

Dynamic memory interleaving for tiered memory

Bandwidth is important too

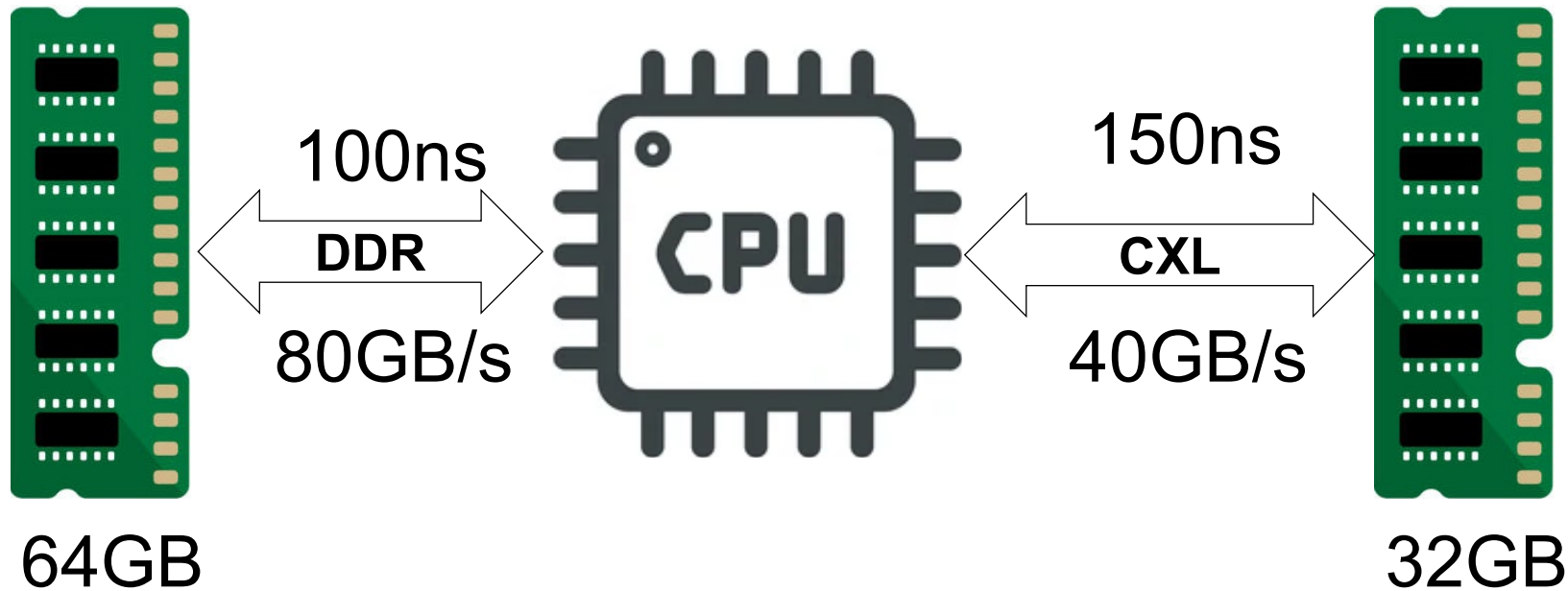
Aug 07, 2025

Eishan Mirakhur

MTS – CXL Product Development



Properties of CXL memory devices

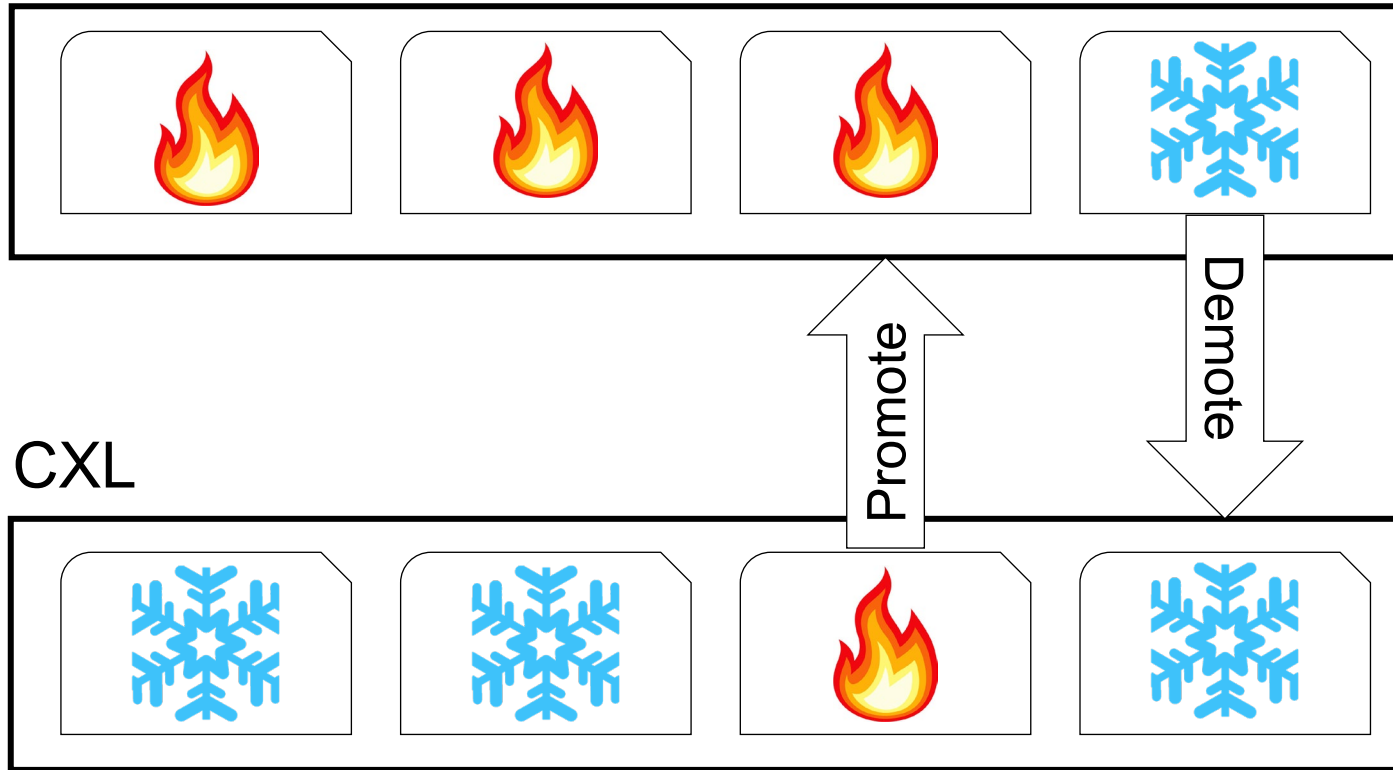


More capacity!
More bandwidth!
Higher latency :|

Memory tiering

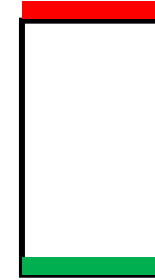
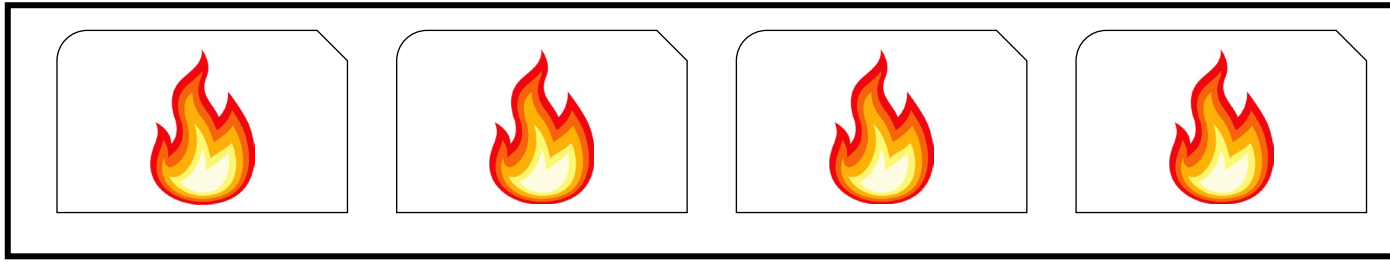
