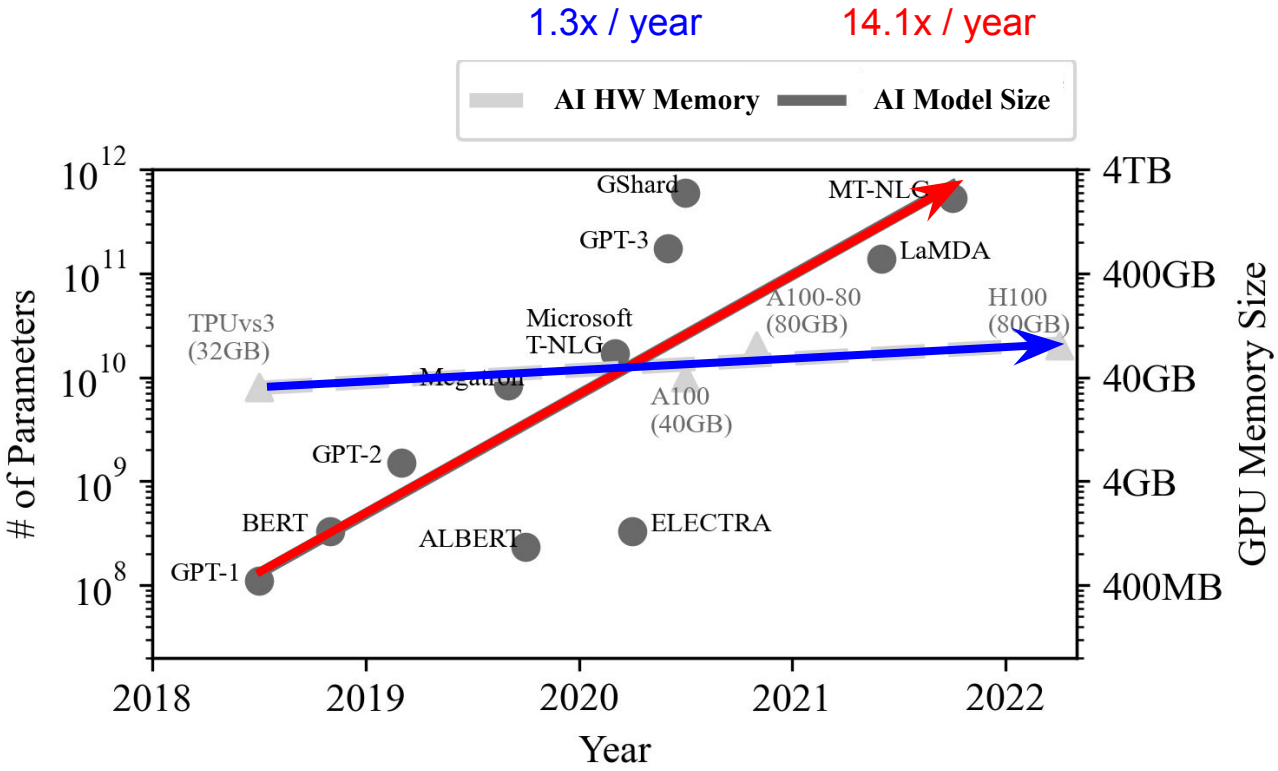
Overcoming the Memory Wall with CXL-Enabled SSDs

Shao-Peng Yang[1], Minjae Kim[2], Sanghyun Nam[3], Juhyung Park[2], Jin-yong Choi[4], Eyee Hyun Nam[4], Eunji Lee[3], Sungjin Lee[2], Bryan S. Kim[1]

[1]Syracuse University, [2]DGIST, [3]Soongsil University, [4]FADU Inc.

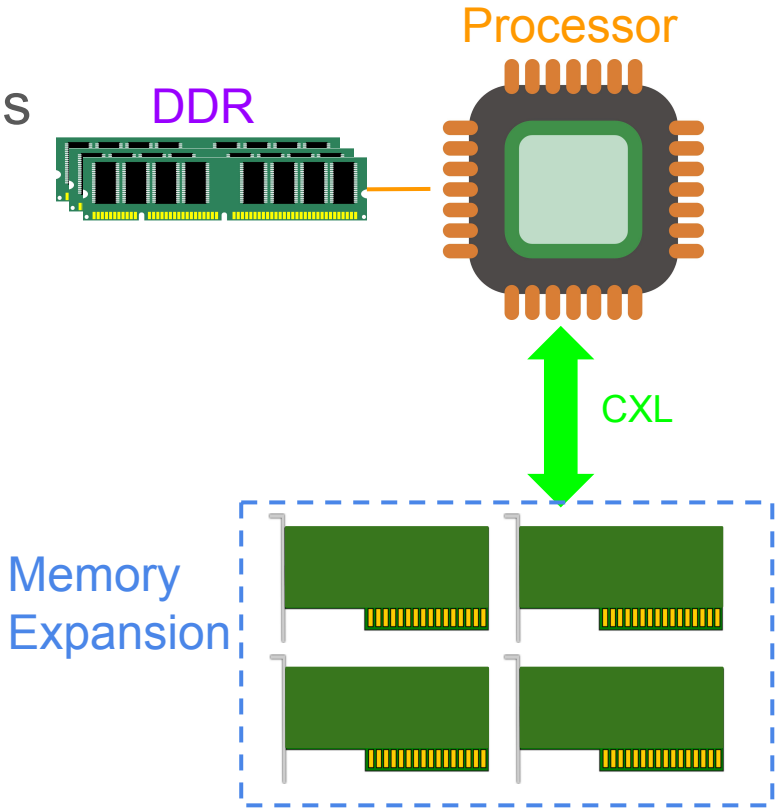
2023 USENIX Annual Technical Conference

What is the memory wall?



Compute Express Link (CXL) and CXL-flash

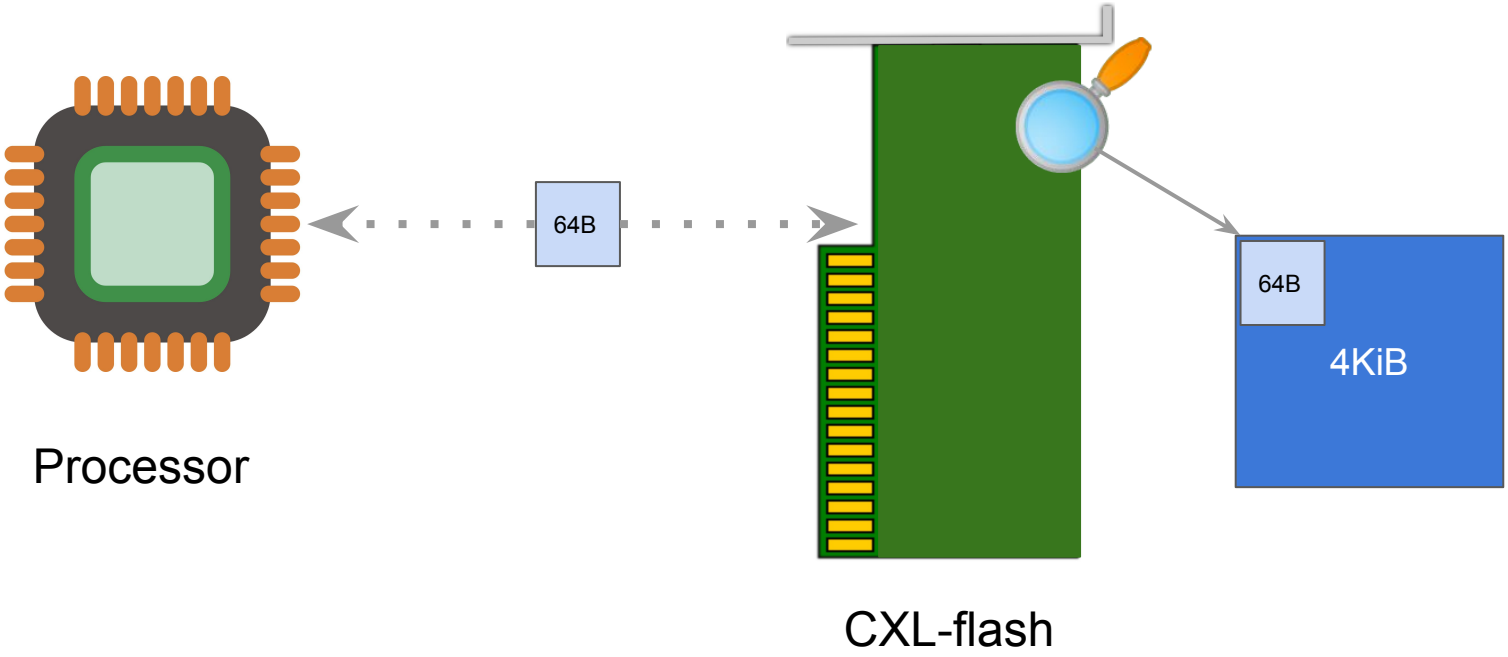
- CXL enables direct memory access between CPU and endpoints
- Samsung Memory-semantic SSD[1] and CXL-SSD[2] are examples of CXL-flash



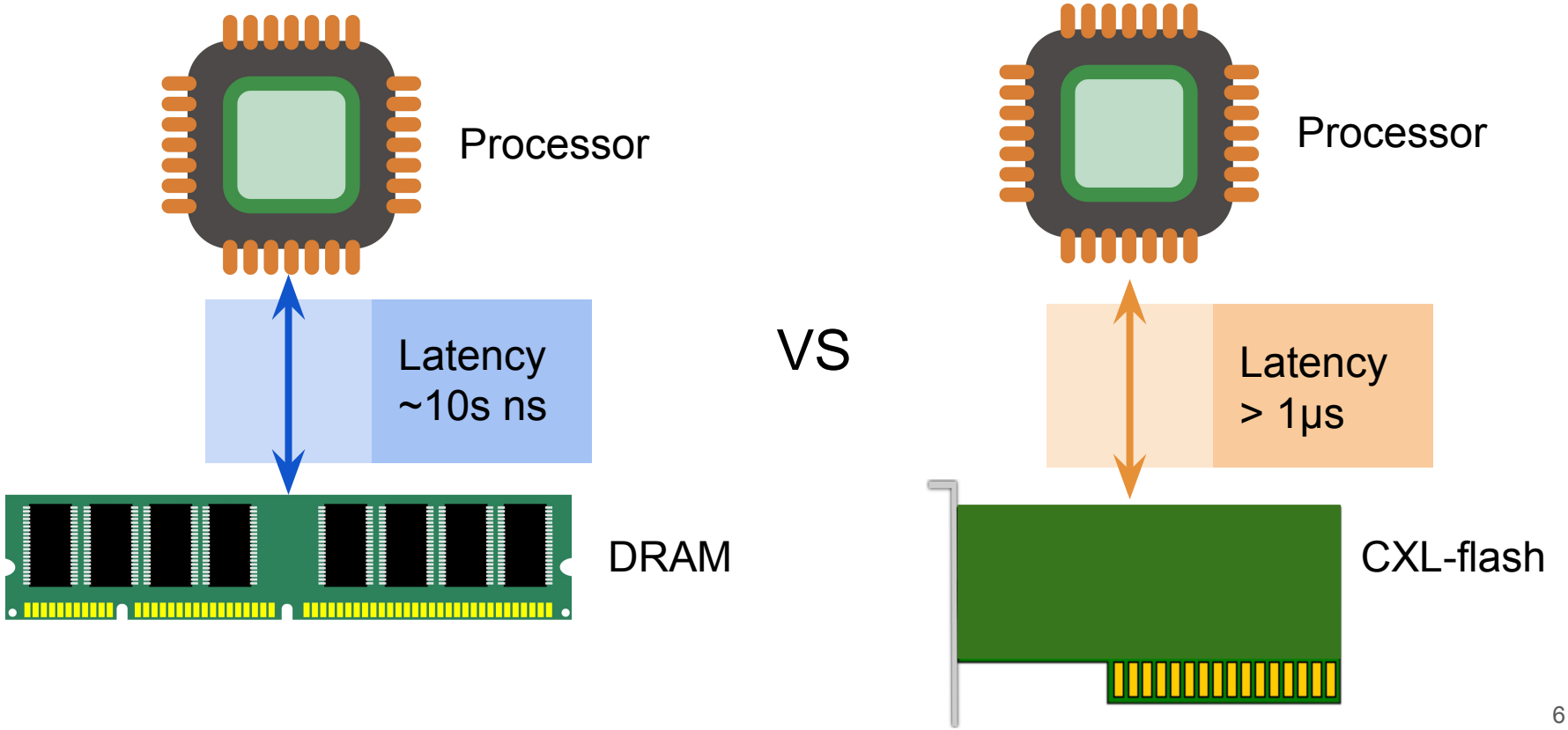
1 Memory-Semantic SSD. https://samsungmsl.com/ms-ssd/.
2 Myoungsoo Jung. Hello bytes, bye blocks: PCIe storage meets compute express link for memory expansion (CXL-SSD). In HotStorage '22, page 45–51. Association for Computing Machinery, 2022. https://doi.org/10.1145/3538643.3539745.

Can flash memory handle the intensity of memory requests?

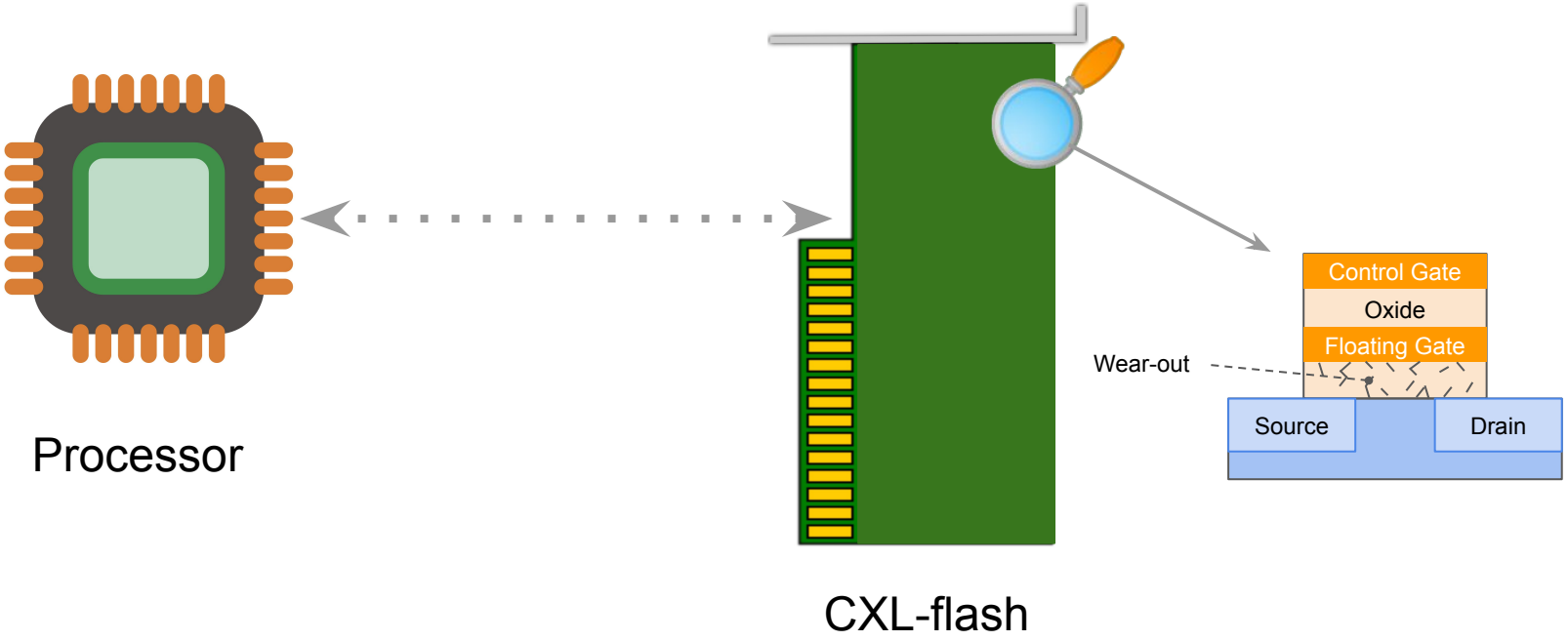
Challenge #1 - granularity mismatch



Challenge #2 - microsecond latency



Challenge #3 - limited endurance



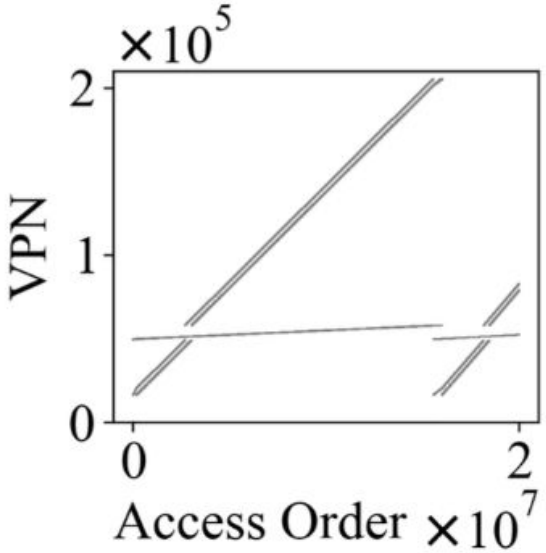
Contributions

- CXL-flash design tools
  - Physical memory tracer
  - CXL-flash simulator
- Design space of CXL-flash
  - Optimization techniques
- Analysis on CXL-flash performance
  - Effectiveness of algorithms
  - System-level change

Outline

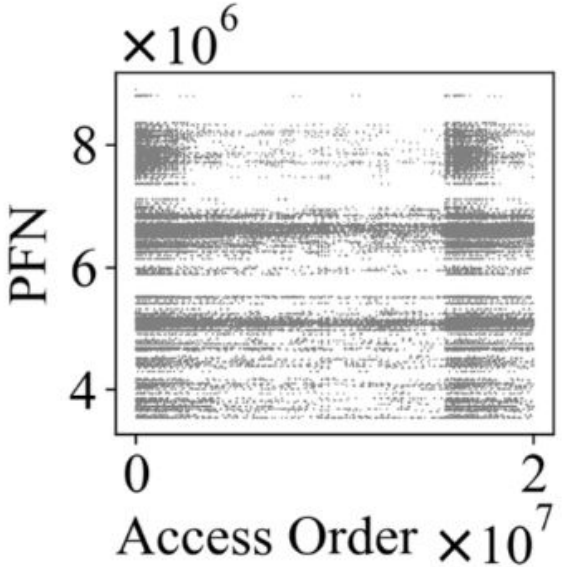
- **Memory Tracing**
- Design of CXL-flash
- Evaluation and Observations
- Final Thoughts

Virtual vs physical memory trace



Matrix mult. (V)

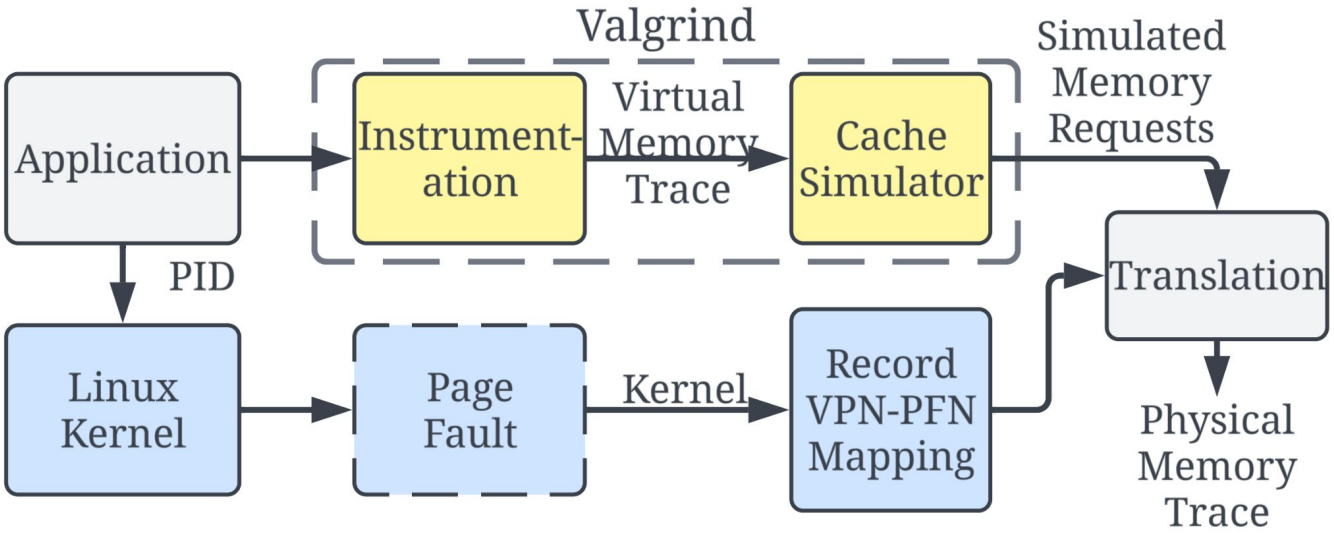
VS



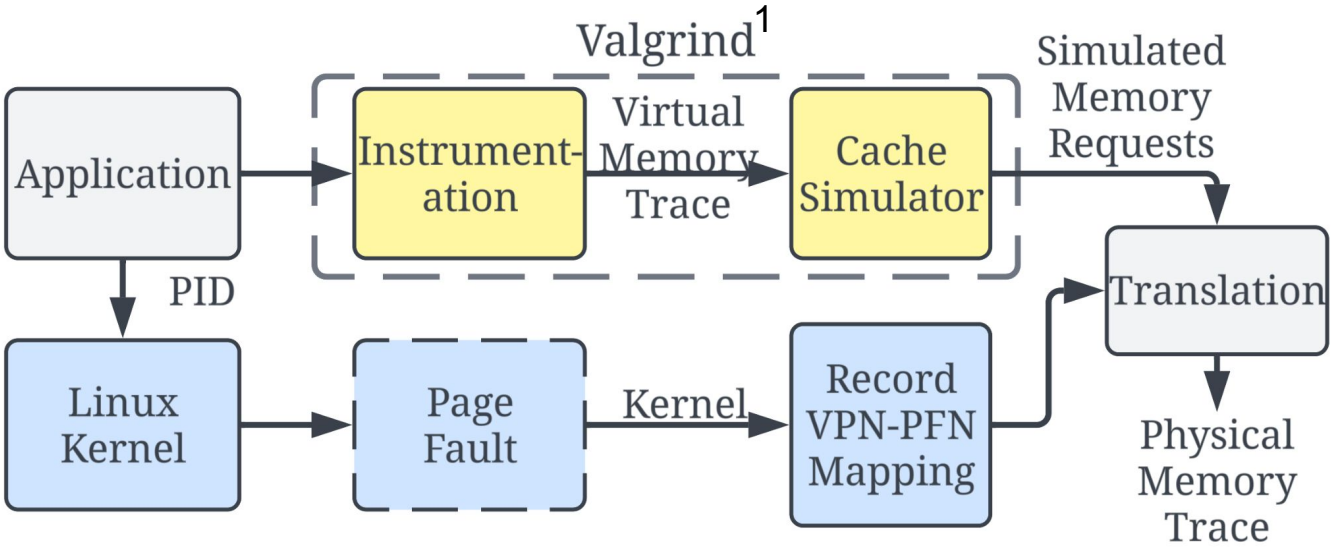
Matrix mult. (P)

Overview of physical memory tracer

- Independent of hardware or tools
- Capture physical memory accesses instead of virtual ones



Physical memory tracing tool



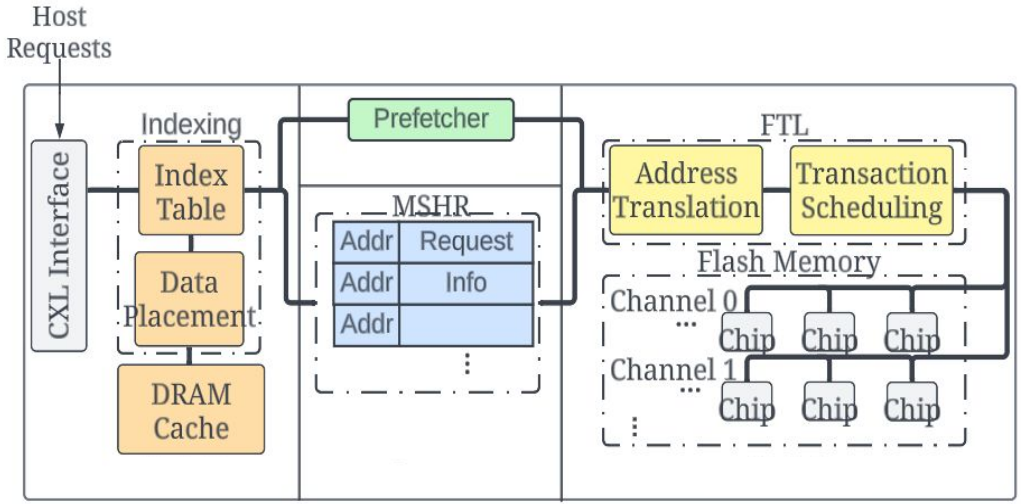
1 Nicholas Nethercote and Julian Seward. Valgrind: A framework for heavyweight dynamic binary instrumentation. In Proceedings of the 28th ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI '07, page 89–100. Association for Computing Machinery, 2007. https://doi.org/10.1145/1250734.1250746.

