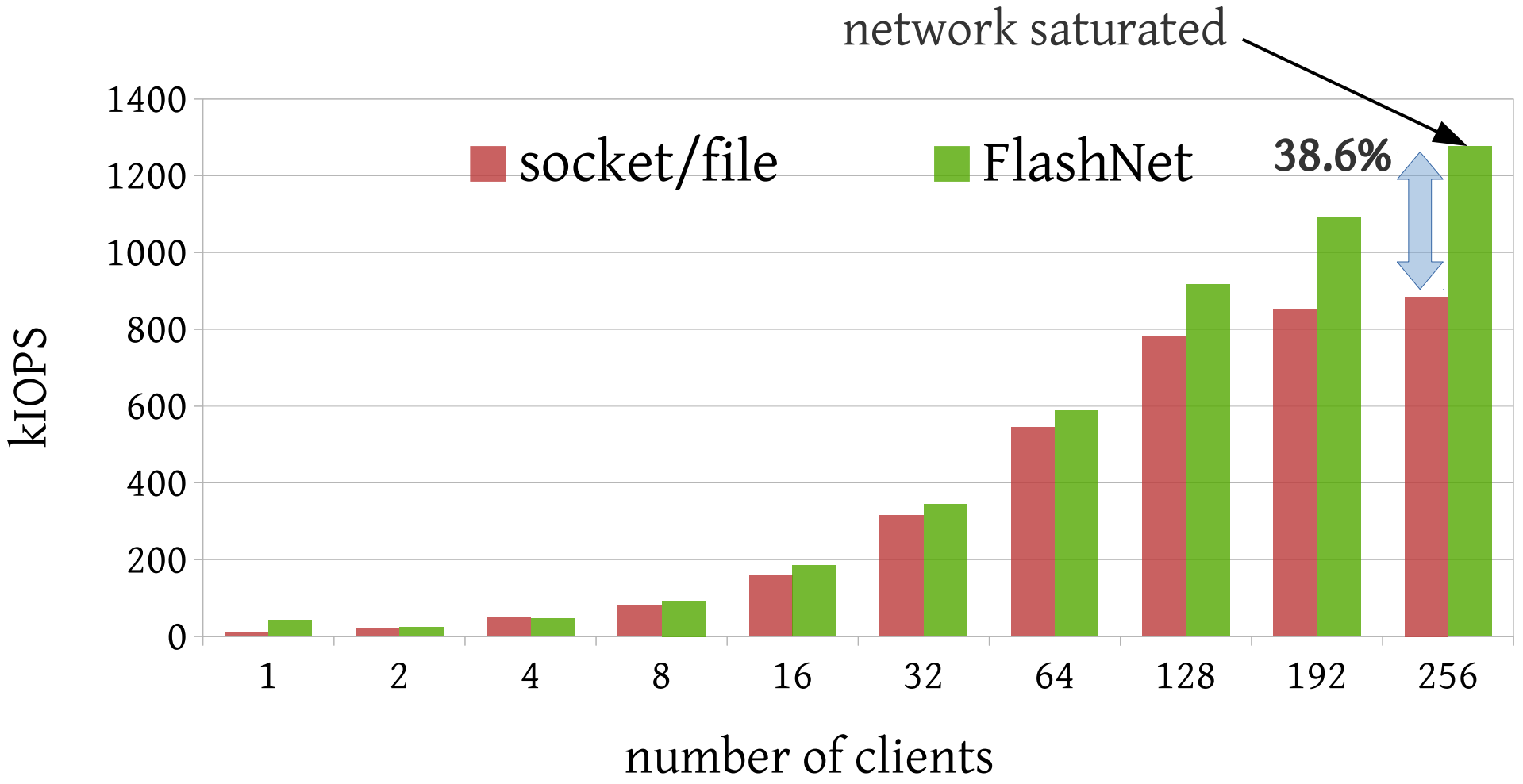
Performance - IOPS Efficiency



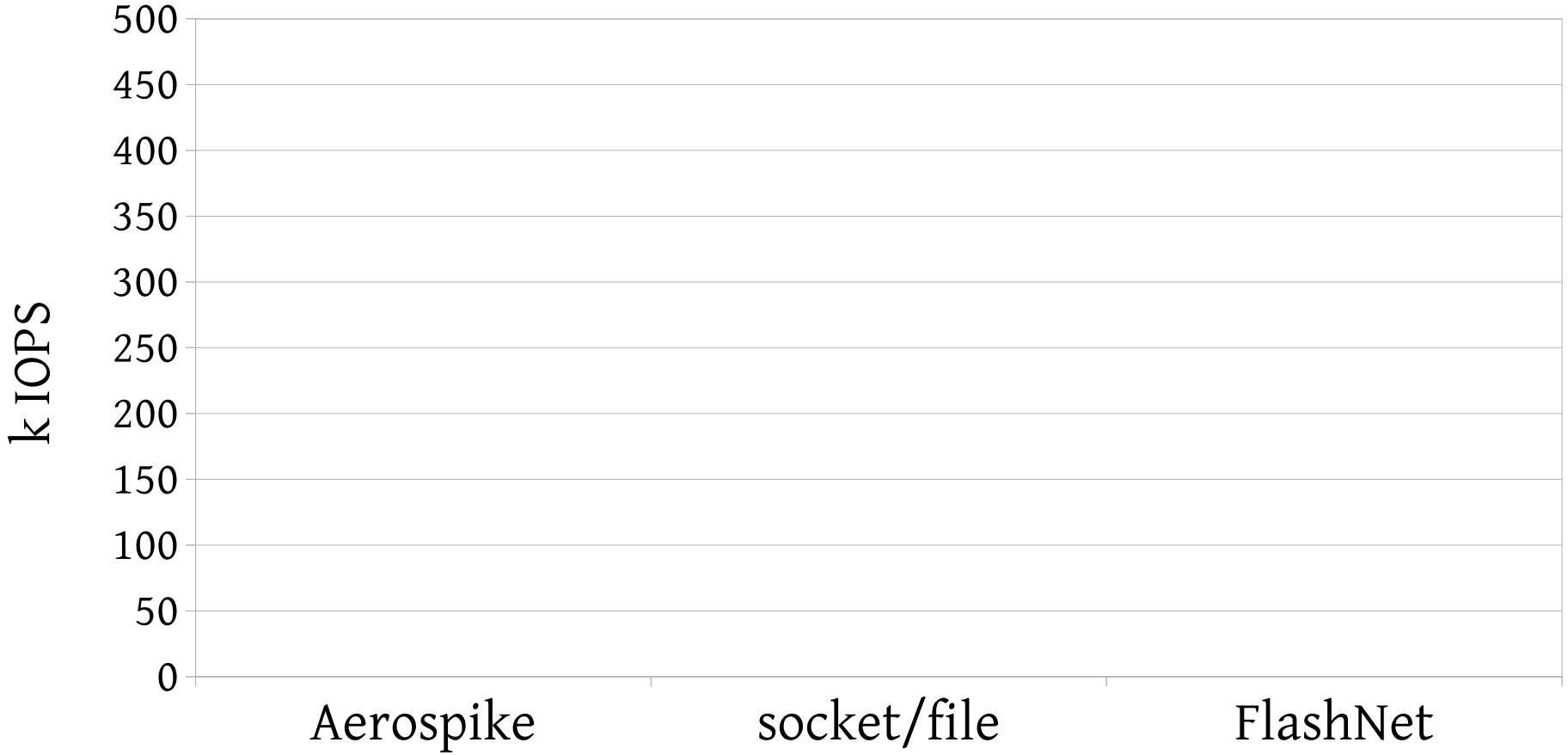
Performance - IOPS Efficiency



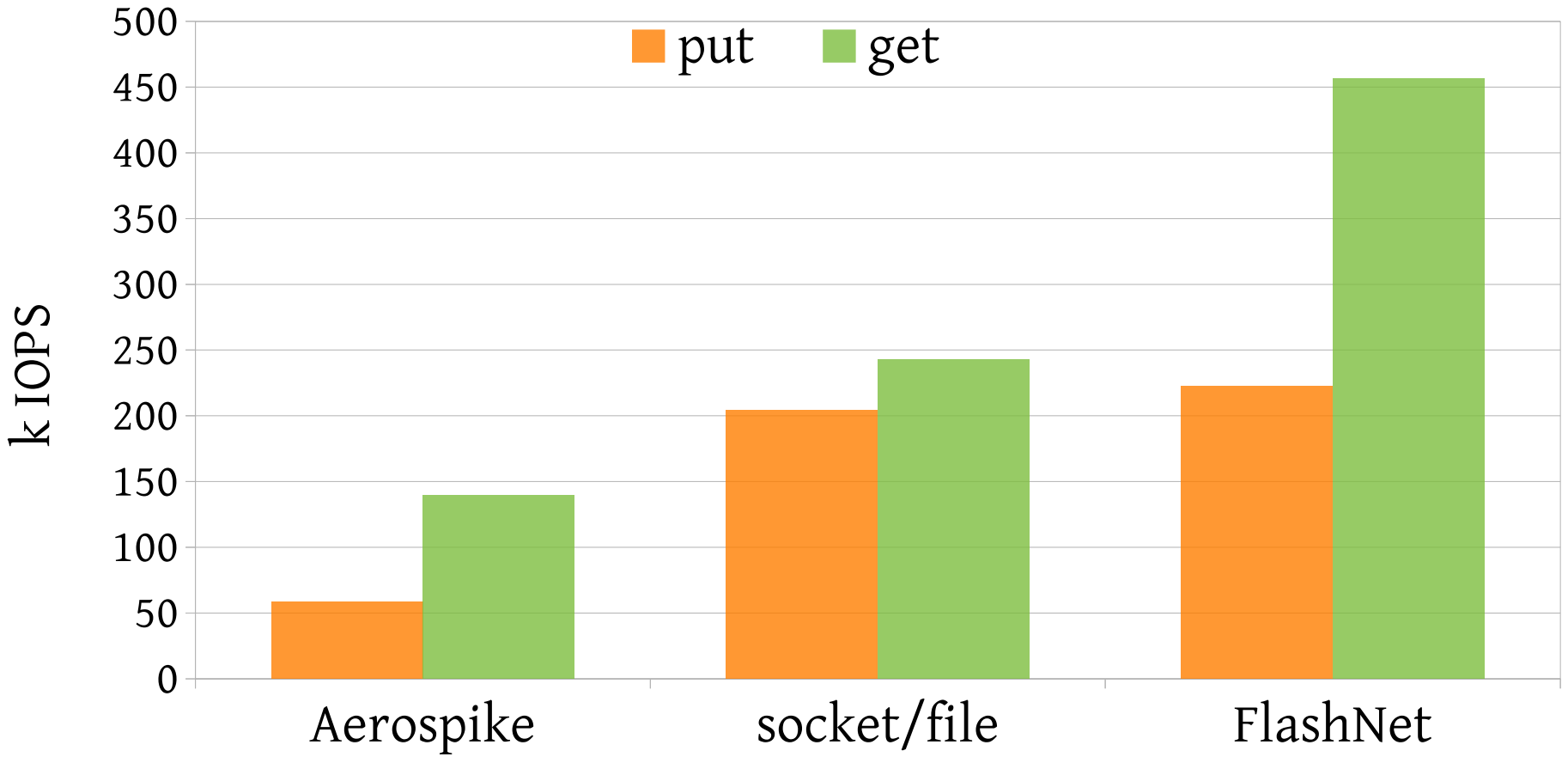
