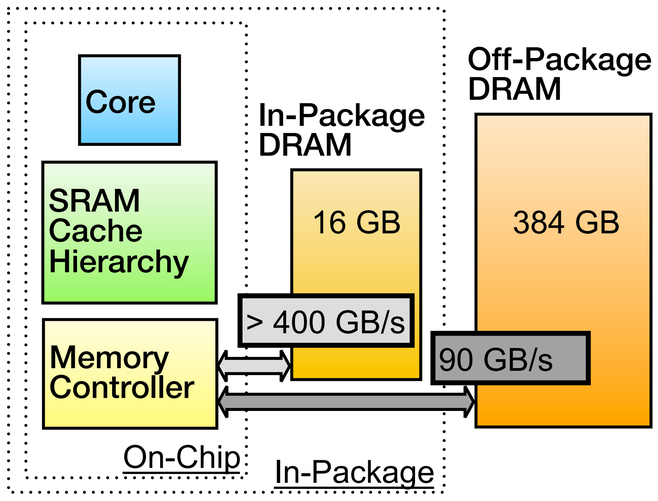
Banshee: Bandwidth-Efficient DRAM Caching via Software/Hardware Cooperation

Xiangyao Yu[1], Christopher Hughes[2], Nadathur Satish[2], Onur Mutlu[3], Srinivas Devadas[1]
[1]MIT [2] Intel Labs [3]ETH Zürich

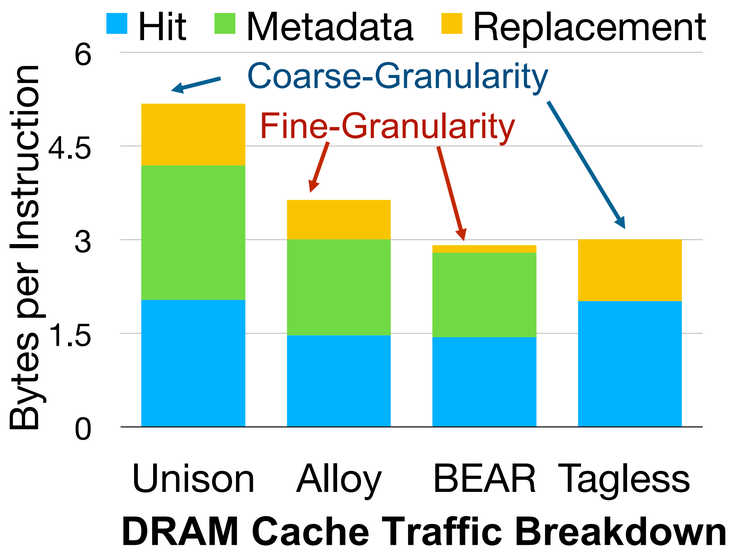
Motivation

- In-package DRAM has
  - **5X higher bandwidth** than off-package DRAM
  - **Similar latency** as off-package DRAM
  - **Limited capacity** (up to 16 GB)
- In-package DRAM can be used as a cache



