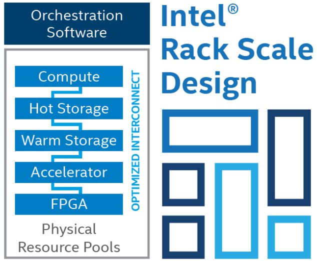
UCLA Samueli School of Engineering

Hermit: Low-Latency, High-Throughput, and Transparent Remote Memory via Feedback-Directed Asynchrony

Yifan Qiao, Chenxi Wang, Zhenyuan Ruan,
Adam Belay, Qingda Lu, Yiying Zhang, Miryung Kim, Guoqing Harry Xu

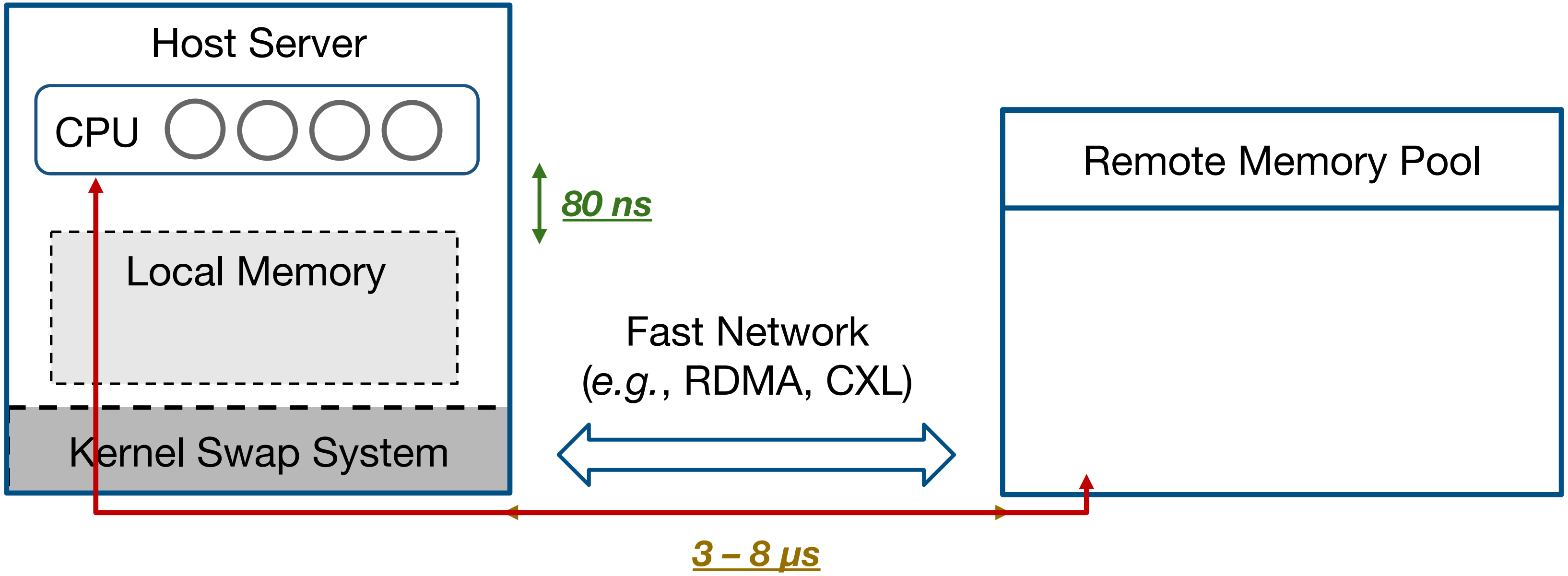
Datacenter Must Balance Latency With Utilization

Latency-Critical Applications



Remote Memory Systems

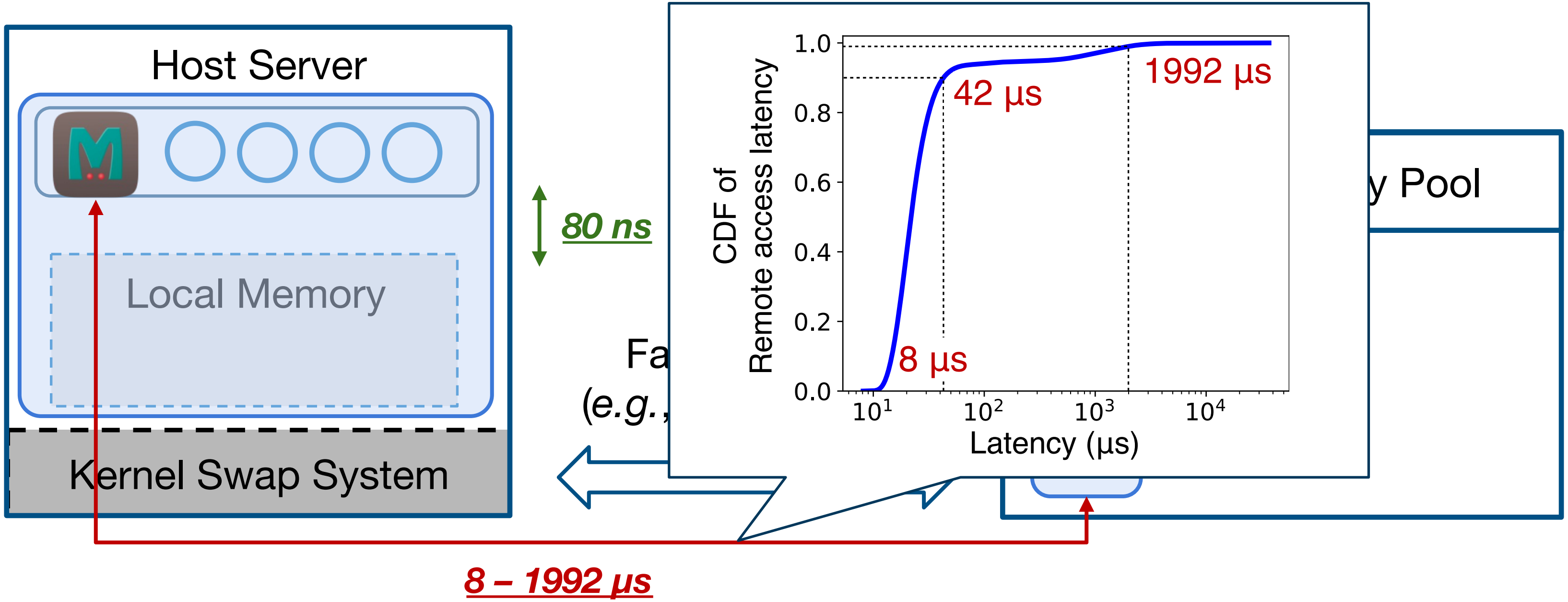
Remote Memory Systems



Network: Mellanox ConnectX-5 Ex (100 GbE, 7 μs for a 4 KB page)

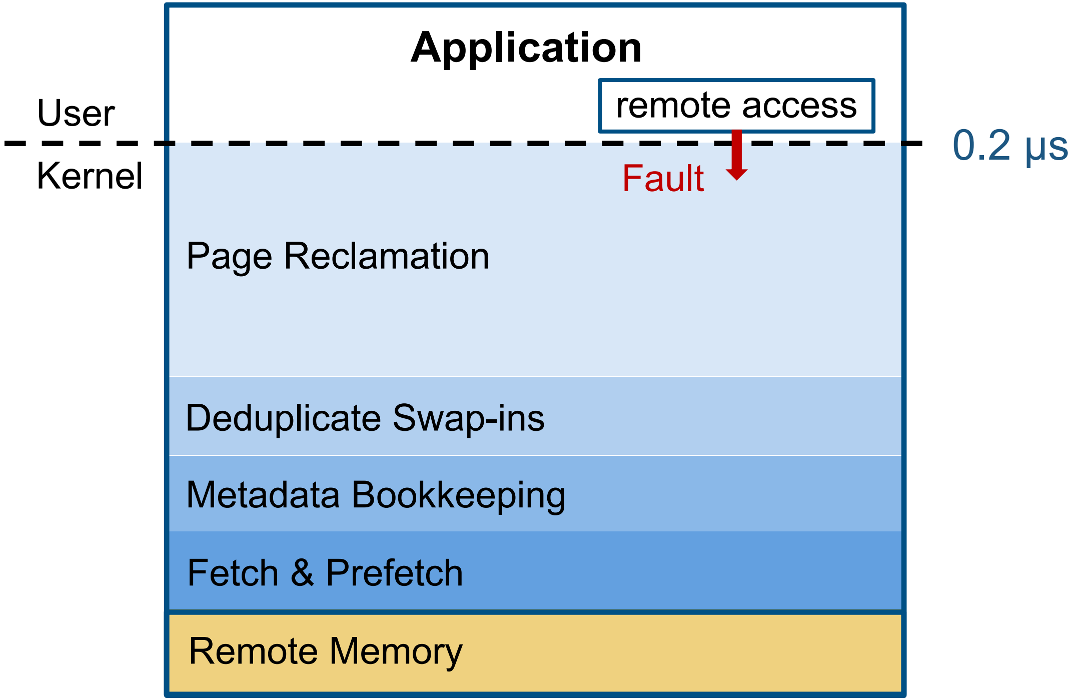
Kernel Swap Can Incur High Tail Latency

1914x higher tail latency!



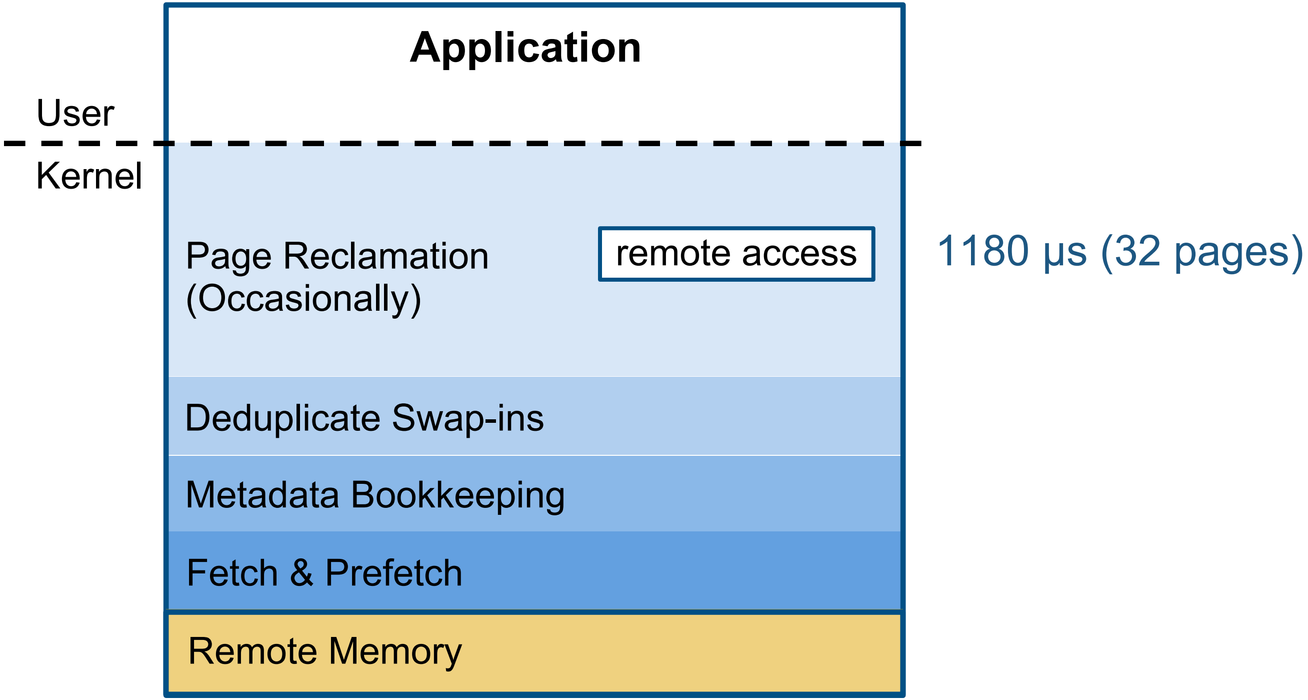
Network: Mellanox ConnectX-5 Ex (100 GbE, 7 µs for a 4 KB page)

Where Does the Latency Come From?



UCLA Samueli School of Engineering

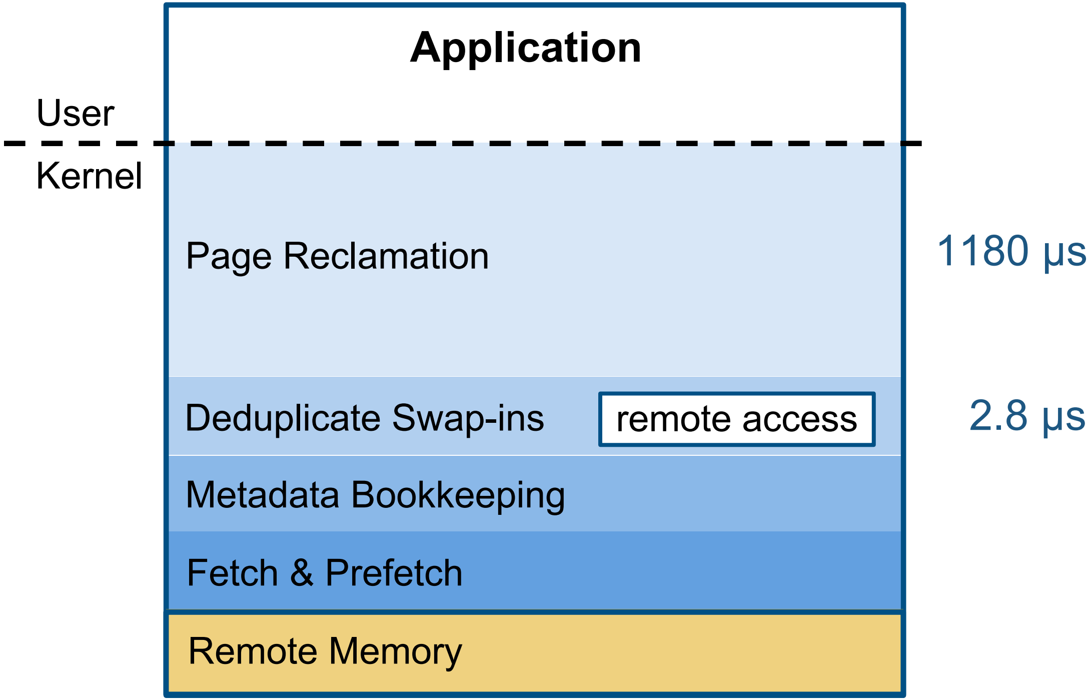
Where Does the Latency Come From?



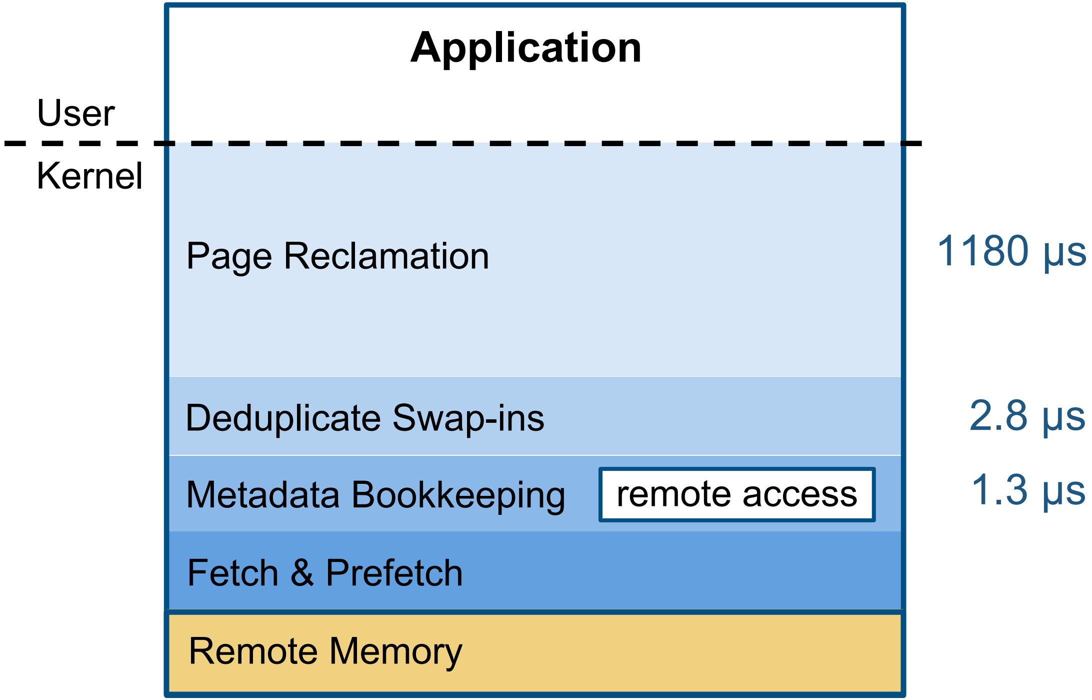
UCLA Samueli School of Engineering

Network: Mellanox ConnectX-5 Ex (100 GbE)

Where Does the Latency Come From?

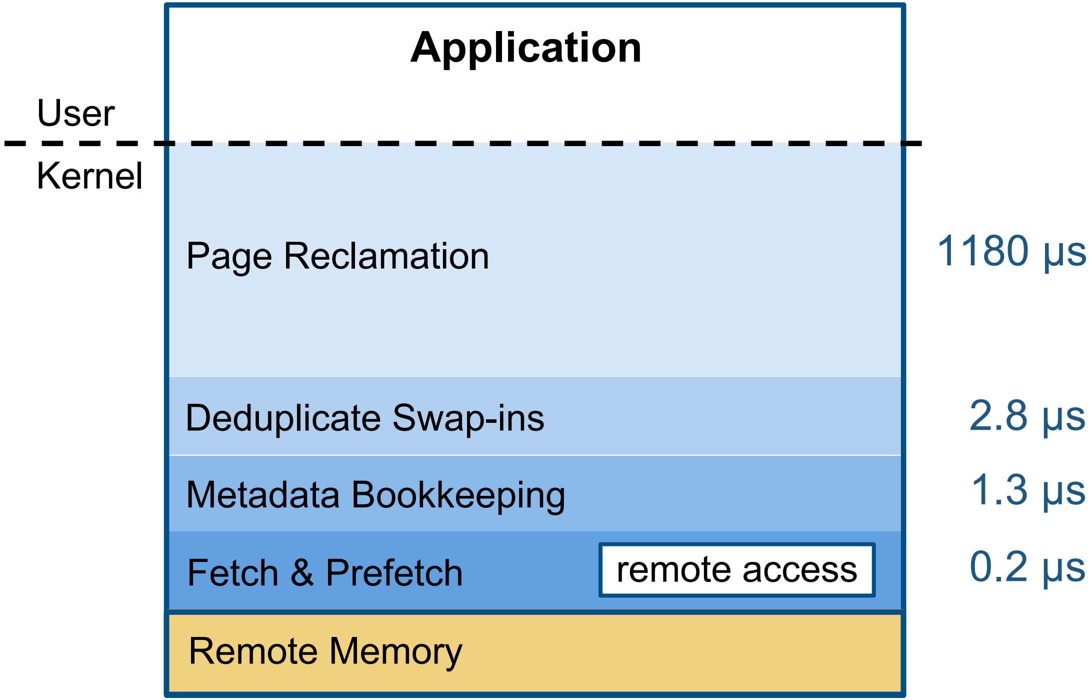


Where Does the Latency Come From?

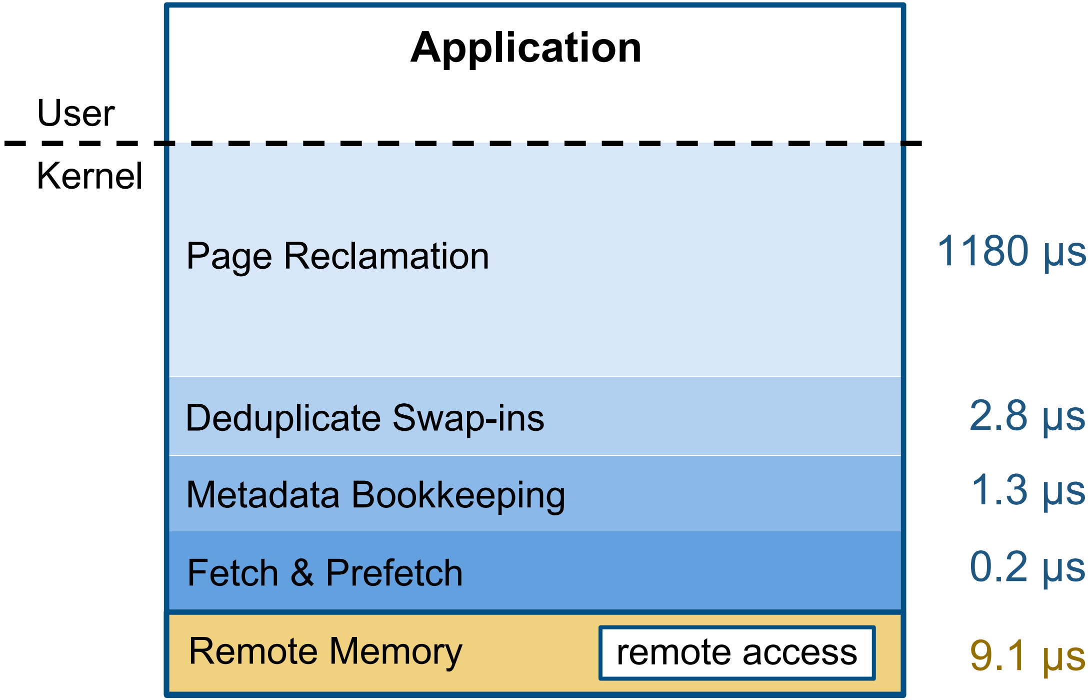


Network: Mellanox ConnectX-5 Ex (100 GbE)

Where Does the Latency Come From?

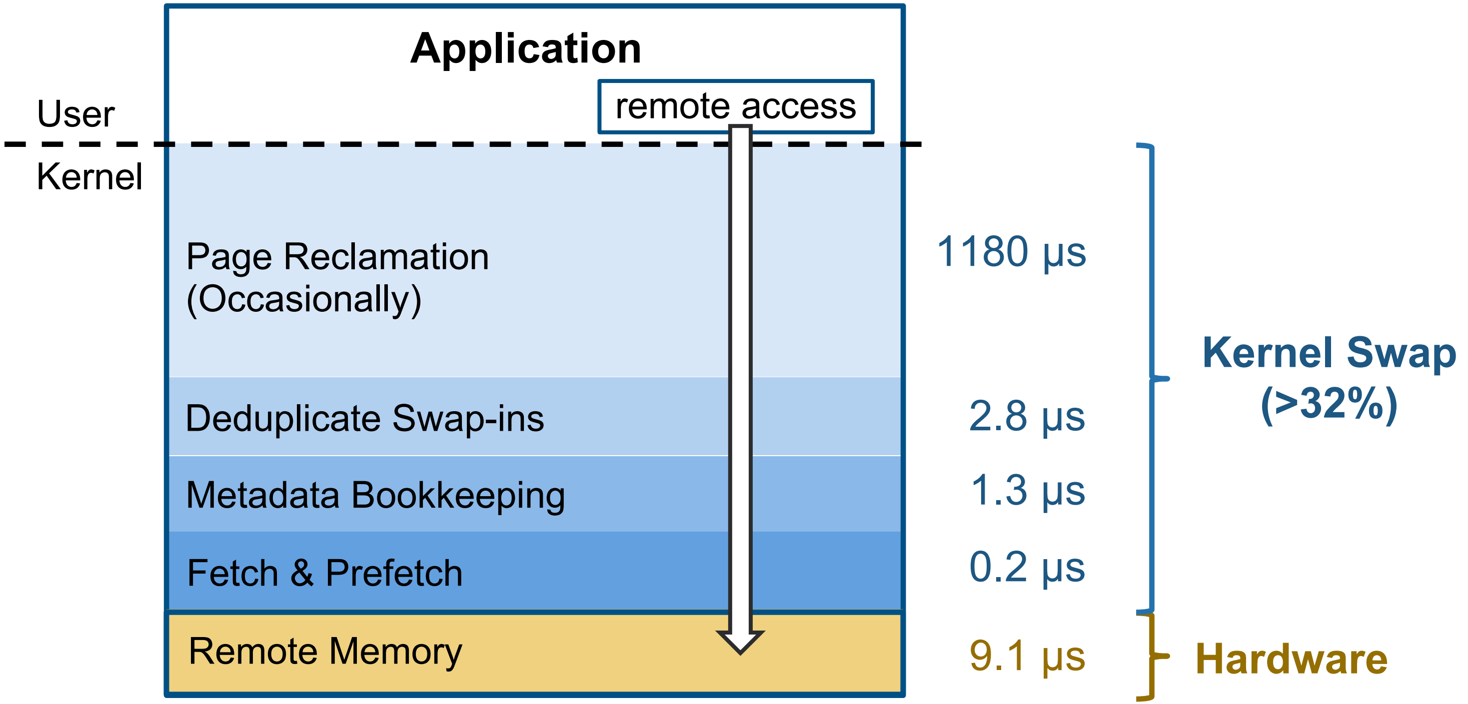


Where Does the Latency Come From?



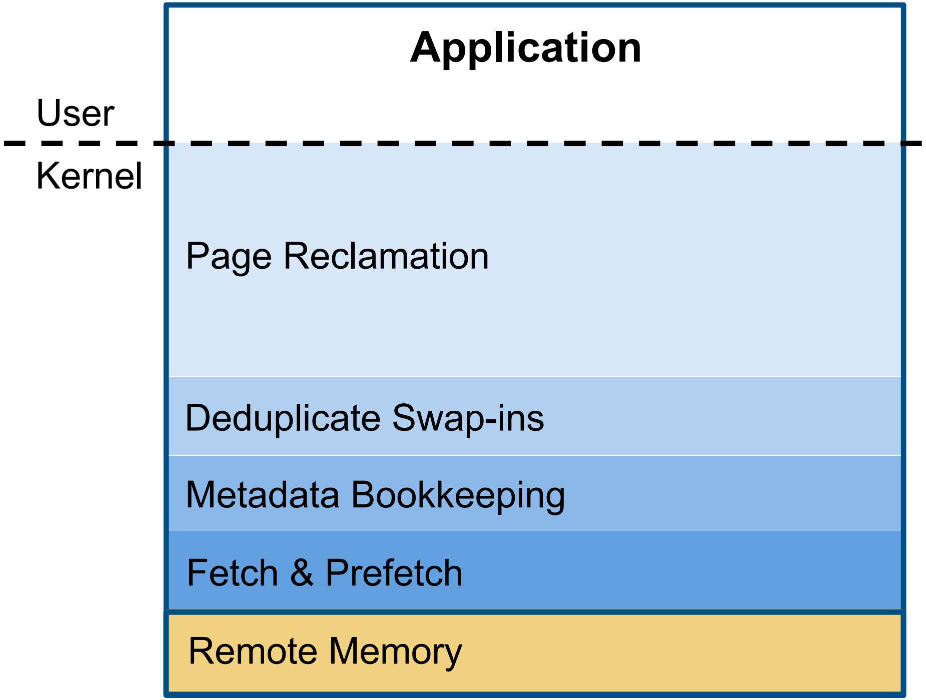
Network: Mellanox ConnectX-5 Ex (100 GbE)

Where Does the Latency Come From?

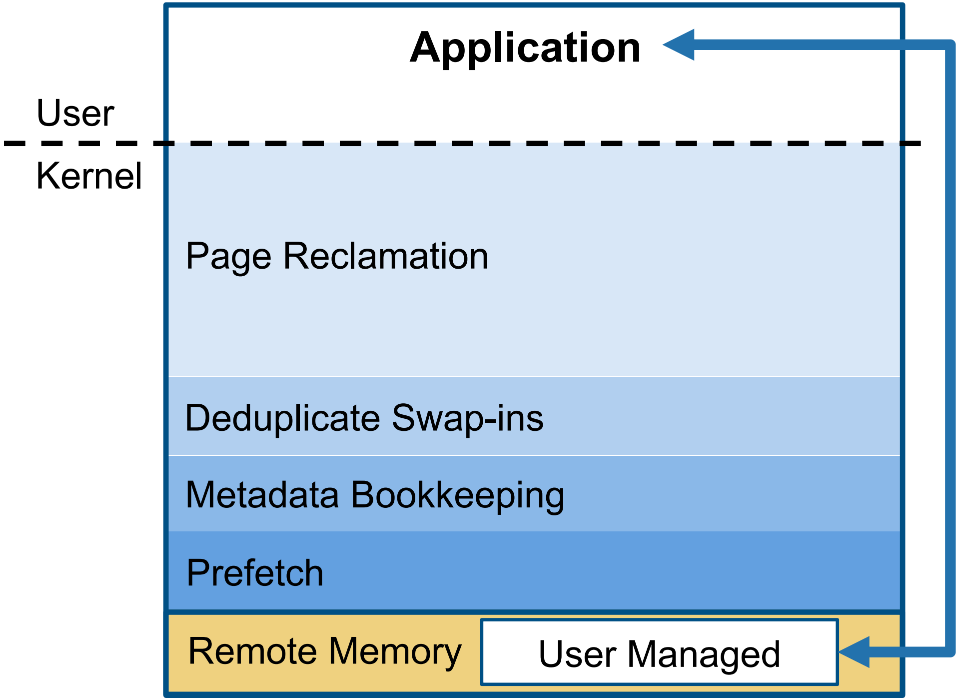


Network: Mellanox ConnectX-5 Ex (100 GbE)

How To Reduce Latency?



Kernel Bypassing Is Not a Panacea

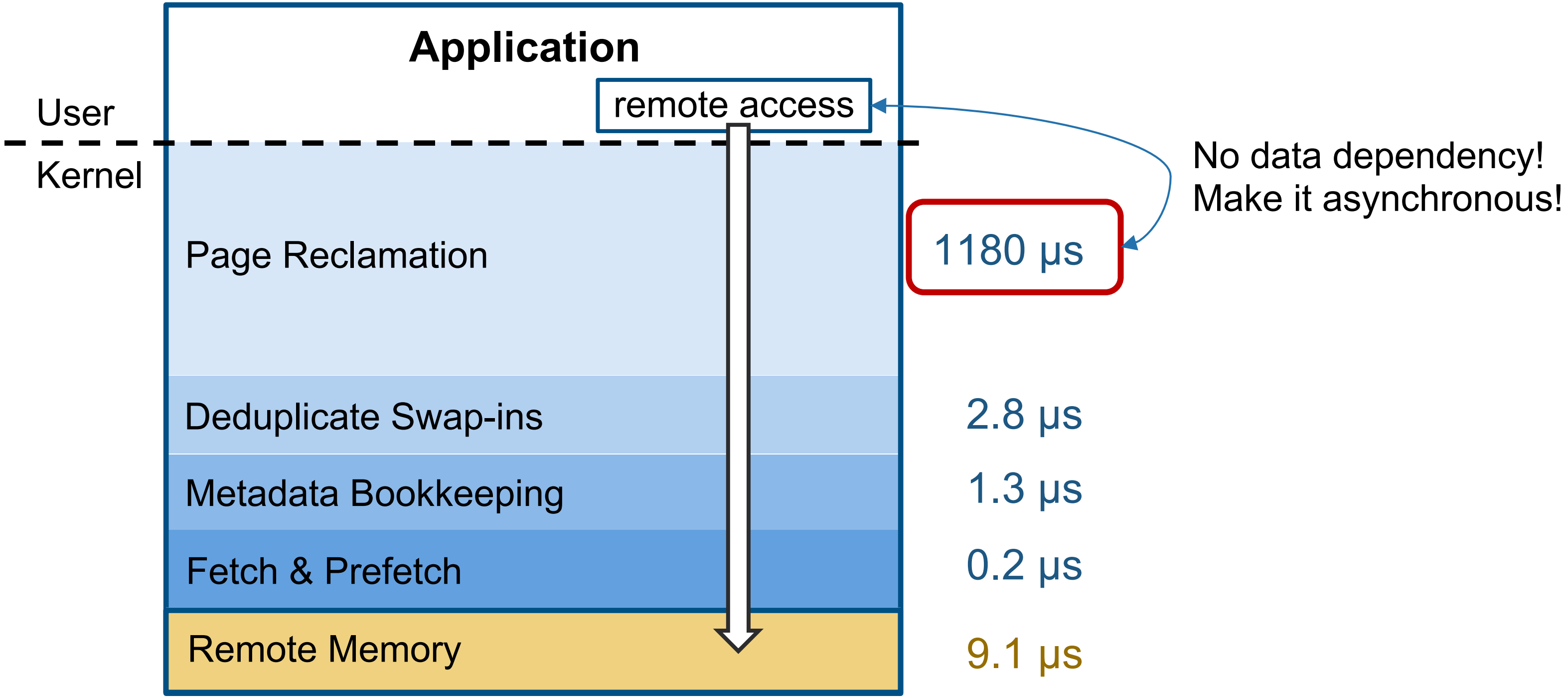


Lose Transparency

Lose kernel protection & isolation

Can we eliminate performance bottlenecks in the kernel directly?

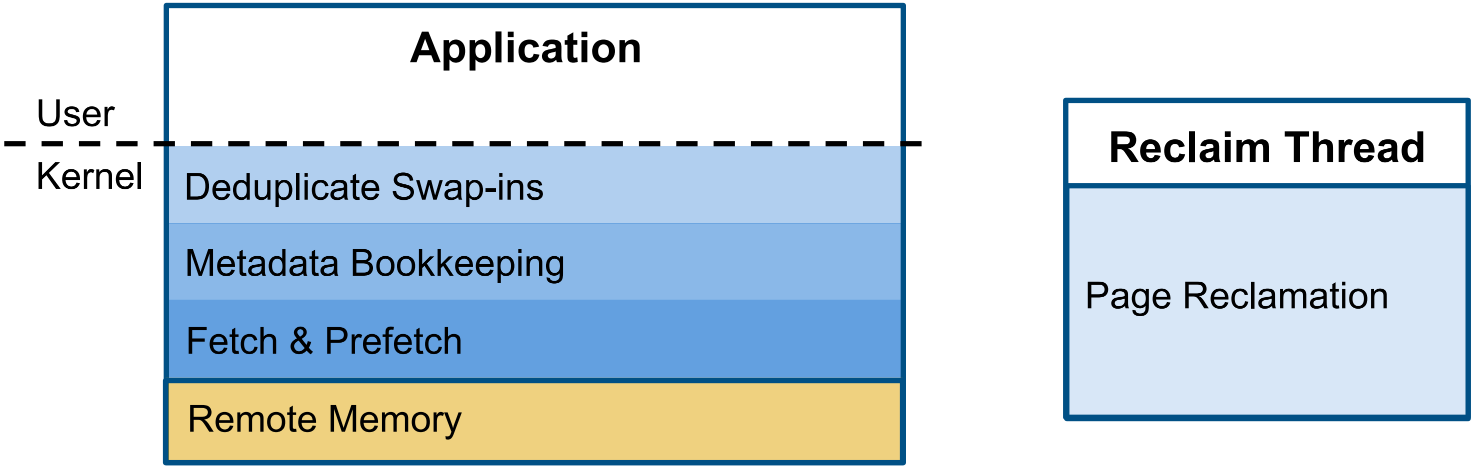
Can Asynchrony Reduce Latency?



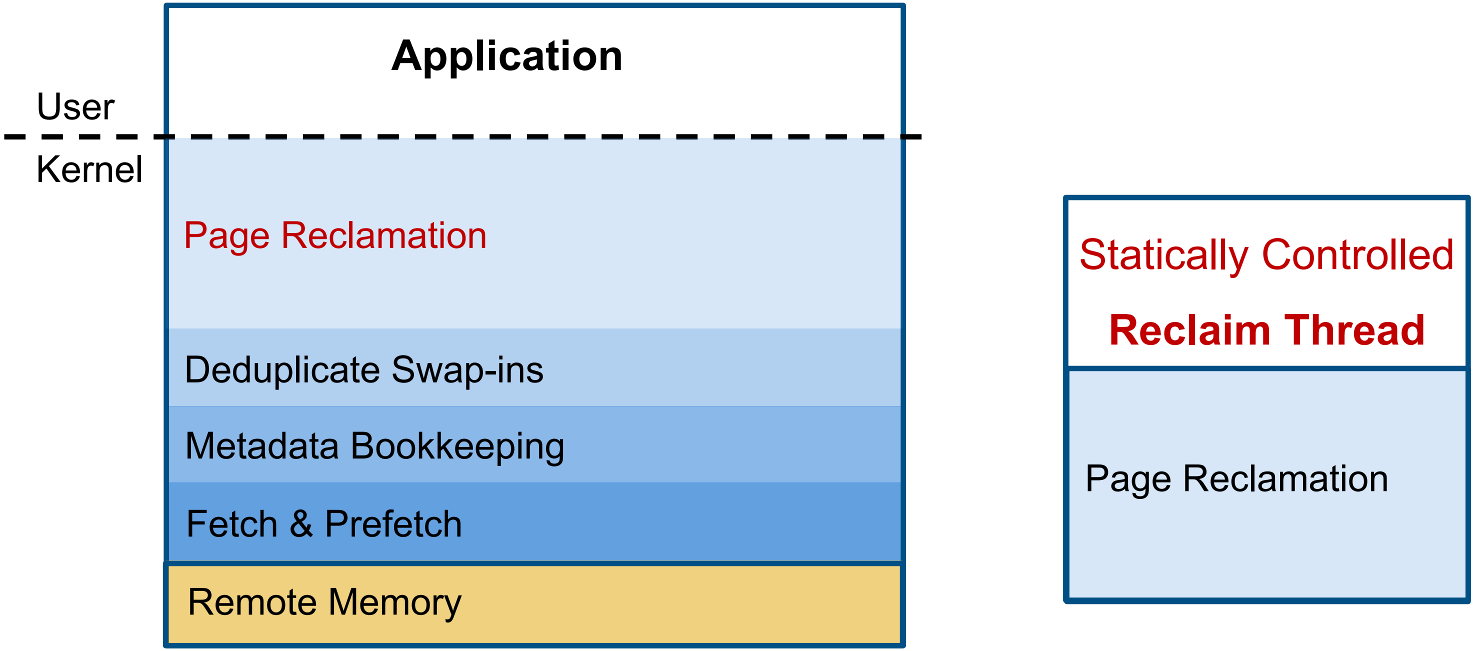
Network: Mellanox ConnectX-5 Ex (100 GbE)

Naive Asynchrony Is Not Enough

- Linux: kswapd
- Fastswap [EuroSys'20]: dedicated core

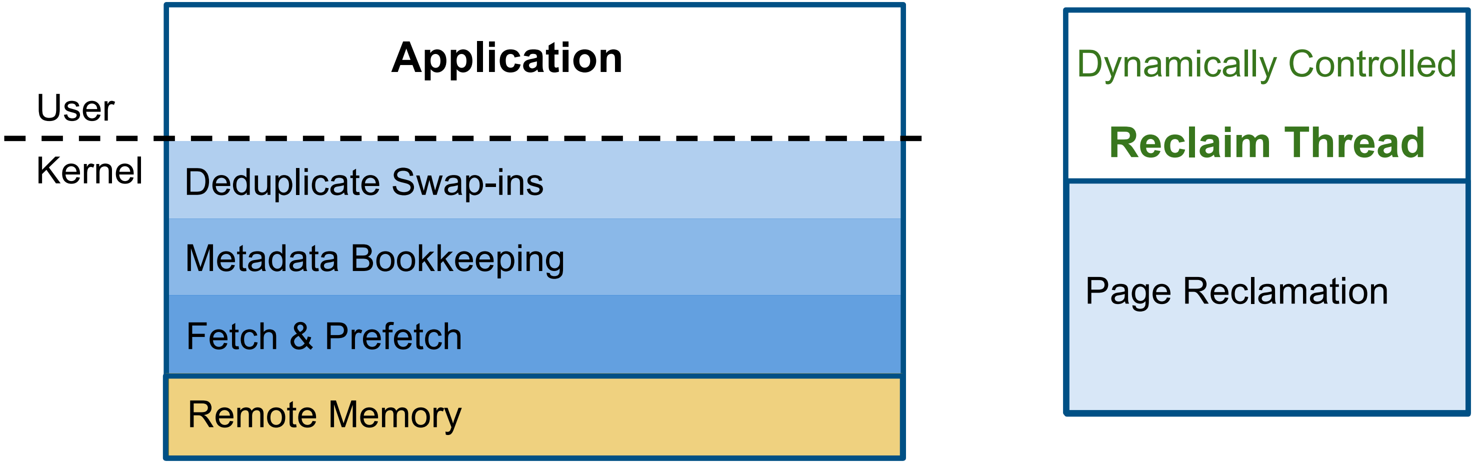


Naive Asynchrony Is Not Enough



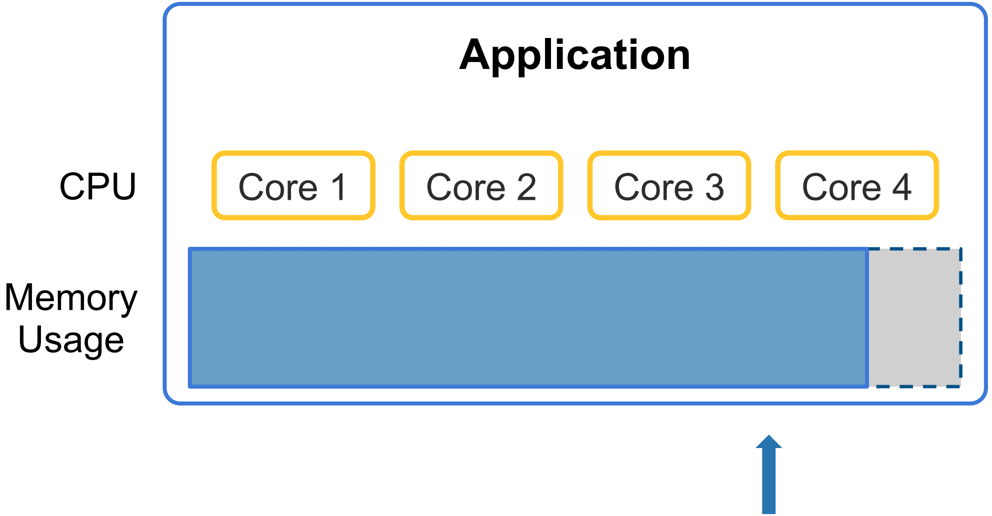
Must Have Controlled Asynchrony

- **When** to start reclamation?
- **How** many cores for reclamation?



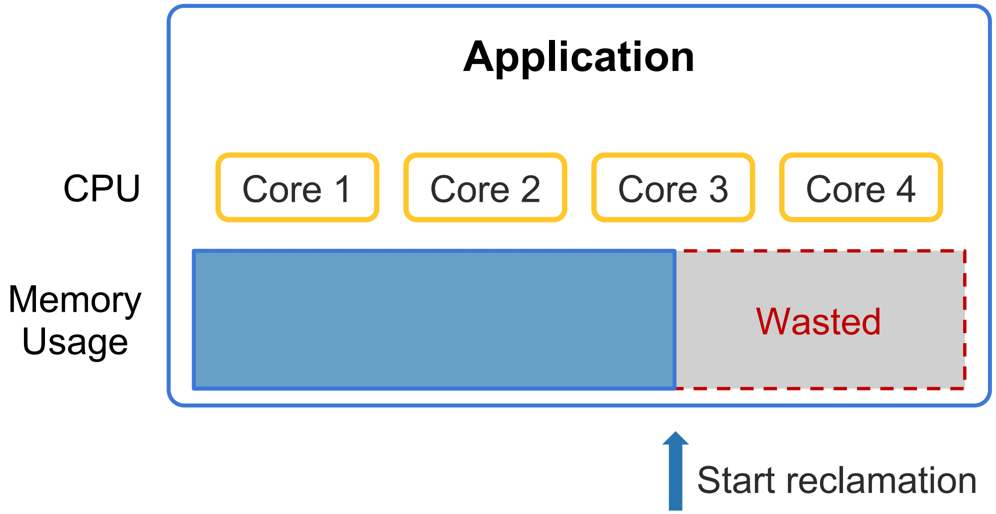
UCLA Samueli School of Engineering

Challenge #1: When To Start Reclamation



When to start reclamation?

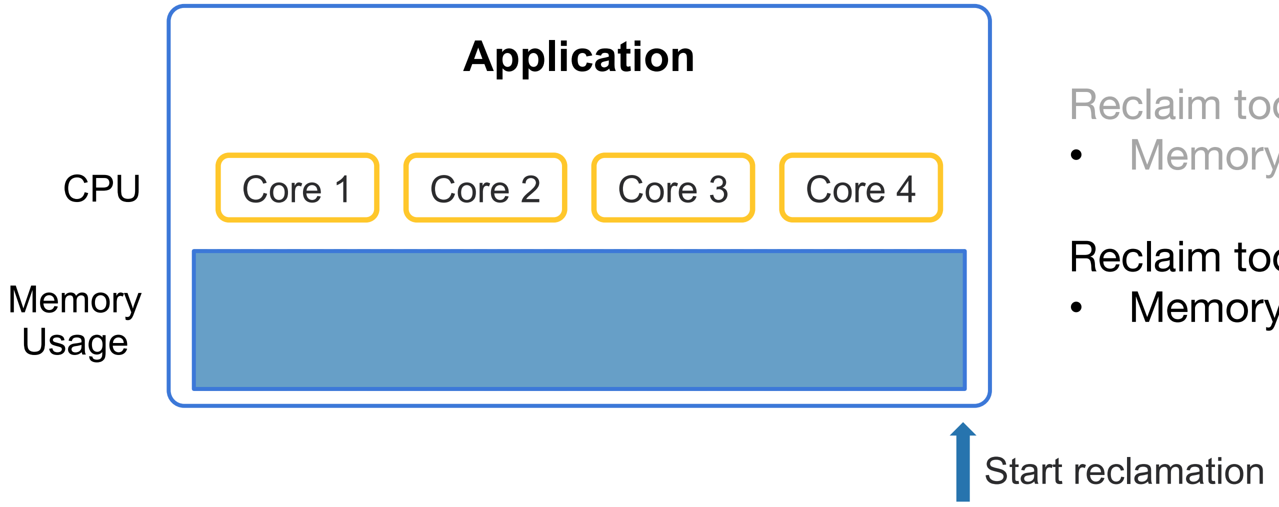
Challenge #1: When To Start Reclamation



Reclaim too early:

- Memory underutilization

Challenge #1: When To Start Reclamation



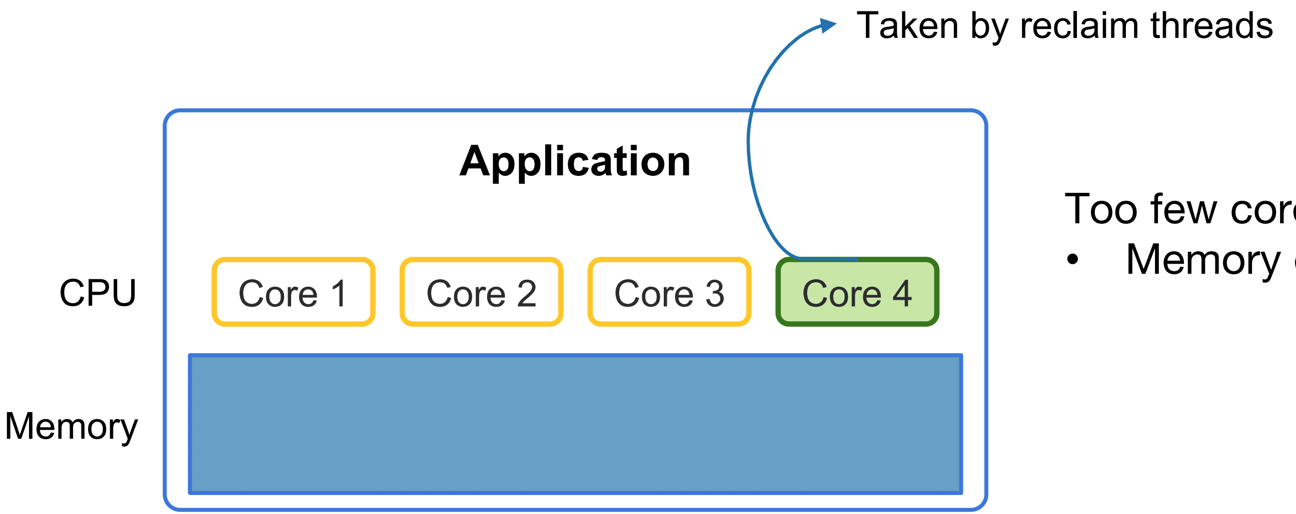
Reclaim too early:

- Memory underutilization

Reclaim too late:

- Memory exhaustion

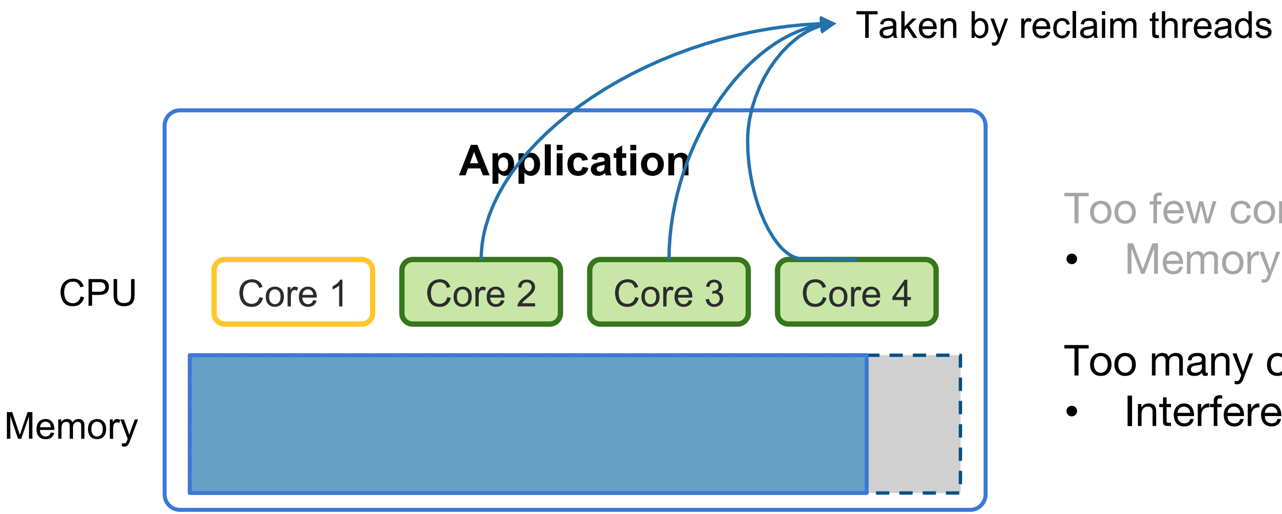
Challenge #2: How Many Cores For Reclamation



Too few cores:

- Memory exhaustion

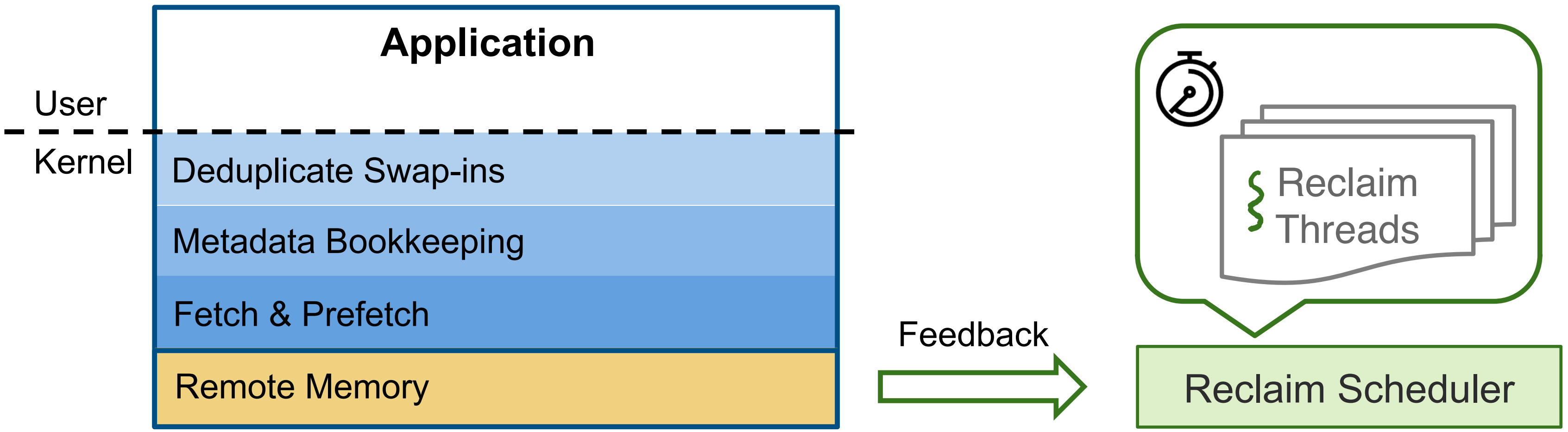
Challenge #2: How Many Cores For Reclamation



Too few cores:
- Memory exhaustion

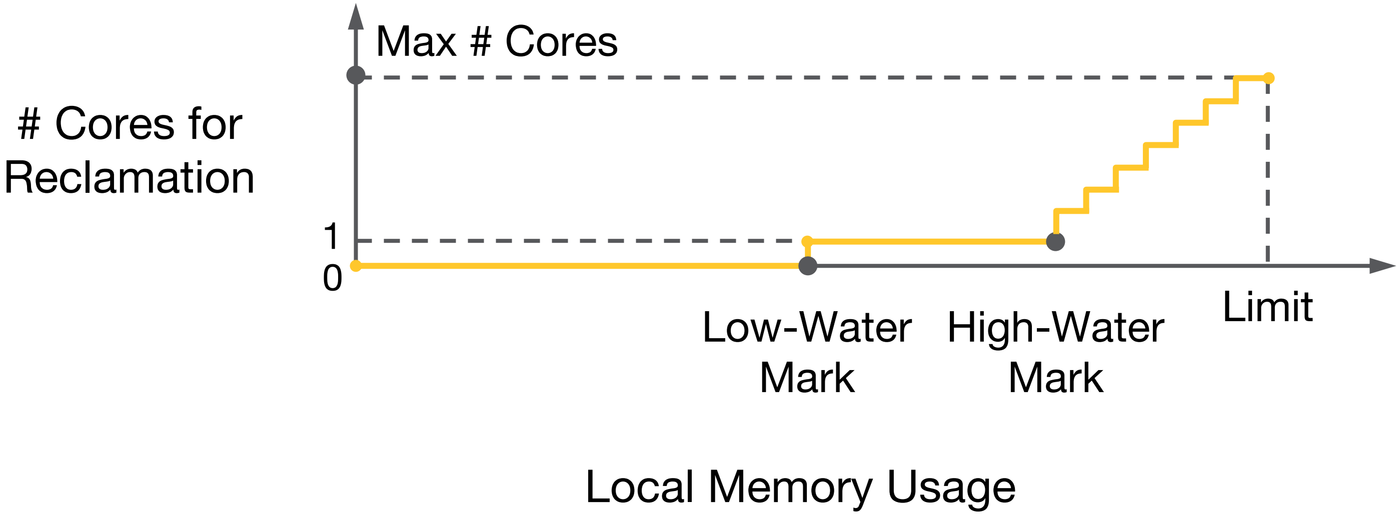
Too many cores:
- Interfere user threads

Hermit Design: Feedback-Directed Asynchrony



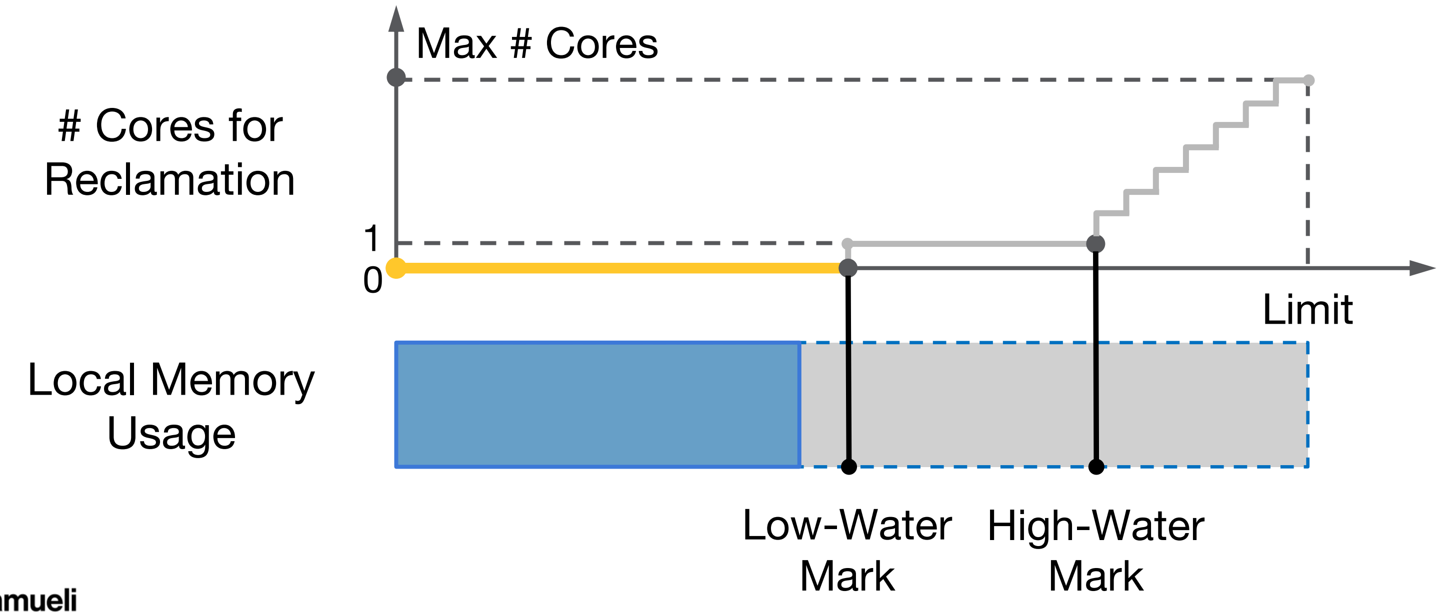
UCLA Samueli School of Engineering

Hermit's Adaptive Reclaim Scheduling



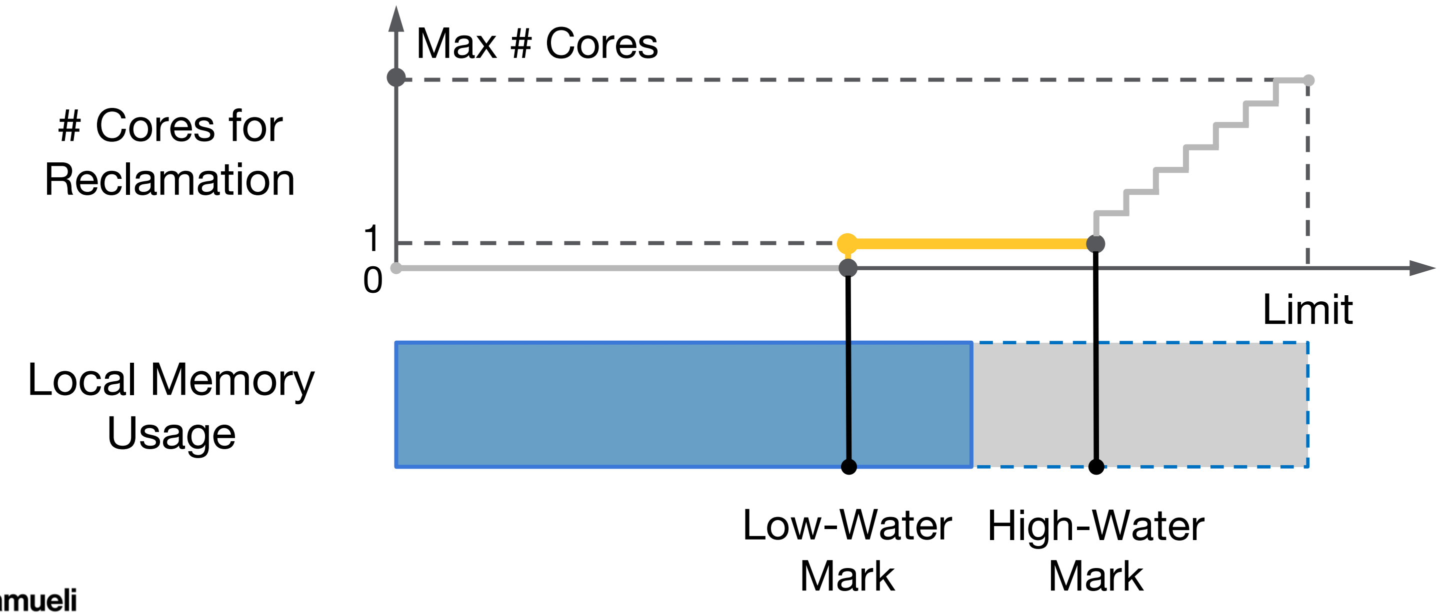
Hermit's Adaptive Reclaim Scheduling

❖ **When** to start reclamation?



