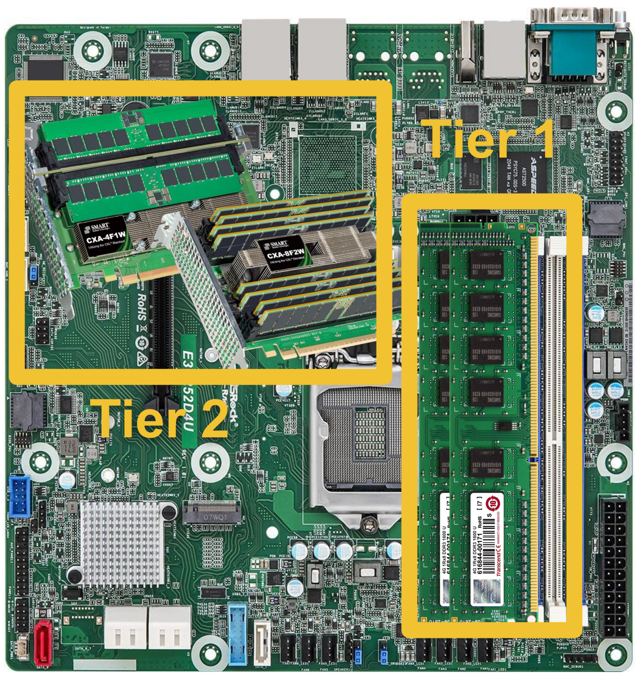
Handling Latency in Tiered Memory with Prefetching

Musa Unal, Vishal Gupta, Yueyang Pan, Yujie Ren, Sanidhya Kashyap

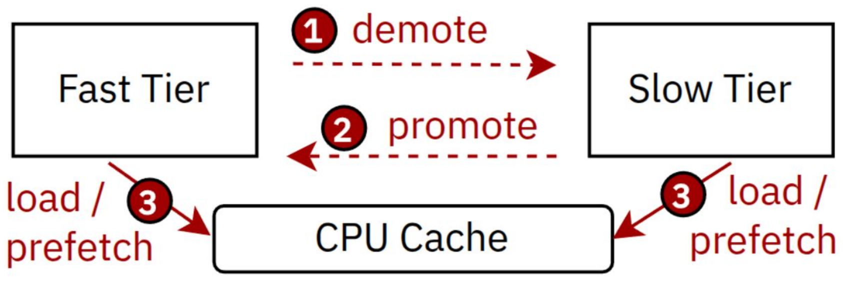
EPFL

Problem: Tiering systems do not consider how application is accessing the data



Data centers use different types of memories which have different characteristics. Trade-off between

- Latency/Bandwidth
- Price
- Energy


Fast Tier

Slow Tier

1 demote

2 promote

load / 3 prefetch

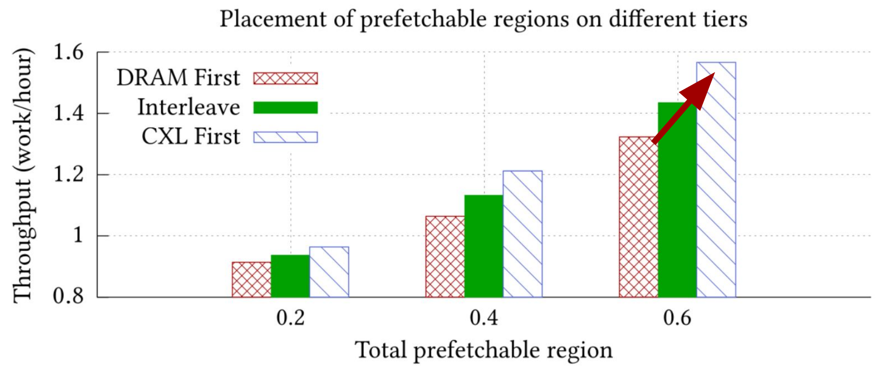
3 load / prefetch

CPU Cache

Current tiering systems are trying to optimize load time (3) by demoting (1) and promoting (2) between the nodes.

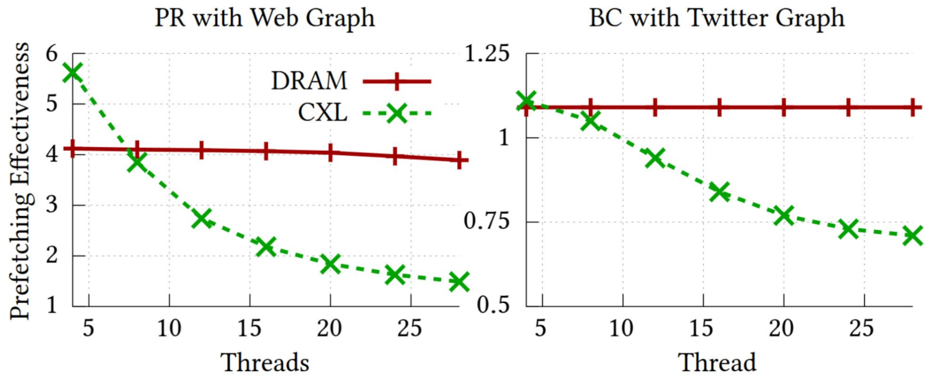
Insight: Data prefetchers can tolerate CXL latency.