* Numbers from Intel Knights Landing

Bandwdith Inefficiency in Existing DRAM Caches



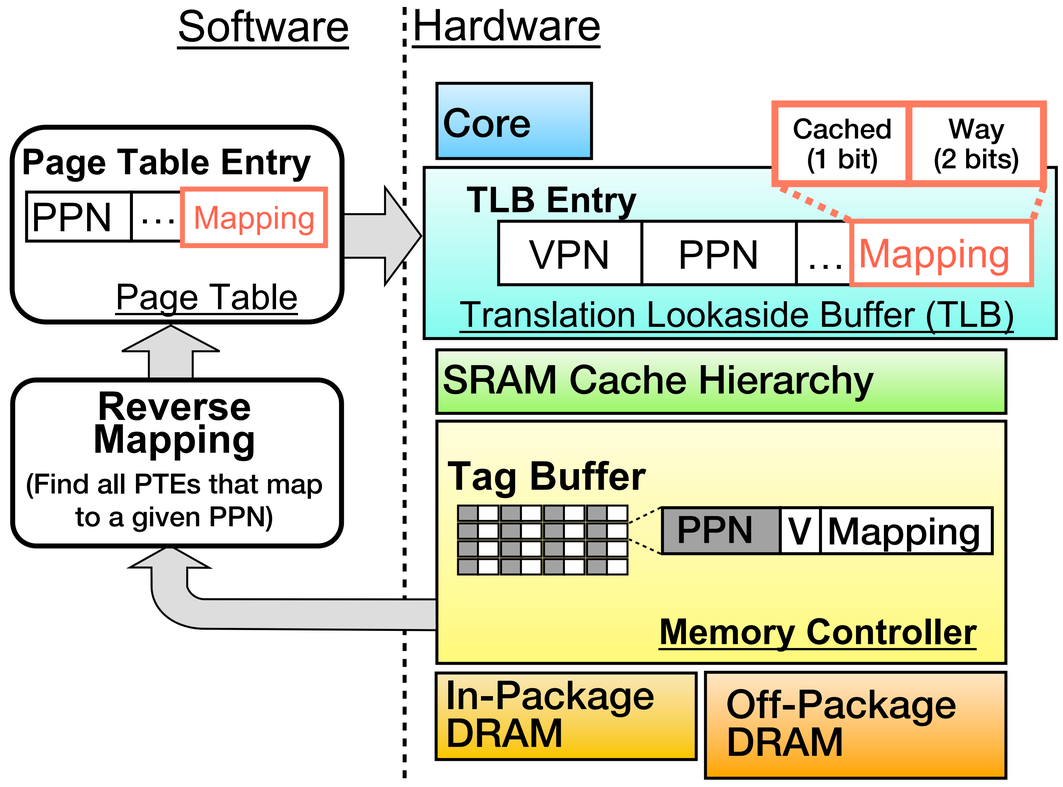
- **Drawback 1**: Metadata traffic (e.g., tags, LRU bits, frequency counters, etc.)
- **Drawback 2**: Replacement traffic
  - Especially for coarse-granularity (e.g., page-granularity) DRAM cache designs

Banshee Contribution

- Bandwidth efficiency as a first-class design constraint
- High Bandwidth efficiency without degrading latency

Idea 1: Efficient TLB coherence for Page-Table-Based DRAM Caches

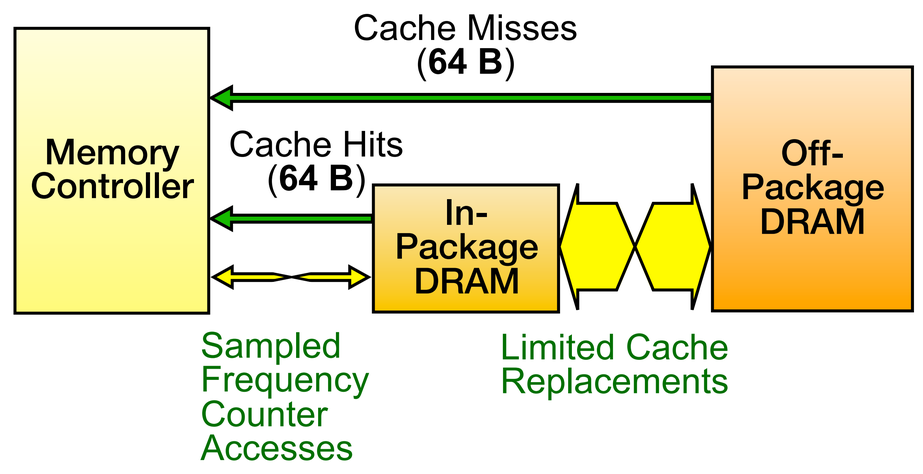
- Track DRAM cache contents using **page tables** and **TLBs**
- Maintain latest mapping for recently remapped pages in the **Tag Buffer**
- Enforce TLB coherence **lazily** when the Tag Buffer is full to amortize the cost