Outline

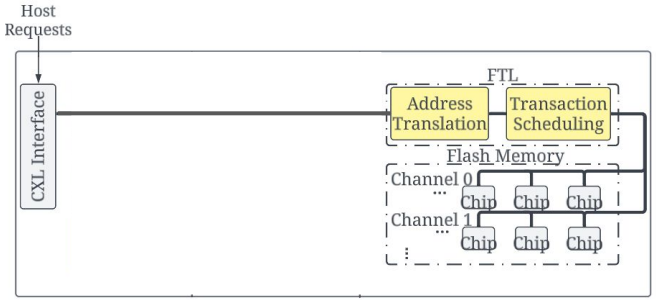
- Memory Tracing
- **Design of CXL-flash**
- Evaluation and Observations
- Final Thoughts

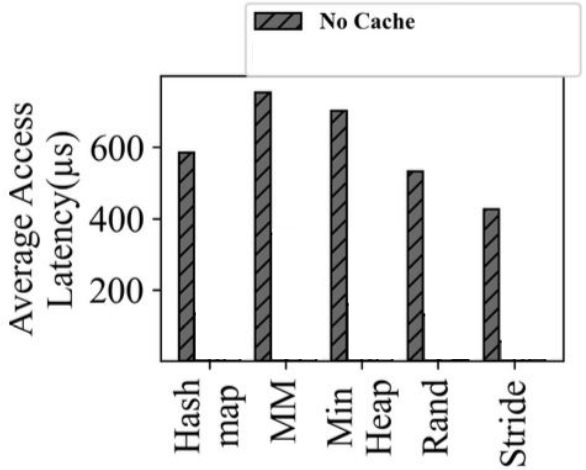
Overview of our design

- Integration of existing techniques
- Experiments with synthetic workloads

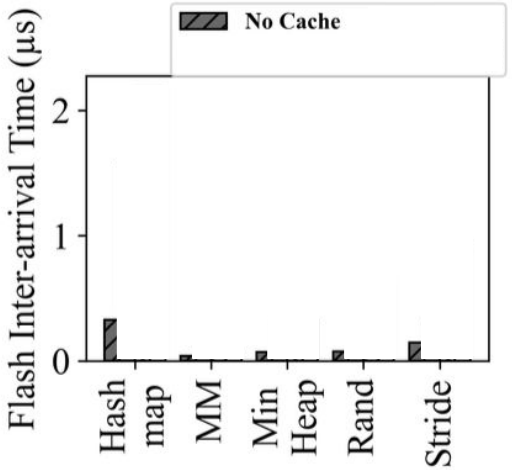


Design of CXL-flash - cache





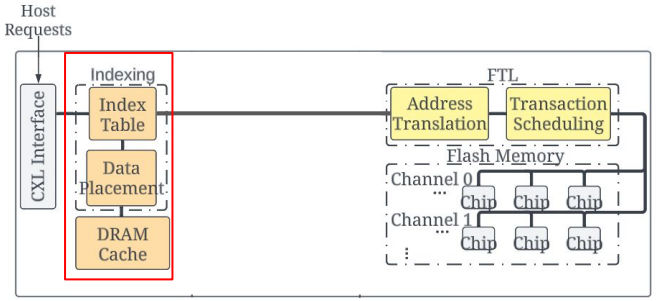
Average access latency

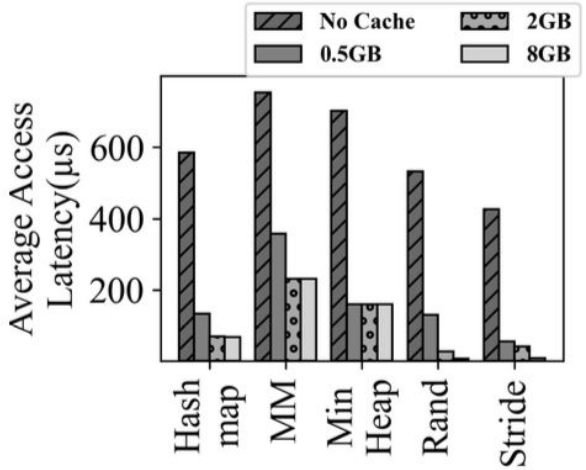


Flash inter-arrival time

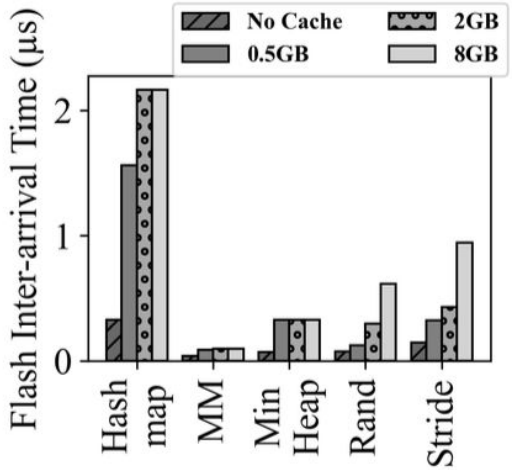
Design of CXL-flash - cache

- DRAM cache reduces latency and traffic





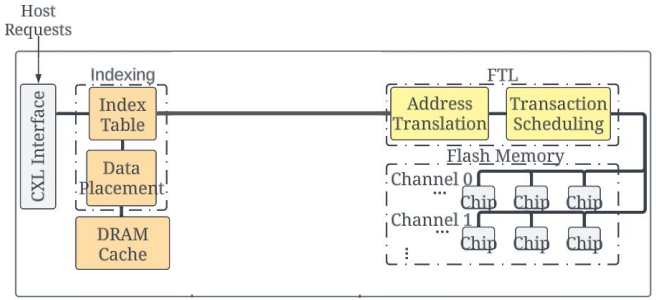
Average access latency

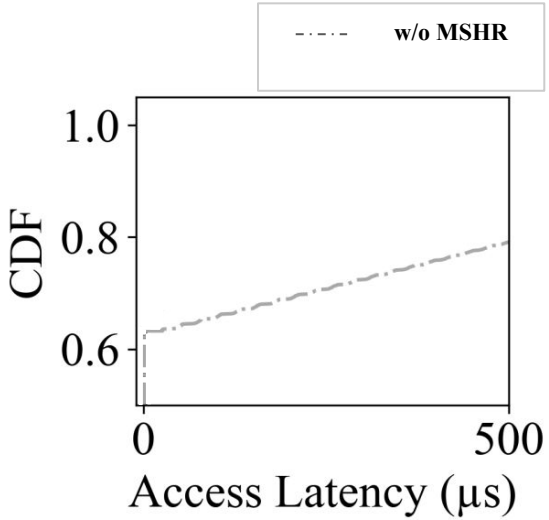


Flash inter-arrival time

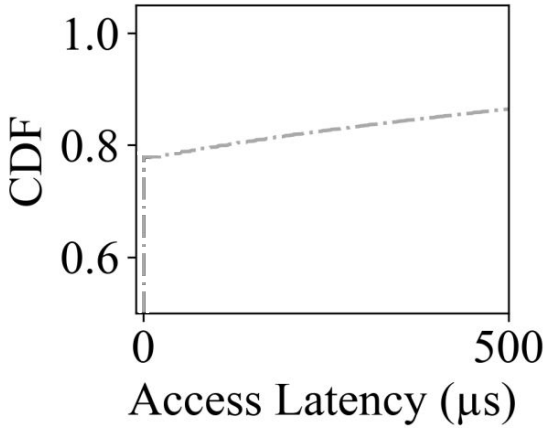
Design of CXL-flash - miss status holding registers (MSHR)