Performance - IOPS Efficiency



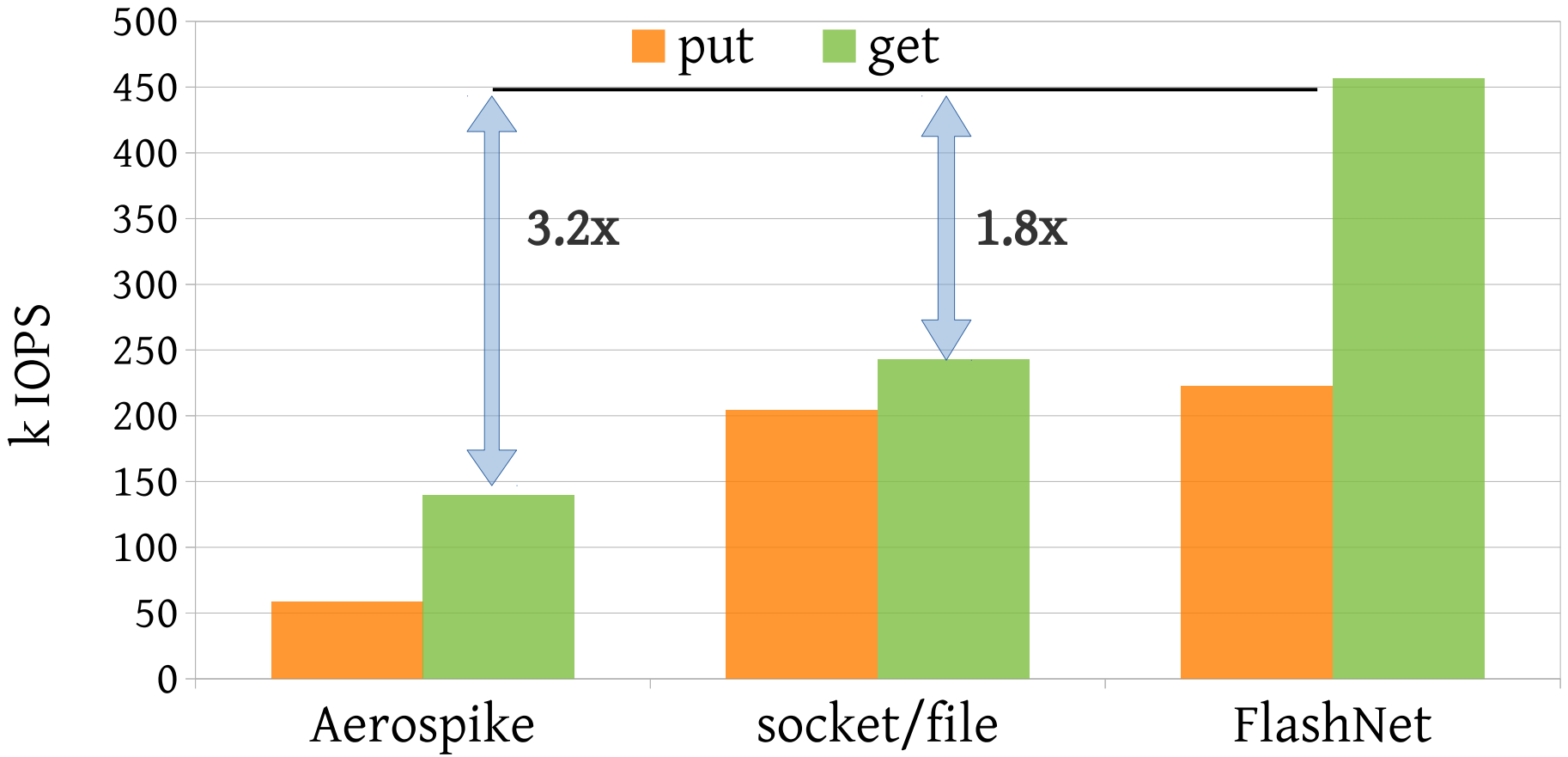
Application-level Performance: KV



Application-level Performance: KV



Application-level Performance: KV



Conclusion

