

# Understanding Performance in Solid State Disks

Milo Polte, Jiri Simsa, Garth Gibson

## Overview

- High performance of Flash-based Solid State Devices motivates their incorporation into the memory hierarchy
  - But they are expensive
- Disks and Solid State are based on technologies with different performance behaviors
  - May interact poorly with application and operating system software designed for disks
- Incorporating Flash SSDs into the memory hierarchy calls for understanding interactions with overlying software layers
- This work investigates how different software decisions on top of Flash SSDs affects their performance

## Evaluated Solid State Devices

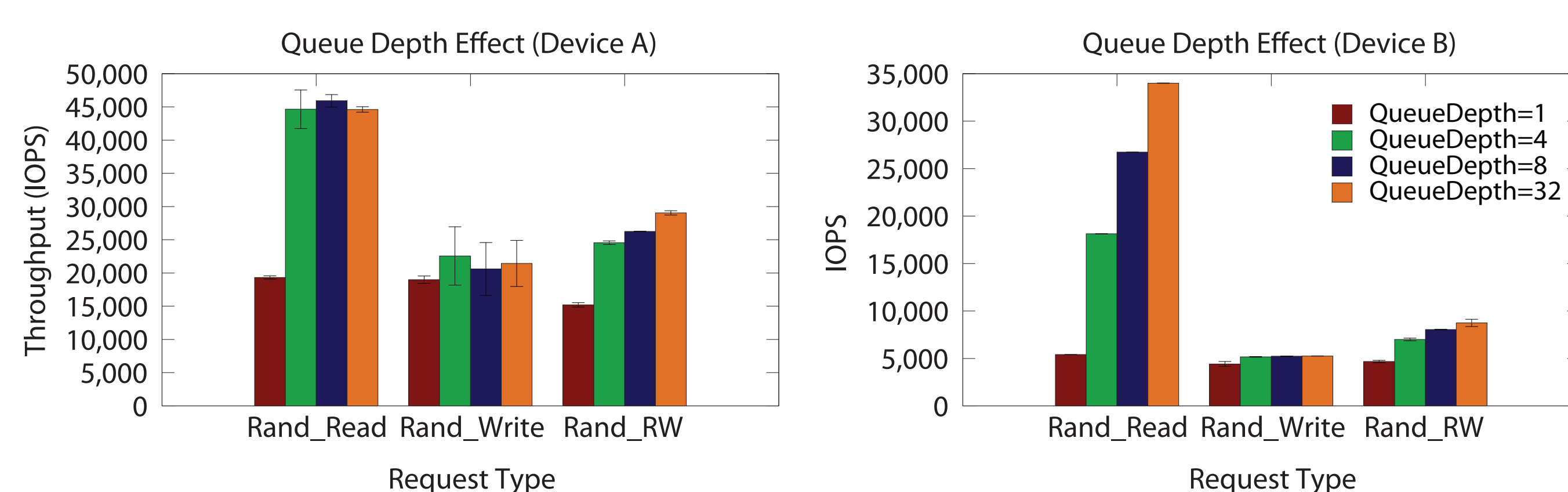
Label	Device A	Device B	VS Disks
Drive Type	PCIe SSD	SATA SSD	
Capacity	80 GB	32 GB	< 1/20 x
Price	\$2400	\$810	~10 x
\$/GB	\$30	\$25.31	~100 x
Access Time	0.05 msec	0.085 msec	~1/100 x
Advertised Bandwidth	700 MB/s read 550 MB/s write	250 MB/s read 170 MB/s write	< 10 x
Advertised Peak IOPS	102k IOPS read 91k IOPS write	35k IOPS read 3.3k IOPS write*	> 100 x

SUMMARY OF SOLID STATE DISKS

\*Write IOPS measured differently  
(See sustained random writes below)

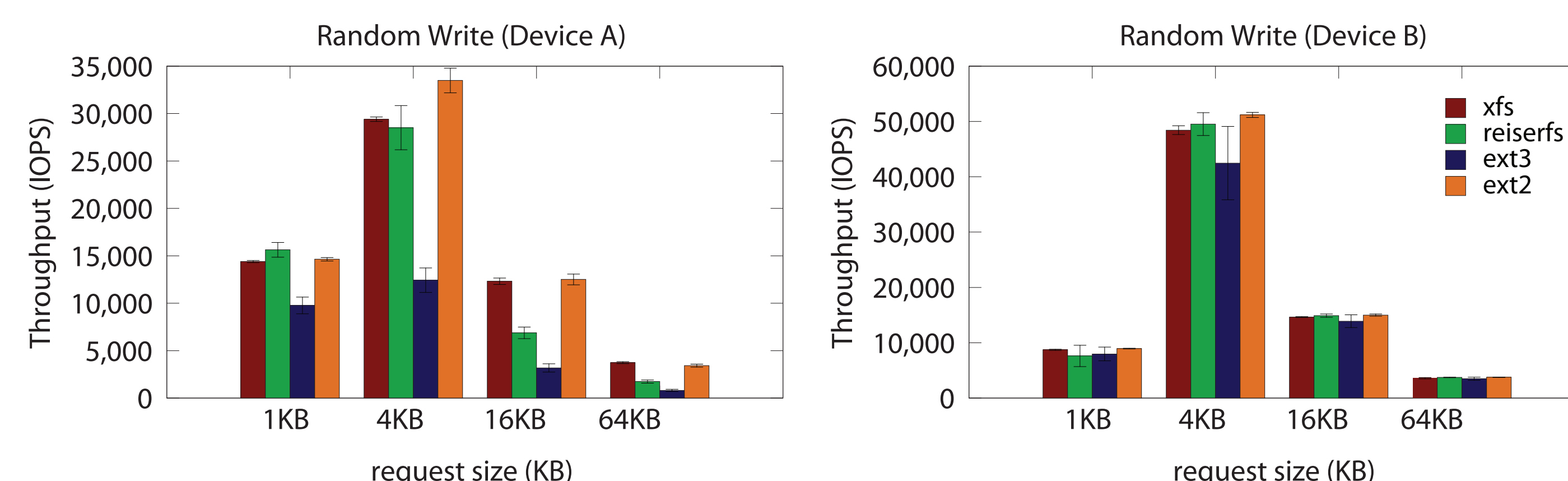
## Effect of Queue Depth on Performance

- Solid State Devices require deep queue depth to achieve advertised performance
- May require higher parallelism from Applications