Hardware prefetchers and software prefetching can ***hide*** the CXL latency

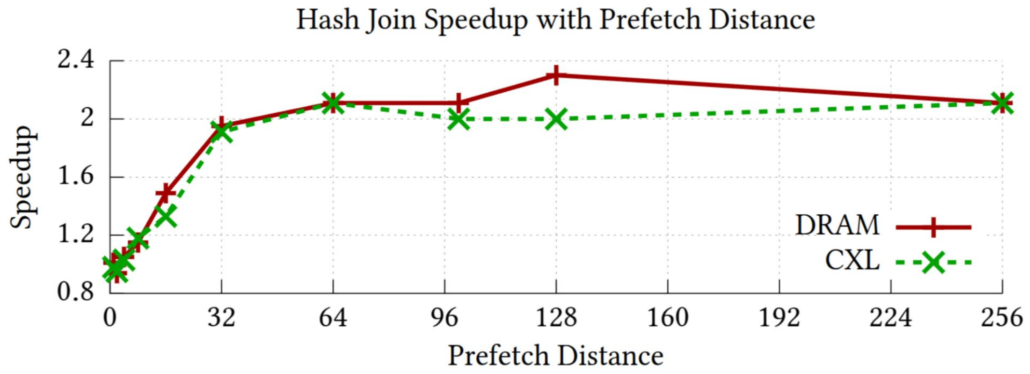

Placement of prefetchable regions on different tiers

Hardware prefetchers can degrade performance under high load.

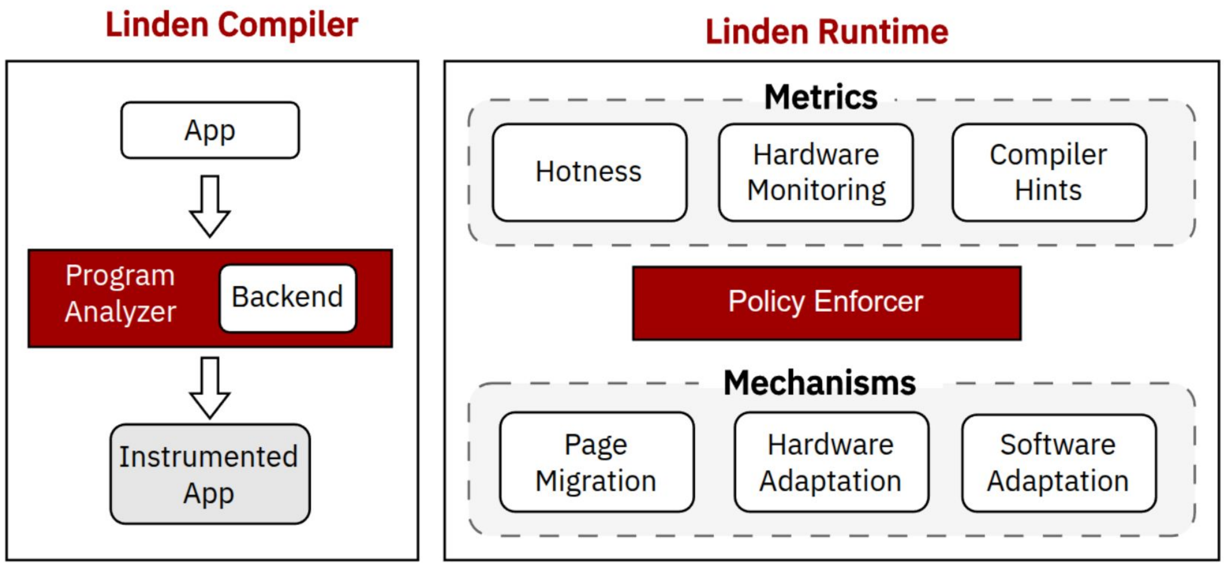
Software prefetches must use tiering-aware prefetch distances to be effective.


PR with Web Graph

BC with Twitter Graph


Hash Join Speedup with Prefetch Distance

Proposal: A runtime system to effectively use prefetchers on tiered memory


Linden Compiler

App

Program Analyzer

Backend

Instrumented App

Linden Runtime

Metrics

Hotness

Hardware Monitoring

Compiler Hints

Policy Enforcer

Mechanisms

Page Migration

Hardware Adaptation

Software Adaptation

Reduce: Minimize latency by increasing the locality.
Tolerate: Hide the latency by prefetching.

1. Detect prefetchable regions
2. Monitor memory bandwidth
3. Enforce different policies regarding to application's needs