Utilize capacity, minimize latency

DDR



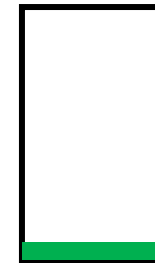
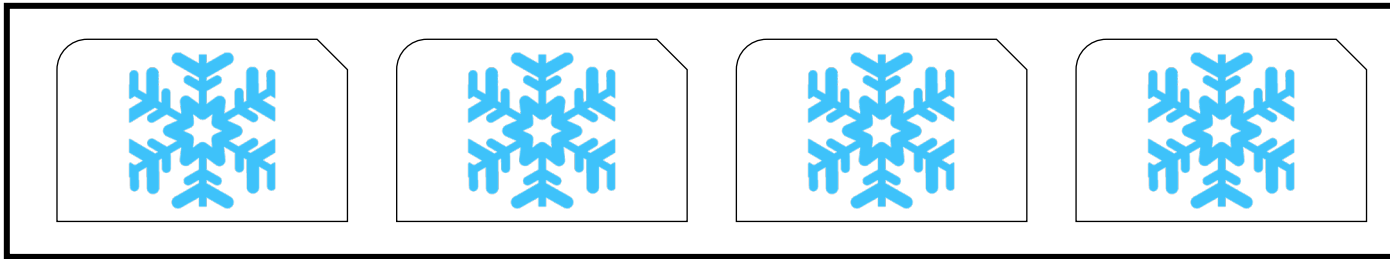
Problem: traditional tiering can saturate local bandwidth

DDR



BW util

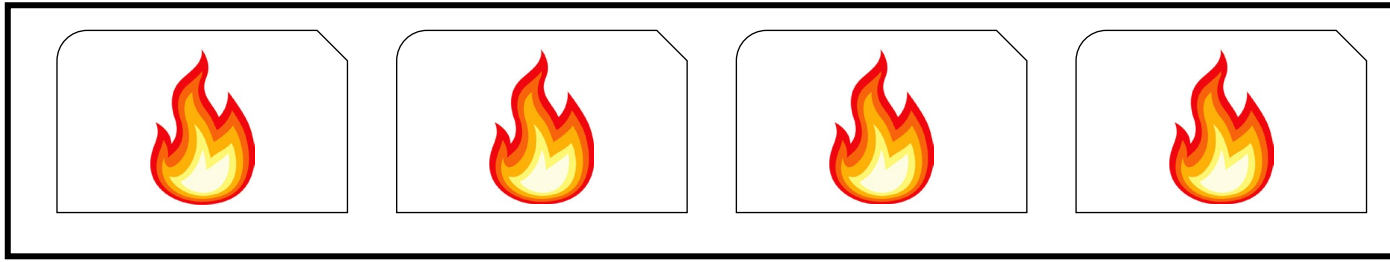
CXL



BW util

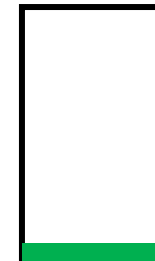
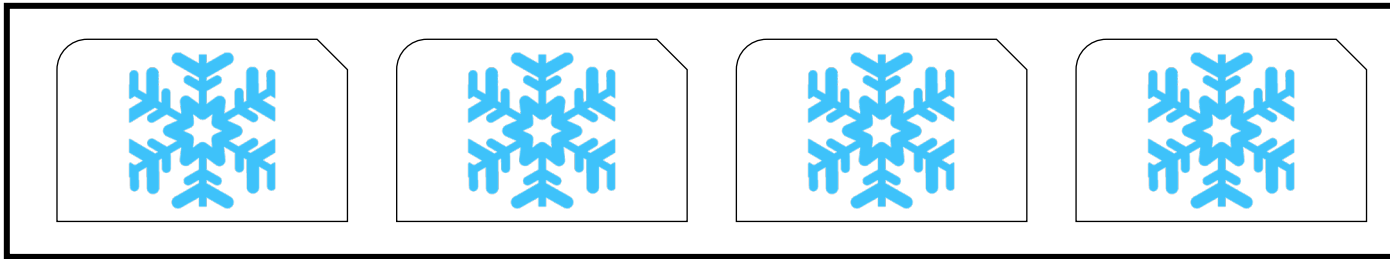
Solution: interleave hot pages

