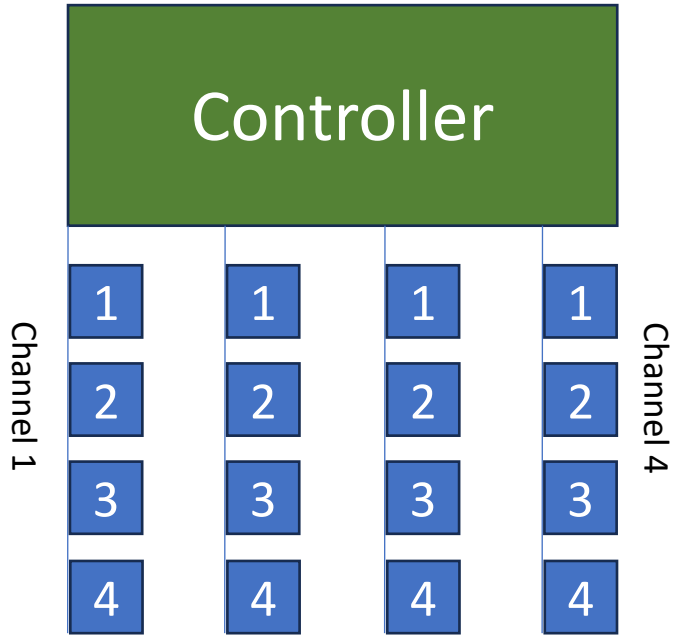
Excessive SSD-Internal Parallelism Considered Harmful

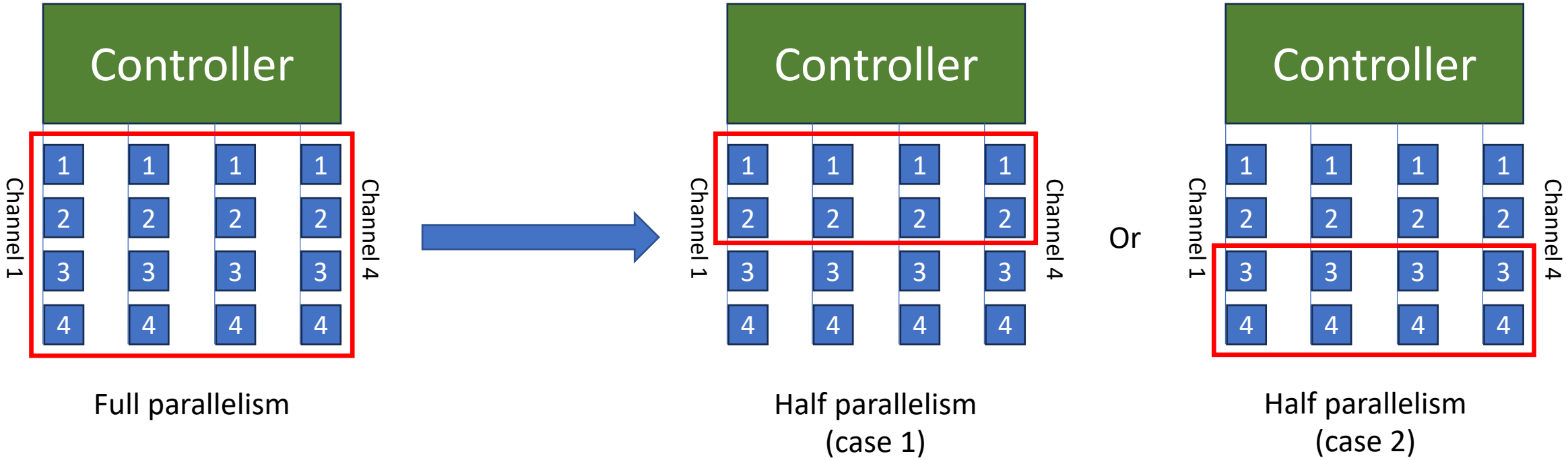
Xiangqun Zhang, Syracuse University
Shuyi Pei, Samsung Semiconductor USA
Jongmoo Choi, Dankook University
Bryan S. Kim, Syracuse University

Parallelism in SSD

- Multiple channels, multiple chips per channel
- Single blocks from all chips form superblocks for the sake of parallelism
  - Controller stripes incoming requests
  - Multiple chips serve the request in parallel
    - Each chip receives a part of the request
  - Superblock is the unit for garbage collection
- Higher level of SSD-internal parallelism provides better SSD performance
  - Or… does it?