- Even with a large cache size (8GB), the average access latency is still high





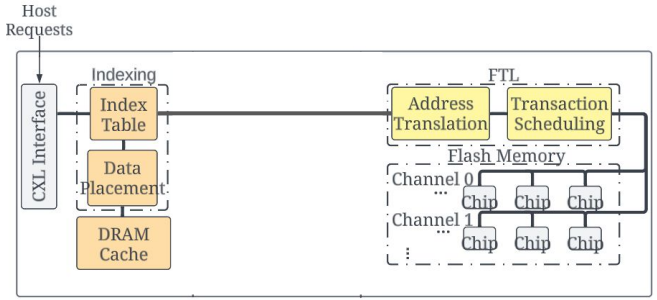
Matrix mult.

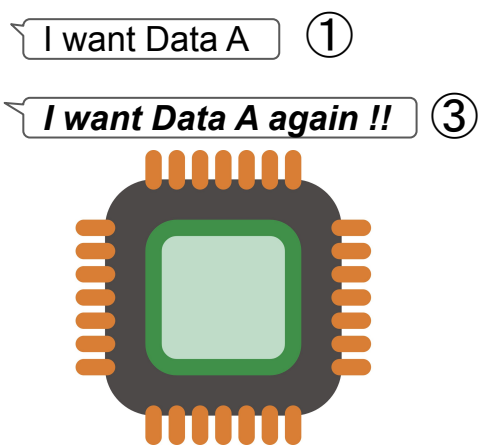


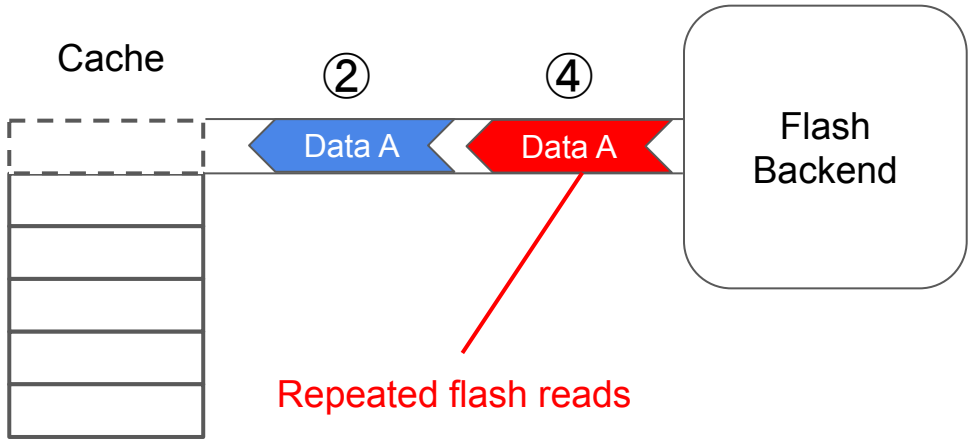
Min heap

Design of CXL-flash - MSHR

- This is due to repeated flash reads

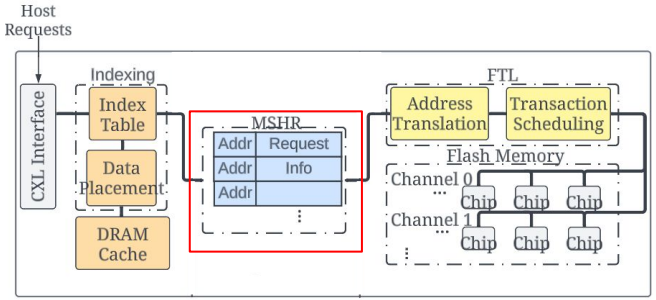


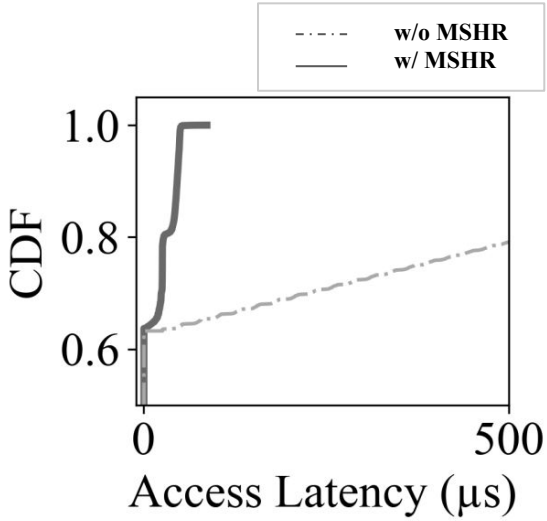




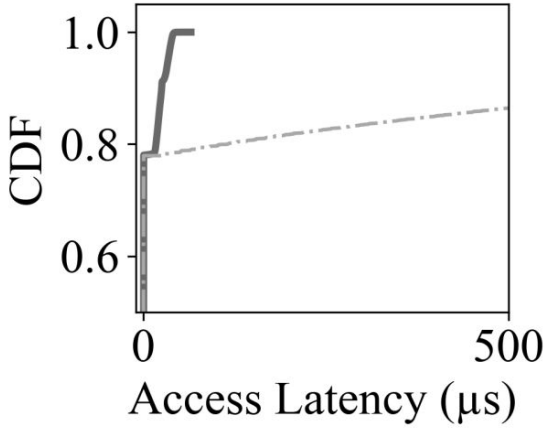
Design of CXL-flash - MSHR

- MSHR prevents repeated flash reads





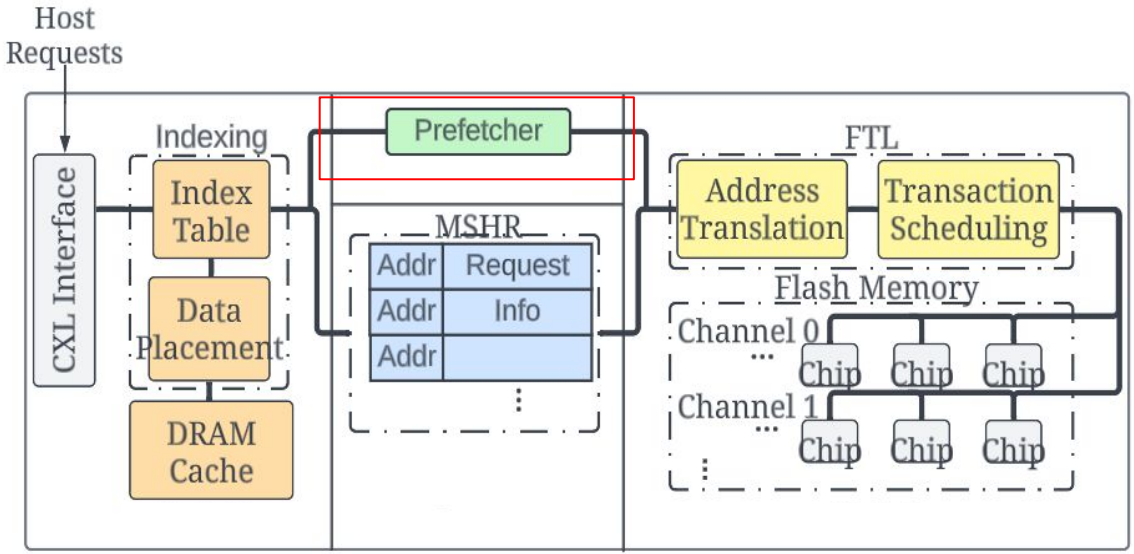
Matrix mult.



Min heap

Design of CXL-flash - prefetcher

- A prefetcher is added to improve the device's performance



Outline

- Memory Tracing
- Design of CXL-flash
- **Evaluation and Observations**
- Final Thoughts

Evaluation objectives

- How effective are the cache policies?
- How effective are the prefetchers?
- Is CXL-flash a good memory expansion option?
- How is the performance difference between virtual and physical traces?