DDR



BW util

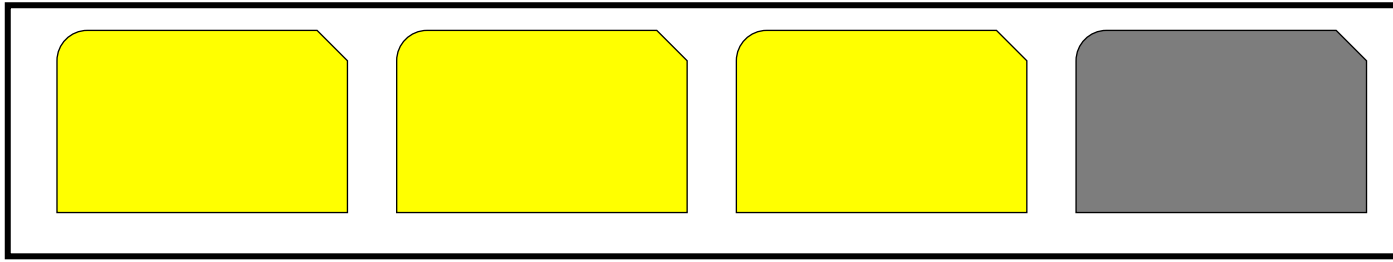
CXL



BW util

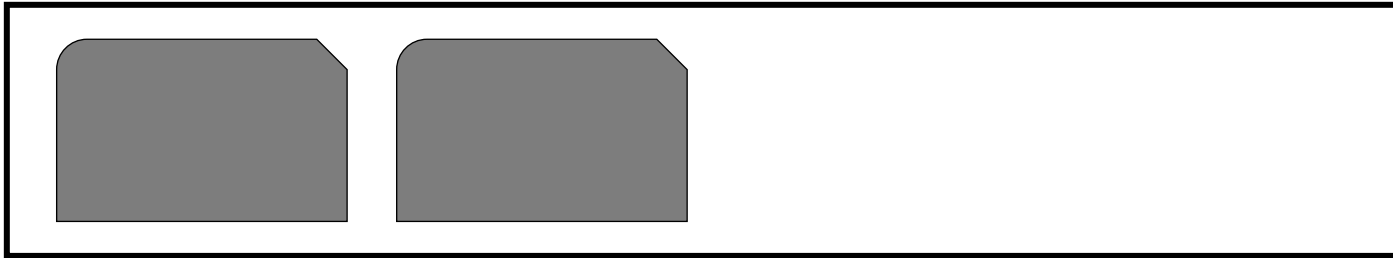
Caveat: static interleaving isn't enough

DDR



BW Util

CXL



BW Util

Want: memory policy to dynamically interleave data based on demand

Our proposal: dynamic memory interleaving (DMI)

Our proposal: dynamic memory interleaving (DMI)

