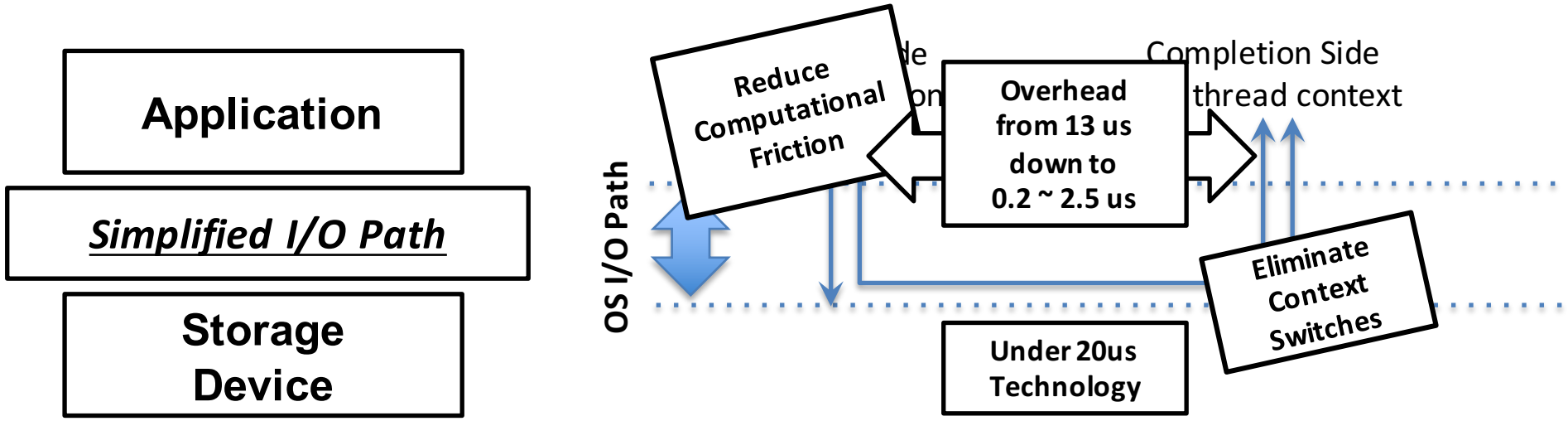
Unblinding the OS to Optimize User-Perceived Flash SSD Latency

Woong Shin[*], Jaehyun Park[**], Heon Y. Yeom[*]
[*]Seoul National University
[**]Arizona State University

USENIX HotStorage 2016
Jun. 21, 2016

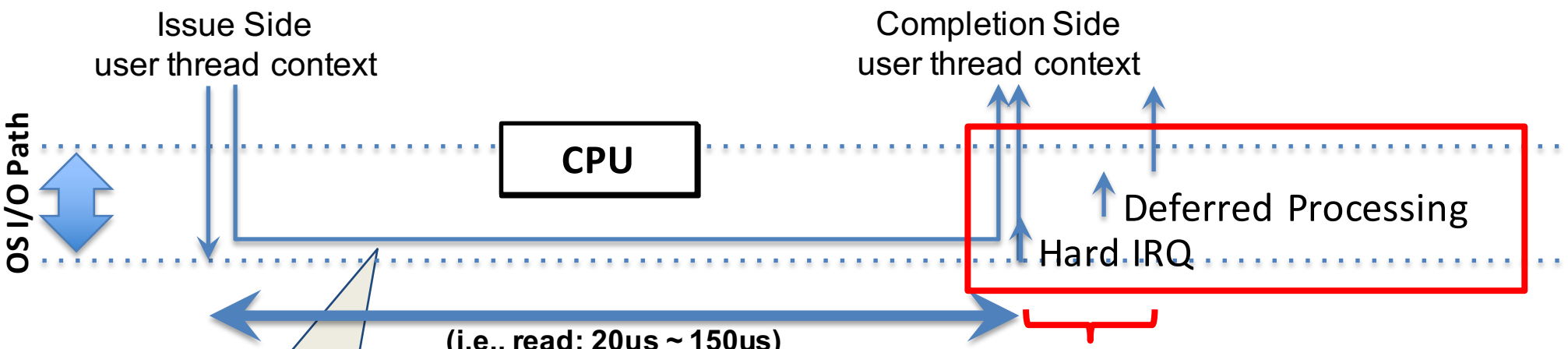
OS I/O Path Optimizations: Reducing S/W Overheads



Memory technology with ultra low (nanoseconds), predictable latencies.

"Block CPU (poll) while waiting for the I/O to complete"

To Block the CPU (sync) or to Yield the CPU (async)



Cannot use polling:
Wastes CPU cycles
Harms system parallelism

H/W context switch := 4 us ~ 5 us
OS involved switch := 7 us ~ 8 us

"Impact of Modern SSDs"

More contexts on a CPU core

Larger scheduling delays

Impact of Modern SSDs

700,000 IOPS NVM-e SSD

Bandwidth:

4kB x 700 kIOPS = 2.8 GB/s

Latency:

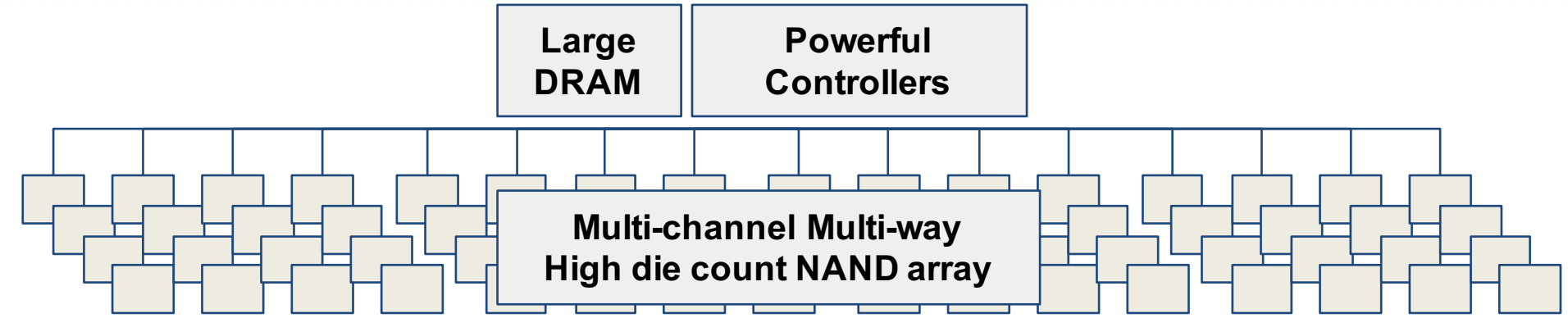
~~**1 sec / 700 kIOPS = 1.42 µs**~~ **?!?**

Impact of Modern SSDs

700,000 IOPS NVM-e SSD

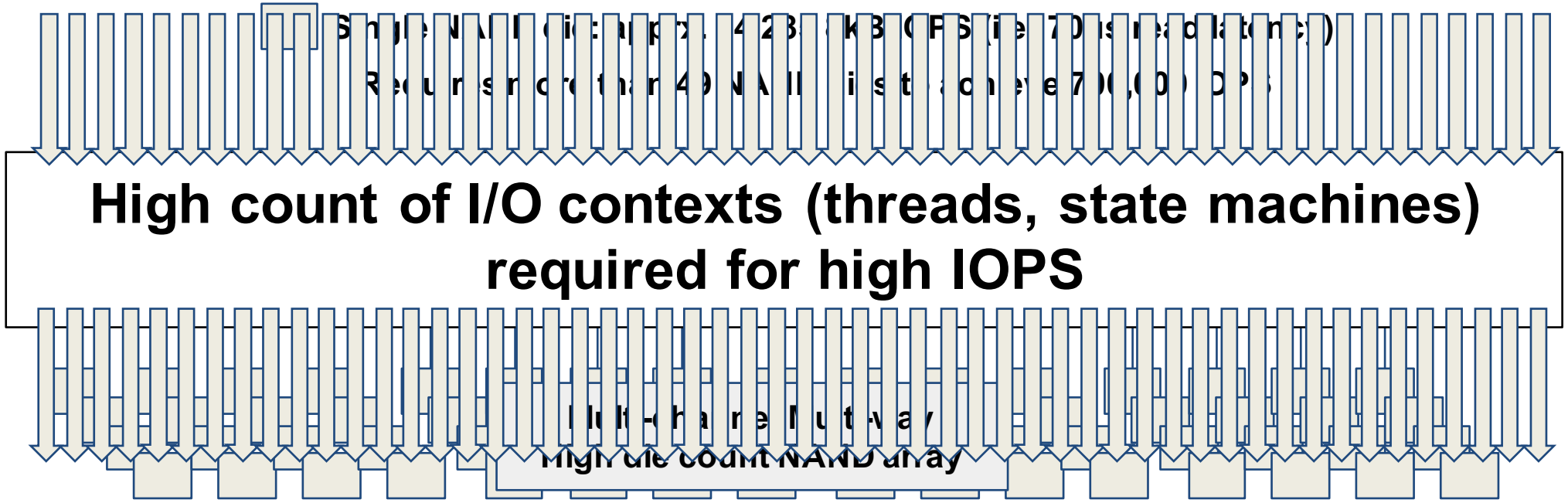
Single NAND die: apprx. 14,285 8kB IOPS (i.e, 70us read latency)