Hermit’s Adaptive Reclaim Scheduling

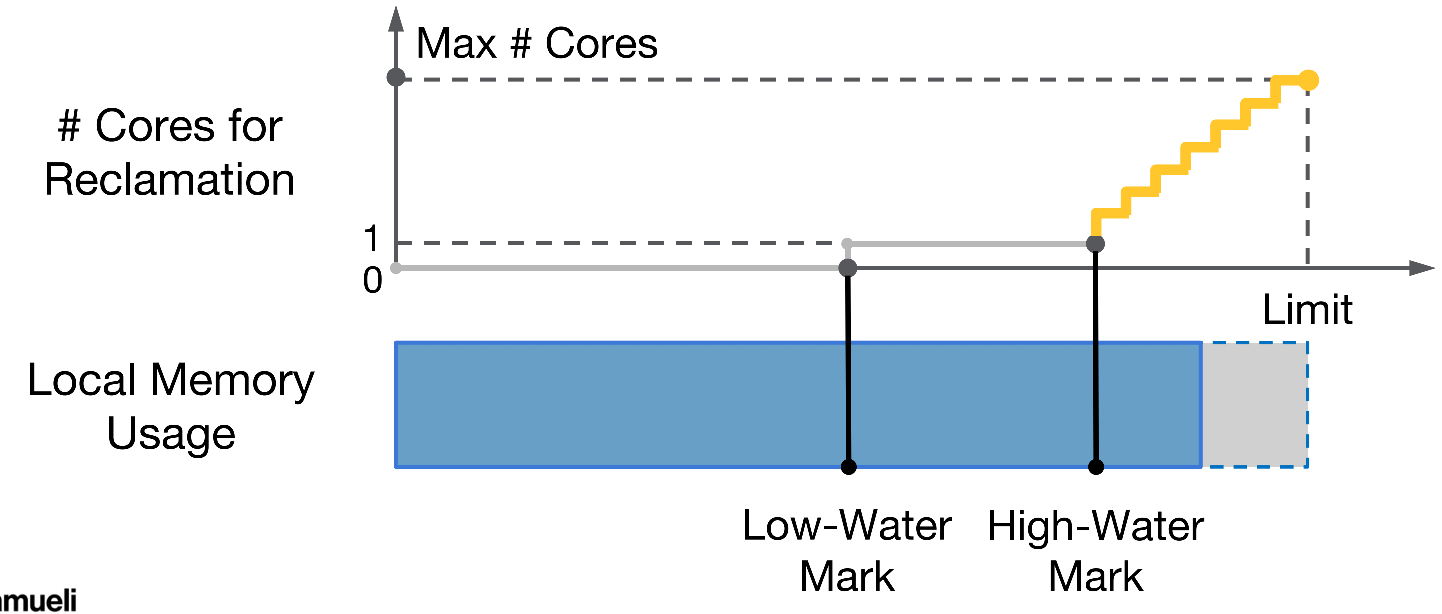
❖ **When** to start reclamation?



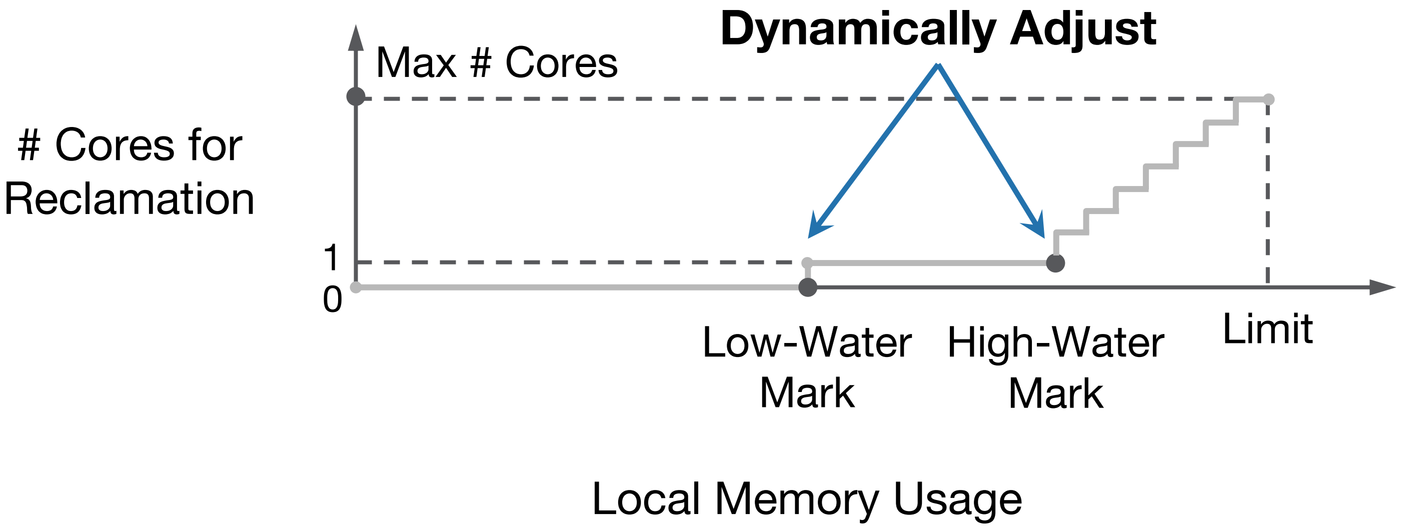
UCLA Samueli School of Engineering

Hermit's Adaptive Reclaim Scheduling

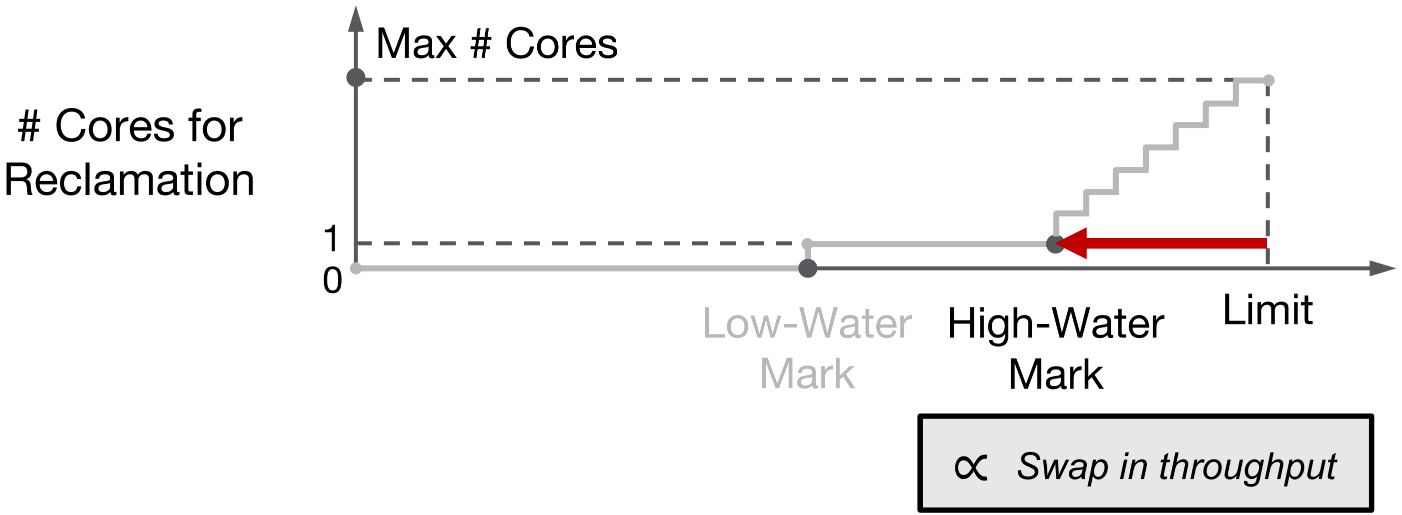
❖ **How** many cores for reclamation?