Goal: maximize bandwidth utilization

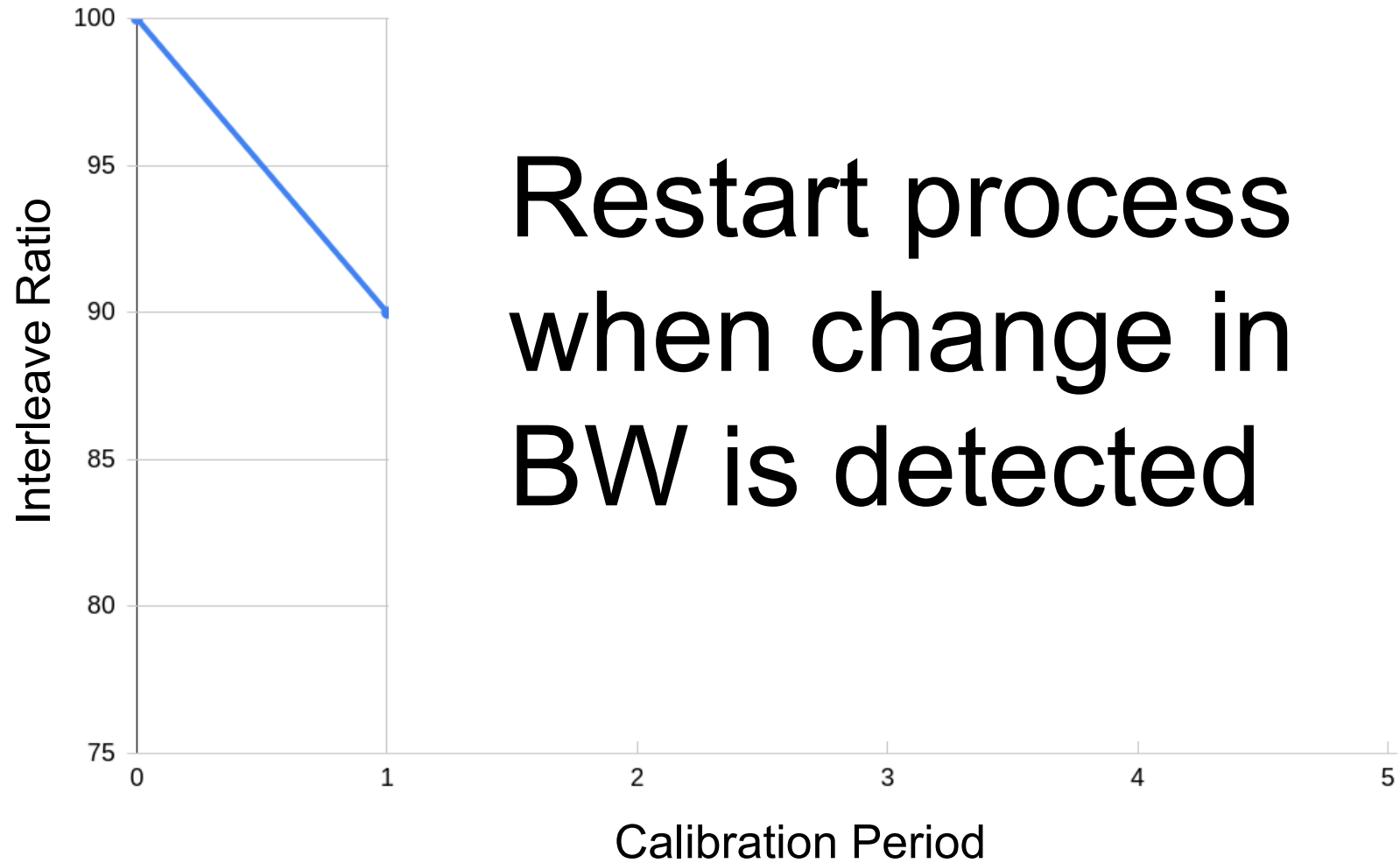
Calibrator

- Monitors system bandwidth utilization
- Determines a target interleave ratio for the system

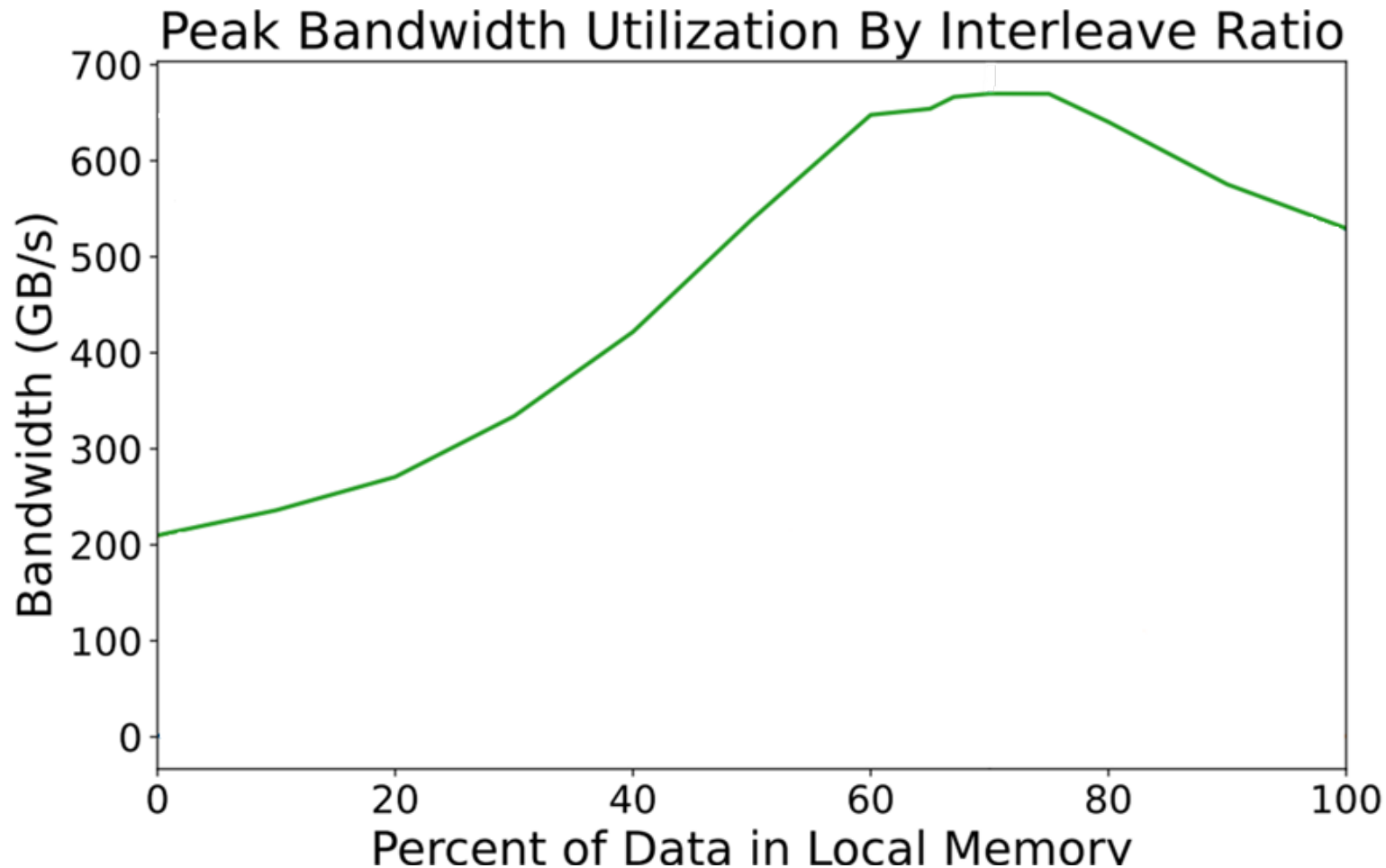
Migrator

- Reads target interleave ratio from calibrator
- Migrates application pages to achieve the target interleave ratio

Calibrator: determining the interleave ratio



Calibrator: what's a “good” step?

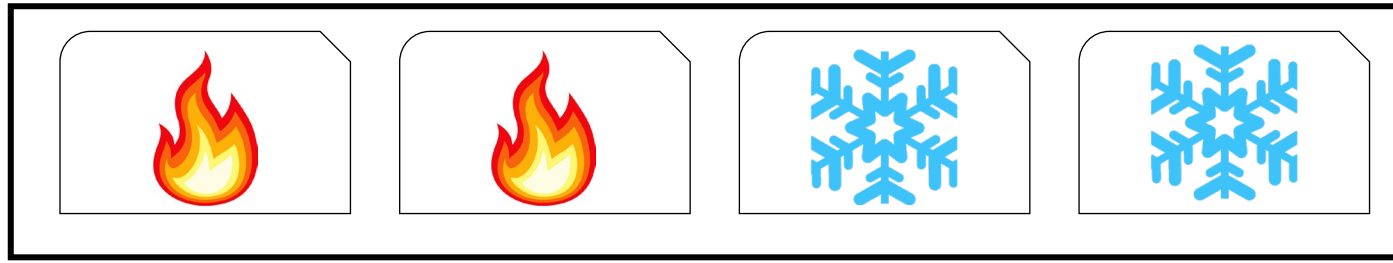


Calibrator: implementation

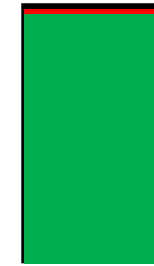
- User space application
- Reads bandwidth via perf counters
- Samples bandwidth every 200ms
 - Ignore samples during migration
 - Select 90th percentile sample to use for the period
- Calibration period: 6s

Migrator: demotion

DDR

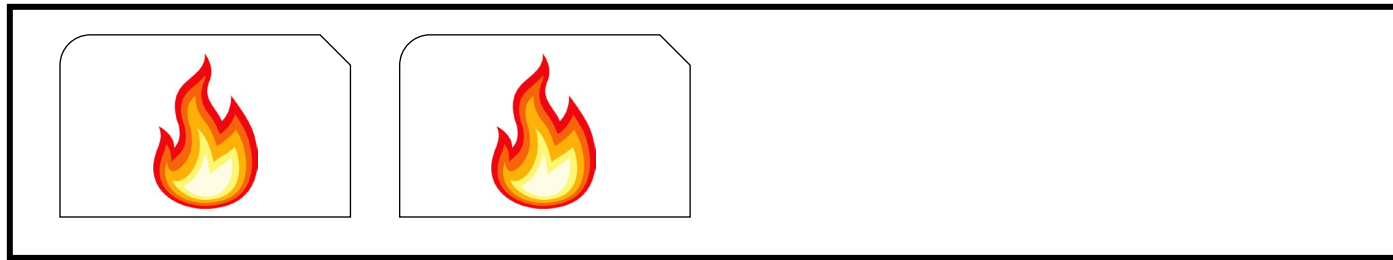


BW Util



Capacity

CXL



BW Util

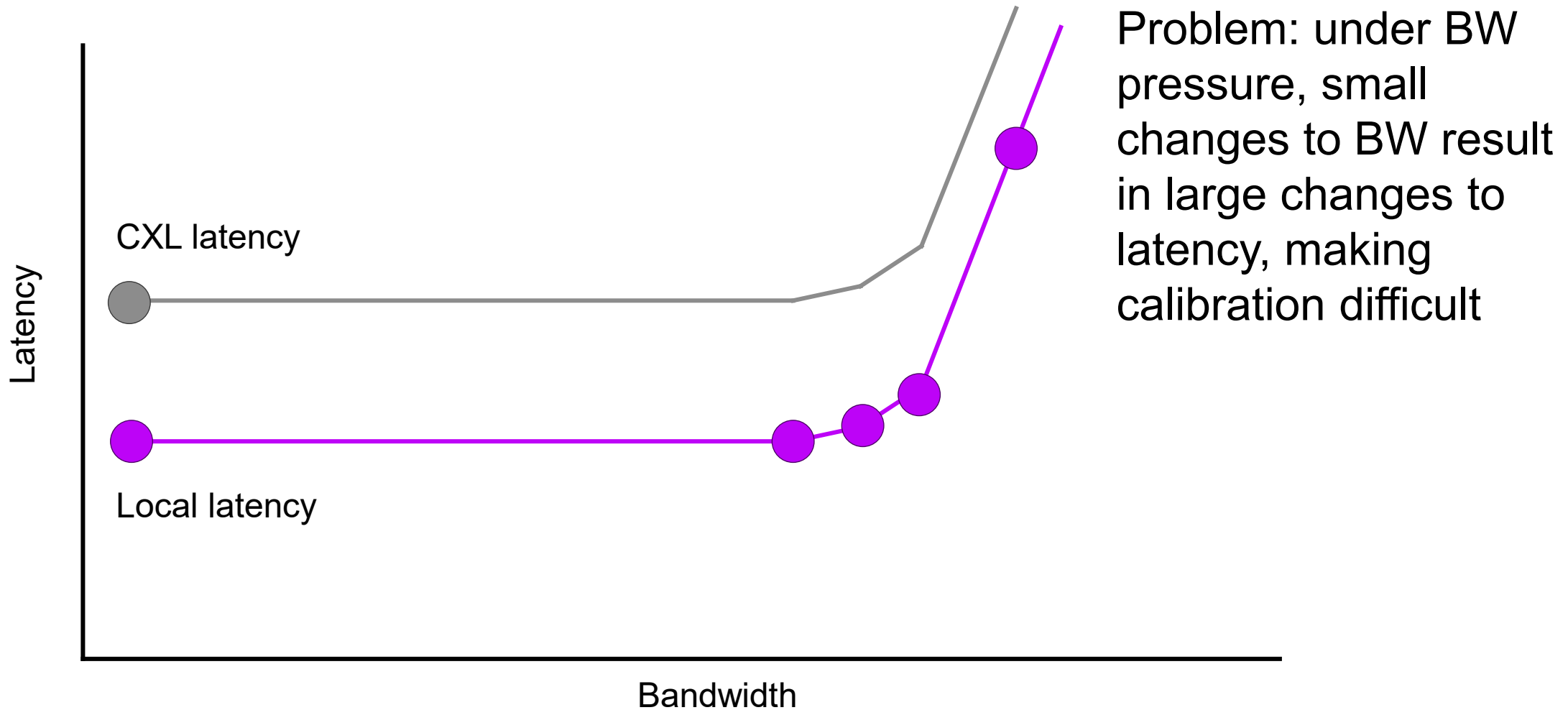
Property: When the interleave ratio is 100% local, DMI behaves the same as traditional tiering