Requires more than 49 NAND dies to achieve 700,000 IOPS

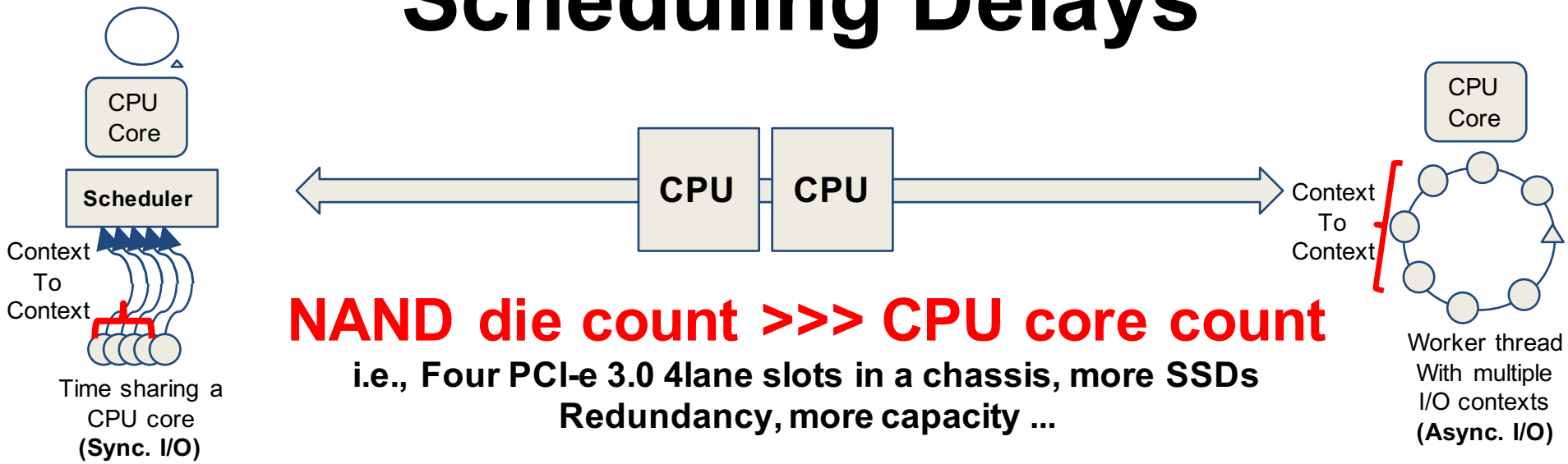


Impact of Modern SSDs

700,000 IOPS NVM-e SSD

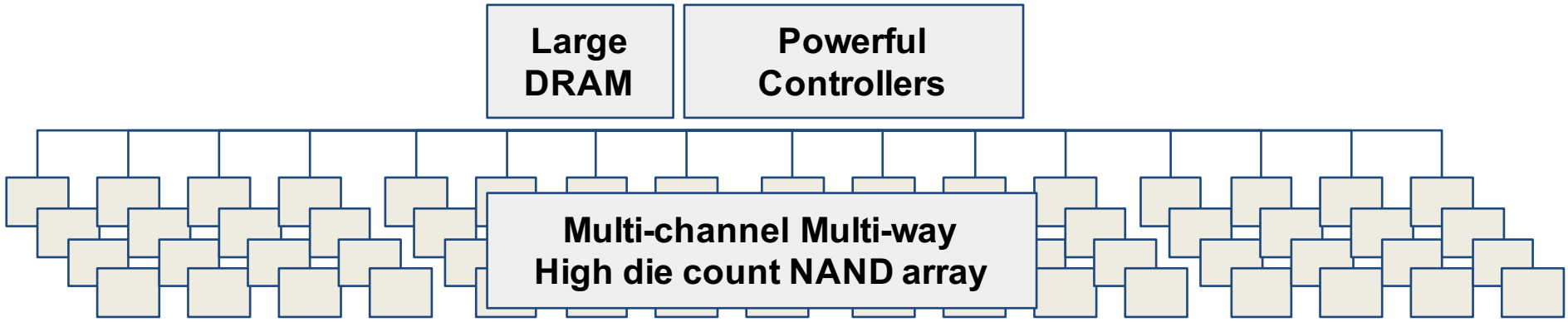


Higher Context Multiplexing Cost "Scheduling Delays"

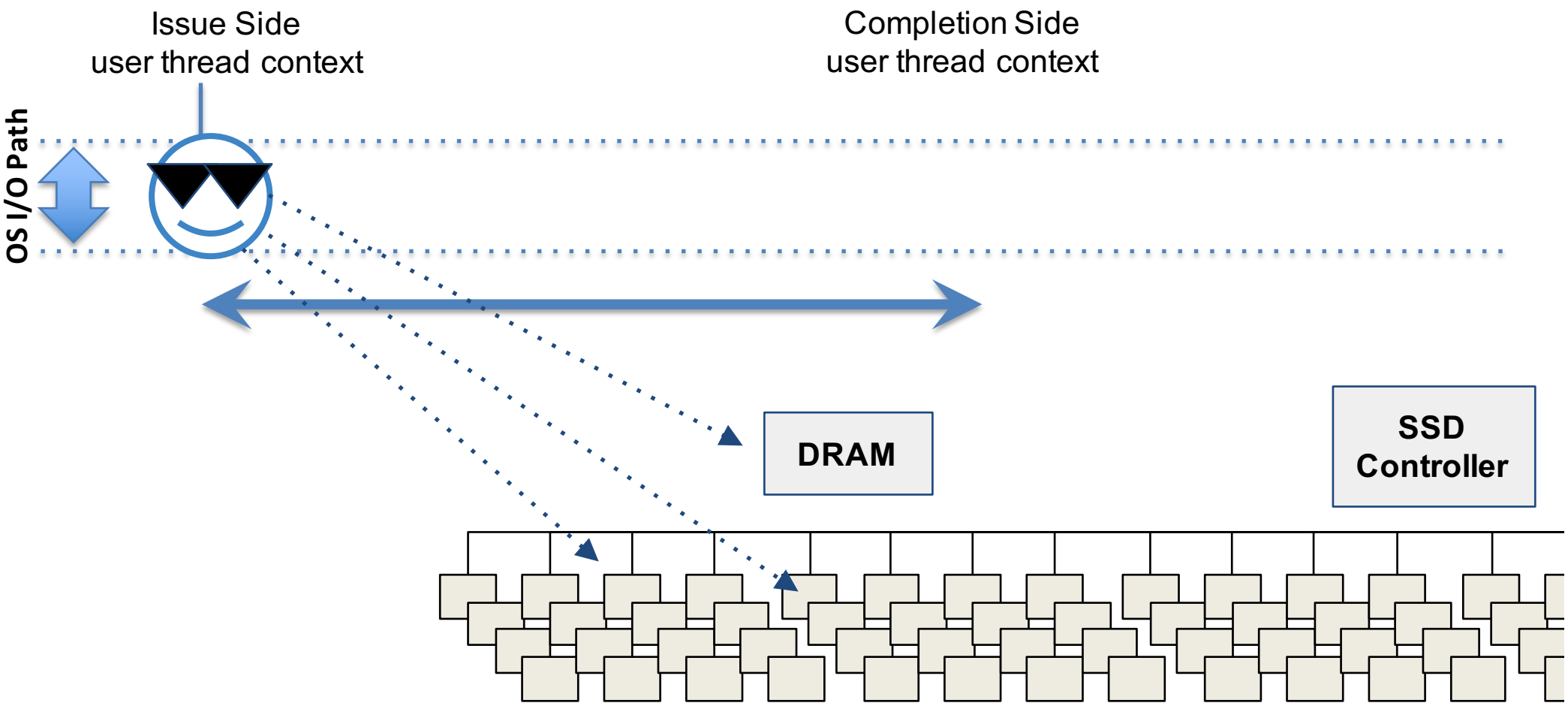


NAND die count >>> CPU core count

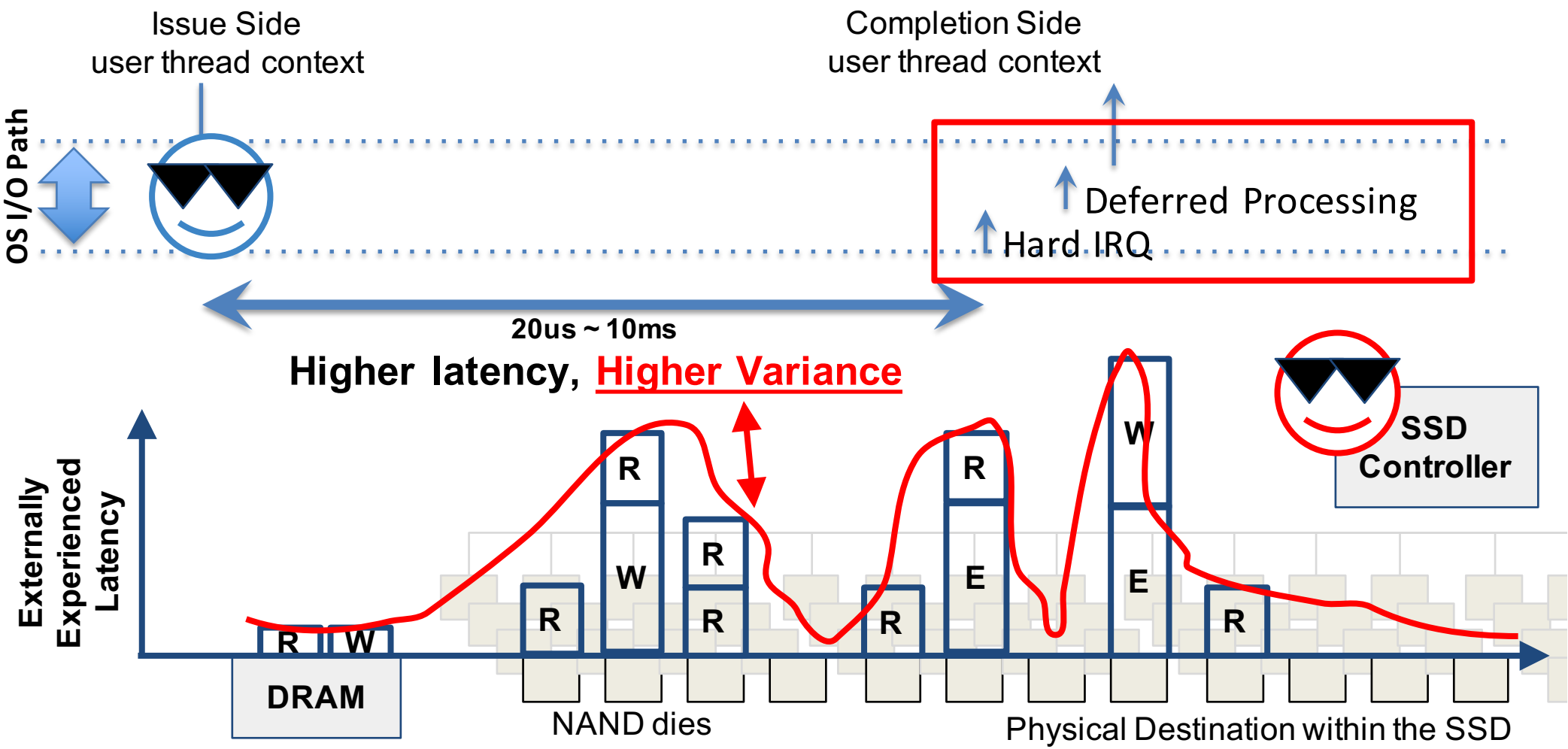
i.e., Four PCI-e 3.0 4lane slots in a chassis, more SSDs Redundancy, more capacity ...



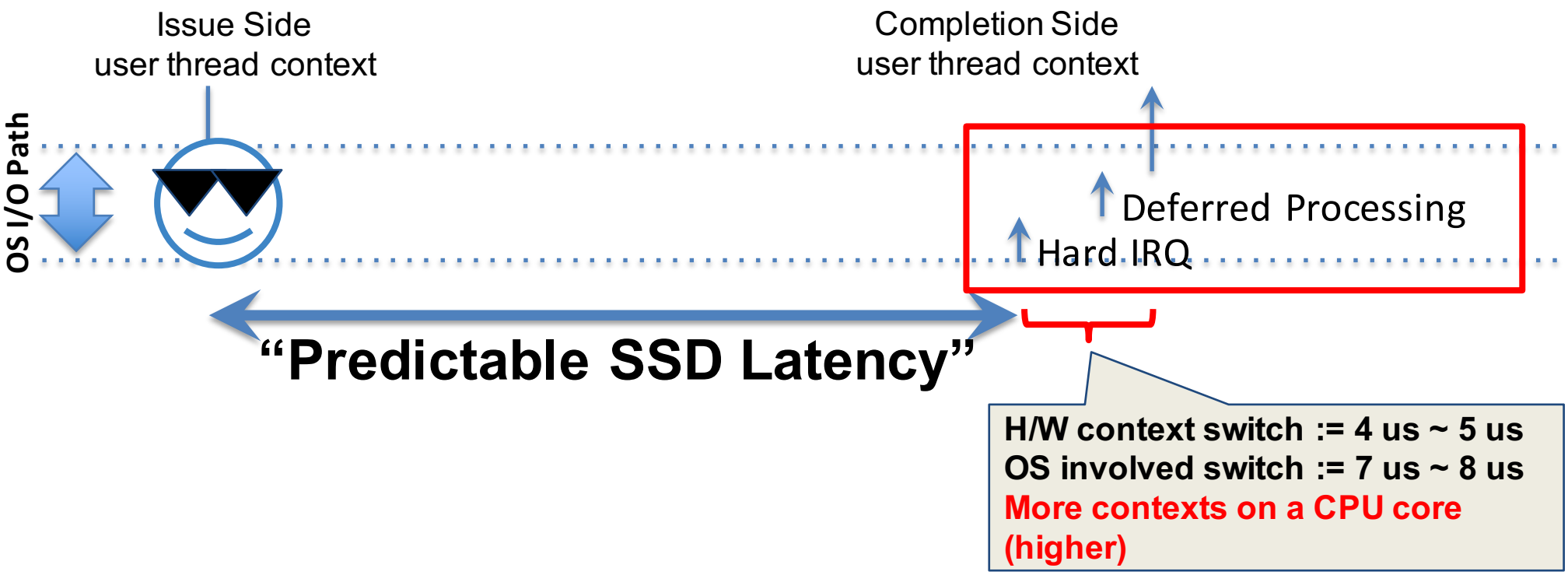
The OS is Blind: Conservative Strategies



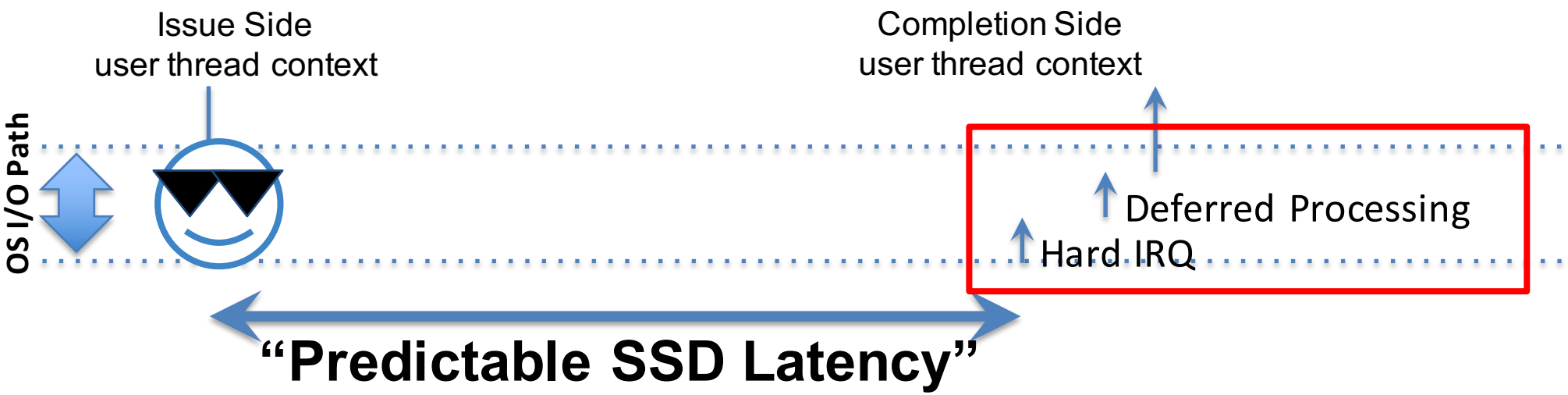
The OS is Blind: Conservative Strategies

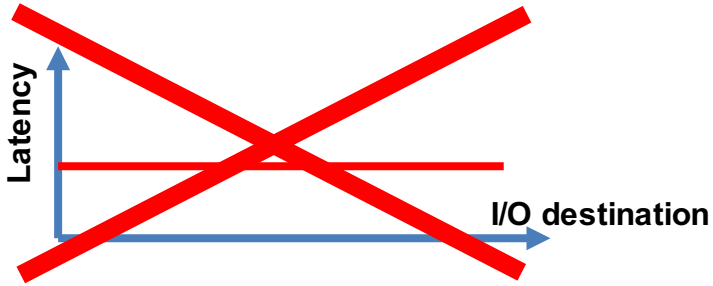


This Work: Unblinding the OS



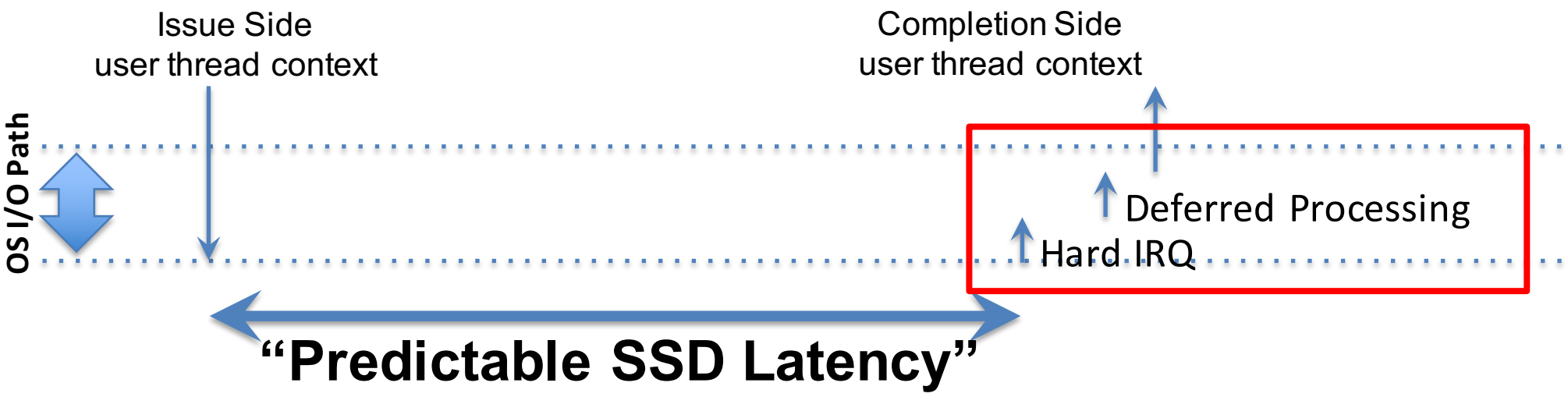
This Work: Unblinding the OS



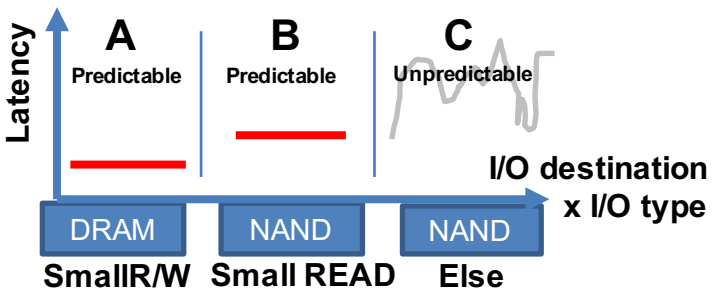


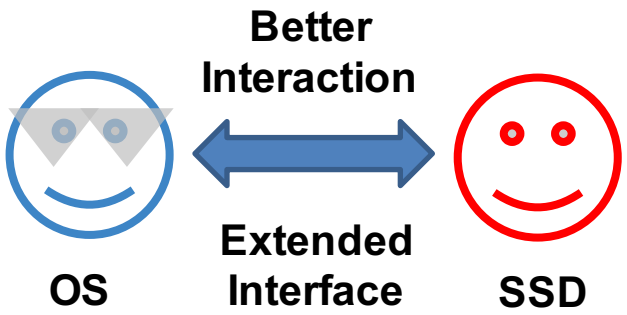
OS SSD

This Work: Unblinding the OS

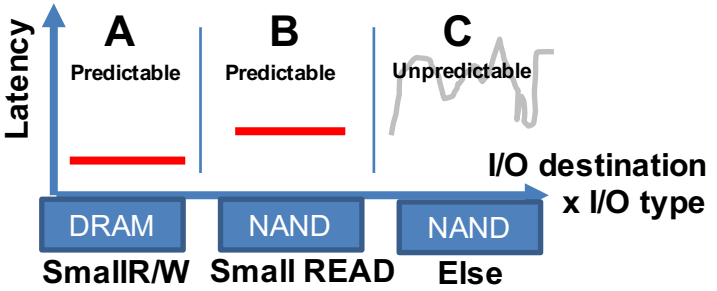


Exposed Knowledge
"A simplified model"





Internal Knowledge

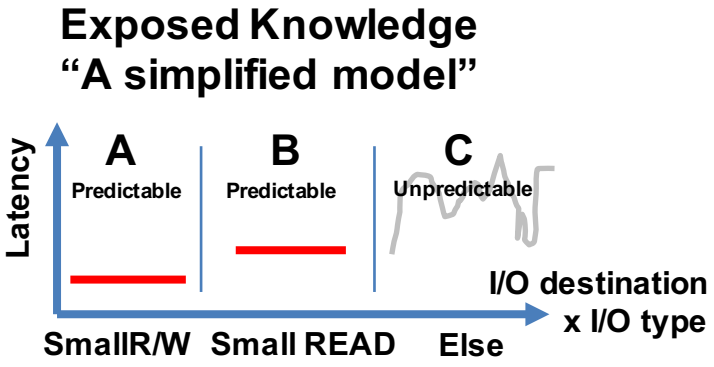


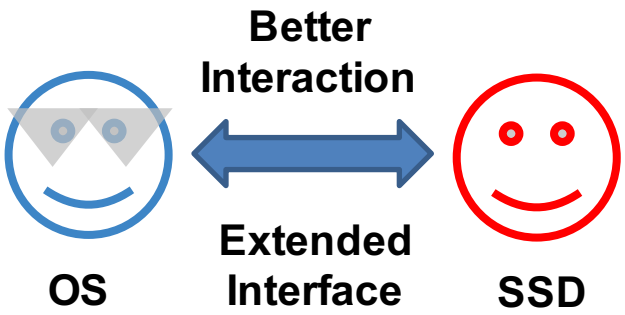
This Work: Unblinding the OS

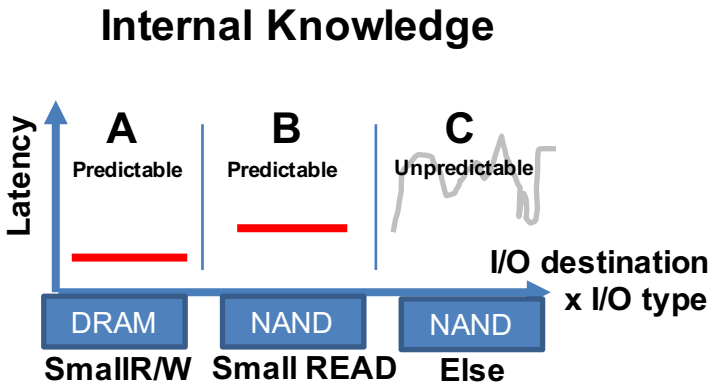
Exploiting SSD Internal Information

Towards a "Predictable SSD"

New Optimization Opportunities





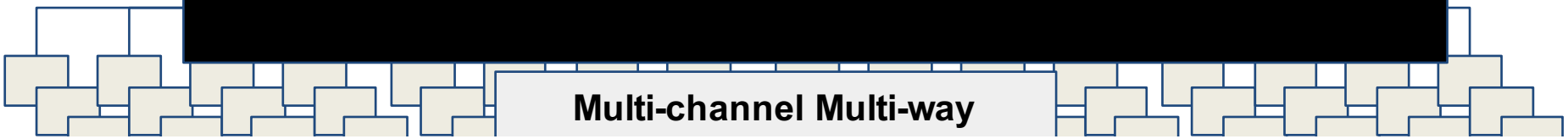


Exploiting SSD Internal Information

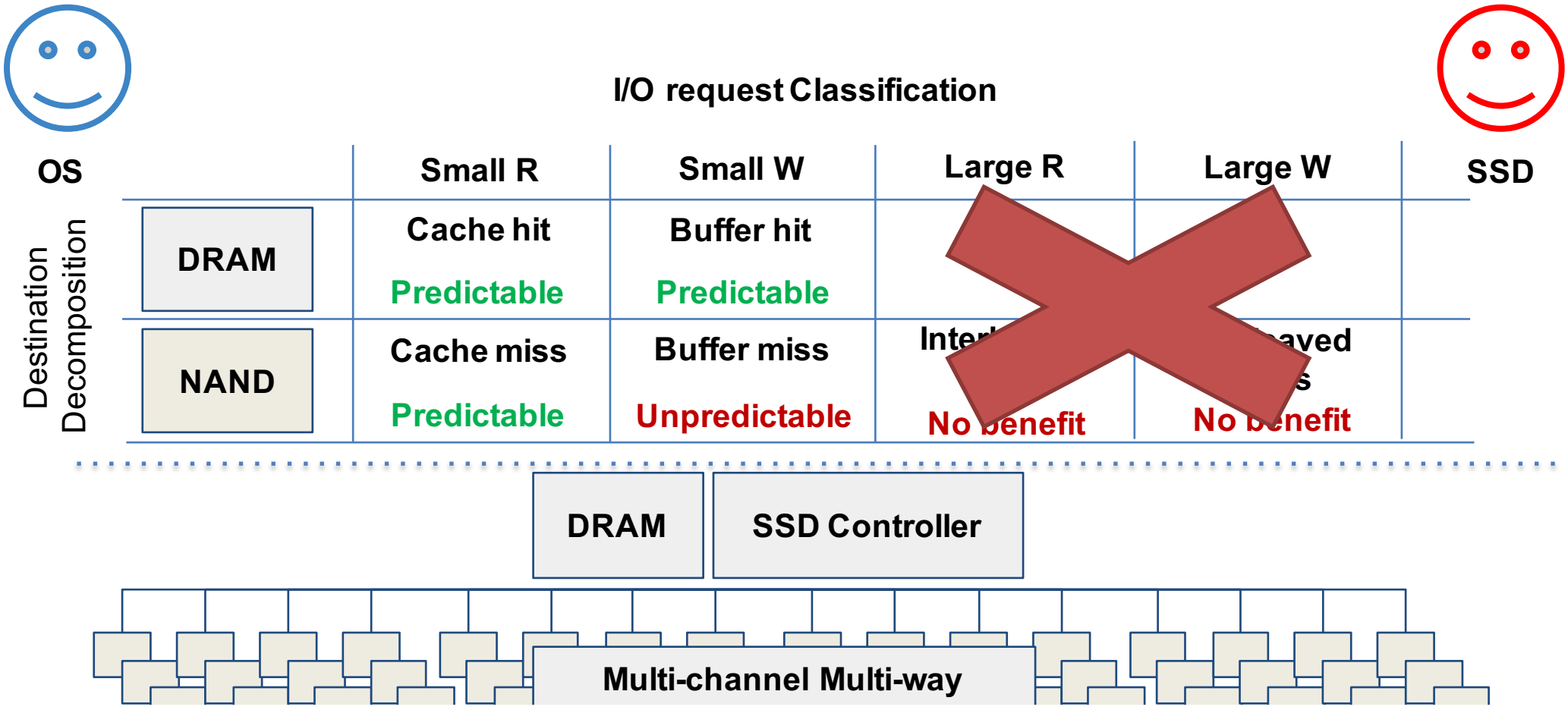
OS

SSD

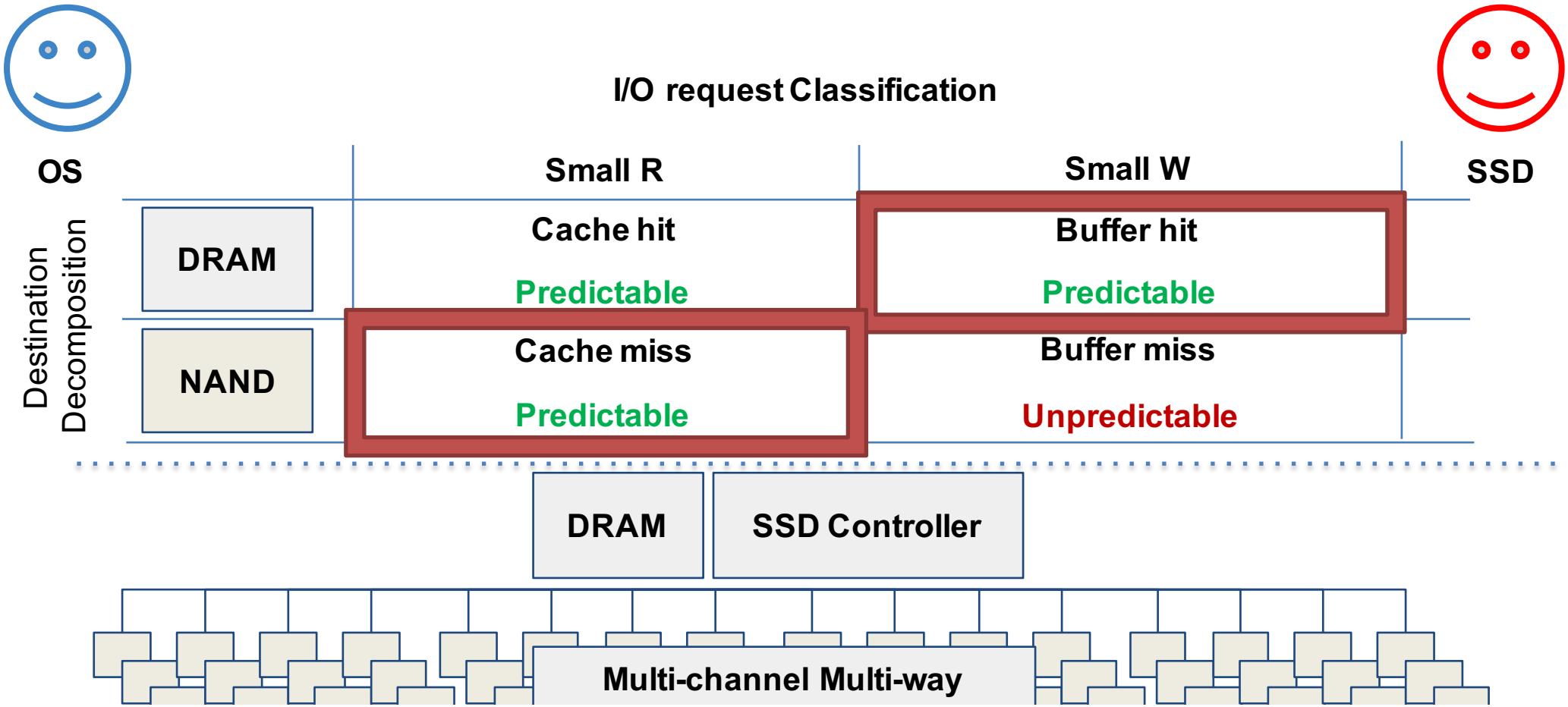
The Unpredictable SSD



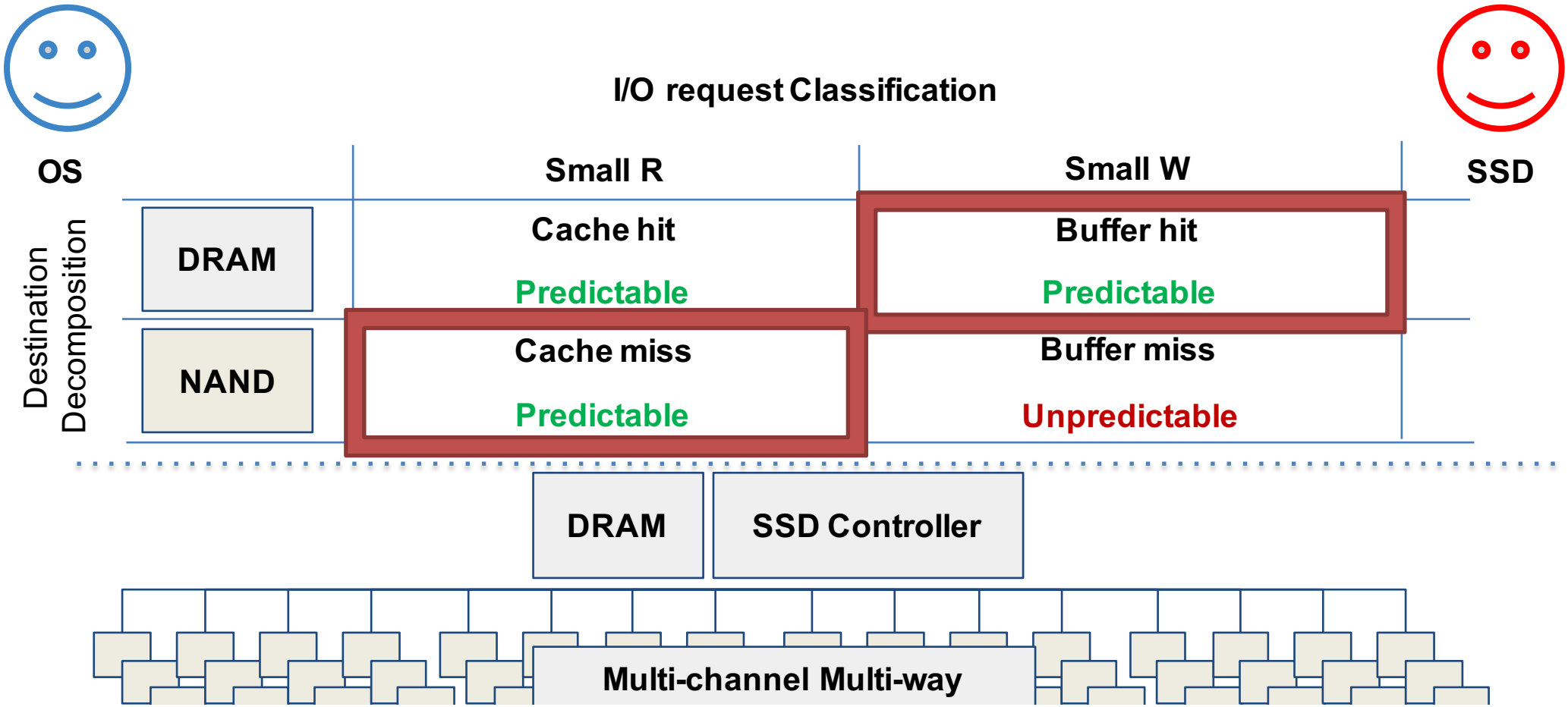
Exploiting SSD Internal Information
“Decomposition & Classification”



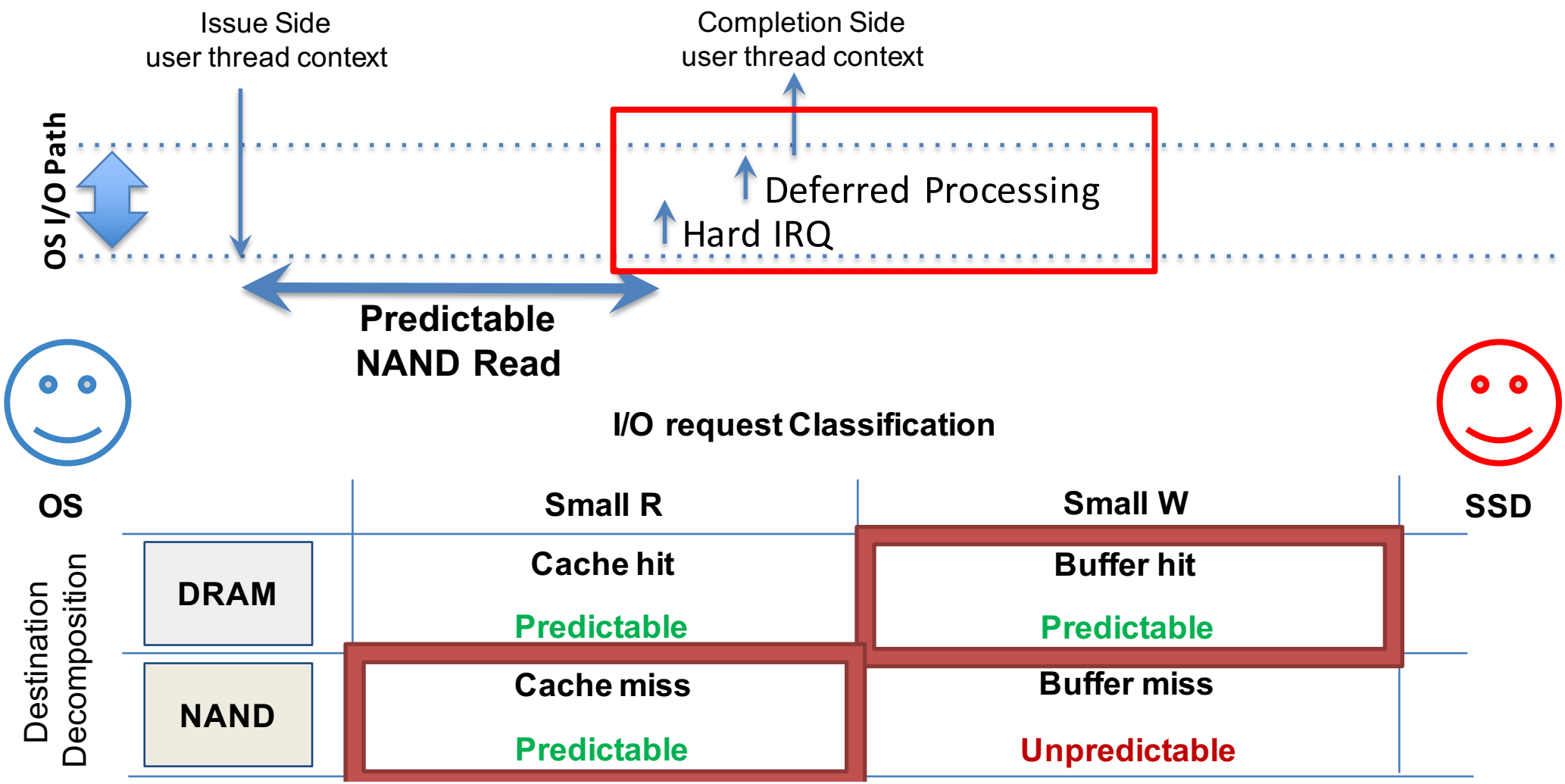
Exploiting SSD Internal Information
“Decomposition & Classification”



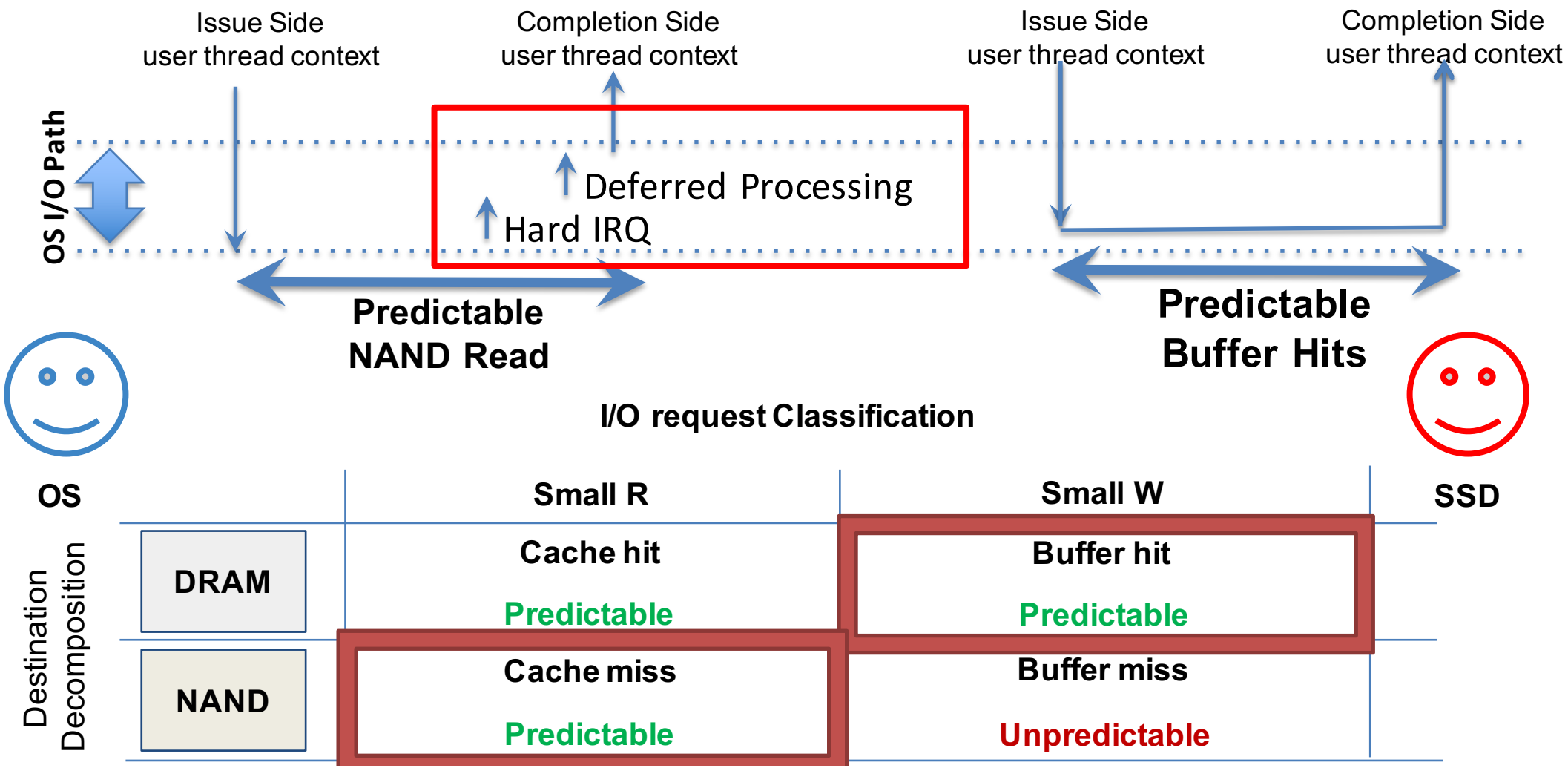
Exploiting SSD Internal Information
“Decomposition & Classification”



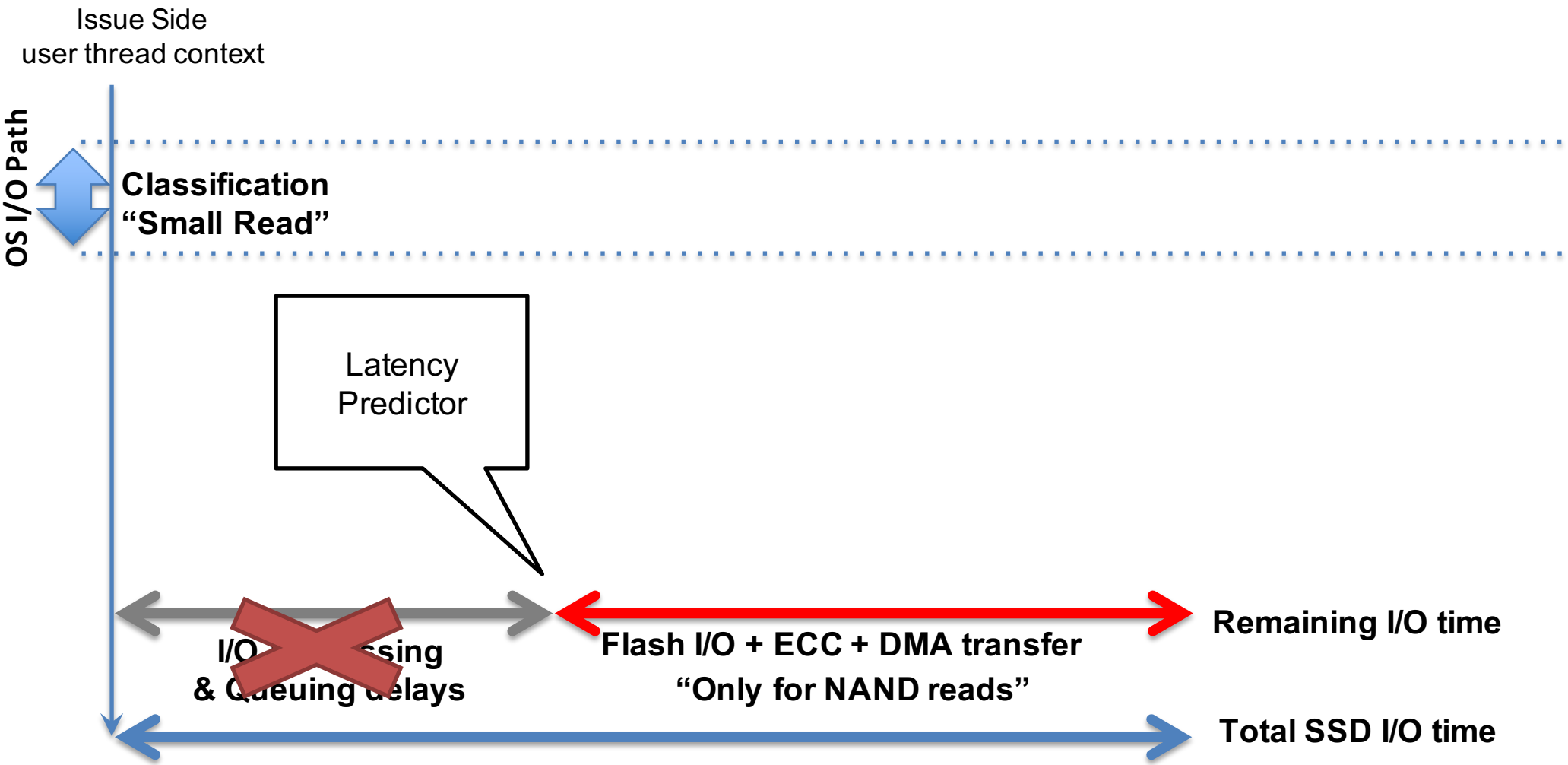
Mitigating the Impact of Scheduling Delays



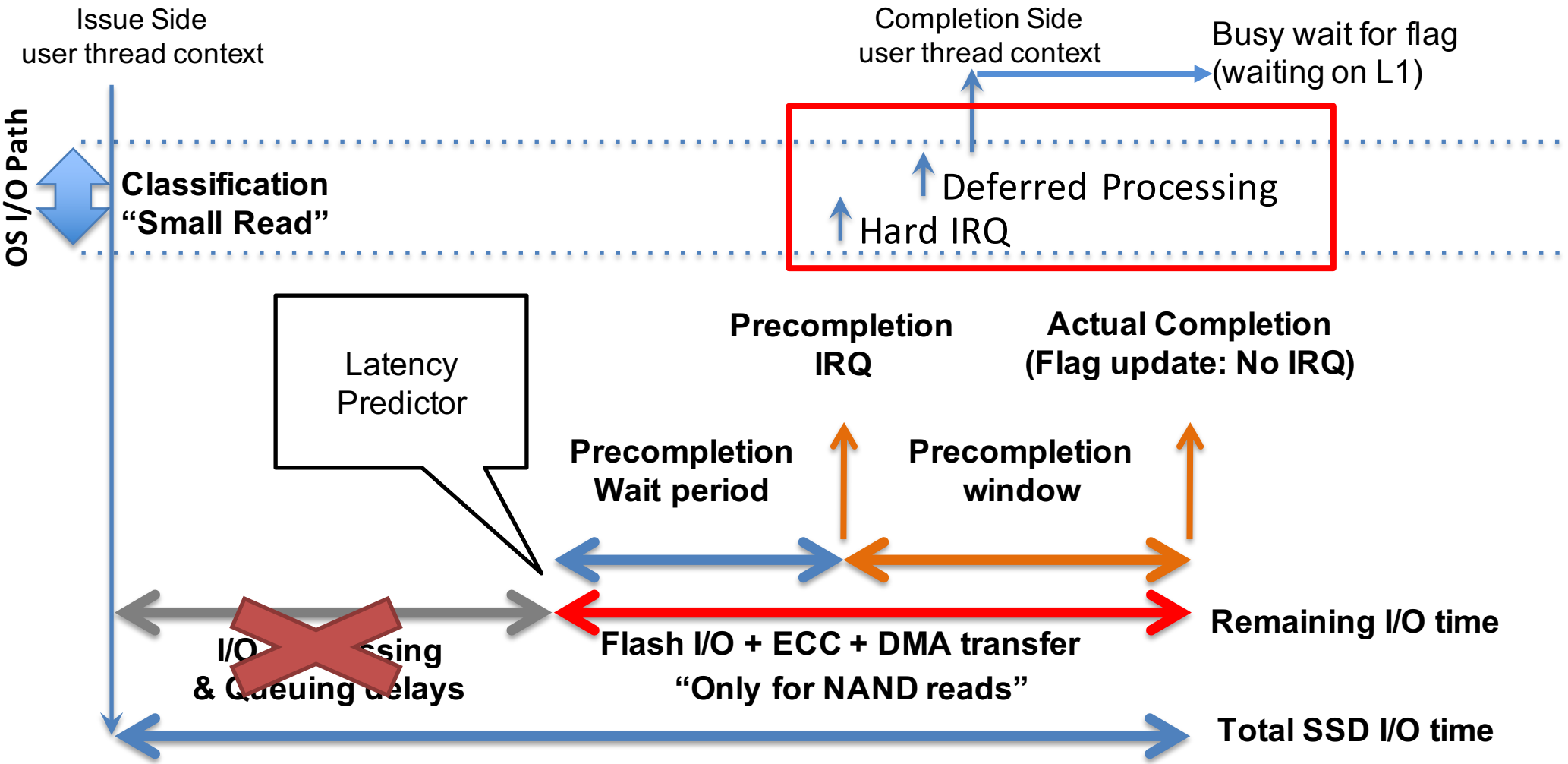
Mitigating the Impact of Scheduling Delays



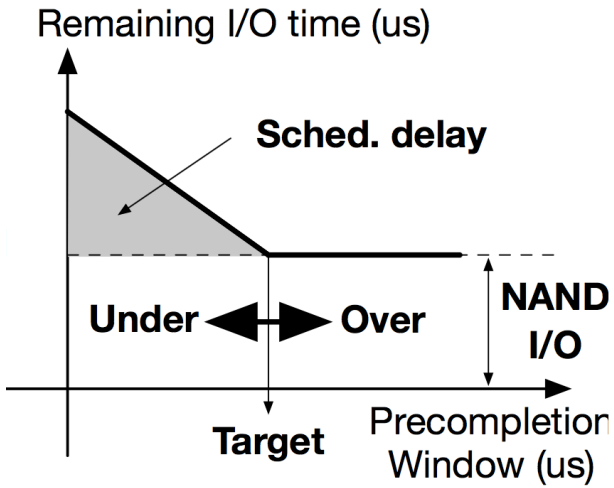
Accurate Latency Prediction: Remaining I/O Time

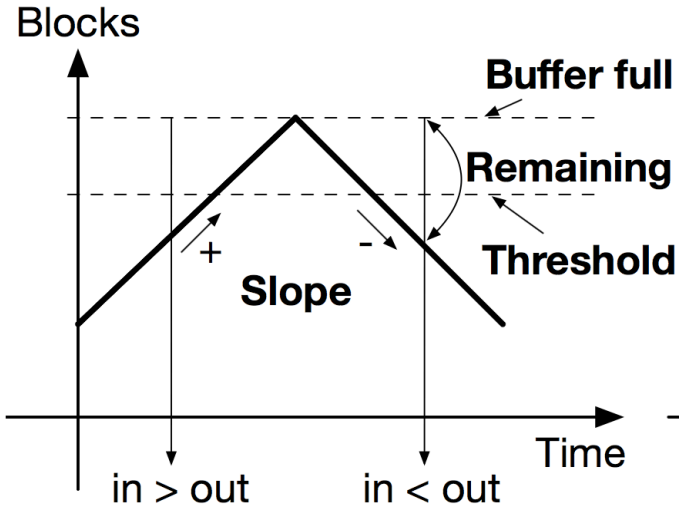


Precompletions: Overlapping I/O & Scheduling Delay



OS & SSD Interaction: Simple Behavioral Models

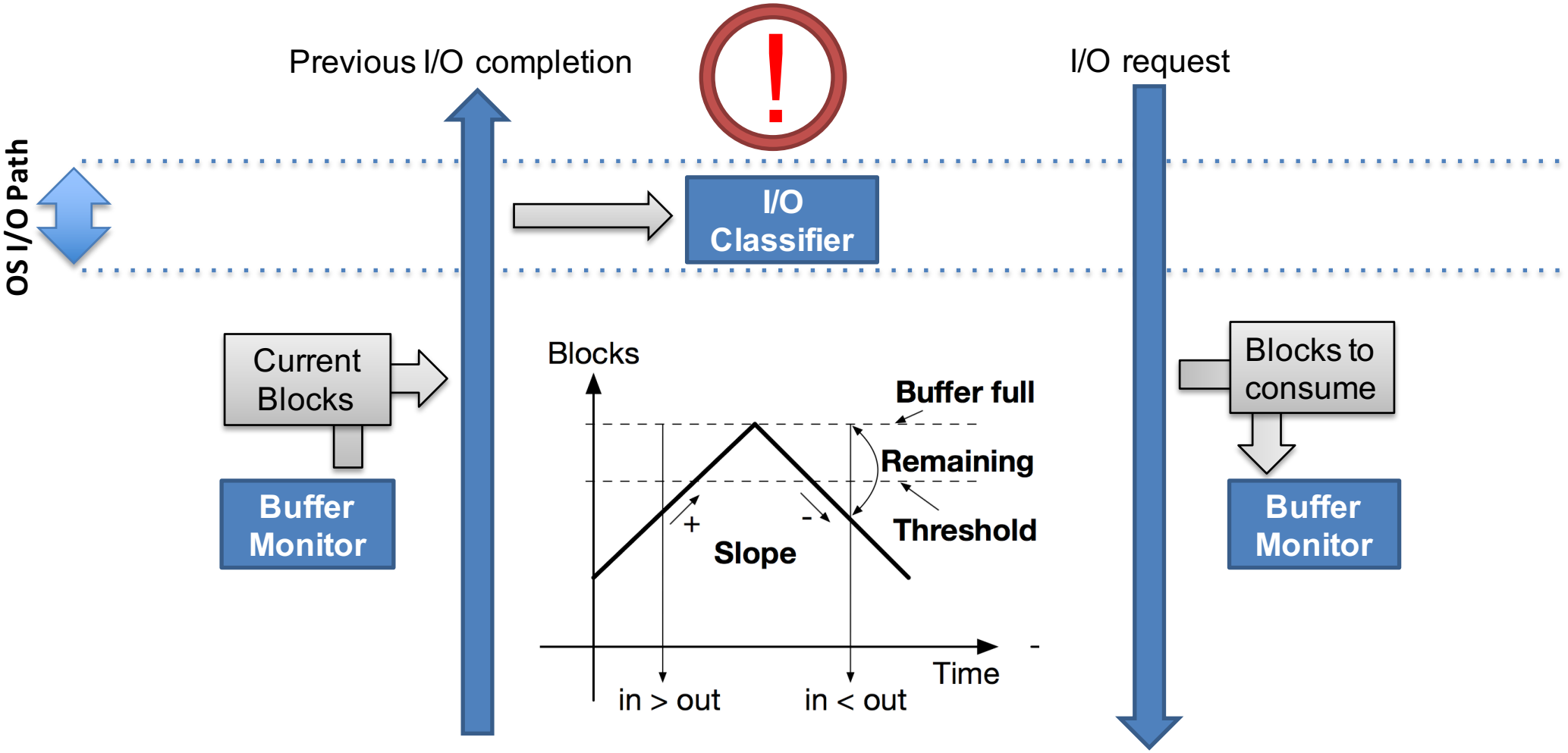




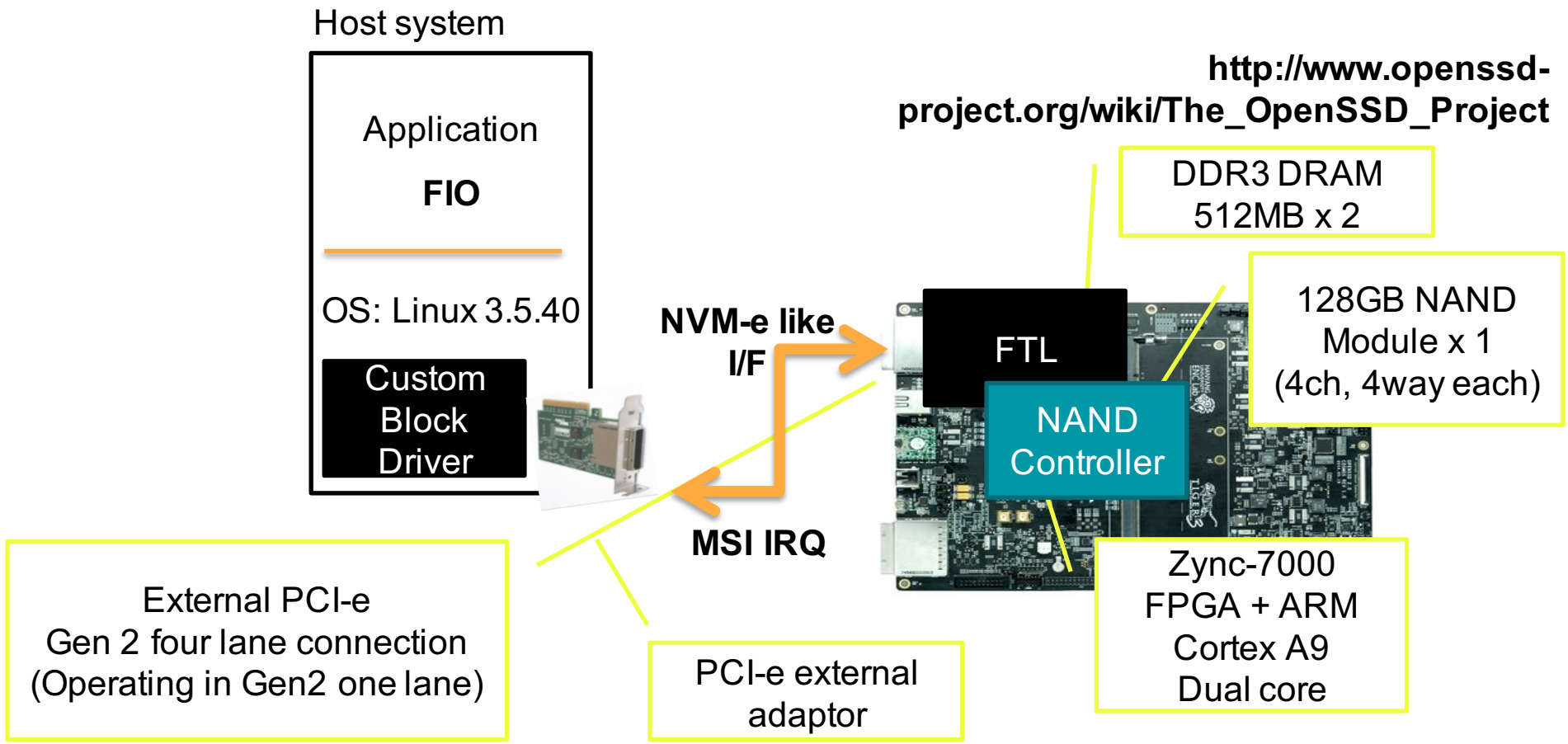
I/O request Classification

	OS	Small R	Small W	SSD
Destination Decomposition	DRAM	Cache hit Predictable	Buffer hit Predictable	
	NAND	Cache miss Predictable	Buffer miss Unpredictable	

In-band Communication Channel
“Piggybacked Information”



Implementation & Evaluation



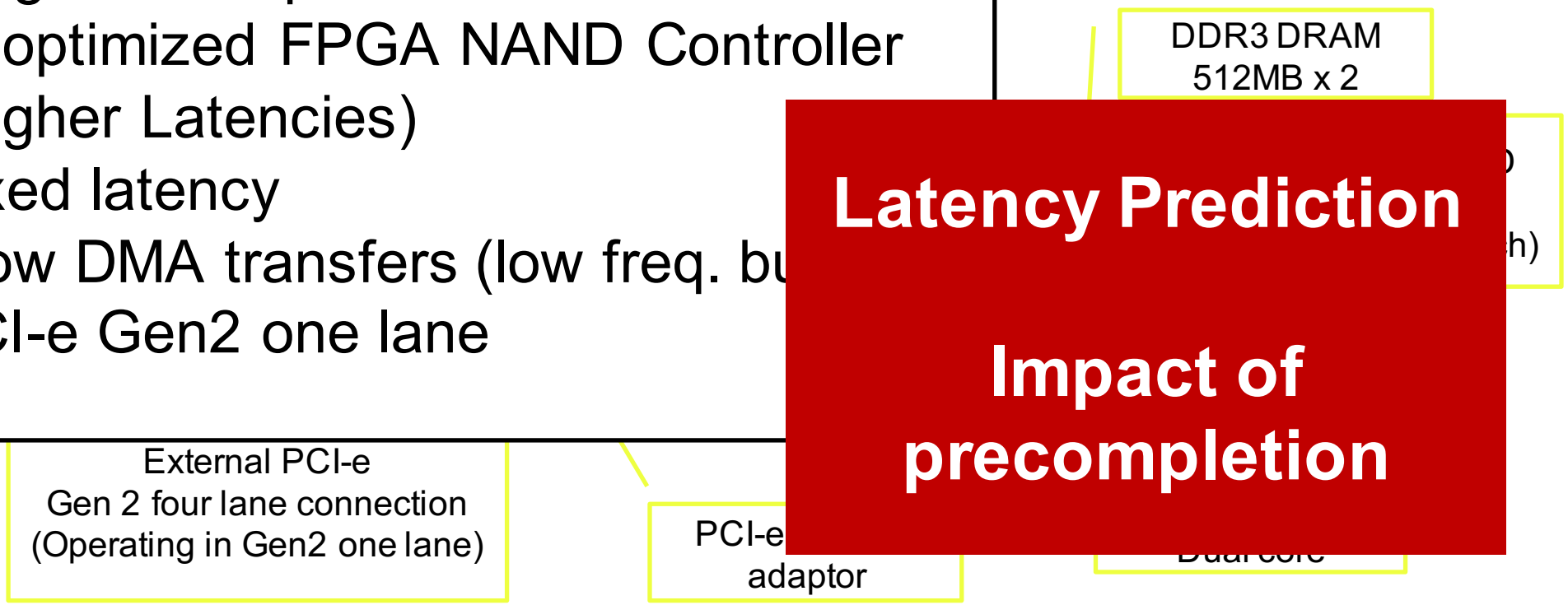
Implementation & Evaluation

Limitation

- Single I/O Depth
- Unoptimized FPGA NAND Controller (Higher Latencies)
- Fixed latency
- Slow DMA transfers (low freq. bu
- PCI-e Gen2 one lane

Latency Prediction

Impact of precompletion



Predicting SSD Latency (Small NAND Read)

Flash: NAND I/O + ECC

Prediction: three value moving average

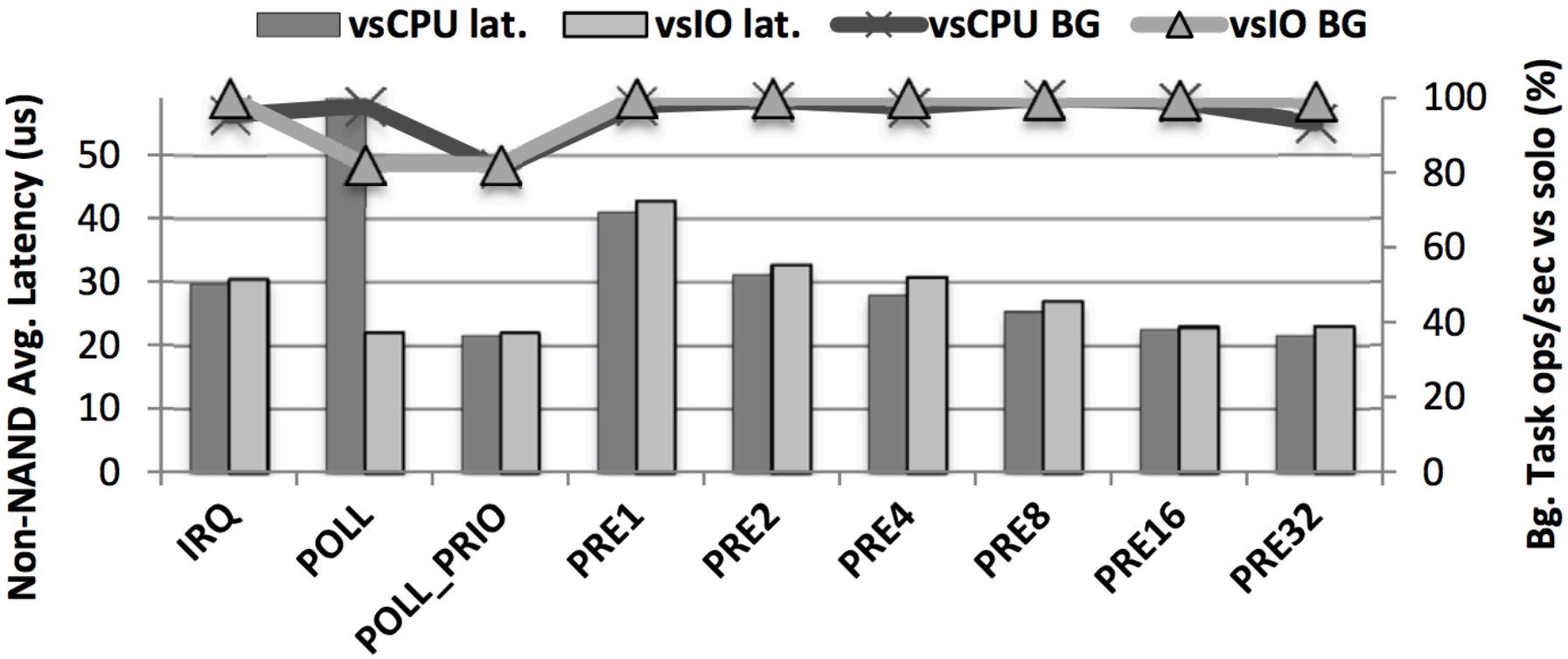
H/W	Measured	Std. dev	Predicted	Error
Flash	352 µs	0.66 µs	352 µs	0.94 µs
DMA	9 µs	0.26 µs	9 µs	0.56 µs

DMA: device to host transfer (4kB)

Low variance
Low Error
“Predictable”