Hermit's Adaptive Reclaim Scheduling

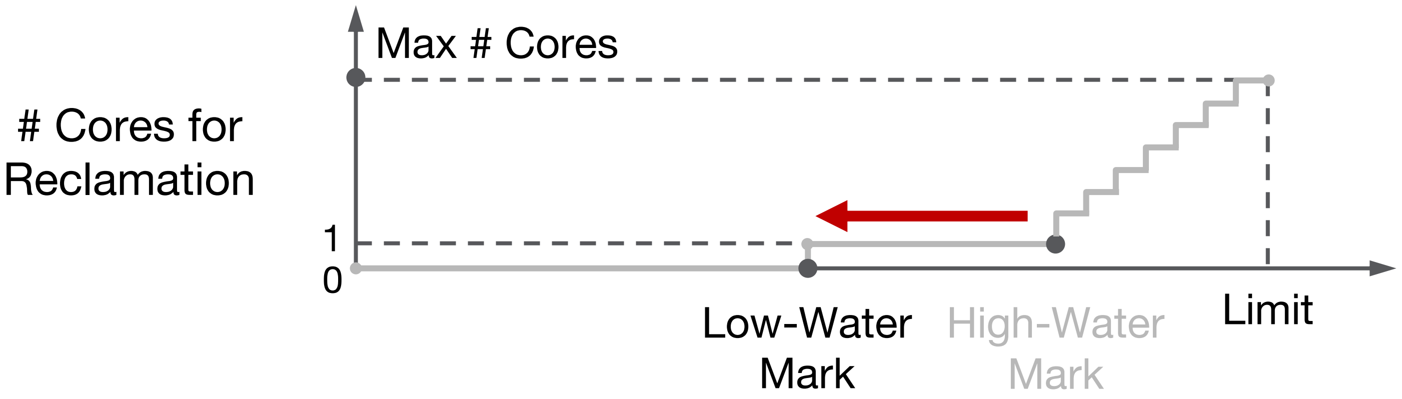


Hermit's Adaptive Reclaim Scheduling



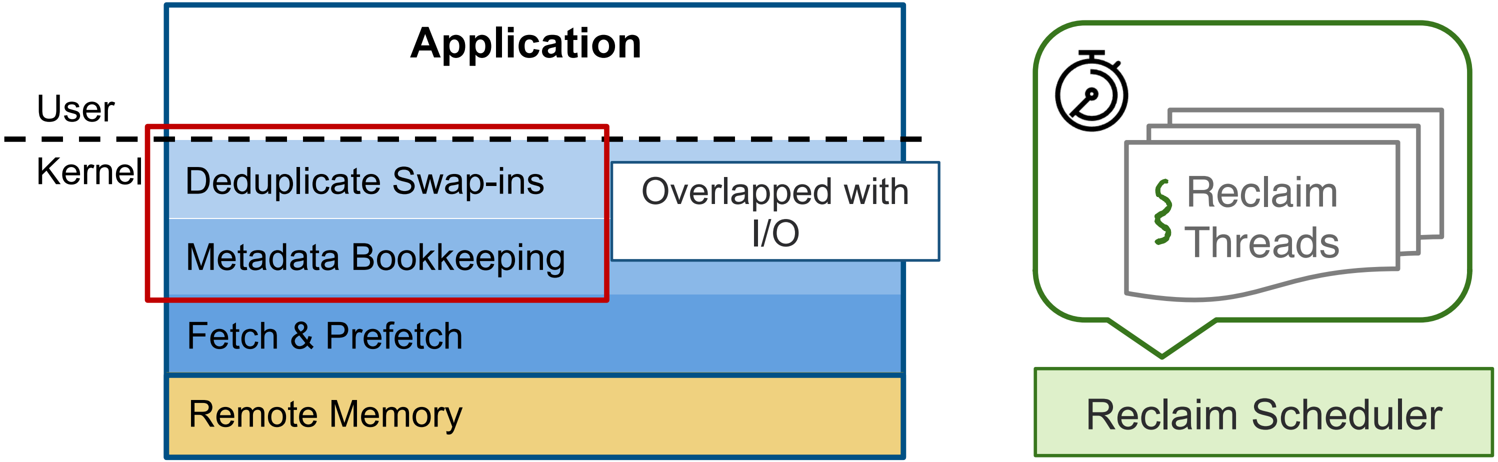
Hermit's Adaptive Reclaim Scheduling

***Page Turnaround (PT)*: how long a swapped-out page remains untouched**



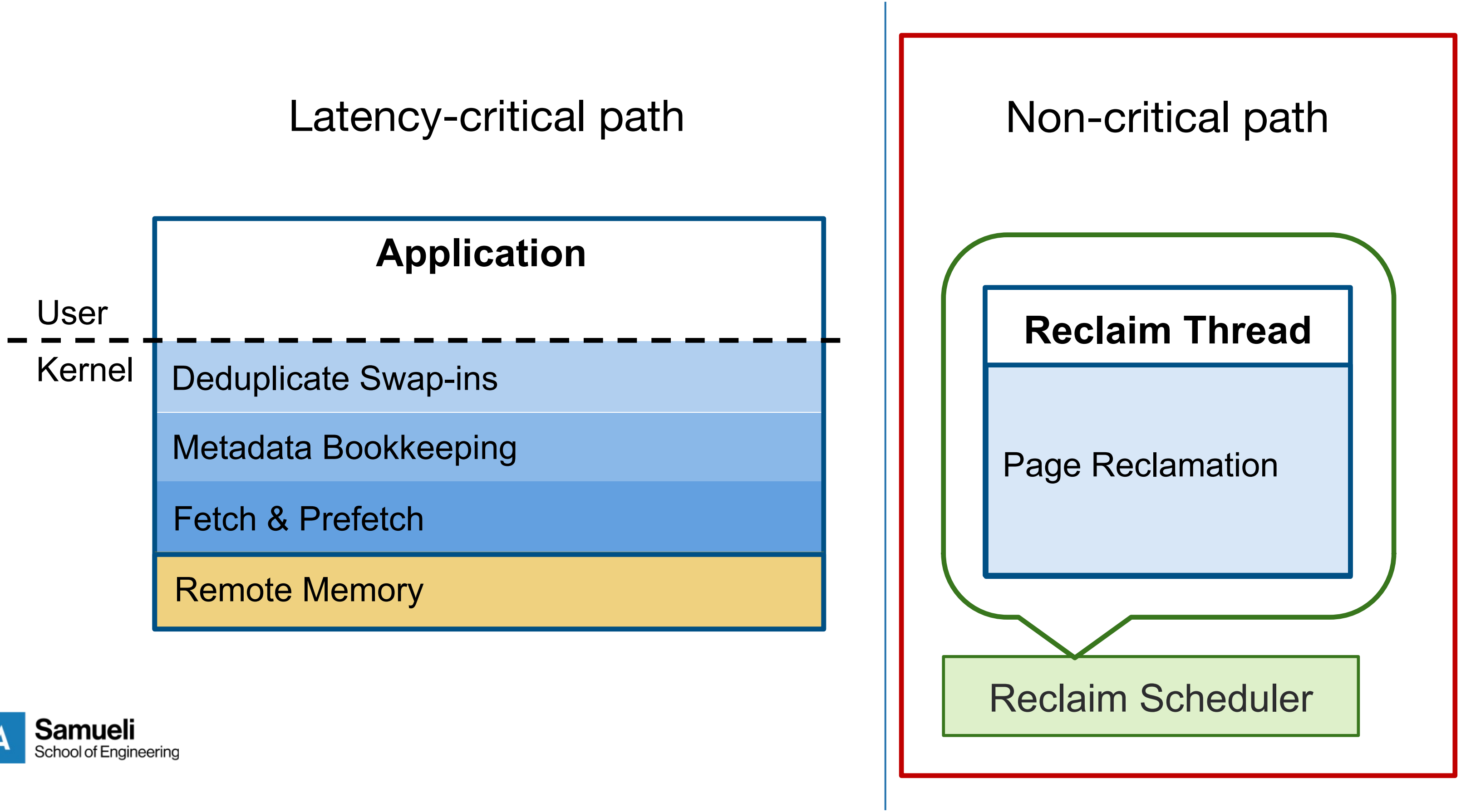
When PT remains high

Hermit Achieves Low Latency



UCLA Samueli School of Engineering

How To Improve Throughput?



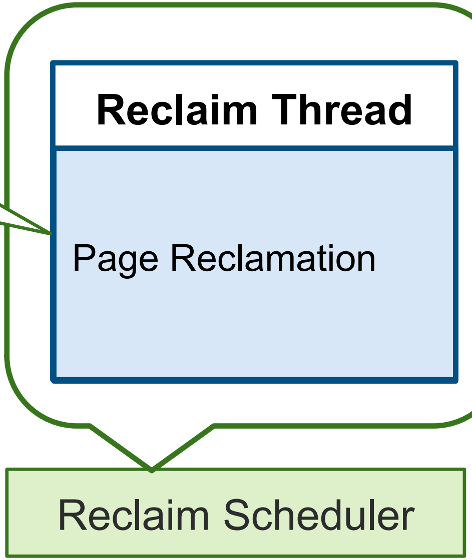
Aggressive Batching For Async. Reclamation

Batched operations for a group of pages:

- *TLB Shootdown*
- *Page I/O Write*
- *`cgroup` Accounting*
- *etc.*

2.9x more CPU efficient

Non-critical path



UCLA Samueli School of Engineering

Evaluation

Evaluated 6 real-world cloud applications with varying local memory ratios

- **Latency-Critical**: Memcached, SocialNet, Gdnsd
- **Batch-Processing**: Spark, XGBoost, Cassandra

State of the art: Fastswap [EuroSys'20]

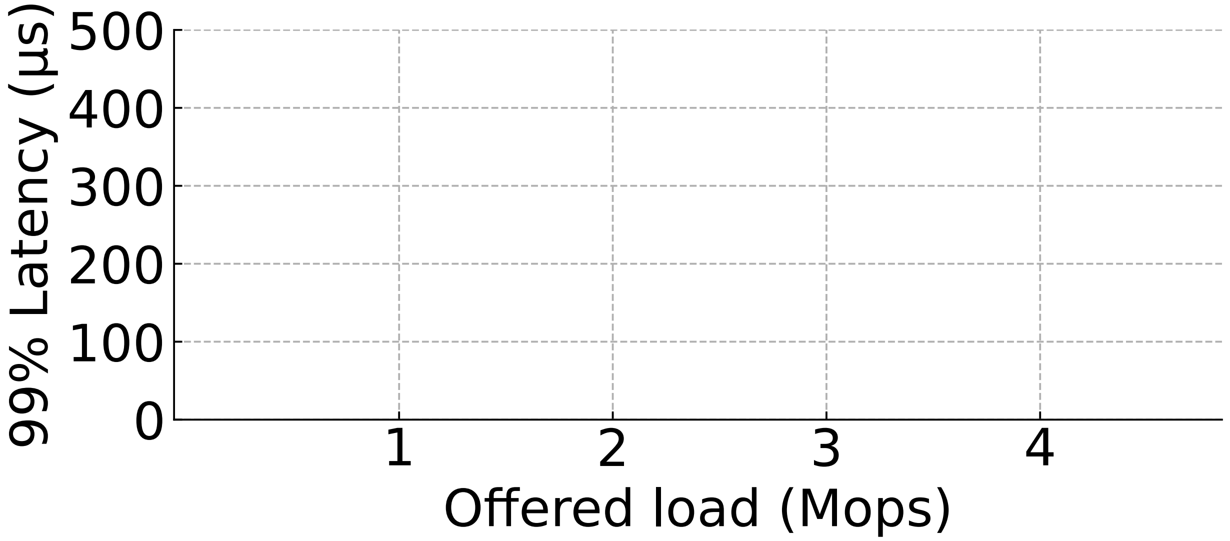
- Offload page reclamation to a single dedicated core

➢ How does Hermit maintain low end-to-end tail latency?

➢ How does Hermit improve application throughput?

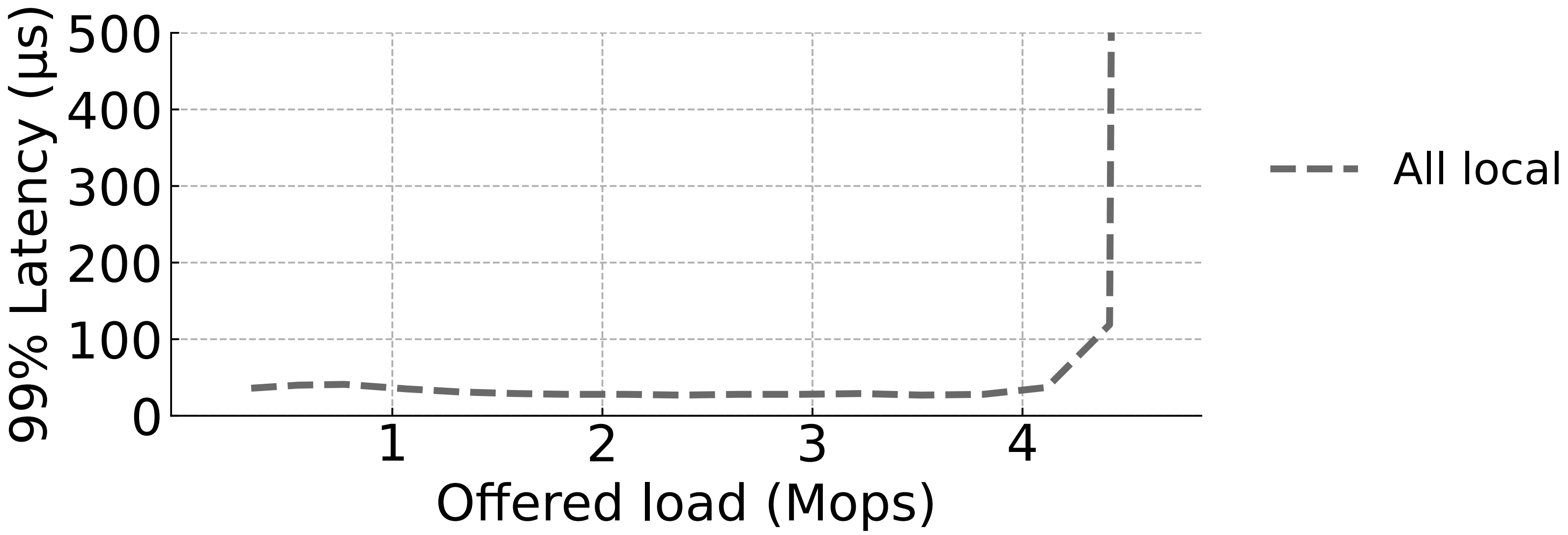
Low Tail Latency

Memcached with Facebook USR workload.



Low Tail Latency

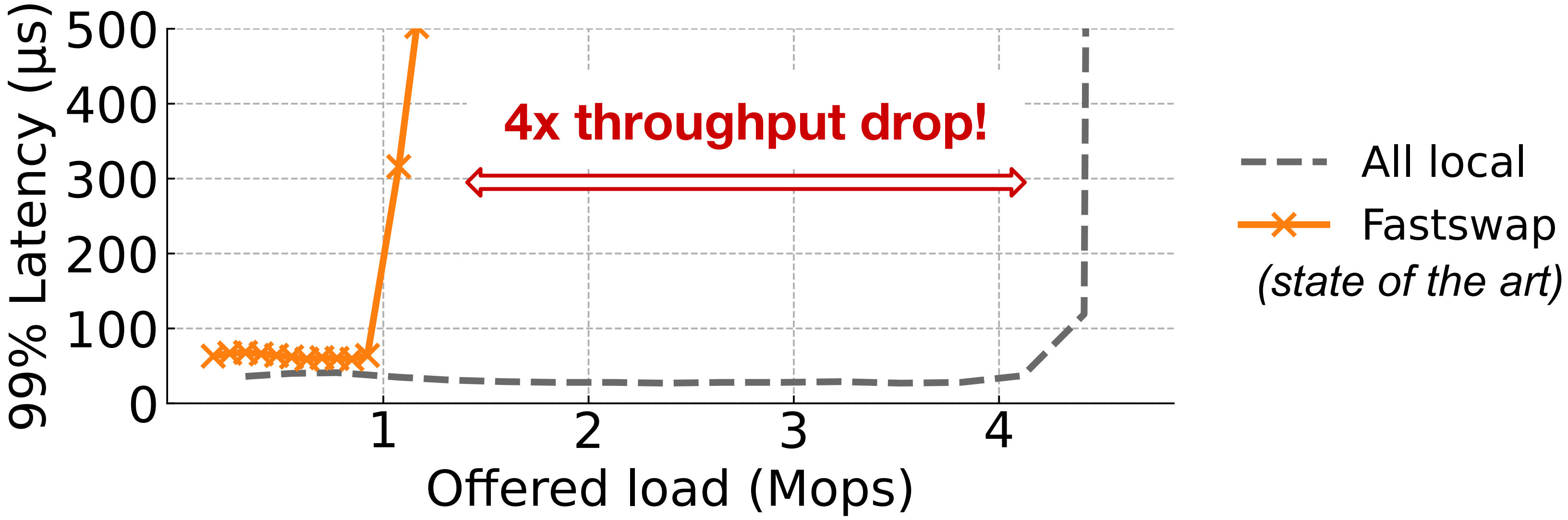
Memcached with Facebook USR workload.



Low Tail Latency

Memcached with Facebook USR workload.

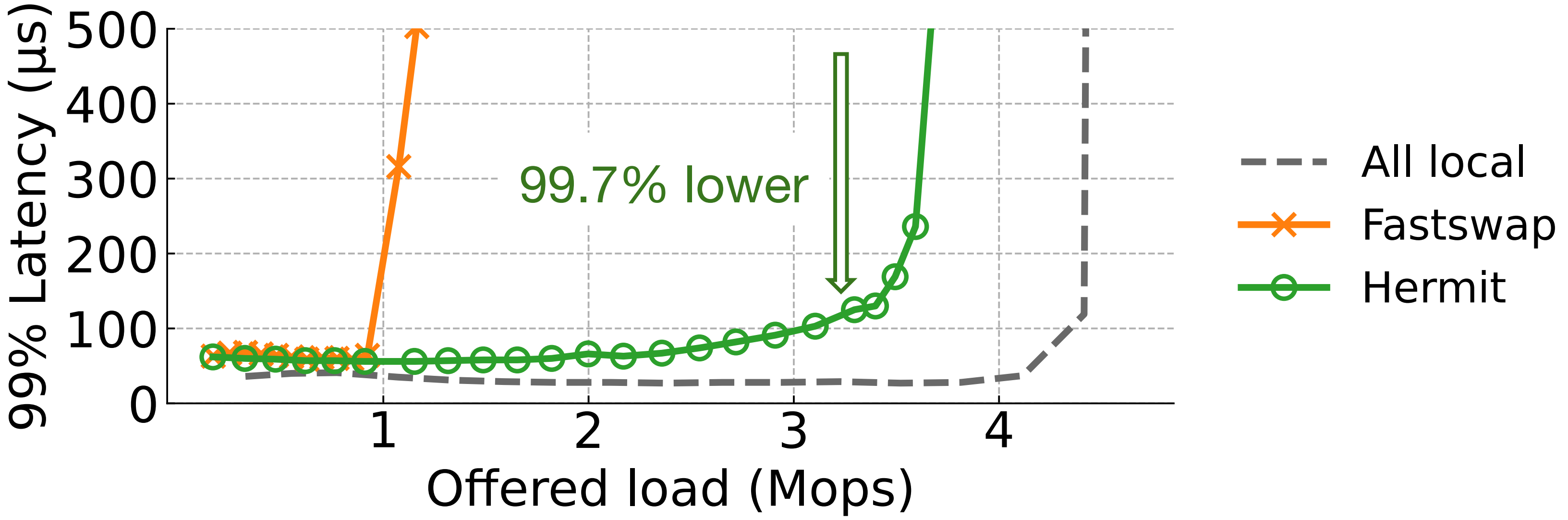
❖ Cache 70% of working set in local memory (i.e., 30% in remote memory).



Low Tail Latency

Memcached with Facebook USR workload.

❖ Cache 70% of working set in local memory (i.e., 30% in remote memory).

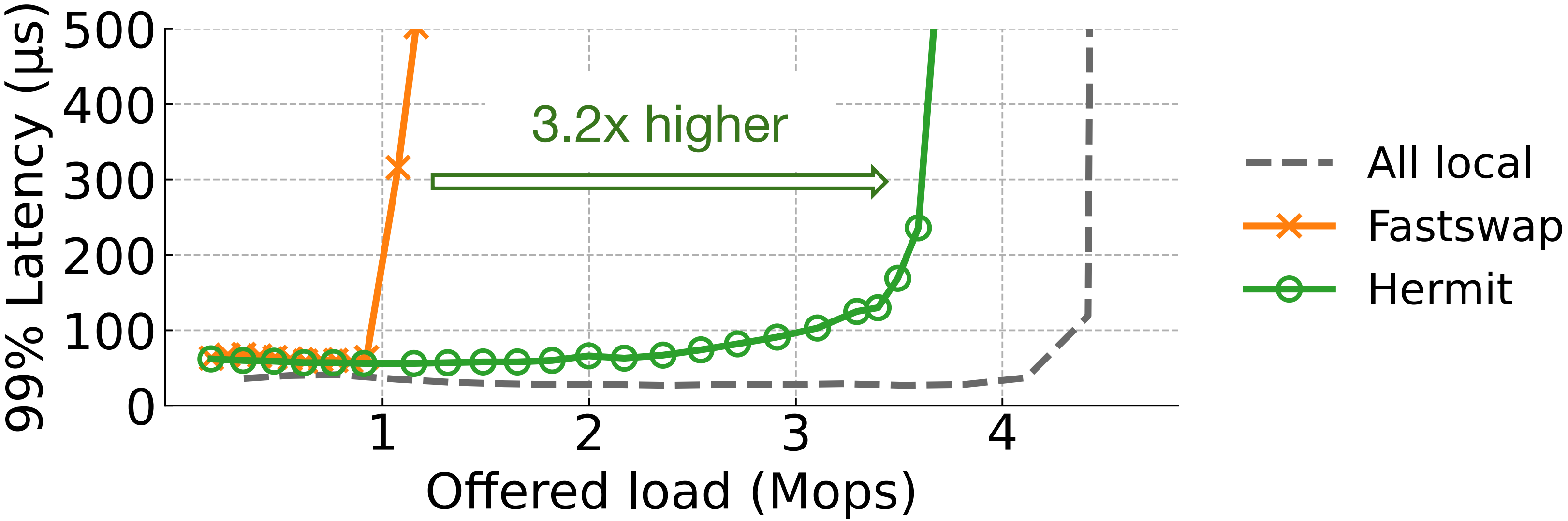


UCLA Samueli School of Engineering

High Throughput

Memcached with Facebook USR workload.

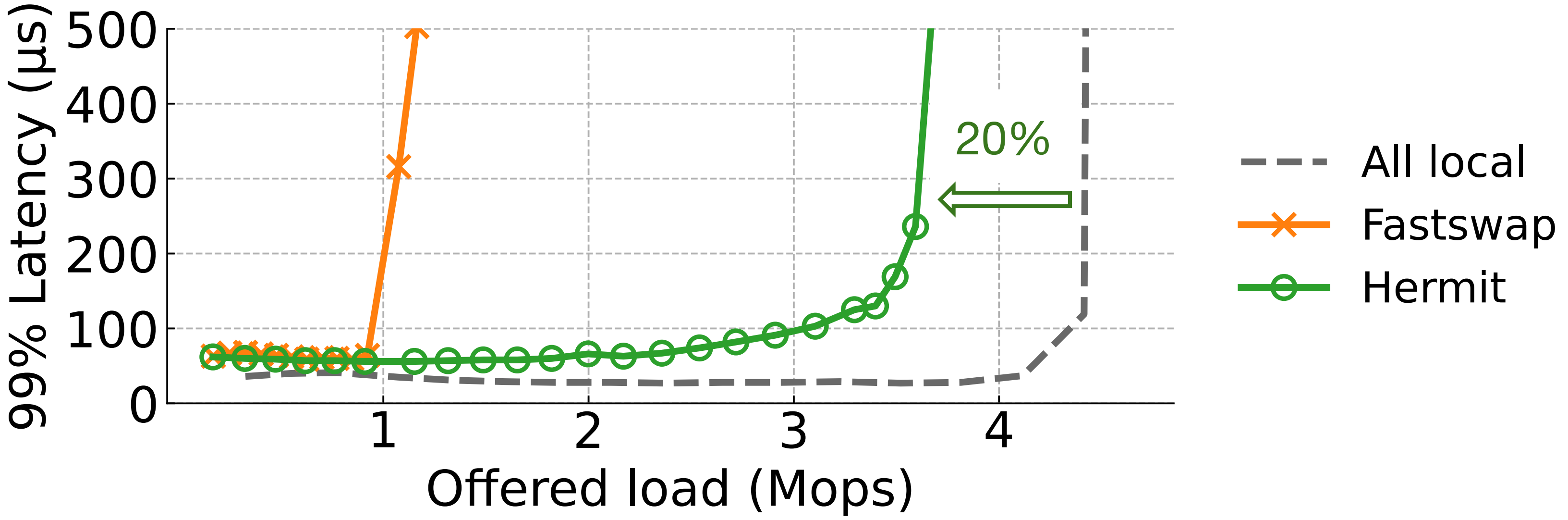
❖ Cache 70% of working set in local memory (i.e., 30% in remote memory).



High Throughput

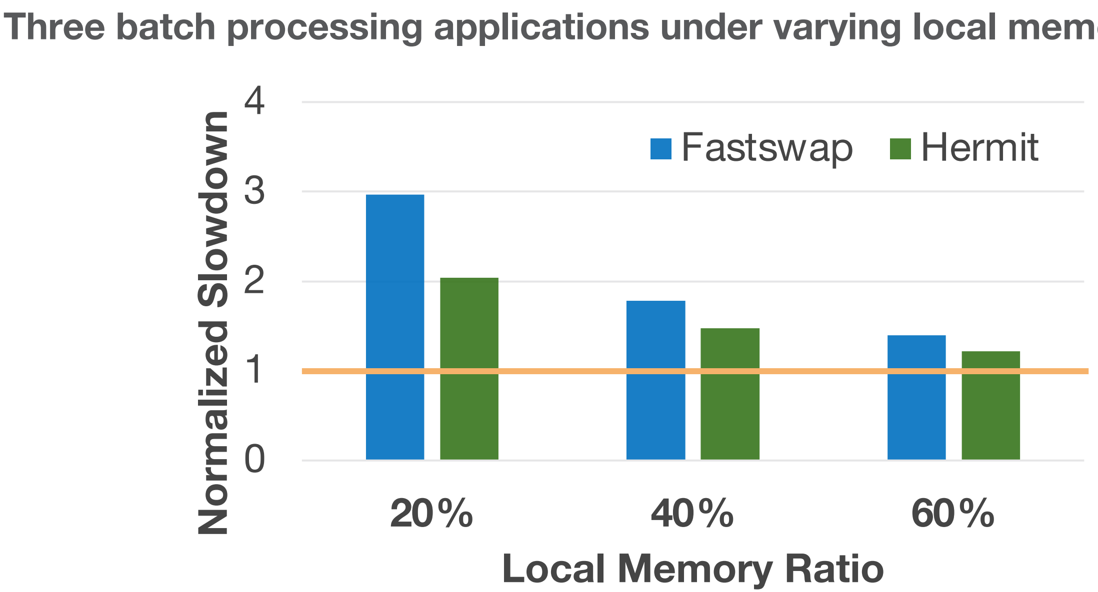
Memcached with Facebook USR workload.

❖ Cache 70% of working set in local memory (i.e., 30% in remote memory).



UCLA Samueli School of Engineering

High Throughput For Batch Applications



Hermit offers 1.24x higher throughput (up to 1.87x)

UCLA Samueli School of Engineering

Conclusion

Low latency, high throughput, and transparency can be achieved simultaneously!

- ❖ Asynchrony reduces latency and improves throughput
- ❖ Feedback loop is critical to the effect of asynchrony
- ❖ Design can be generalized to other kernel components such as page migration for CXL-attached memory

- ❖ Hermit offers up to 99.7% lower latency and 1.24x higher throughput without changing a single line of user code

https://github.com/uclasystem/hermit

Thank You!

UCLA Samueli School of Engineering