Migrator: implementation

- Utilize DAMON - Existing Linux Tool
 - Tracks hot/cold pages
 - Migrates pages
- Use existing DAMON action for demotion
- Added DAMON action for interleaving
- Gets interleave ratio from calibrator via sysfs
- [RFC submitted to Linux kernel maintainers](#)

Similar system: colloid

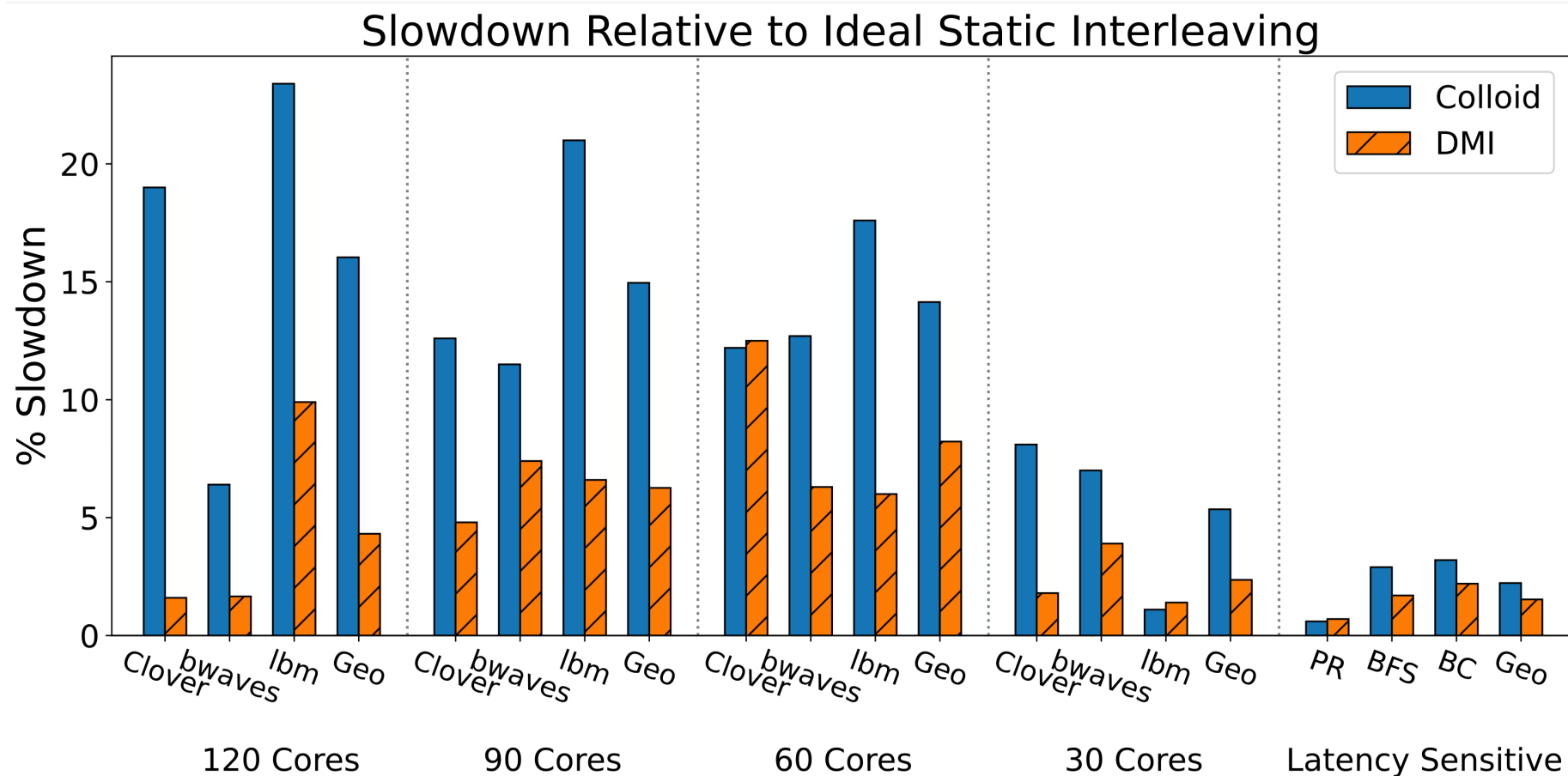
Goal: Minimize the maximum access latency



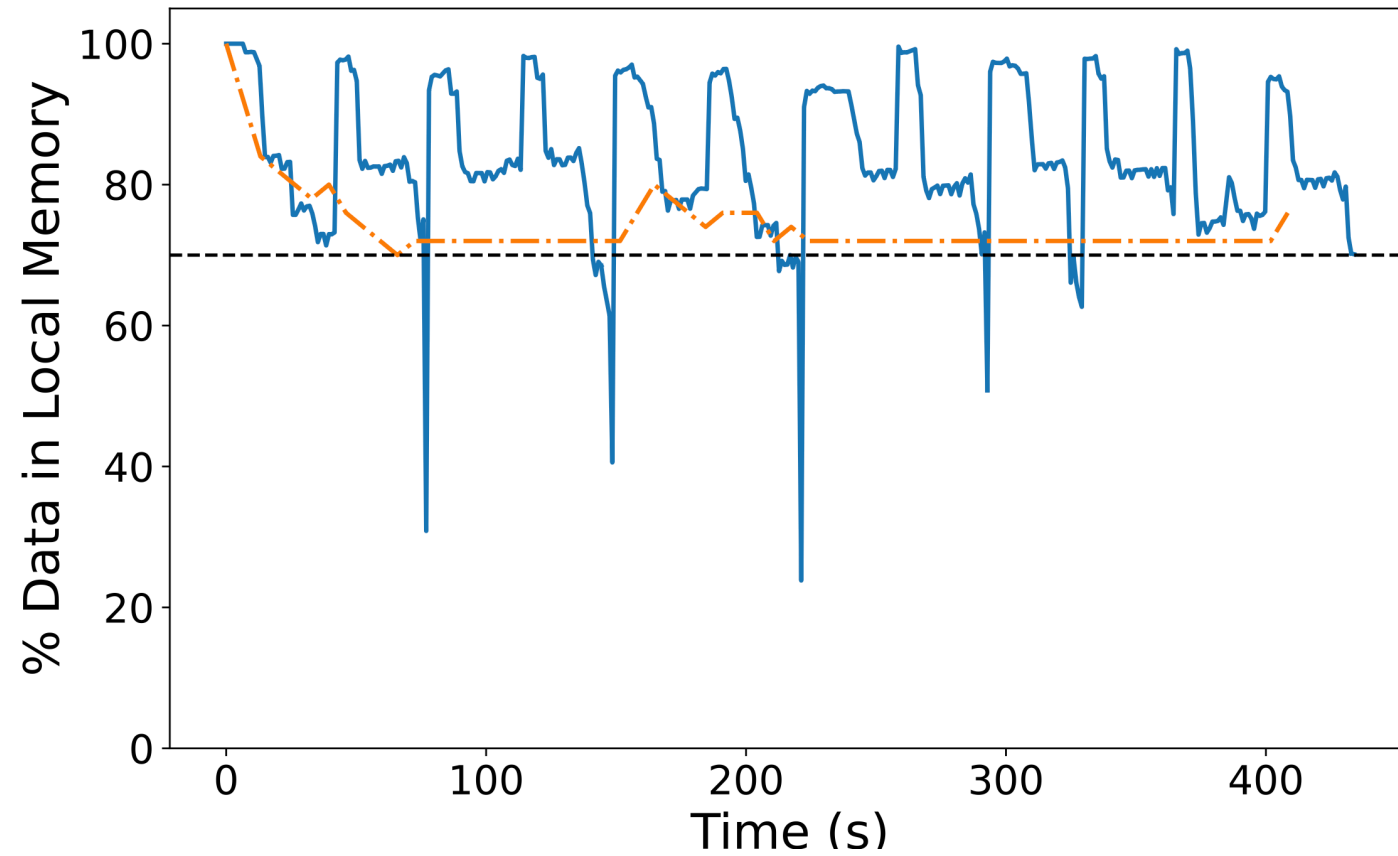
Evaluation setup

- 120 core Intel Granite Rapids CPU
- 12 DDR5 channels with 500GB/s max BW
- 8 Micron CZ122 CXL memory devices with 300GB/s max BW
- BW intensive workloads: CloverLeaf, lbm, bwaves
- Latency sensitive workloads: page rank, breadth first search, betweenness centrality