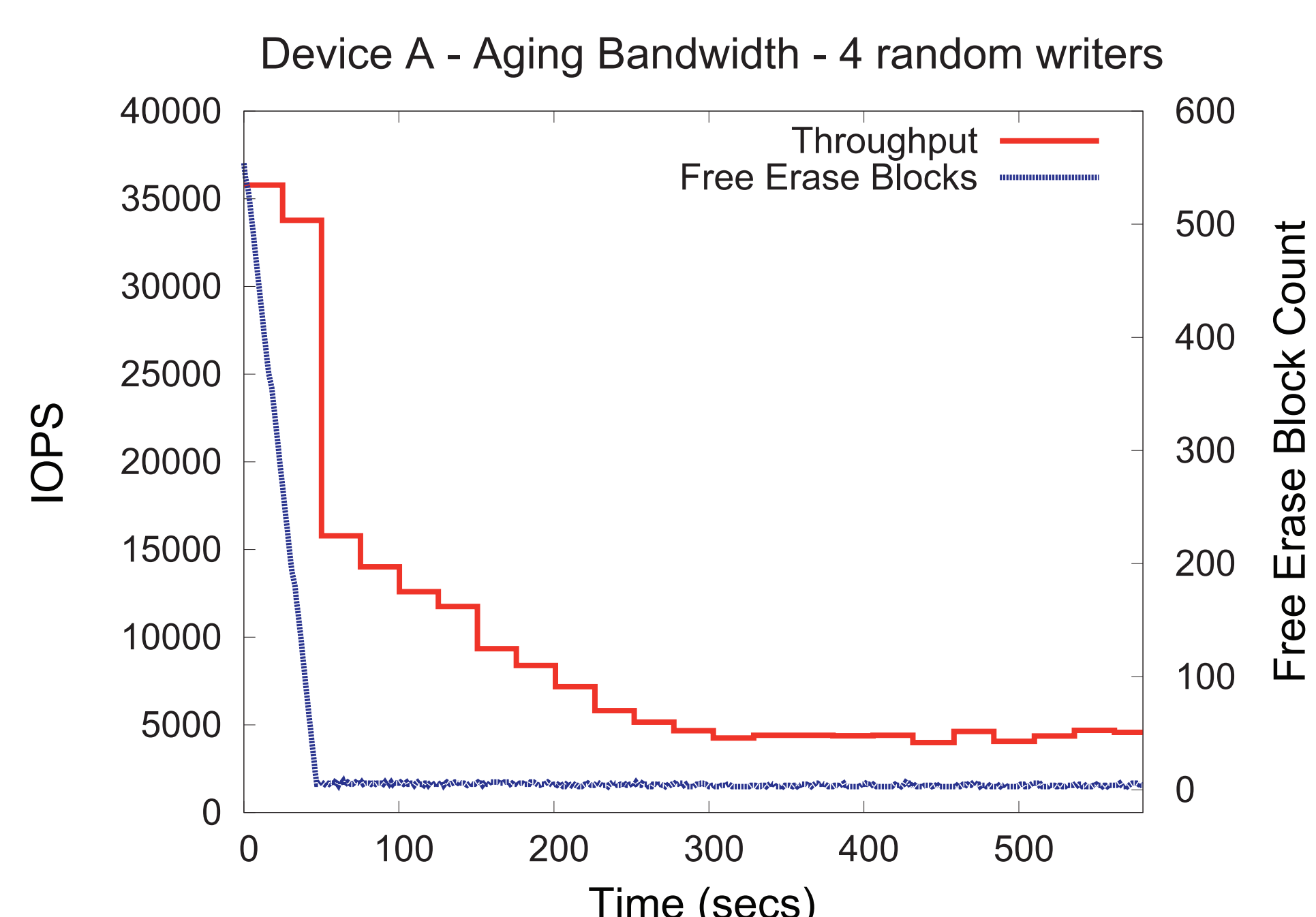
## Filesystems and Request Size

- Some (journaling) filesystems are not tuned for high SSD performance
- SSDs can suffer from poor performance at random writes smaller than the SSD page size (typically 4 KB)



## Effect of Sustained Random Writes

- Sustained random writes drains pool of pre-erased blocks
  - Without pre-erased blocks, writes proceed at erase speed rather than write speed
- Can be ameliorated with more over-provisioning, using less of address space



## Future Work

- Continued testing of newer SSDs
- What is the best use of an 'idealized' SSD?
  - Designing hybrid disk and SSD architectures
- Caching policy experiments with faster SSDs
  - Policies: metadata in SSD, all small random I/O in SSD
  - Apply file system block type information to partition data
    - Annotated ext2/ext3 so far
      - DualIFS
- How to best incorporate SSDs into a parallel filesystem
  - Metadata, file creation
    - Buffering for disk spin down