Identified performance issues with networked flash

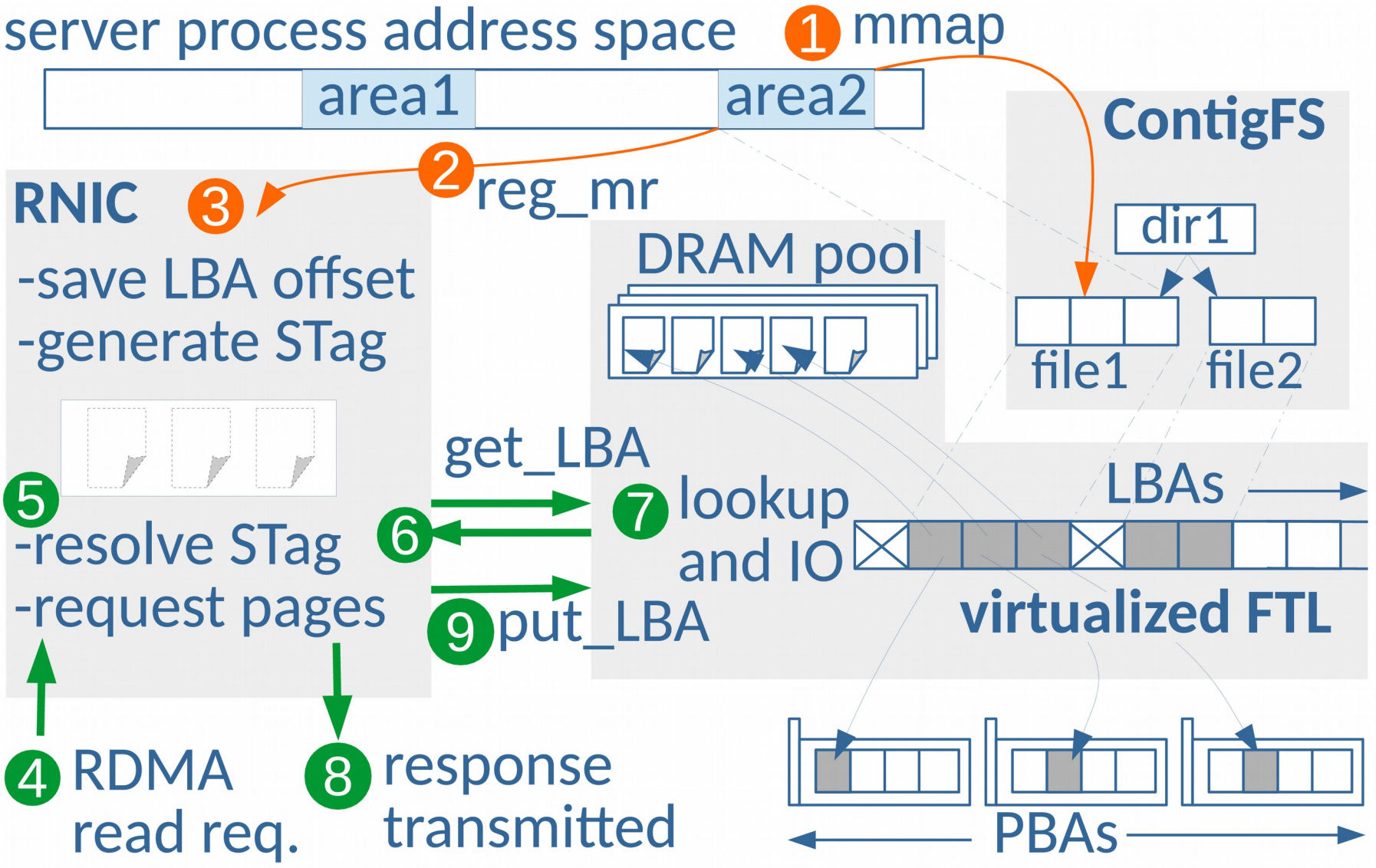
Apply RDMA principles and concepts by extending the path separation idea to a flash controller and a file system

FlashNet is a concrete implementation of this idea
- demonstrated its capabilities in micro-benchmarks and applications

Excited to explore new use-cases for FlashNet

Thank you

Unified I/O Life Cycle



CPU Cycles Breakdown

	network	storage	device drivers	scheduling	kernel	request processing	misc
Socket/file	19.3%	7.3%	6.7%	15.8%	40.1%	4.7%	6.1%
FlashNet	20.6%	0.8%	6.4%	8.4%	46.7%	11.7%	5.4%