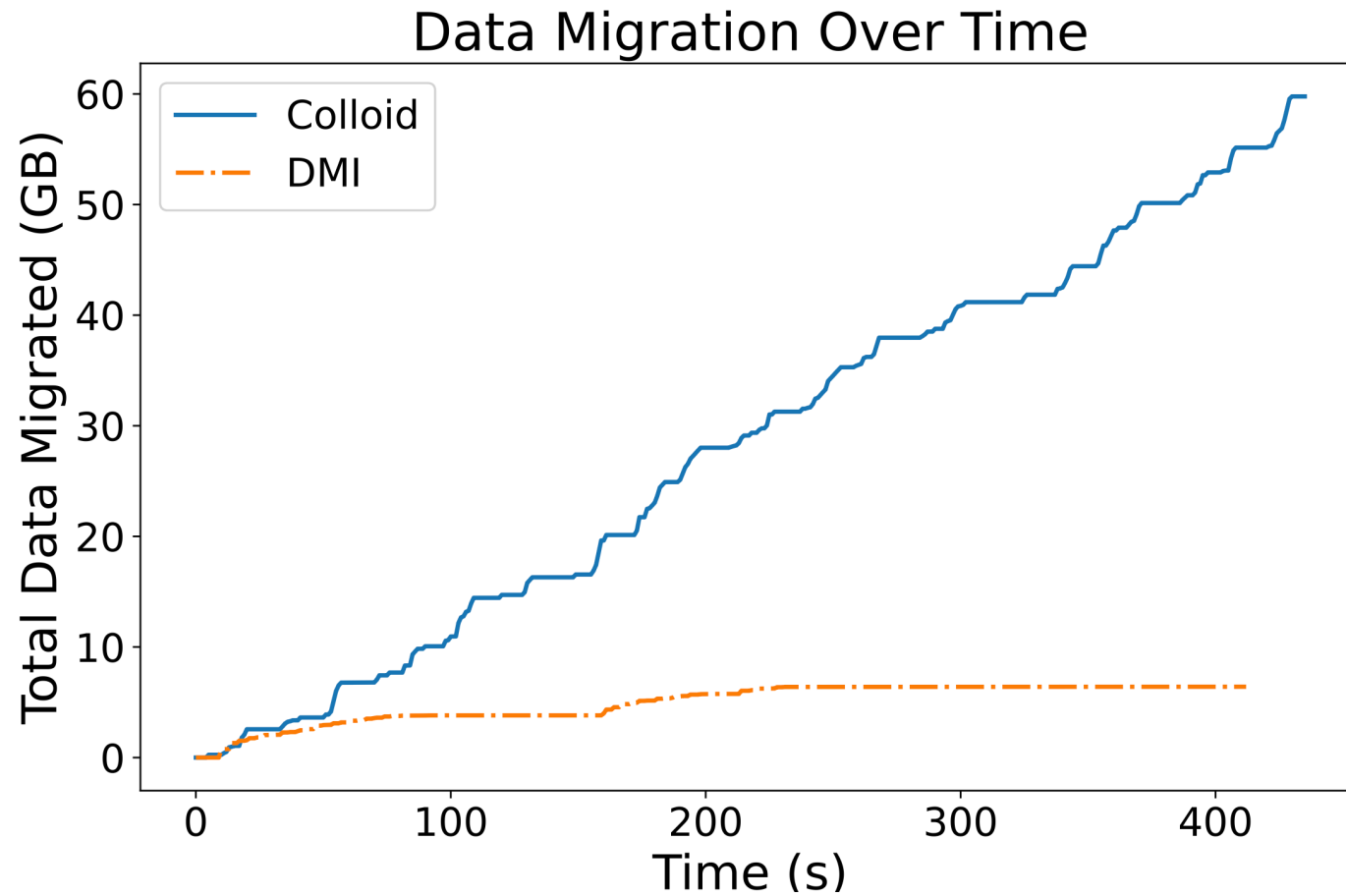
Single workload performance



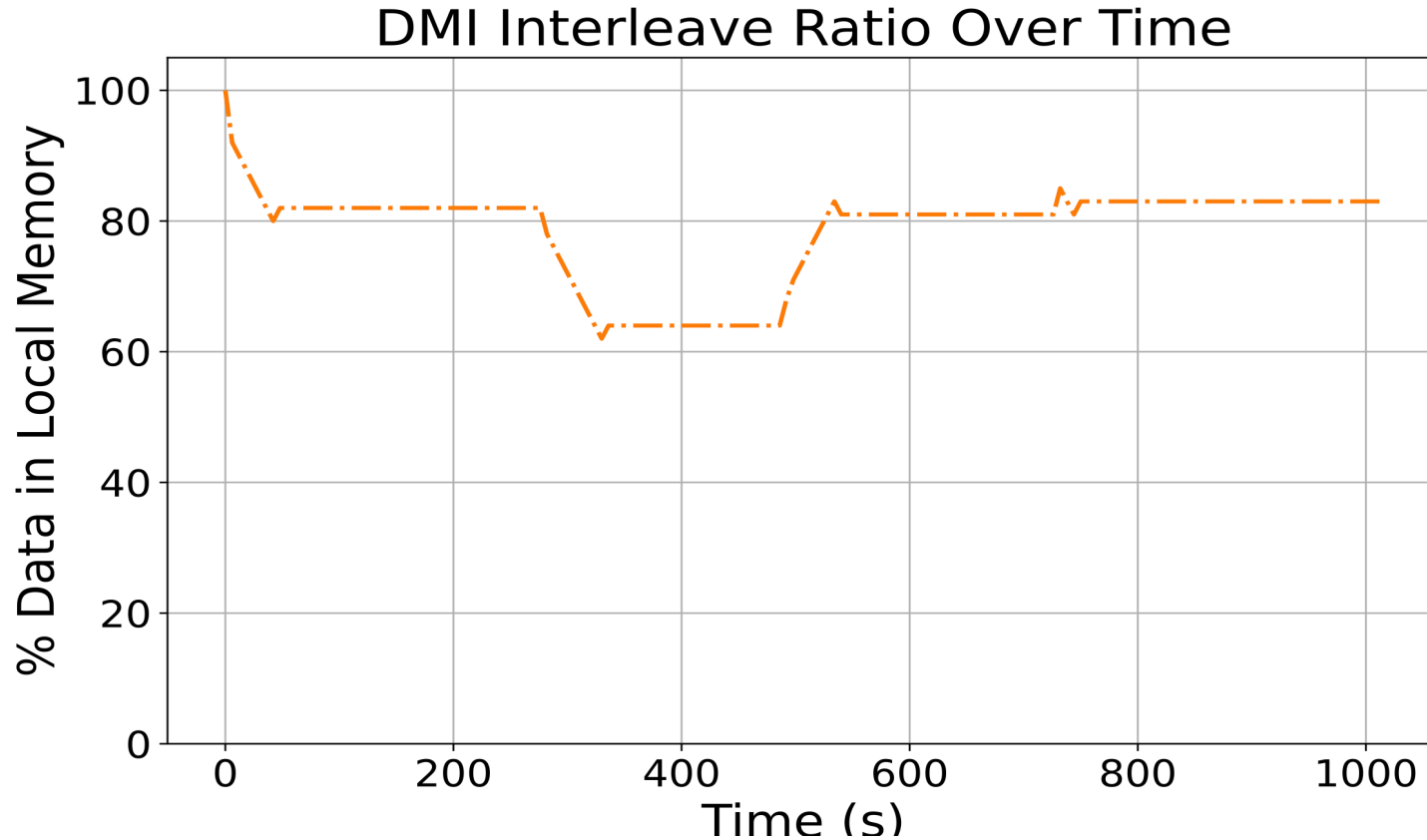
Interleave ratio: bwaves



Migration over time: bwaves



Shifting bandwidth demand



First workload

2% faster than static interleaving

Second workload

5% slower than static interleaving (similar to single workload perf)

This graph represents how well DMI handles shifting demands in bandwidth with lbm workloads with varying cores and iterations

Key takeaways and conclusion

- Making effective use of bandwidth is important for application performance
- Static interleaving is insufficient for changing bandwidth demand
- DMI dynamically interleaves data based on bandwidth utilization to efficiently utilize system bandwidth



© 2025 Micron Technology, Inc. All rights reserved. Information, products, and/or specifications are subject to change without notice. All information is provided on an “AS IS” basis without warranties of any kind. Statements regarding products, including statements regarding product features, availability, functionality, or compatibility, are provided for informational purposes only and do not modify the warranty, if any, applicable to any product. Drawings may not be to scale. Micron, the Micron logo, the M logo, Intelligence Accelerated™, and other Micron trademarks are the property of Micron Technology, Inc. All other trademarks are the property of their respective owners.