Georgia Tech

CREATING THE NEXT

The Case for Optimizing the Frequency of Periodic Data Movements over Hybrid Memory Systems

<u>Thaleia Dimitra Doudali</u>, Daniel Zahka, Ada Gavrilovska

@ MEMSYS 2020

Problem Space

1. Hybrid Memory Systems:

- Massive memory capacity at a reasonable cost.
- Use non volatile memory hardware. (e.g., Intel Optane DC persistent memory)
- Disaggregate the available memory.



Emerging Hybrid Memory Systems.

2. Challenge:

- High variability in the access speeds of the heterogeneous memory units.
- Significant application performance loss from traditional DRAM-only systems.

3. Approach:

Dynamically move frequently accessed data in the fastest memory component.



Problem Statement

Existing Solutions:

Optimize the selection of <u>which</u> and <u>how much</u> data to dynamically move, by monitoring and collecting data access history information.

When to move data?

- According to execution phases.

For example, Unimem relies on MPI phases and Tahoe on task-based execution.

- In fixed time intervals, i.e., periods. (focus of this work)

Empirically set across majority of existing application-agnostic operating system-level solutions.

Problem Statement: What should be the frequency of data movements, so as to boost application performance in return for minimal monitoring and migration costs? What application-level properties hint towards a more sophisticated, rather than empirical, choice?





Why it is important



Methodology: Periodically monitor data access and move as many hot pages as DRAM fits. **Experiment:** Vary the data movement frequency (period length).



Observations:

- Tuning the frequency leads to performance improvements (70% for PMEM, 5x for disaggregated on average)
- A certain frequency may not work across different applications or different platforms.
 - Same platform: different applications may benefit most from completely different frequencies.
 - Same application: different platforms may better offset any data movement costs.

Takeaway: it is important to choose the right frequency, but it is not intuitive how to choose one.



Data Access Behavior



Page Reuse Distance = Number of memory accesses to other pages, between two consecutive accesses to that particular page. **Takeaway:** data reuse indirectly shows data access behavior, with no prior information.



Memory Access Patterns

Data Reuse Distance vs. Application Performance



Observation: When period length < median page reuse distance, the effectiveness of page scheduler *may* drop.

Takeaway: data reuse information can <u>hint towards</u> which frequencies may be good. The <u>final choice</u> of frequency highly depends on the effectiveness of the page scheduler and the way the platform can hide the data movement costs under a higher fast memory hit rate, depending on its memory access speeds.



Future Work

After establishing the importance of data reuse information and the parameters which affect application performance on hybrid memory systems, we are building.

Cori, a profiler that synthesizes:

- data reuse distance
- data access patterns
- page scheduler efficiency
- platform configuration

into the selection of the data movement frequency, that will further boost application performance, irrespective of the effectiveness of the data movement selection itself.

... Stay tuned!