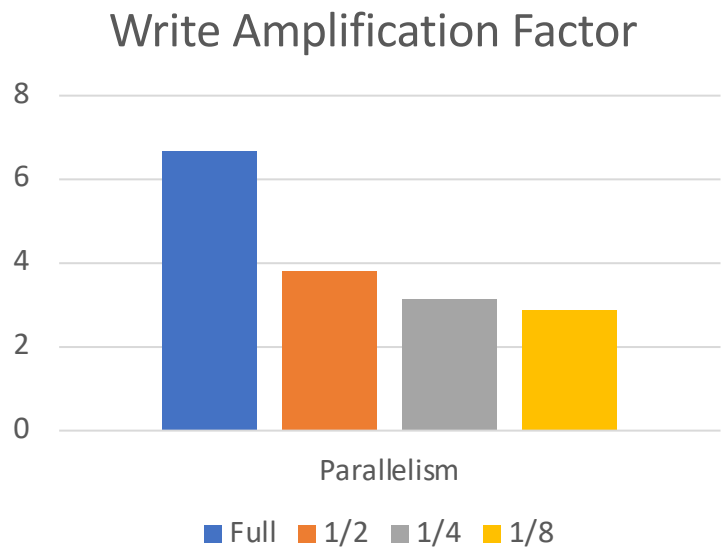


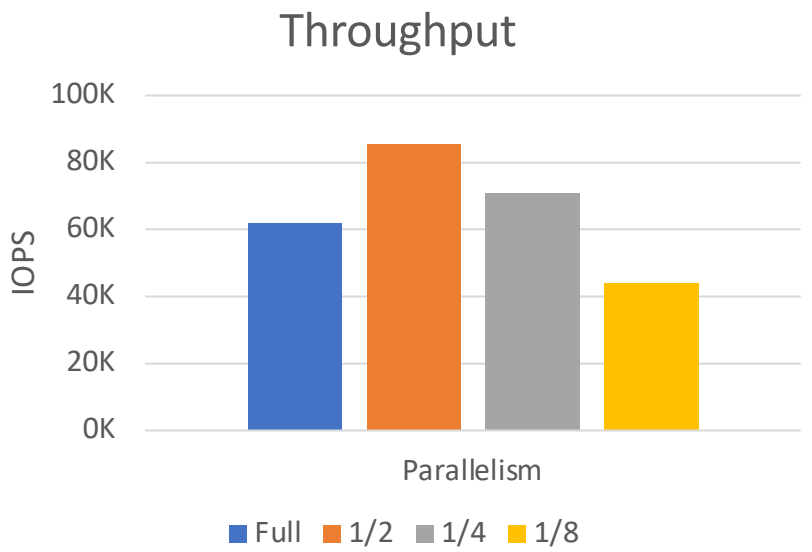
Is Higher Parallelism Always Beneficial?



- Keep the same hardware configuration
- Reduce parallelism by using a portion of all available chips

Is Higher Parallelism Always Beneficial?





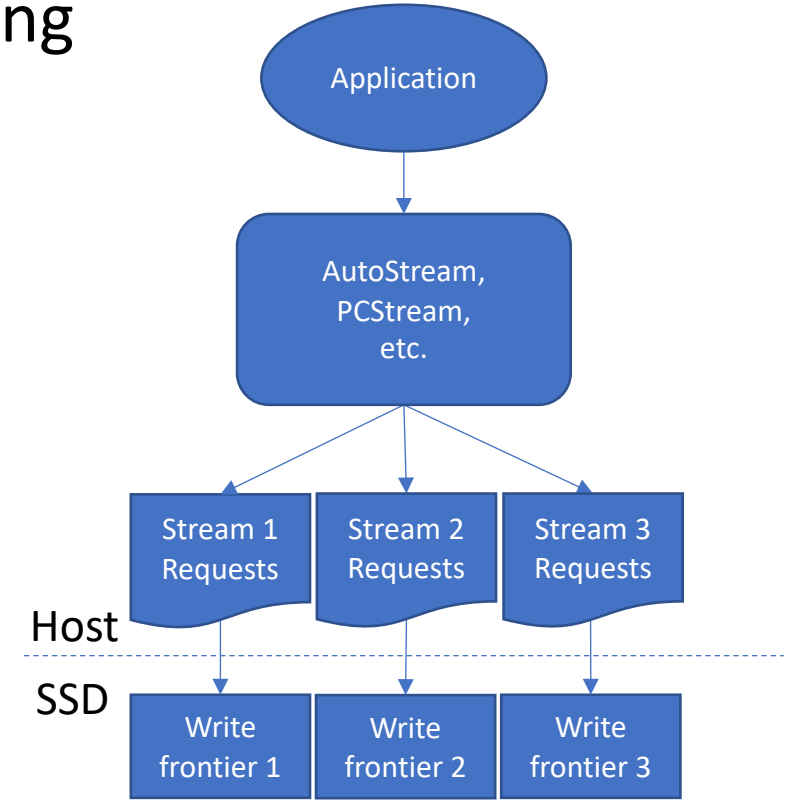
- WAF reduced using lower parallelism
- Throughput improved when reducing parallelism to some extent

Higher parallelism is not always beneficial!

(8 channels, 8 chips/channel in eval)

Other Ways to Improve SSD Efficiency

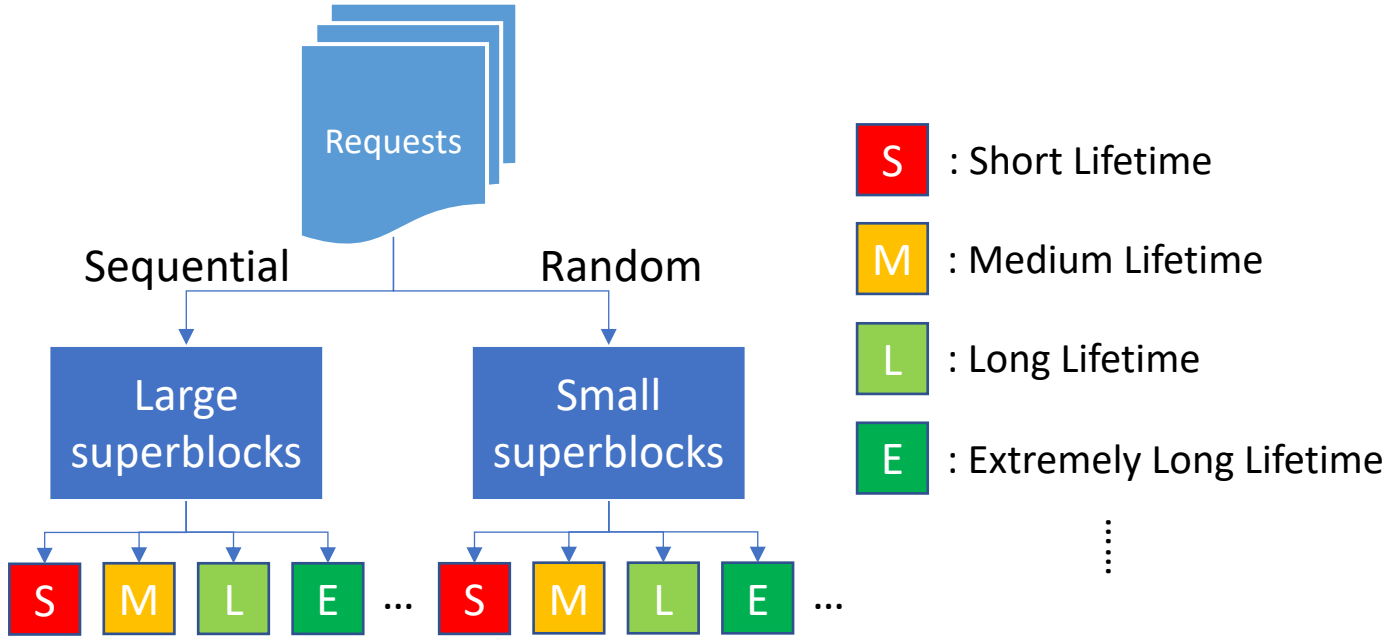
- Reducing WAF and improving throughput by passing hints
  - Multi-stream*
  - AutoStream※, PCStream†, etc.
- Prior works are usually implemented on the host
  - Host side has more I/O information
  - SSD resource is limited
  - Multi-stream *was* supported by Linux



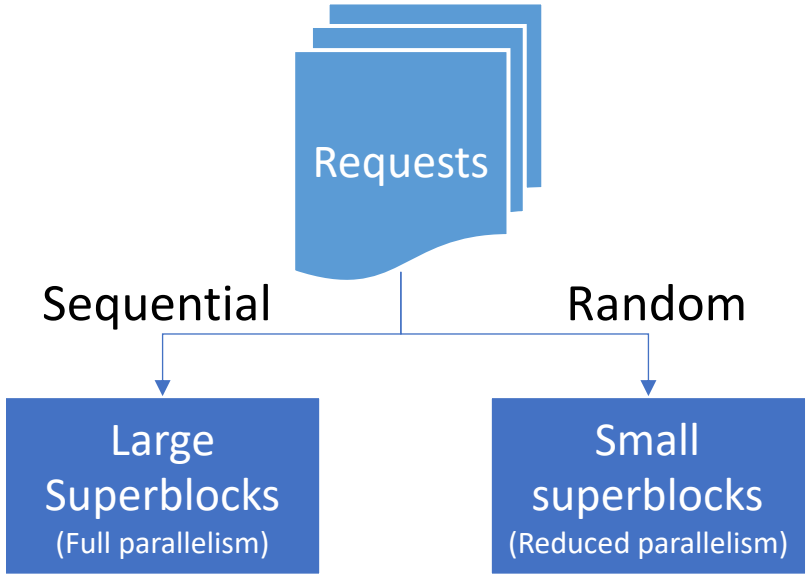
*: HotStorage 14
※: SYSTOR 17
†: HotStorage 18, FAST 19

Parallelism- and Lifetime-aware Allocation (PLAN)

- Reduce write amplification by reducing superblock size & parallelism
- Place data based on data lifetime



Flexible Superblocks

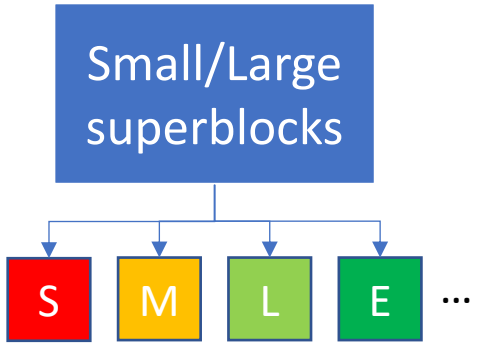


- Sequential requests are more sensitive to SSD parallelism
- Random requests are not as sensitive to SSD parallelism

- Ensures better GC efficiency for small, random requests while keeping throughput for large, sequential requests

Lifetime Predictor

- PLAN keeps data lifetime information
  - Logical address space is split into chunks
- PLAN uses a decay factor to decide the weight of new/old lifetime information
- Choose target superblock based on predicted lifetime

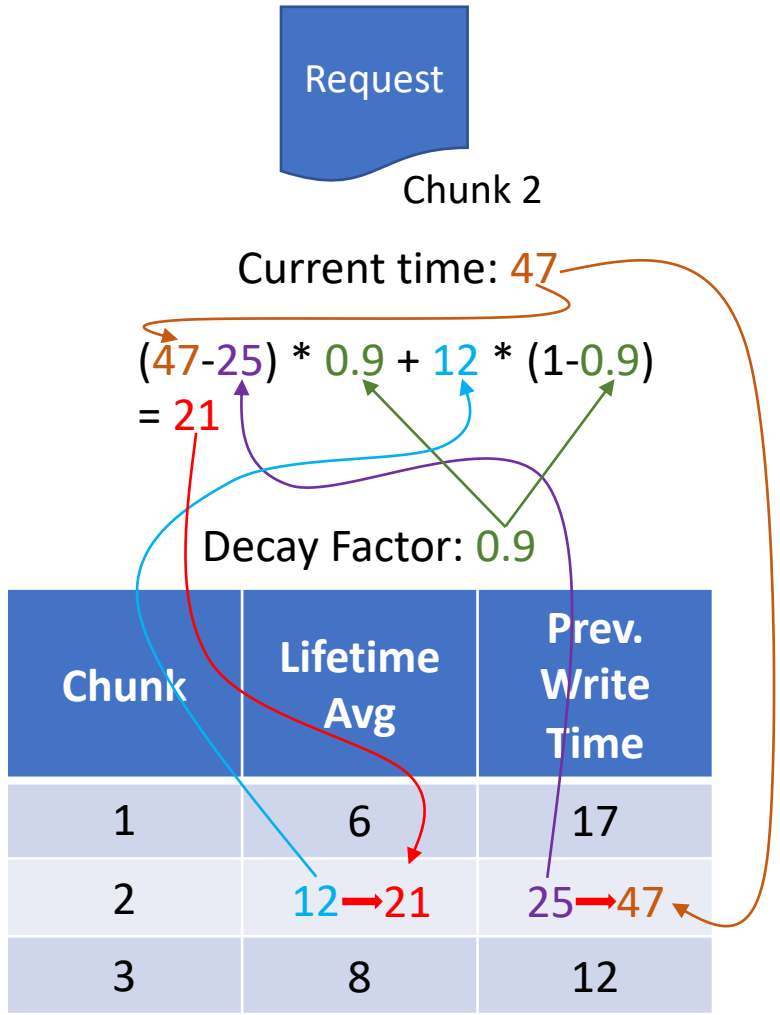


S : Short Lifetime

M : Medium Lifetime

L : Long Lifetime

E : Extremely Long Lifetime



Chunk	Lifetime Avg	Prev. Write Time
1	6	17
2	12→21	25→47
3	8	12

Evaluation – Setup

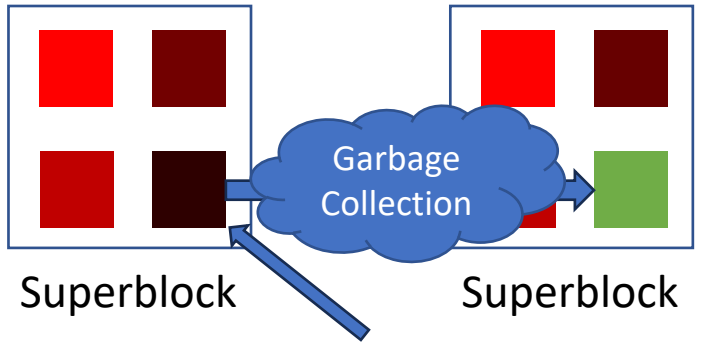
- SSD: 256GiB, FEMU-emulated
  - 8 channels, 8 chips/channel, 1024 blocks/chip, 256 pages/block, 16KiB page

- Benchmarks
  - FIO
  - TPC-C from BenchBase
  - YCSB workload A
  - Fileserver from Filebench
  - GCC Linux kernel compilation

FEMU Settings			
Channels	8	Page size	16 KiB
Chips per channel	8	Physical capacity	256 GiB
Planes per chip	1	Logical capacity	240 GiB
Blocks per plane	1024	Over-provisioning	0.0625
Pages per block	256	Garbage collection	Greedy