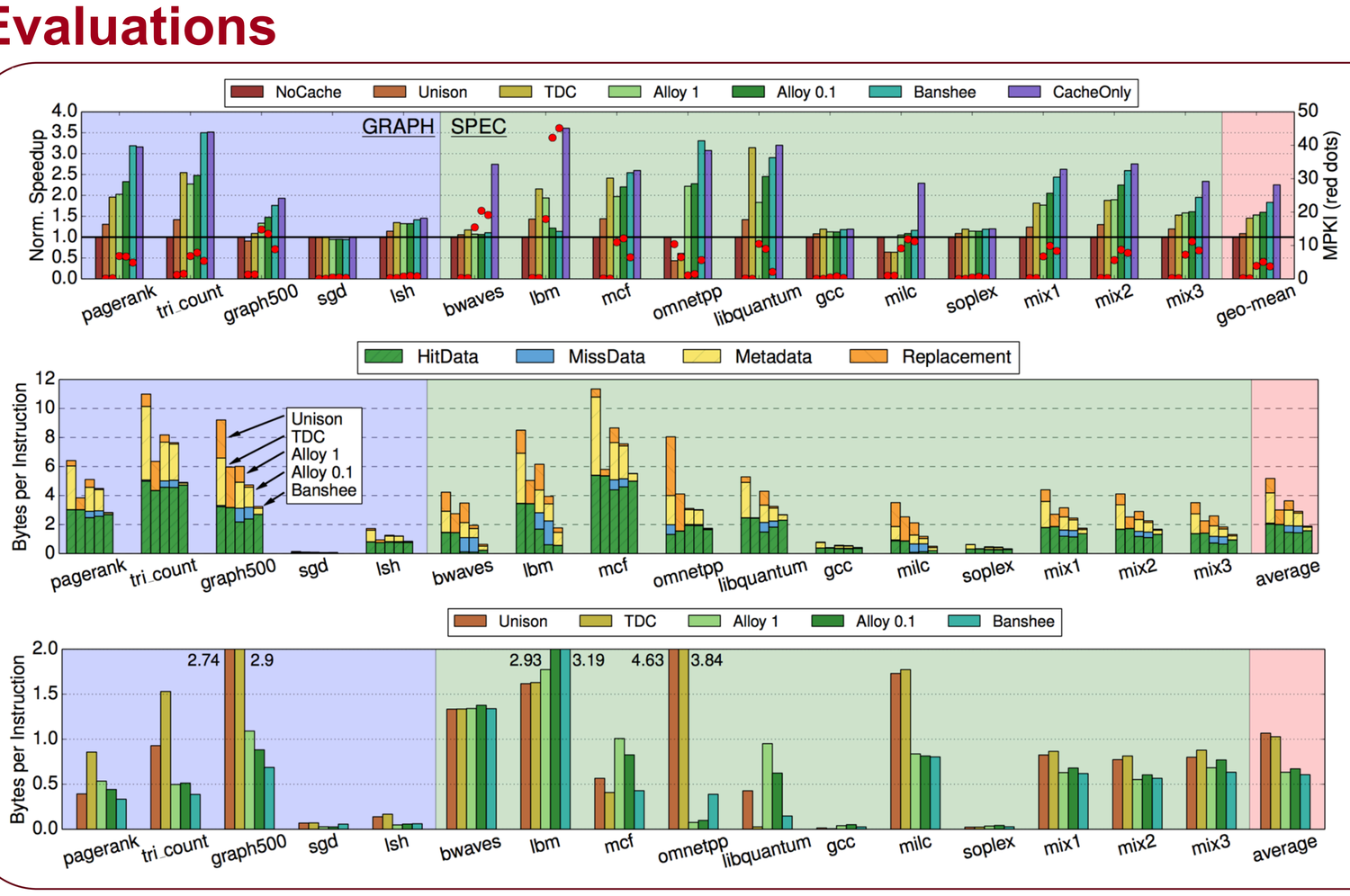
* Assuming 4-way set-associative DRAM cache

Idea 2: Bandwidth-Aware Cache Replacement

- DRAM cache replacement incurs significant DRAM traffic
  - Cache replacement traffic
  - Metadata traffic
- Limit cache replacement rate
  - Replace only when the incoming page's frequency counter is greater than the victim pages's counter by a threshold
- Reduce metadata traffic
  - Access frequency counters for a **randomly sampled** fraction of memory accesses



Evaluations



- Banshee improves performance by **15%** on average over the best-previous (i.e., BEAR) latency-optimized DRAM cache design
- Banshee reduces **36%** in-package DRAM traffic over the best-previous design
- