

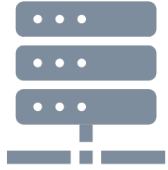


A Tale of Two Paths: Toward a Hybrid Data Plane for Efficient Far-Memory Applications

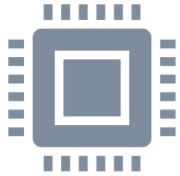
Lei Chen*, Shi Liu* (co-first author), Chenxi Wang, Haoran Ma, Yifan Qiao,
Zhe Wang, Chenggang Wu, Youyou Lu, Xiaobing Feng, Huimin Cui, Shan Lu, Harry Xu



Memory Challenges in Data Centers



Memory-intensive Workloads

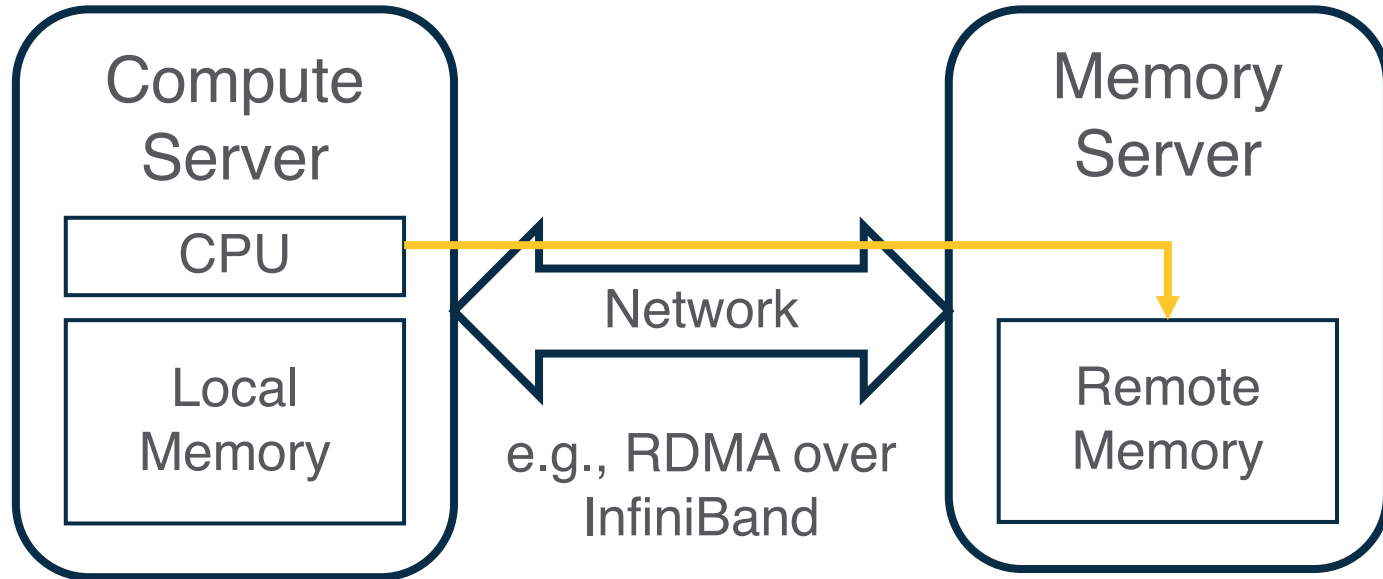


Computation-memory Imbalance

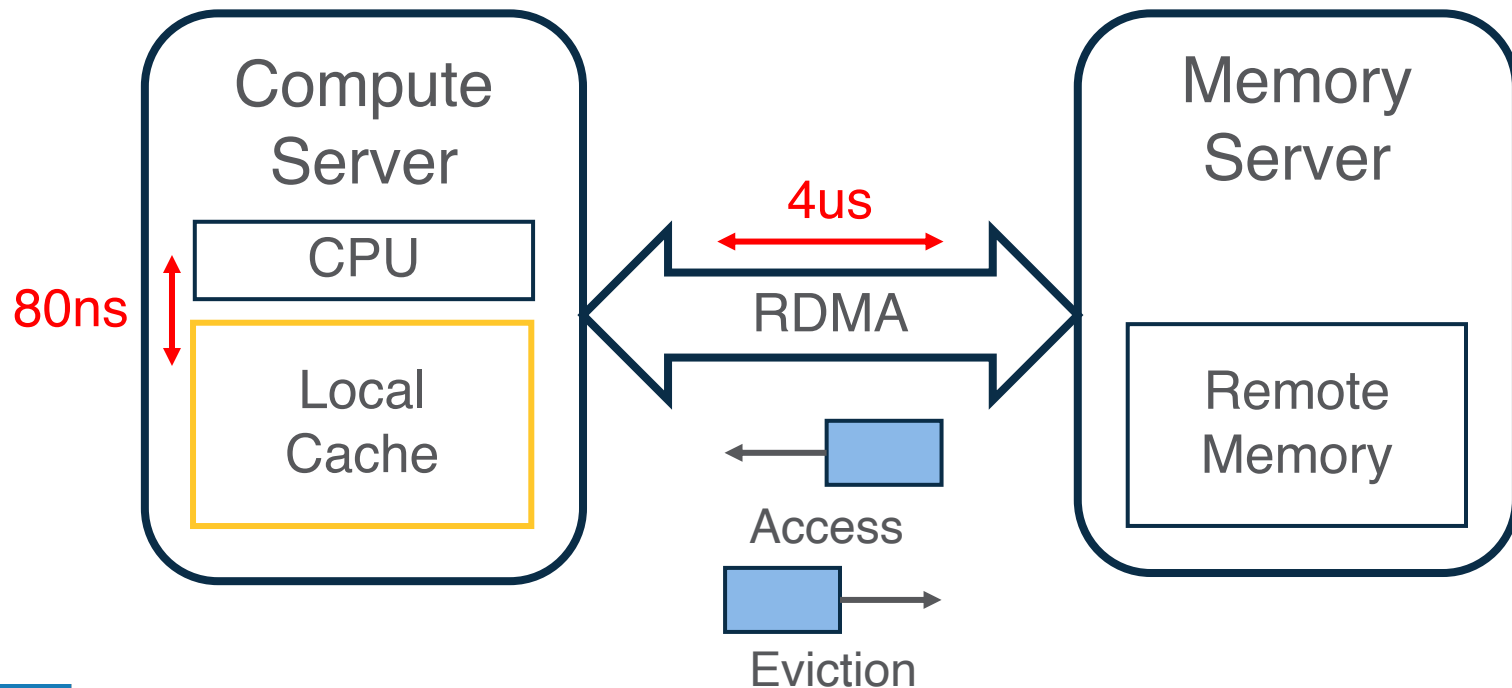


Memory Under-utilization

Remote Memory



Performance Gap



Two Types of Data Paths

Different data transfer granularities

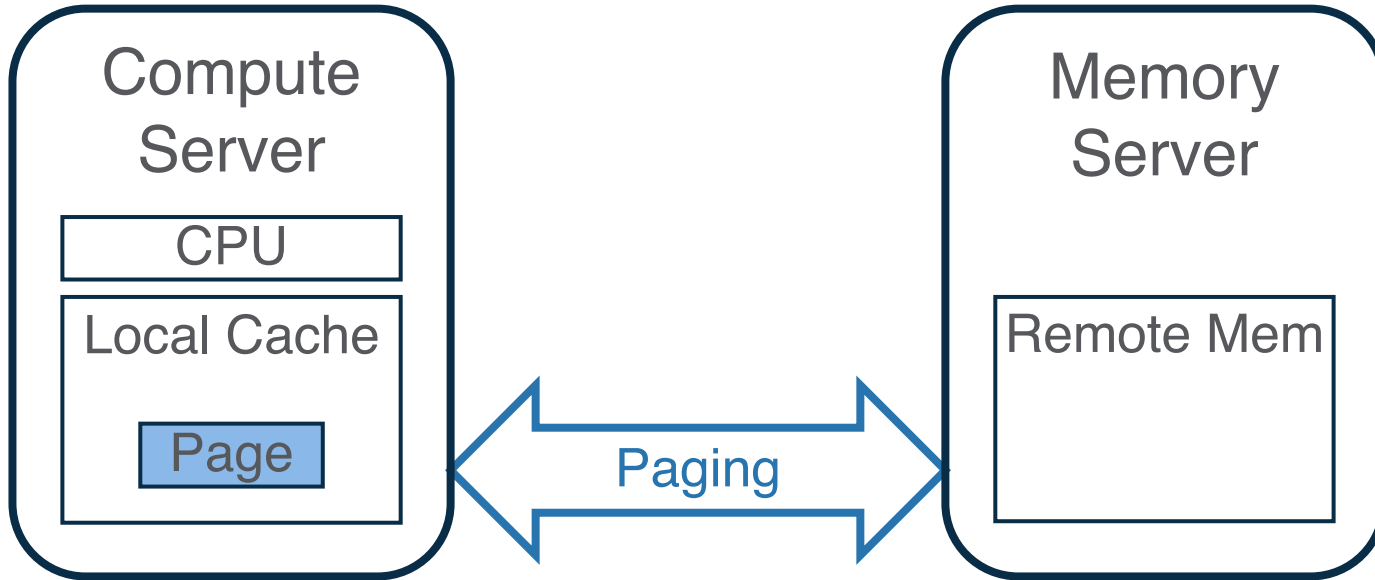


Page
(4 KB)



Object
(e.g., 64 B)

Paging Path



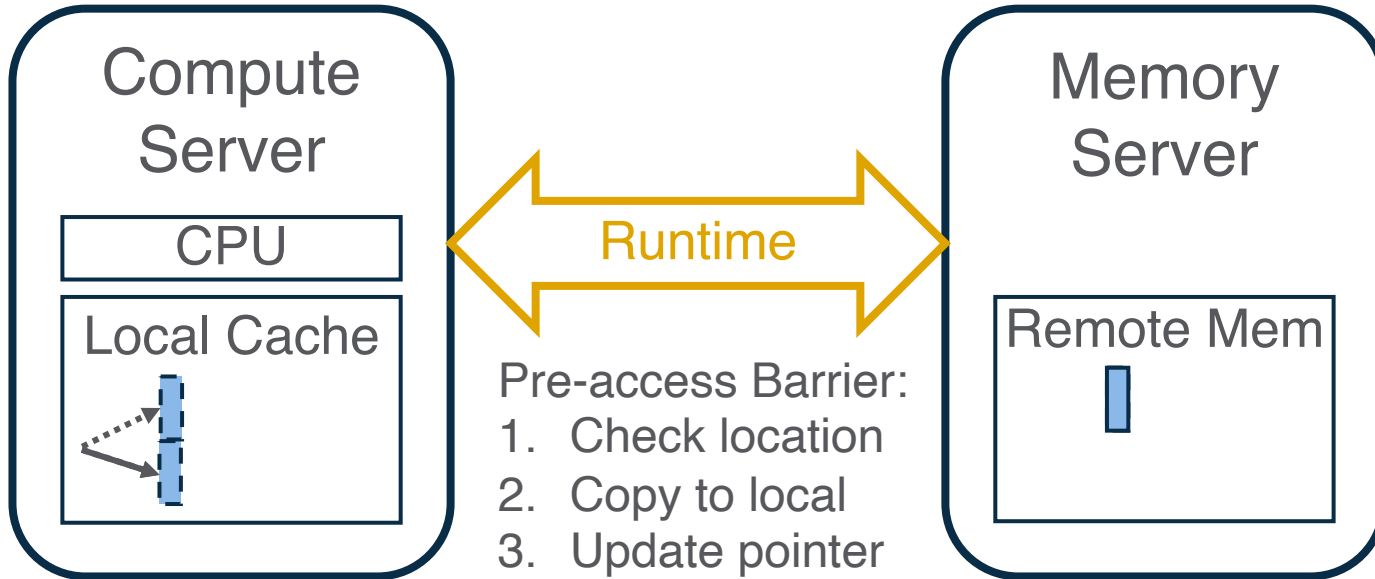
Pros:

- + No runtime overhead
- + High throughput

Cons:

- I/O Amplification

Runtime Path



Pros:

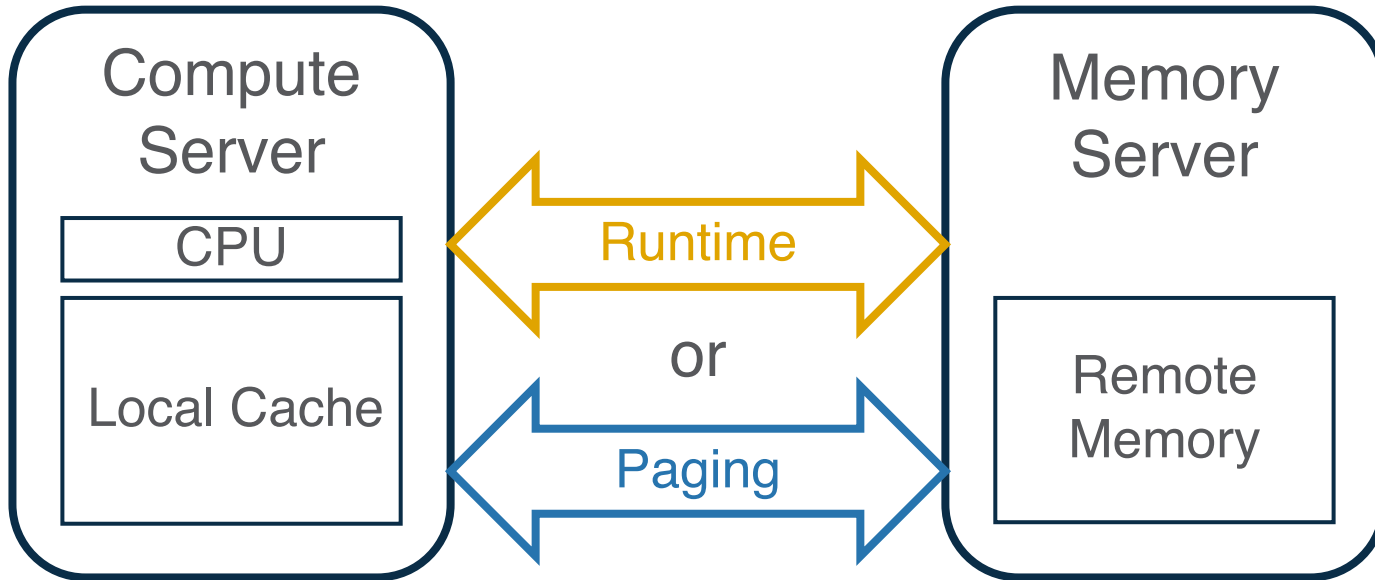
+ No I/O amplification

Cons:

- High runtime overhead

- Low throughput

Existing Systems Only Support One Path

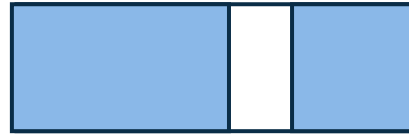


One path cannot fit all!

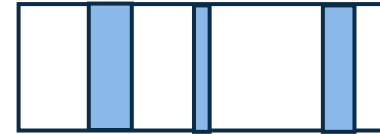
Locality

- Locality: Proportion of a memory block accessed in a short time

Locality



Good Locality



Bad Locality

Exhibited by

Sequential Access

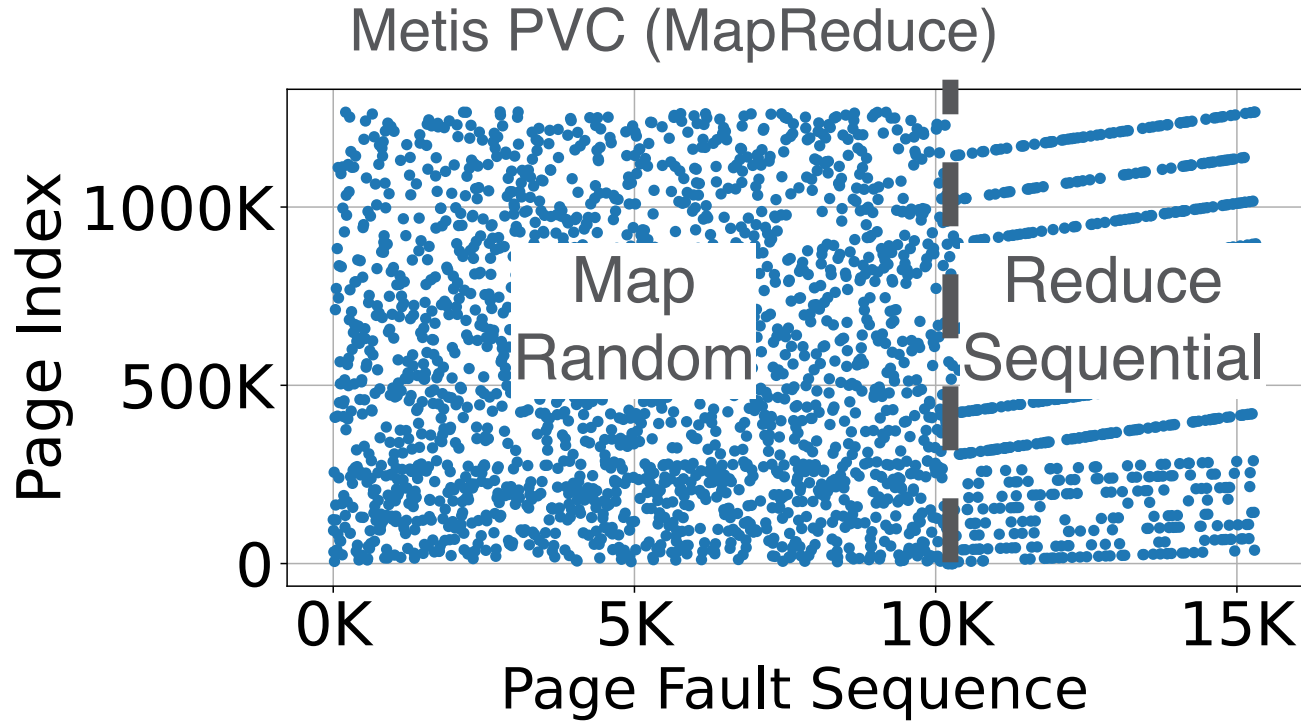
Random Access

Suitable for

Page Fetching

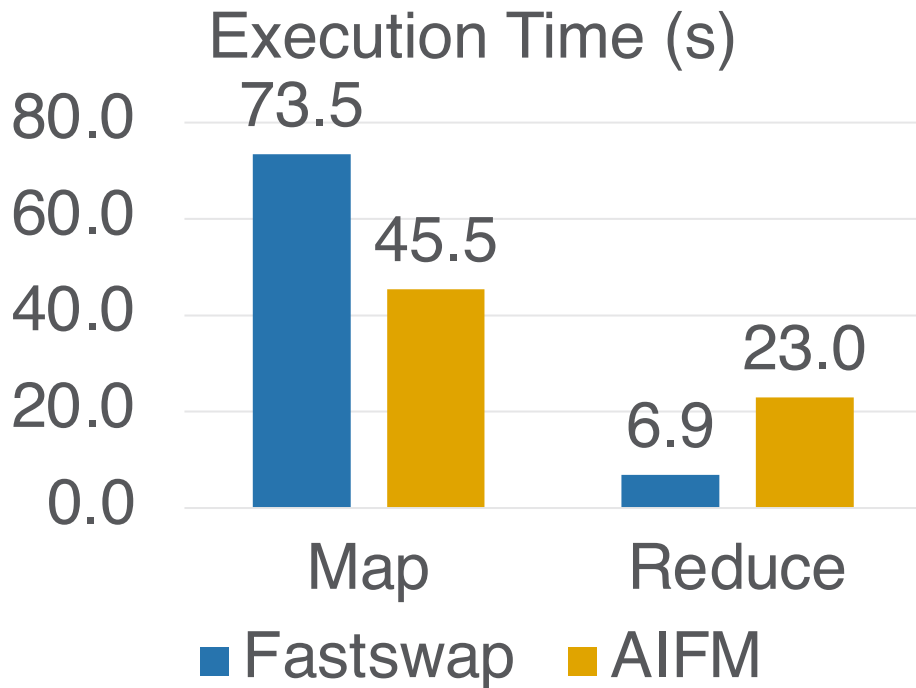
Object Fetching

Dynamic Access Patterns



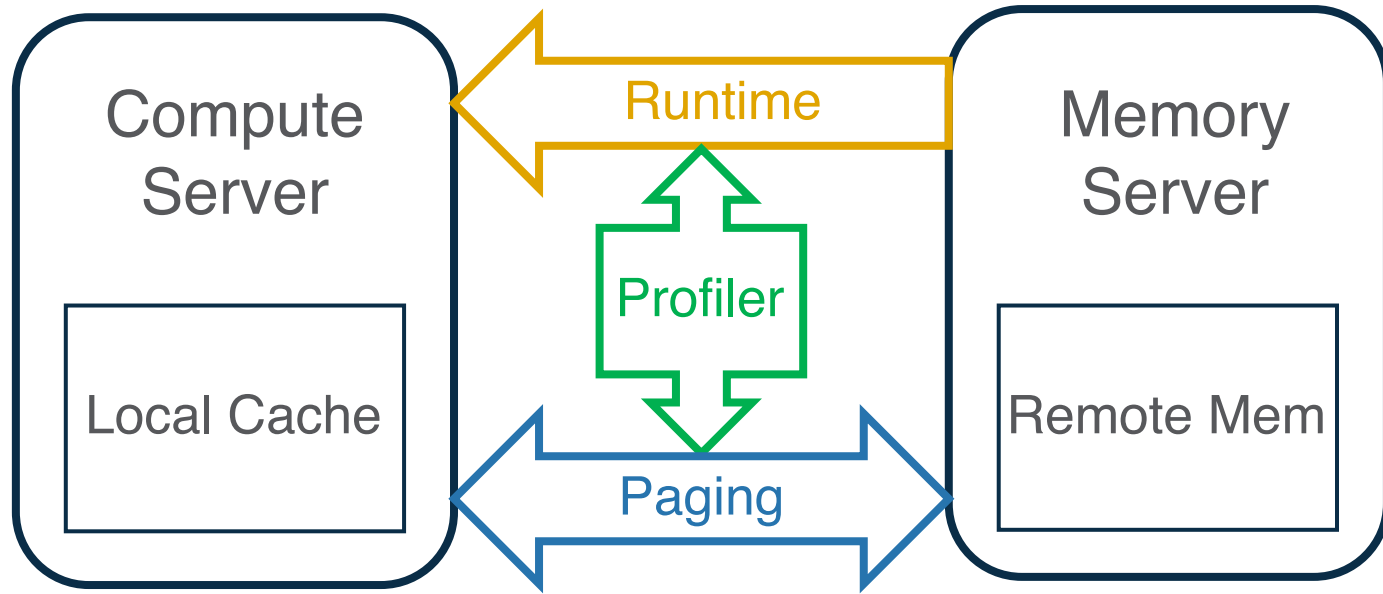
One Path Cannot Fit for All

Metis PVC on
Fastswap^[1] - Paging
AIFM^[2] - Runtime



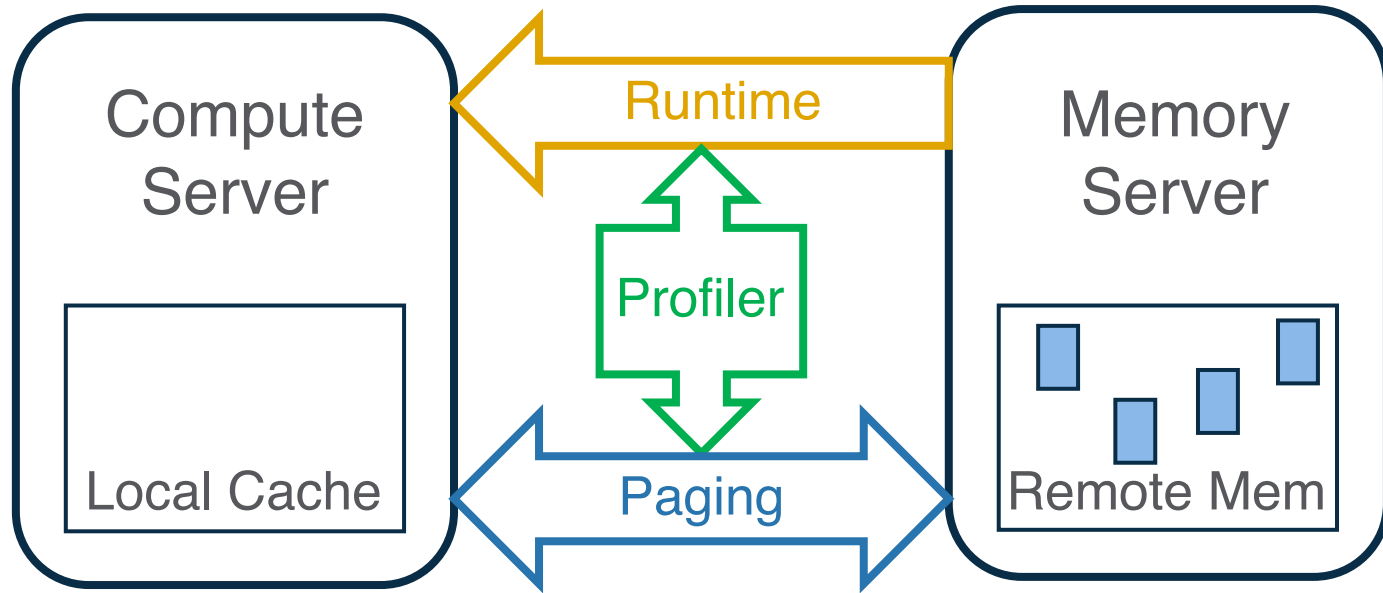
Can a data plane combine the advantages of both paging path and runtime path for optimal performance?

Atlas Overview



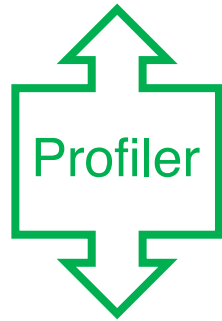
- Hybrid fetching path
- Locality profiler
- Unified eviction path

Benefits

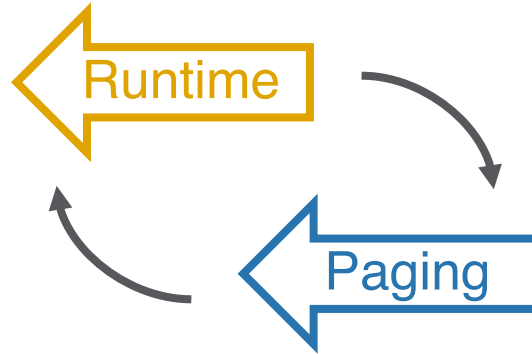


- Optimal data transfer efficiency
- Gradual Locality optimization

Challenges



Profiling
Mechanism



Path
Switching



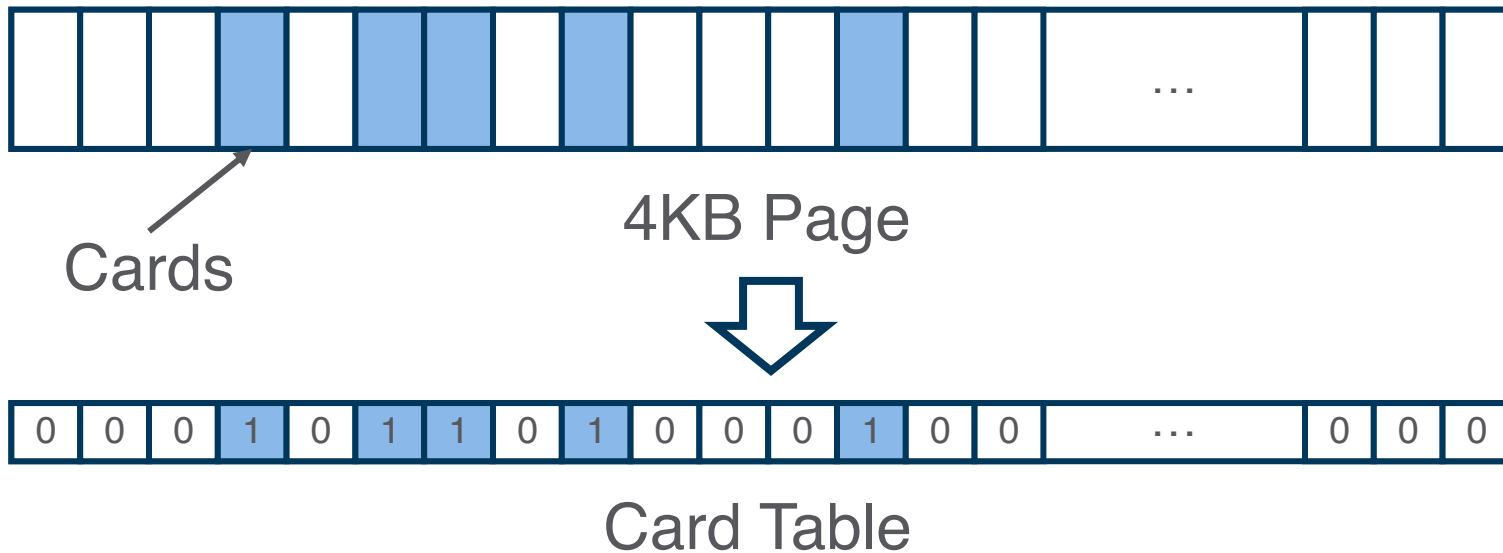
Cross-component
Synchronization

Locality Profiling



4KB Page

Locality Profiling



$$\text{Card Access Ratio} = \frac{\# \text{ Recently Accessed Cards}}{\# \text{ Total Cards in a page}}$$

Path Switching

Good

Page with Good Locality

Good

Bad

Slow

Good

Good

Bad

Slow

Paging

Bad

Page with Bad Locality

Good

Slow

Bad

Good

Slow

Bad

Bad

Runtime

Path Switching

Good

Page with Good Locality

Per-page
Path Flag

=

Paging if

Good

or

Runtime if

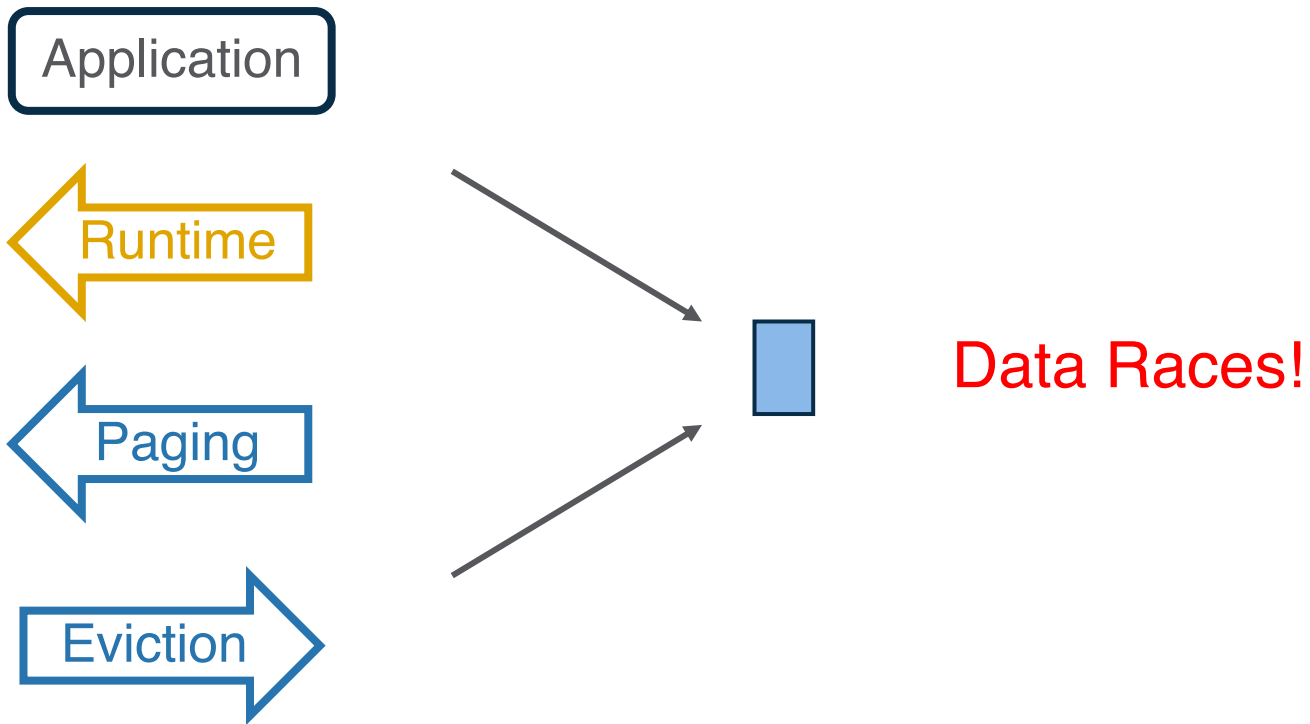
Bad

Bad

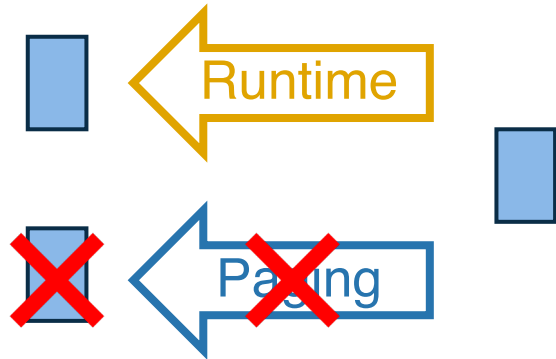
Page with Bad Locality



Synchronization



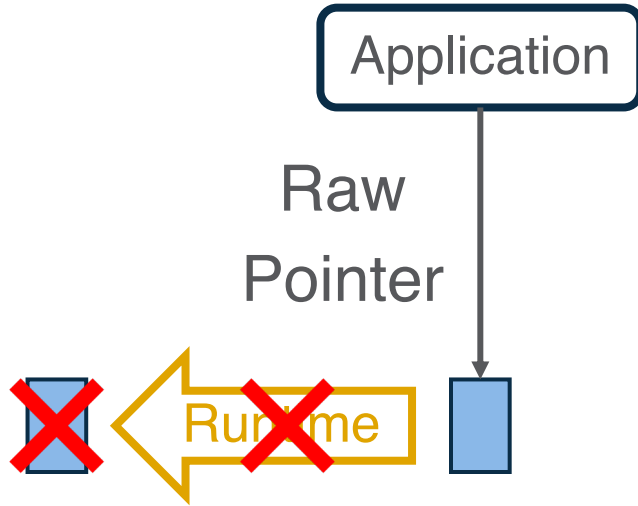
Object Fetching vs. Page Fetching



Invariant:

At any moment, data on the same page must go through one single fetching path.

Application vs. Object Fetching

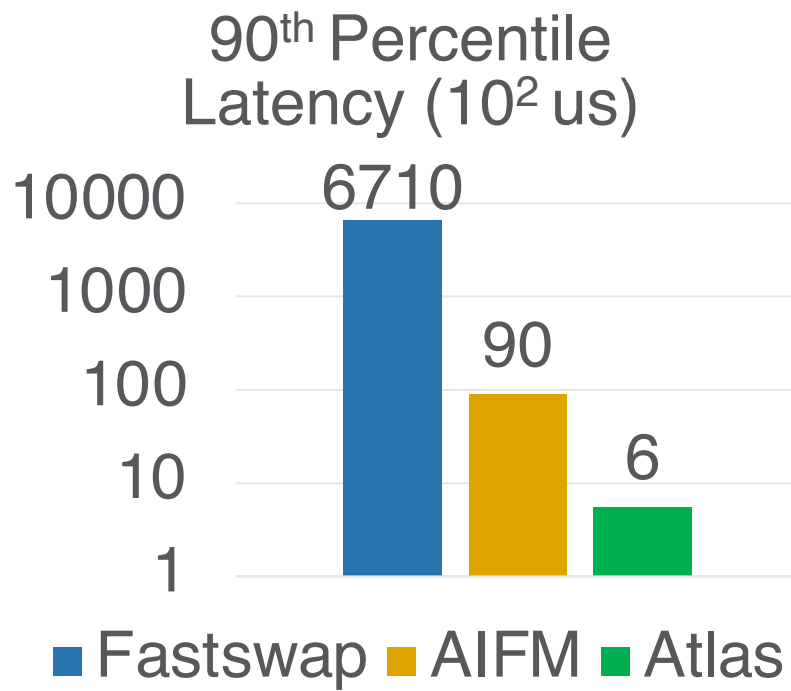
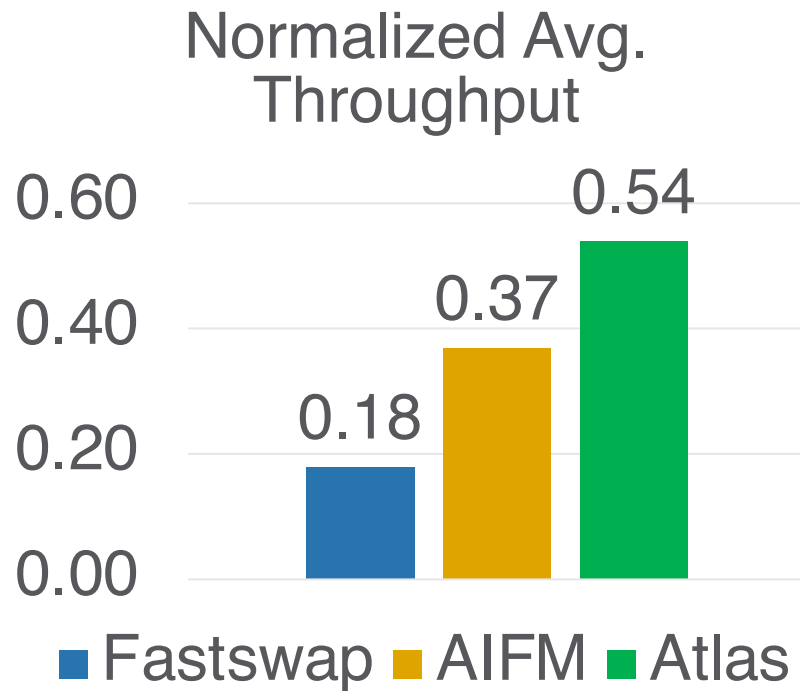


Invariant:
Objects with raw pointers
must be kept in local memory.

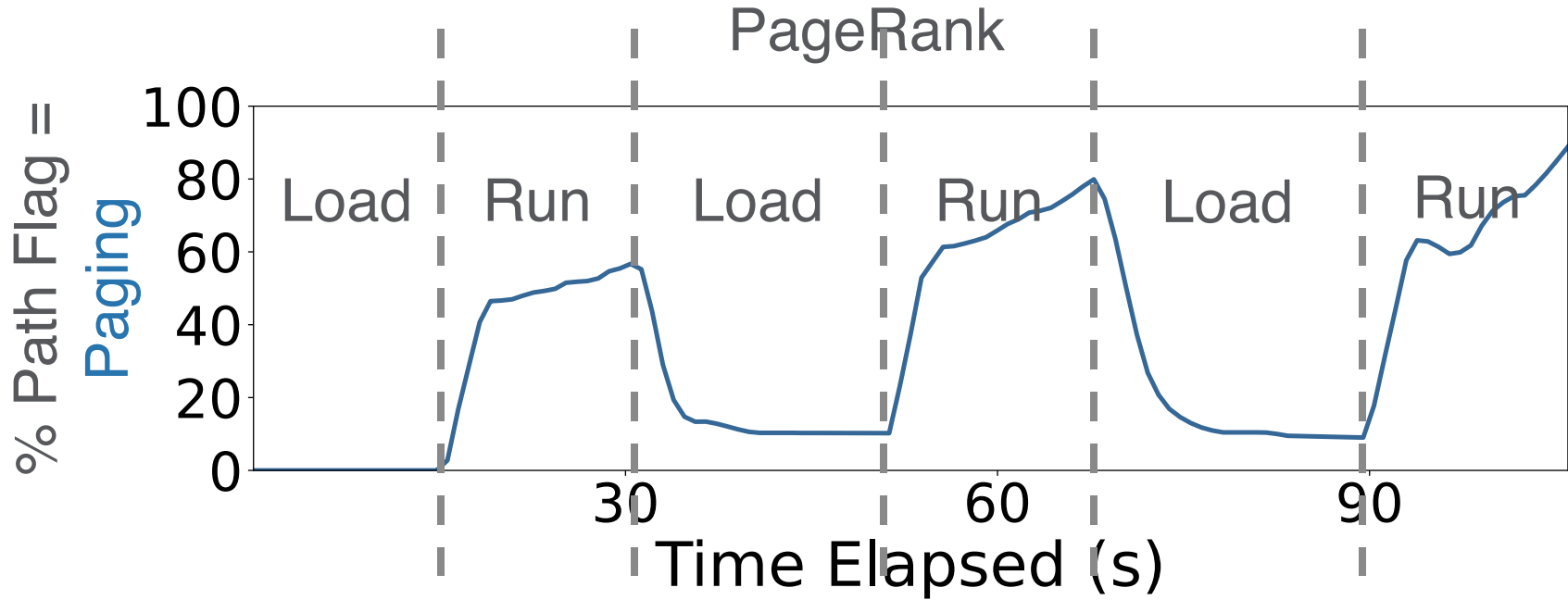
Evaluation: Overview

- Eight Applications:
 - Random/Skewed: Memcached
 - Sequential: DataFrame
 - Phase-changing: MapReduce & dynamic graph analytics
- Baselines: AIFM and Fastswap

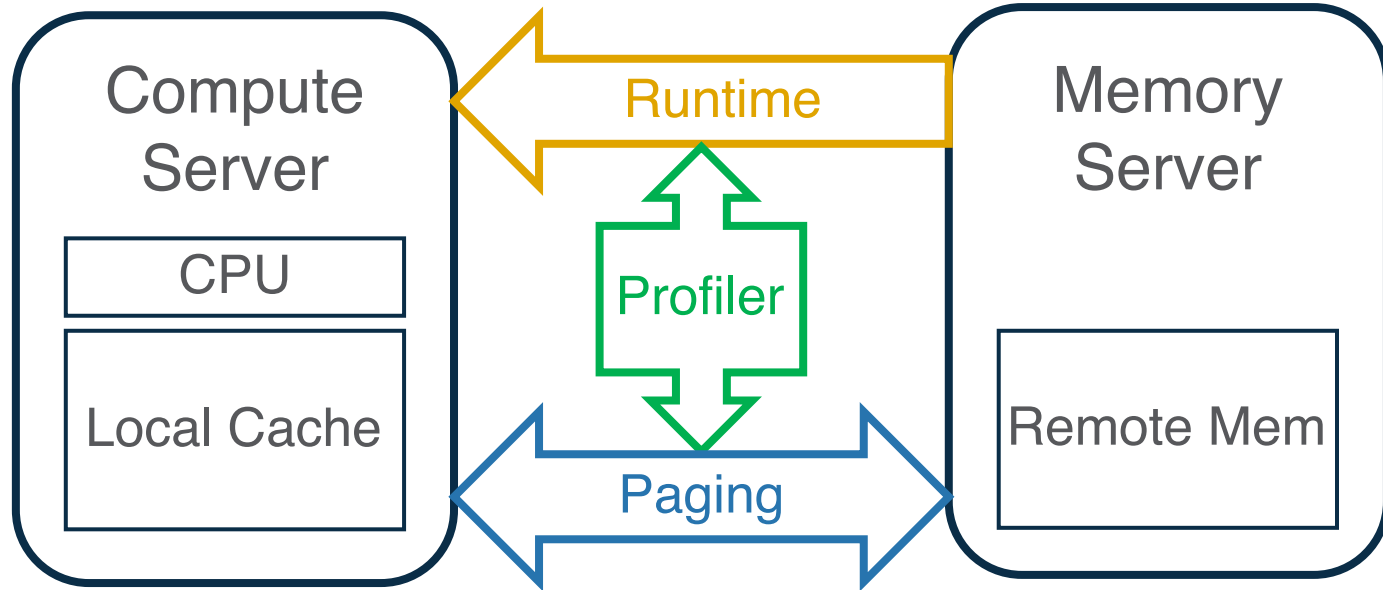
Evaluation: Throughput and Latency



Evaluation: Path Switching



Conclusion



Thank you!

Atlas is available at <https://github.com/wangchenxi7/Atlas>