The Impact of Precompletion

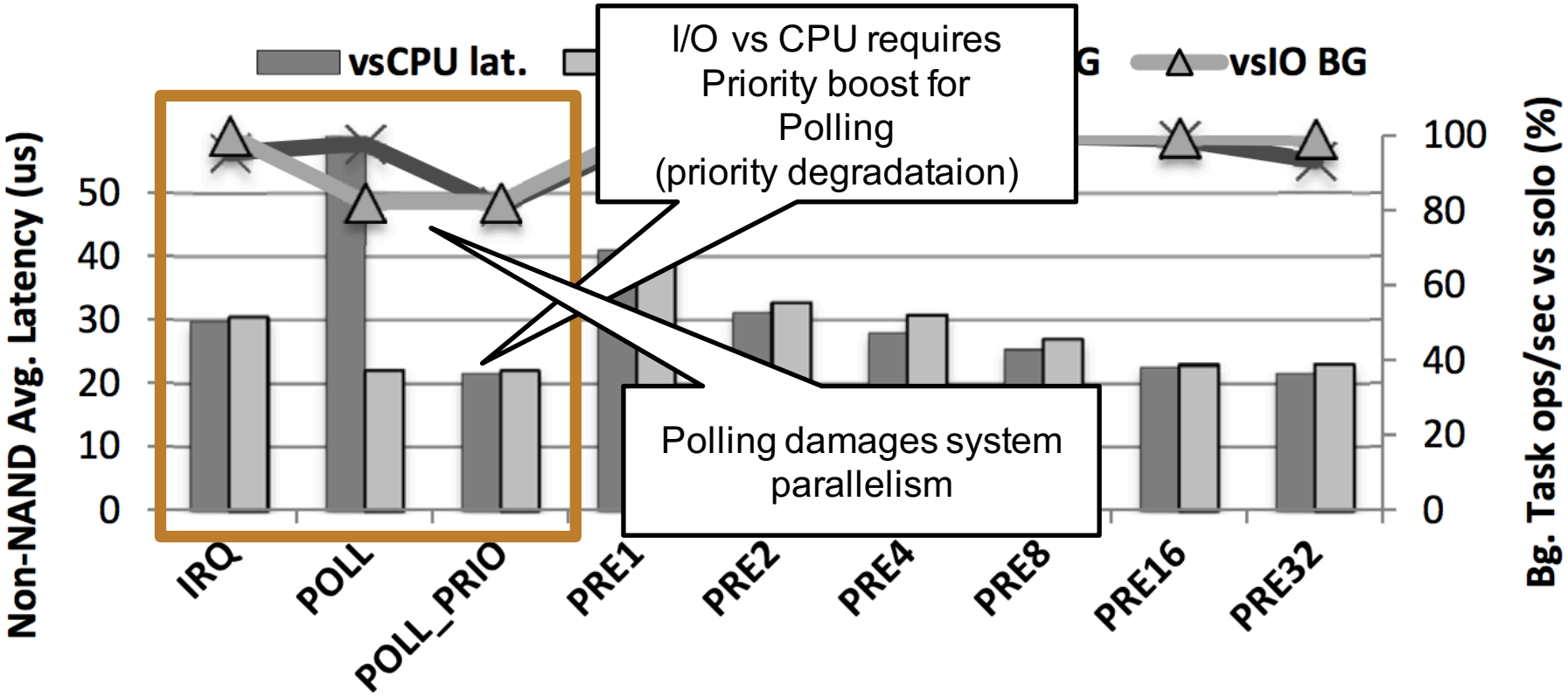
fio I/O thread vs background threads



Non-NAND Avg. latency: Measured AVG latency – 352us (NAND latency)

The Impact of Precompletion

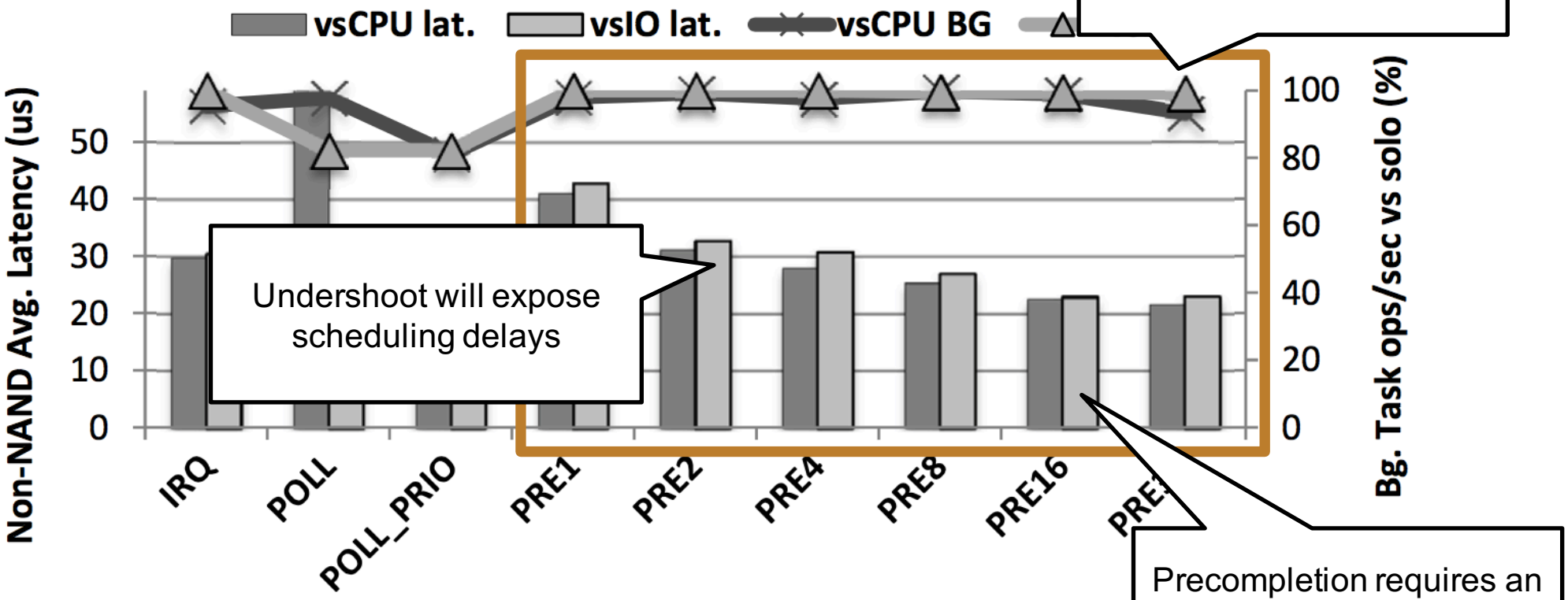
fio I/O thread vs background threads



Non-NAND Avg. latency: Measured AVG latency – 352us (NAND latency)

The Impact of Precompletion

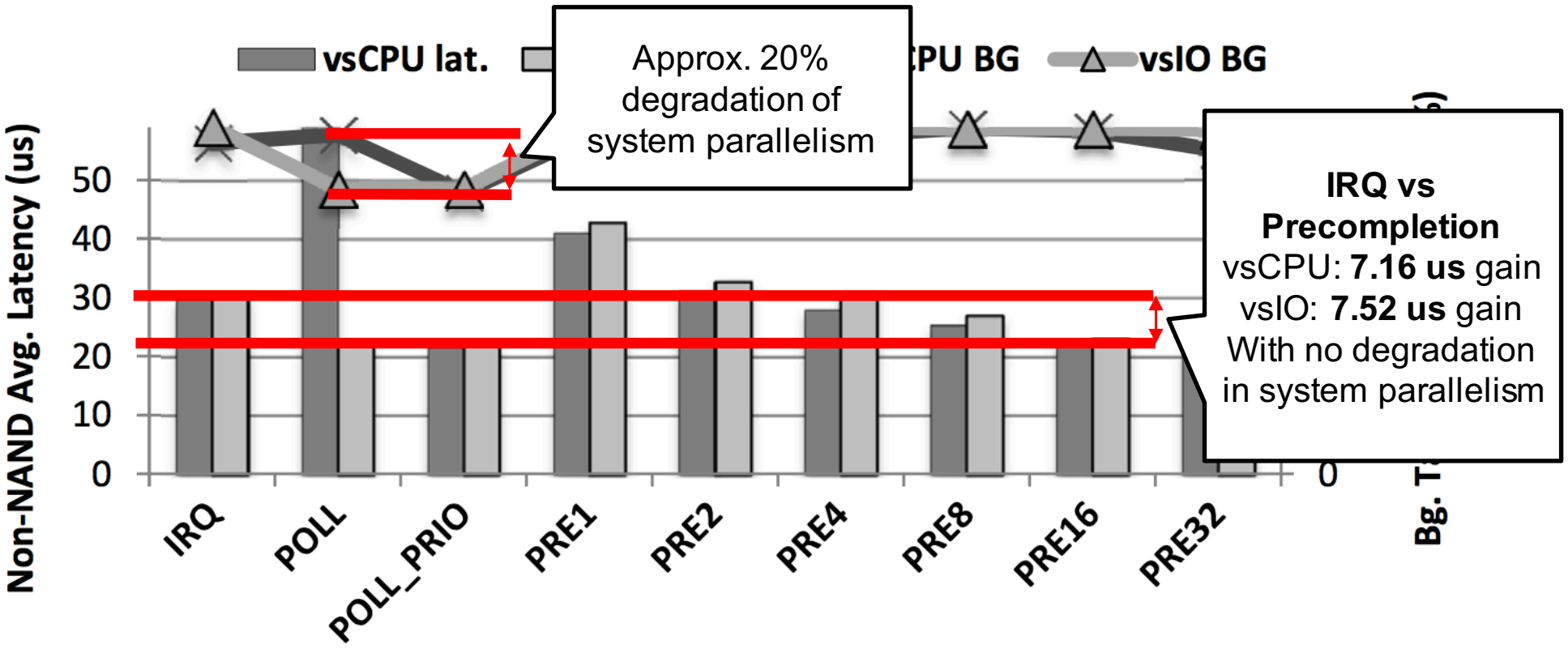
fio I/O thread vs background threads



Non-NAND Avg. latency: Measured AVG latency – 352us (NAND latency)

The Impact of Precompletion

fio I/O thread vs background threads



Non-NAND Avg. latency: Measured AVG latency – 352us (NAND latency)

Summary

- **Unblinding the OS - Cross layer optimization**
  - Achieved a partially predictable SSD ***"decomposition / classification"***
  - Exploit SSD internal information - ***"Remaining I/O time"***
  - Protecting SSD proprietary internals – ***"Abstracted behavioral models"***

- **Mitigating scheduling delays**
  - Exploiting predictability of certain I/O requests
  - Pre-completion - Projection (1 I/O depth vs other threads)

	IRQ	Polling	This work
Latency	**Bad**	**Good**	**Good**
Parallelism	**Good**	**Bad**	**Good**

Future Work

- **Future Implementation & Evaluation**
  - Full blown SSD
  - ***Projection (1 I/O depth – this work)***
    - → Simulation (varying tech latency & etc)
    - → Real implementation

- **Cross layer optimization**
  - More models
  - More use cases
  - More backend technologies rather than flash