Evaluation – Setup

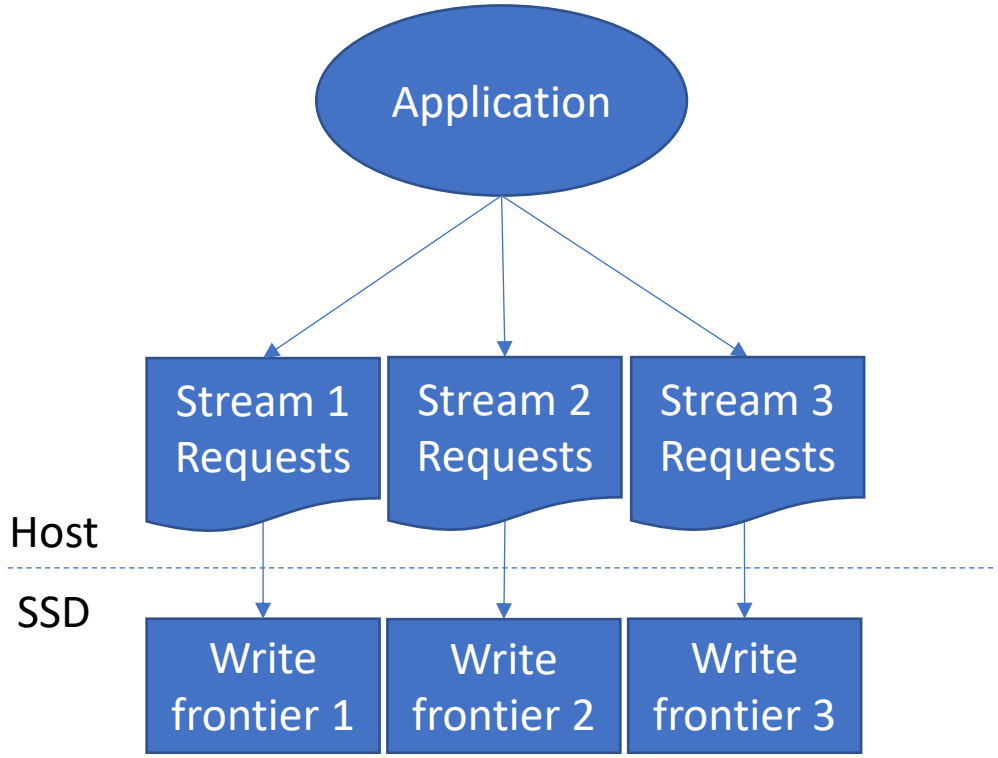
- Existing schemes
  - Baseline
  - Partial GC



Choose the block
in any superblock
with the most invalid
data as the victim

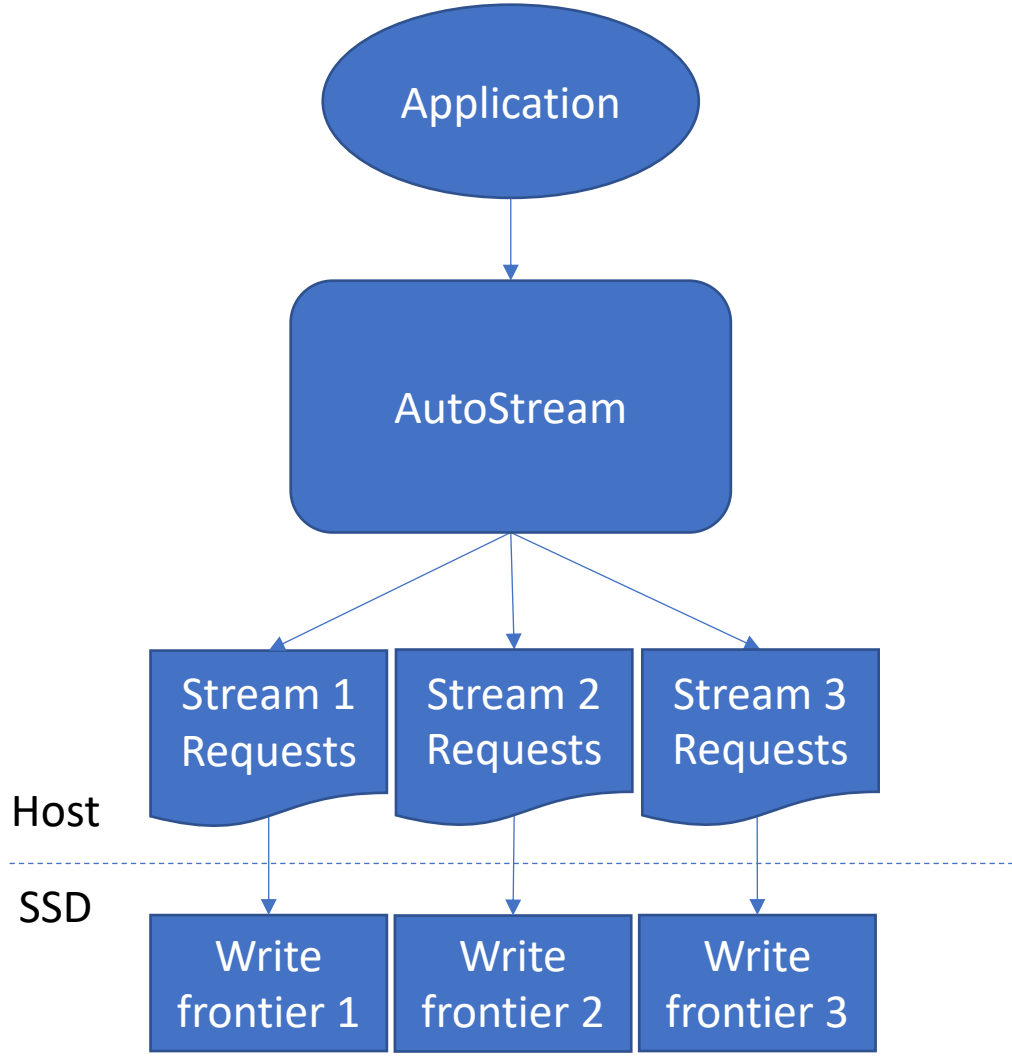
Evaluation – Setup

- Existing schemes
  - Baseline
  - Partial GC
  - Multi-stream (HotStorage 14)



Evaluation – Setup

- Existing schemes
  - Baseline
  - Partial GC
  - Multi-stream (HotStorage 14)
  - AutoStream (SFR method, SYSTOR 17)



Evaluation – Setup

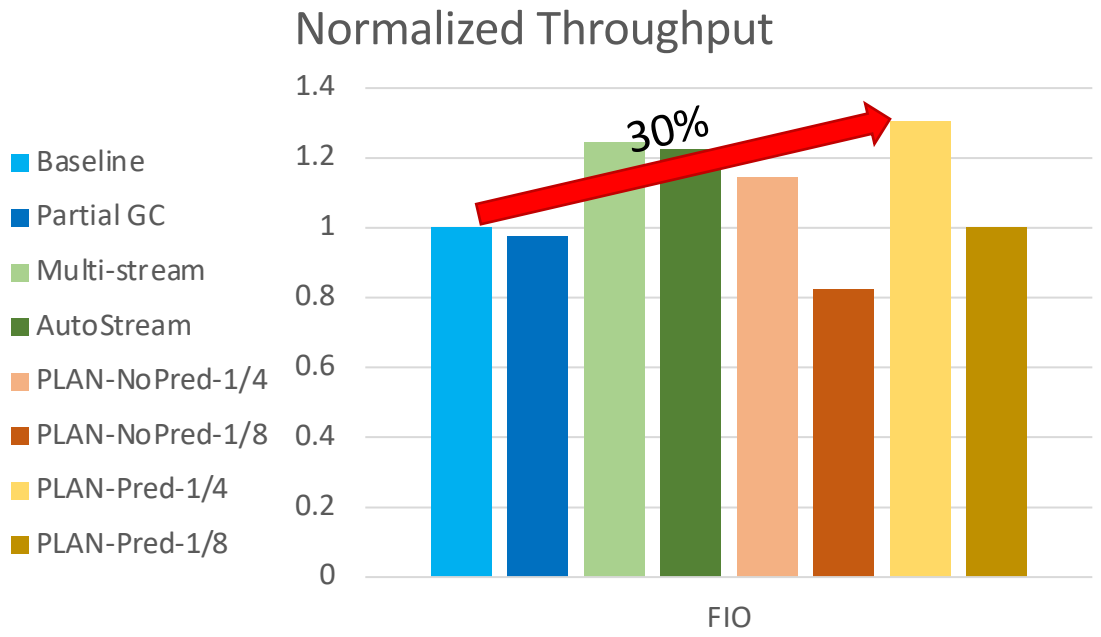
- Existing schemes
  - Baseline
  - Partial GC
  - Multi-stream (HotStorage 14)
  - AutoStream (SFR method, SYSTOR 17)
- PLAN
  - No lifetime prediction, 1/4 parallelism
  - No lifetime prediction, 1/8 parallelism
  - With lifetime prediction, 1/4 parallelism
  - With lifetime prediction, 1/8 parallelism

Two questions:

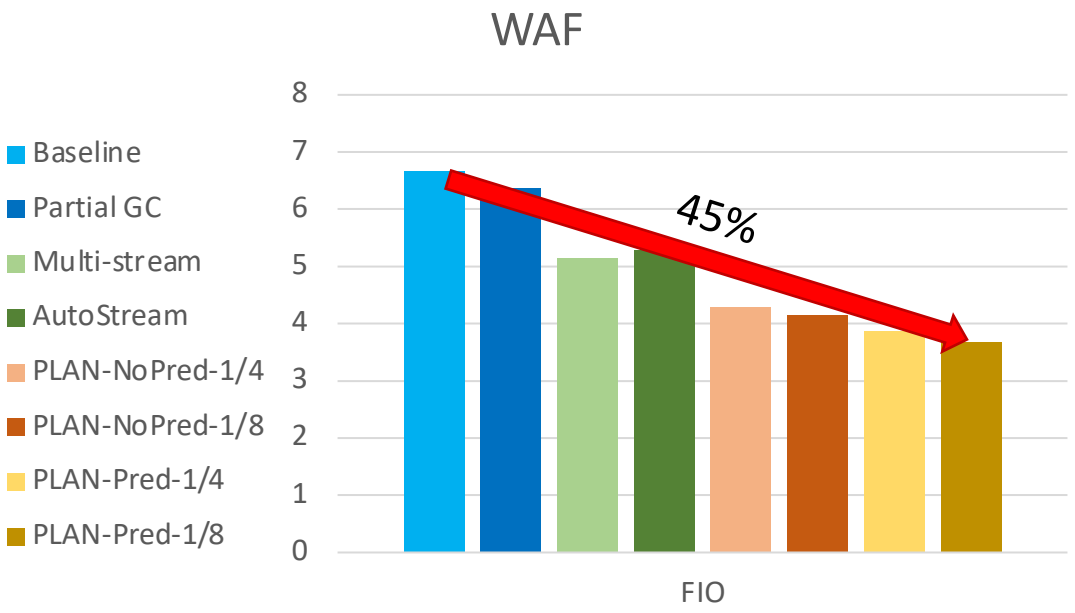
1. How does PLAN work without lifetime prediction?
2. How does PLAN work with different levels of parallelism?

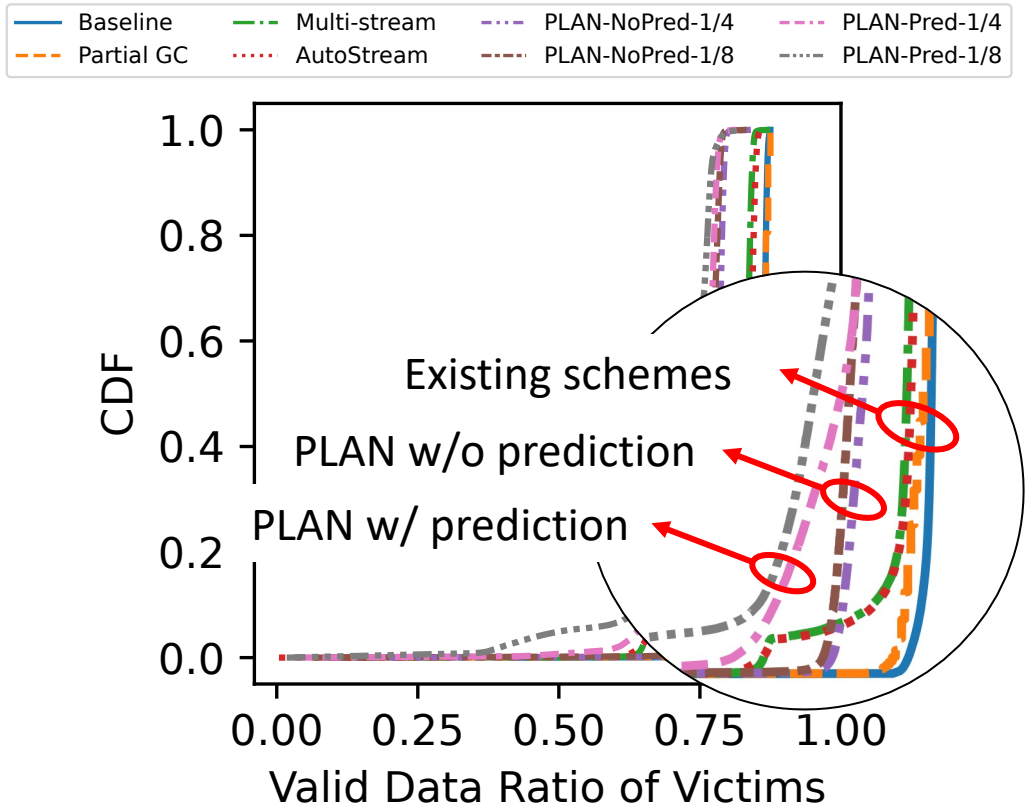
Evaluation – FIO Performance

- PLAN improves throughput for FIO by up to 30%
  - I/O-bounded
    - iodepth = 32