Evaluation Overview

- Cache Policies
  - FIFO
  - Random
  - LRU
  - CFLRU
- Prefetchers
  - Next-n-line (NL)
  - Feedback-directed (FD)
  - Best-offset (BO)
  - Leap (LP)

Evaluation Overview

- The evaluation setup:

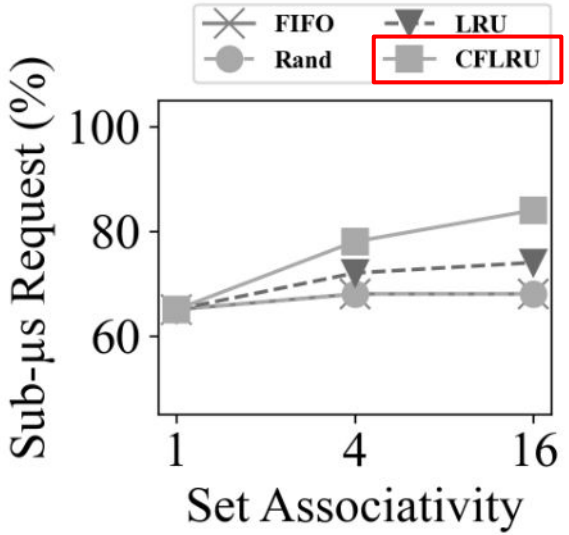
Parameters	Value
DRAM size	64MiB
DRAM latency	46ns
Eviction Policy	CFLRU
Flash parallelism	8×8
Flash technology	ULL

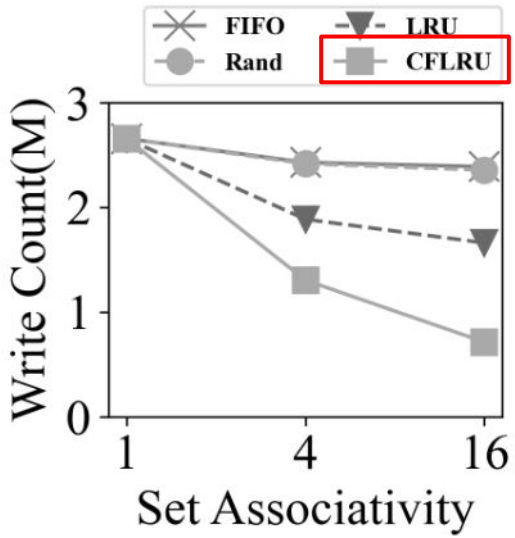
- Real-world applications

Workload	Category	Description
BERT	NLP	Infers using a transformer model
Page rank	Graph	Computes the page rank score
Radiosity	HPC	Computes the distribution of light
XZ	SPEC	Compresses data in memory
YCSB F	KVS	Read-modify-writes on Redis

How effective are the cache policies?

- CFLRU prioritizes evicting clean cache lines

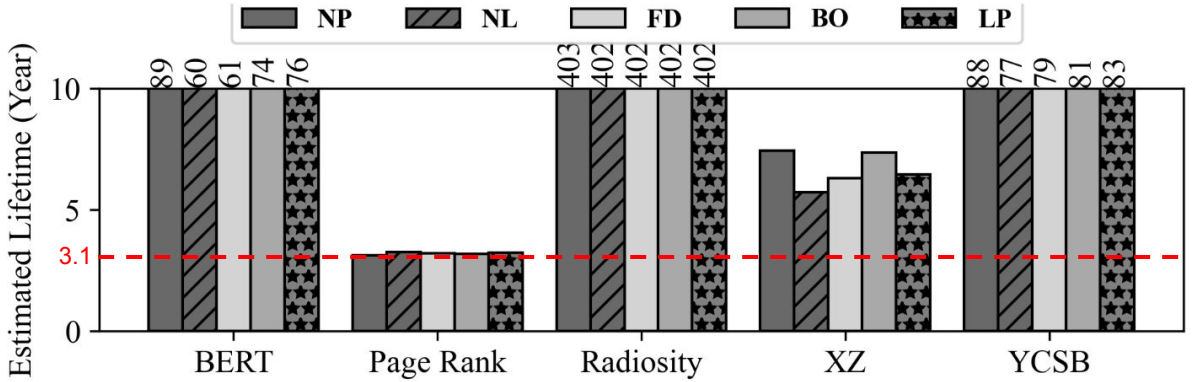




*With BERT

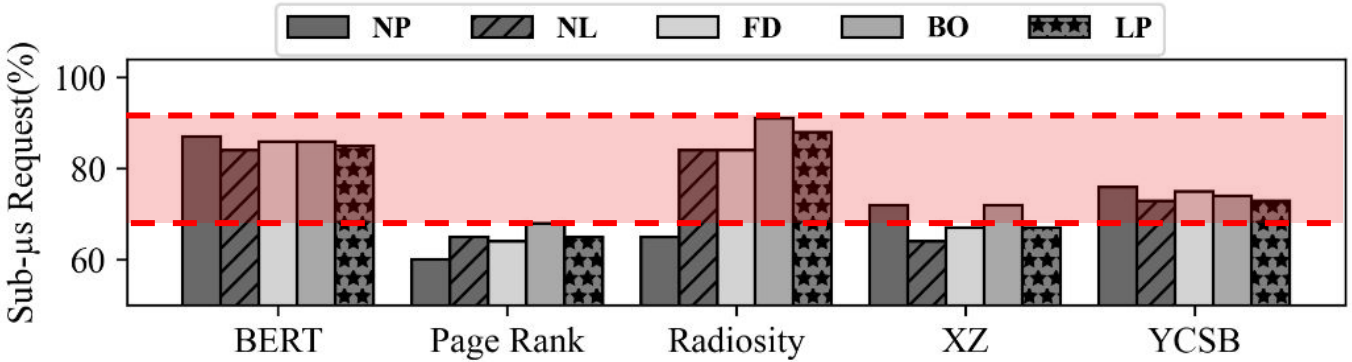
Does CXL-flash have a reasonable lifetime?

- CXL-flash can last for at least 3.1 years



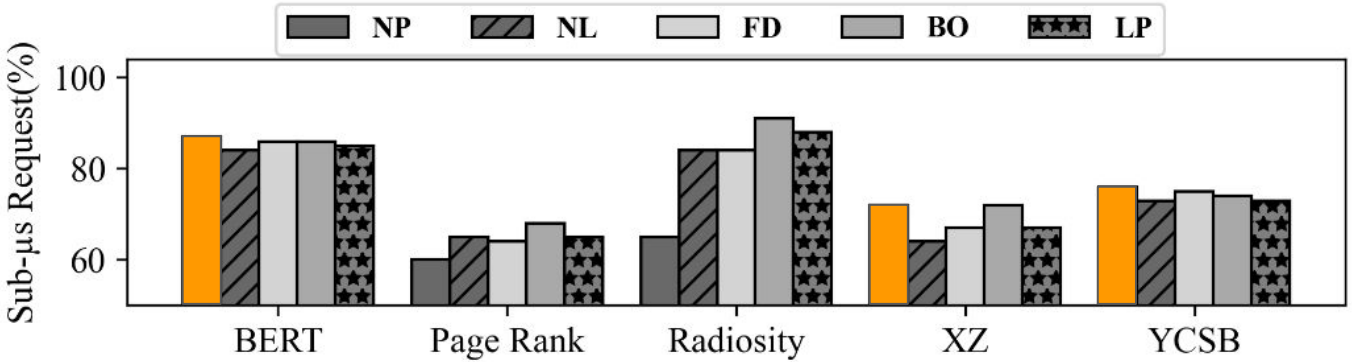
How effective are the prefetchers?

- 68% – 91% of requests experience sub-µs latency

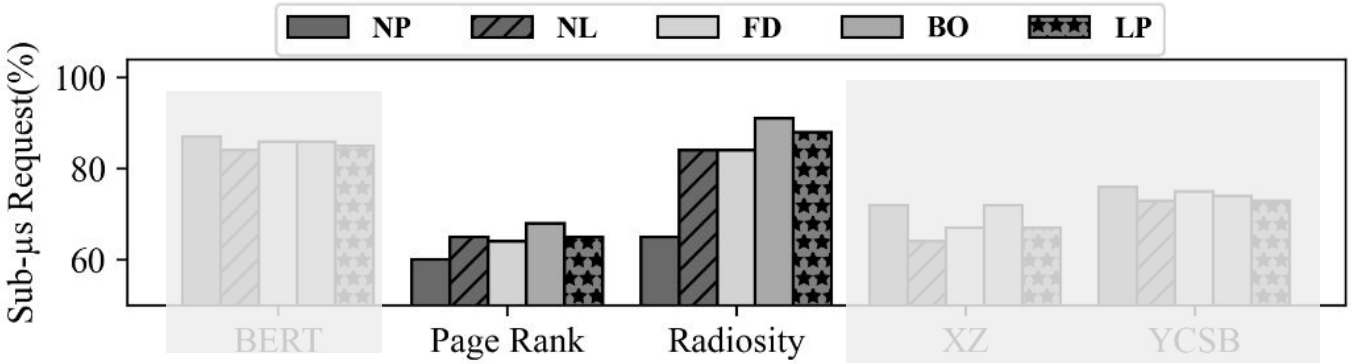


How effective are the prefetchers?

- Using a prefetcher can sometimes hurt performance



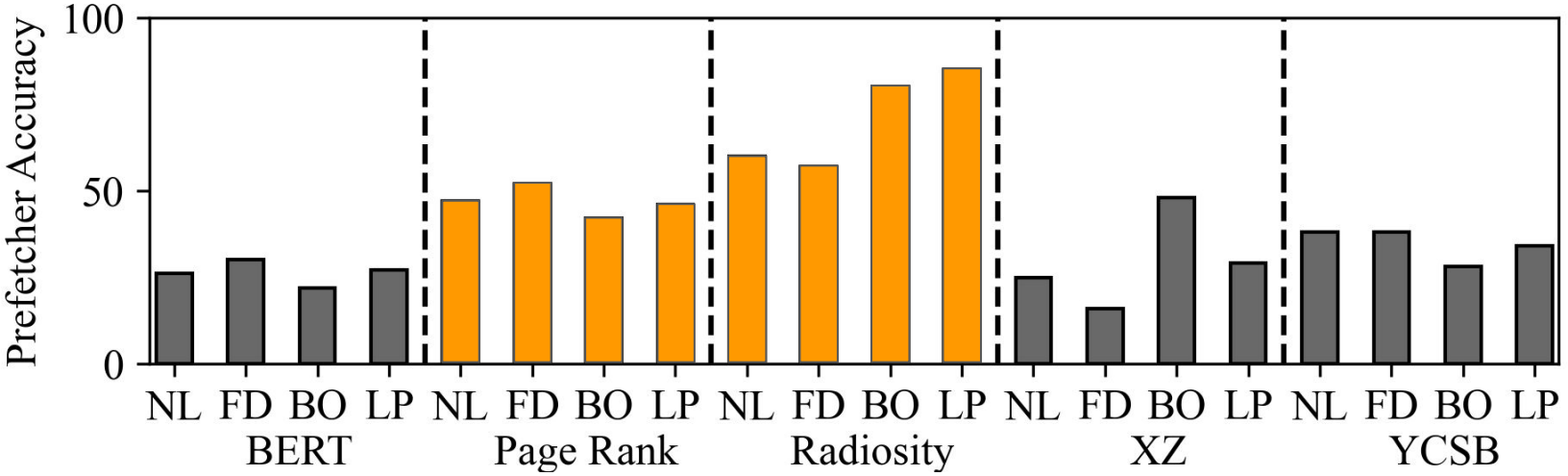
Why does prefetcher improve performance?



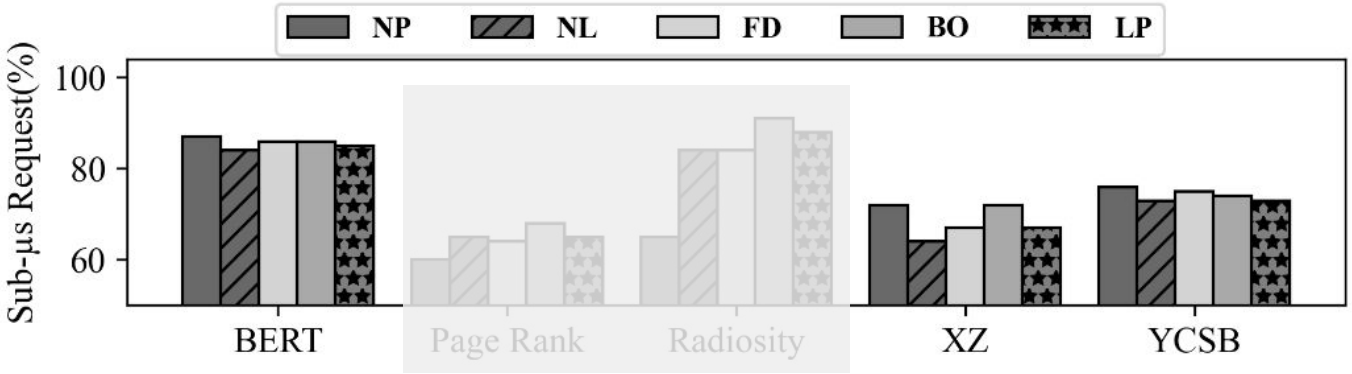
Why does prefetcher improve performance?

- In cases where prefetchers improve performance, it is due to achieving high accuracy

$$\text{Accuracy} = \frac{\text{Accessed prefetched data}}{\text{Total prefetched data}}$$

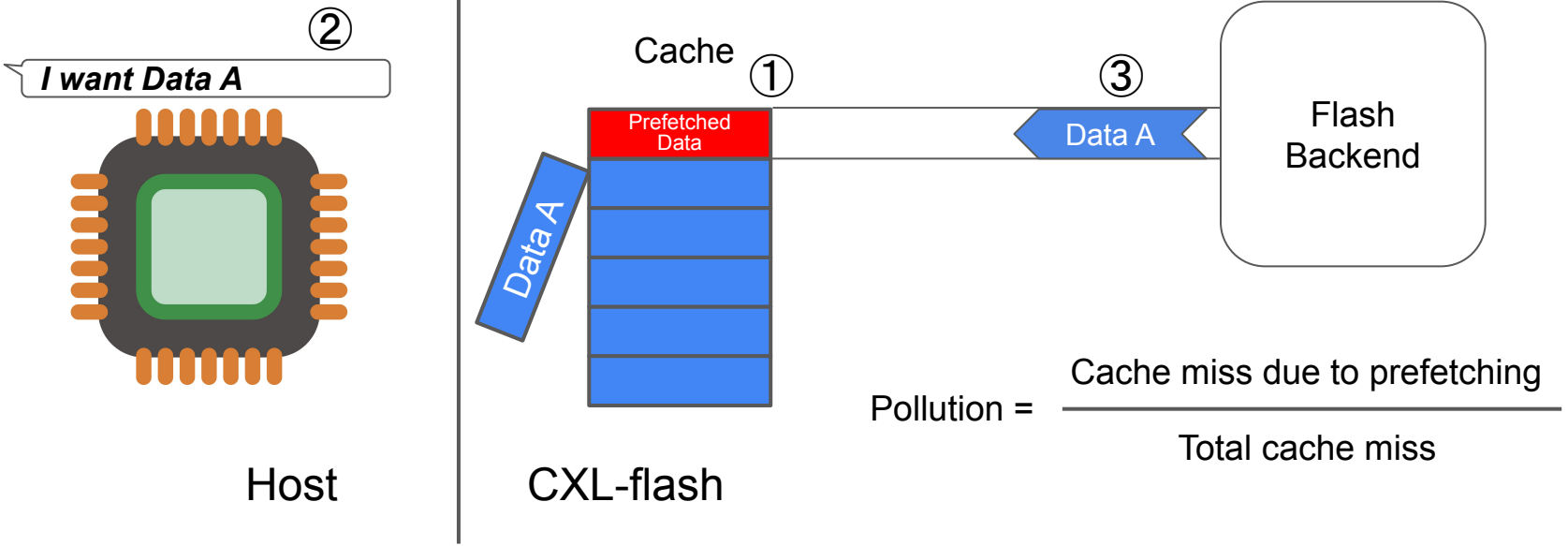


Why does prefetcher degrade performance?