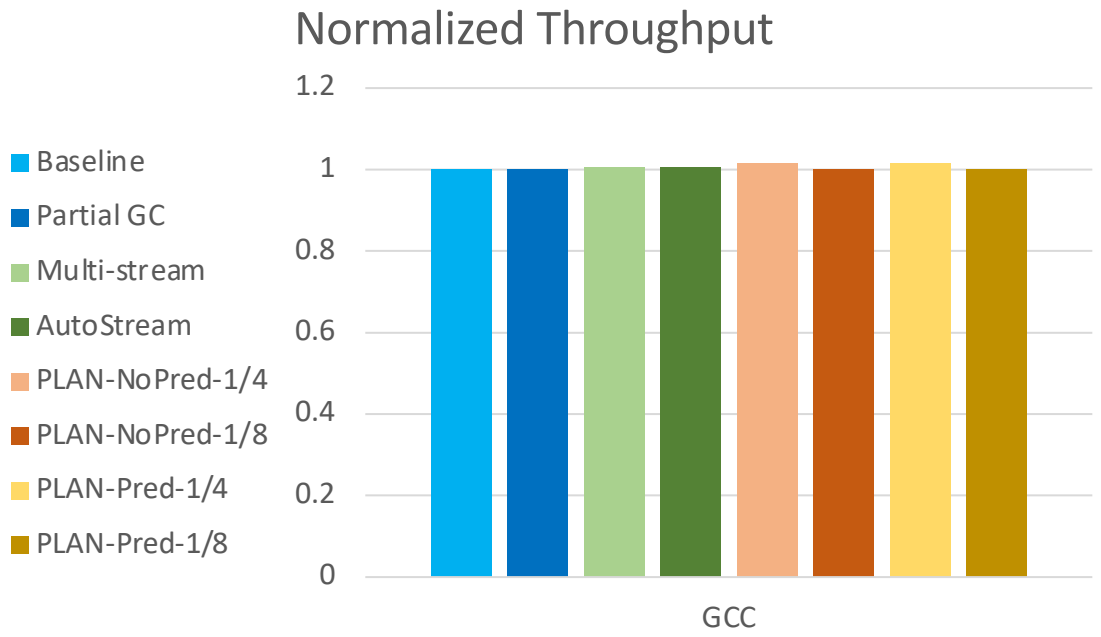
Evaluation – FIO WAF



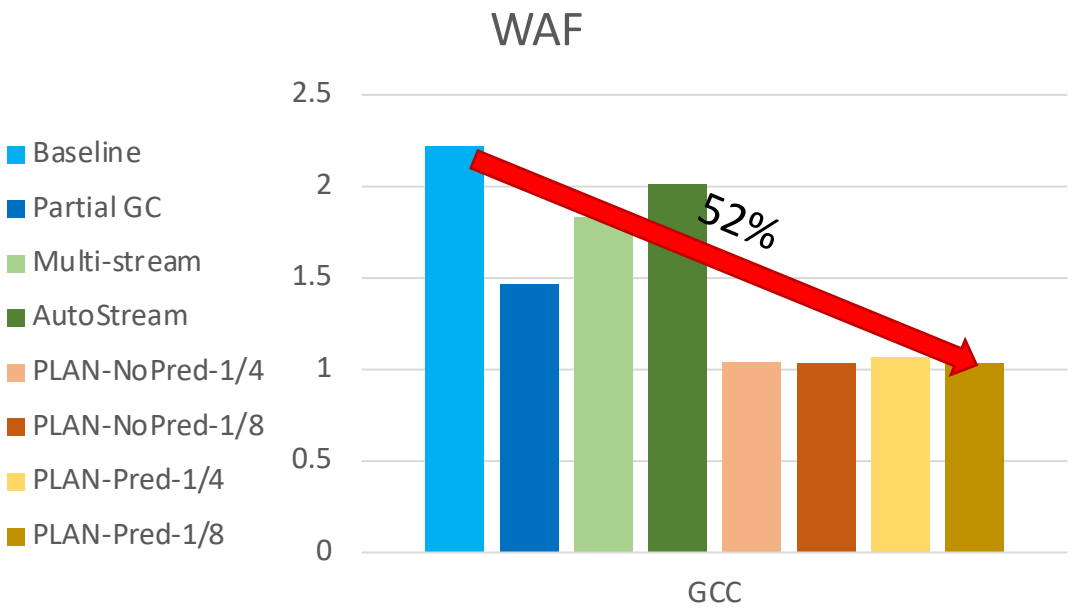


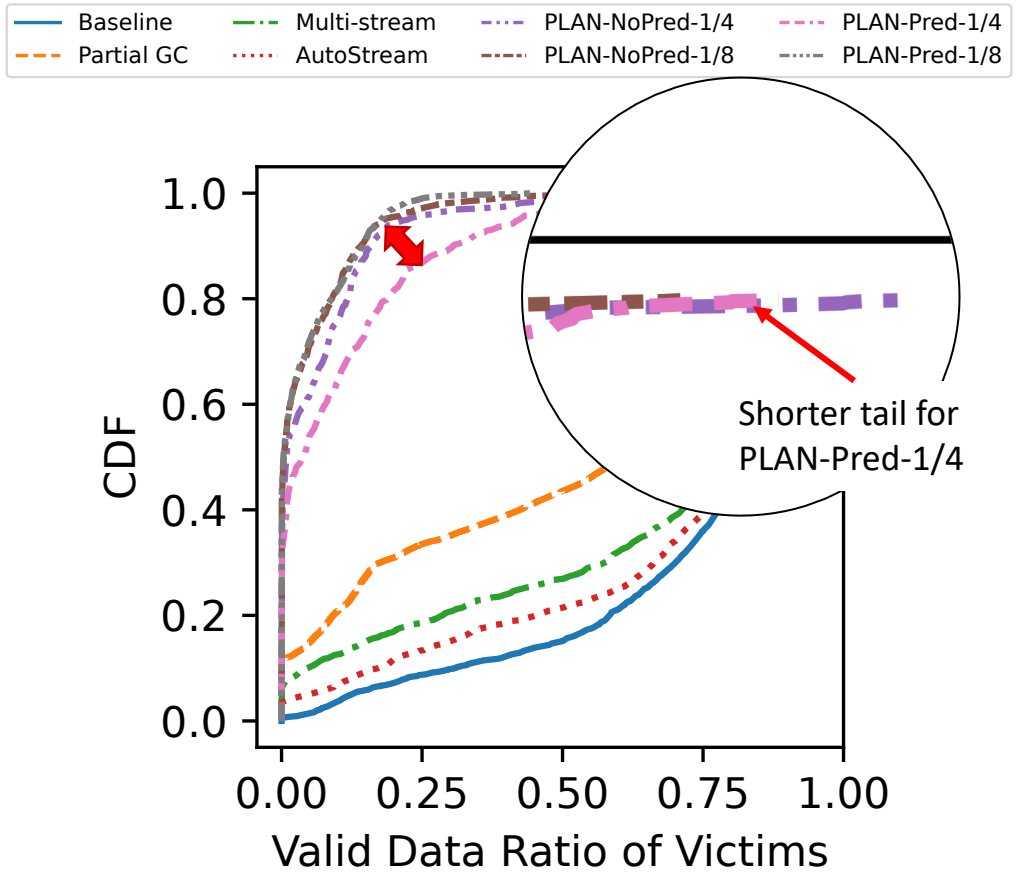
Evaluation – GCC Performance

- PLAN keeps throughput for GCC
  - CPU-bounded
    - CPU constantly at 100%



Evaluation – GCC WAF





Conclusion

- Higher parallelism does not always bring better performance
- PLAN provides better SSD-internal parallelism management
  - High parallelism for large, sequential requests
  - Low parallelism for small, random requests

Q&A

Thank you!

Contact me: xzhang84@syr.edu

https://zhxq.me/