Why does prefetcher degrade performance?

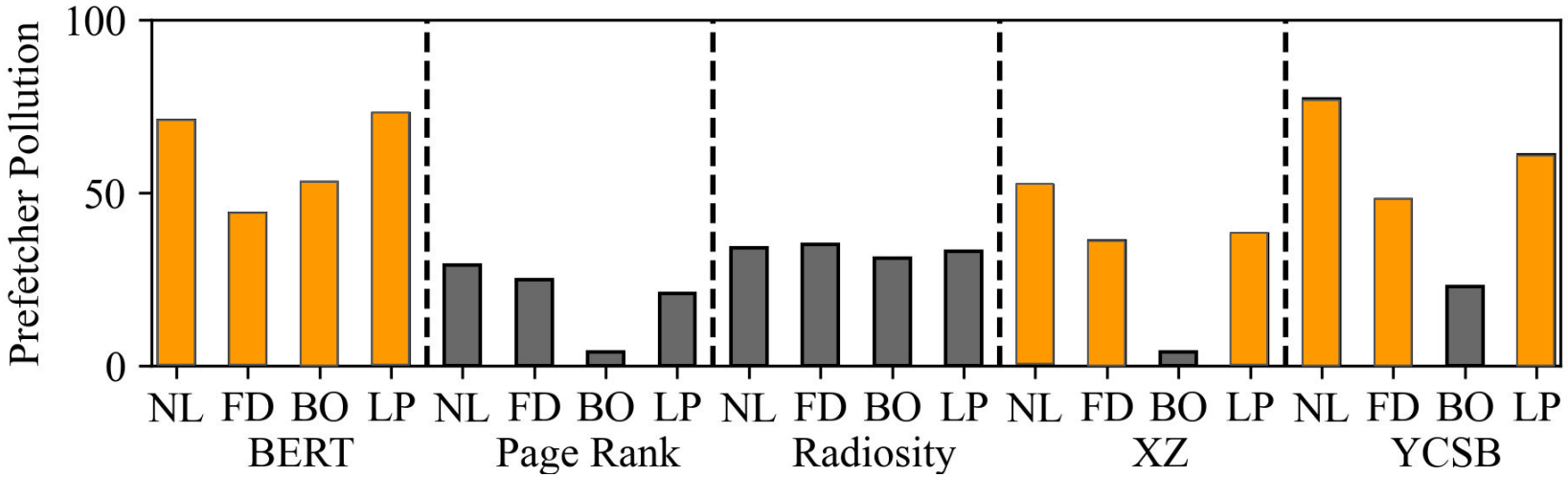
- In cases where prefetchers degrade performance, it is due to cache pollution



$$\text{Pollution} = \frac{\text{Cache miss due to prefetching}}{\text{Total cache miss}}$$

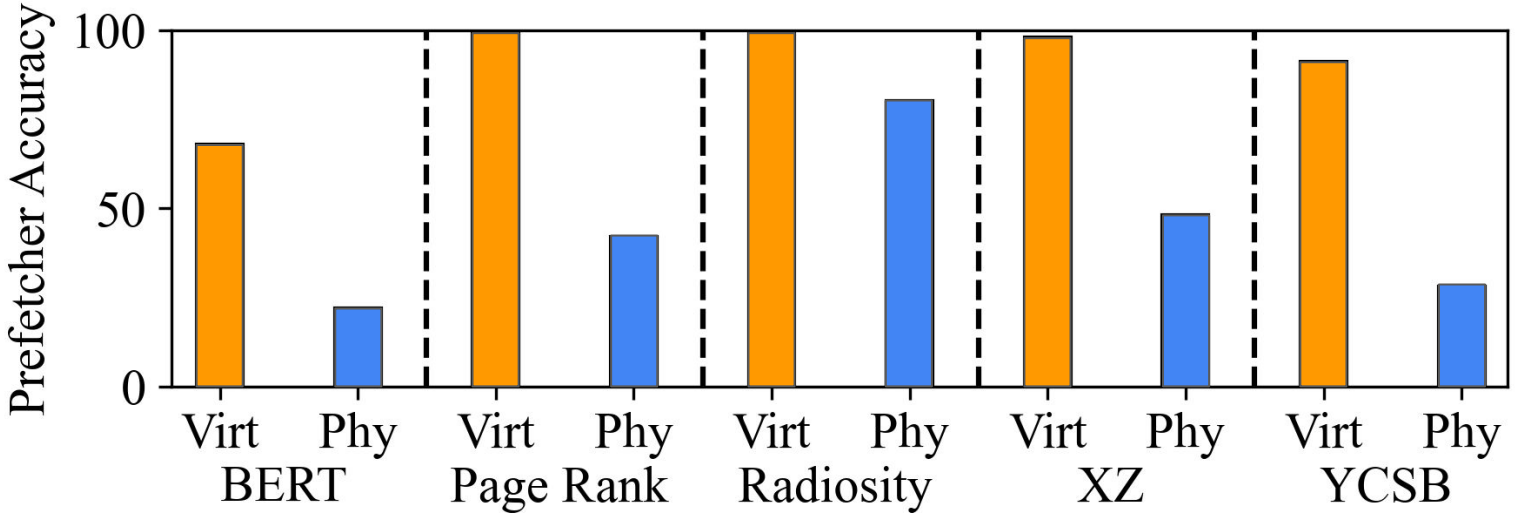
Why does prefetcher degrade performance?

- In cases where prefetchers degrade performance, it is due to cache pollution



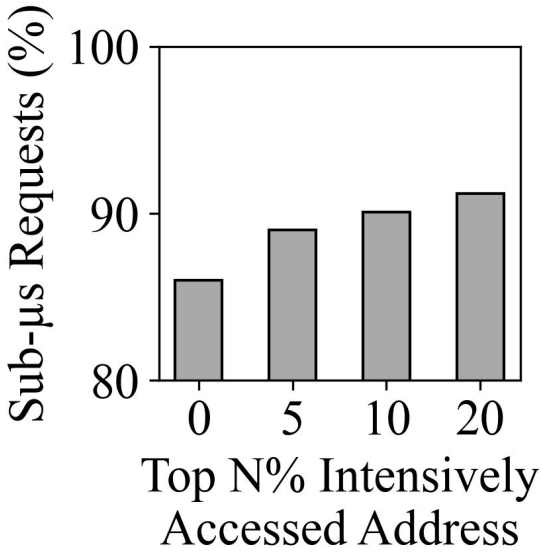
How is the performance difference between traces?

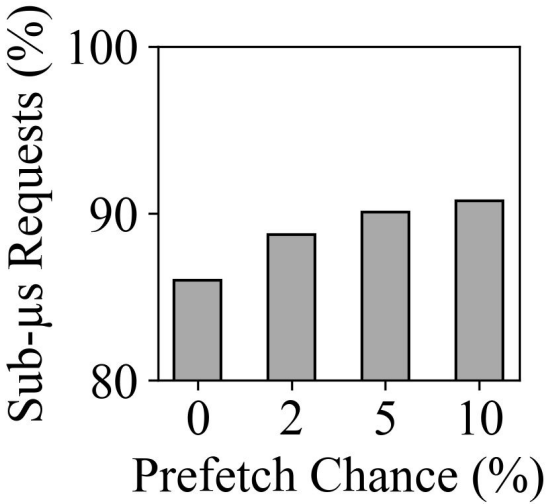
- The V2P address translation
  makes it difficult to accurately prefetch data



How can the performance be further improved?

- Host-generated access pattern hints can improve performance





*With BERT

Outline

- Memory Tracing
- Design of CXL-flash
- Evaluation and Observations
- **Final Thoughts**

Final Thoughts

- CXL-flash has the potential to expand memory
- Future work:
  - DRAM-like performance
    - Flash internal tasks
    - Accuracy and pollution of prefetchers
  - End-to-end performance
    - No existing hardware at the time
    - Interaction between hosts and CXL-flash
- Our work can be a platform for future work to build upon

Thank you
Any questions?

Contact: syang32@syr.edu

Source Code: https://github.com/spypaul/MQSim_CXL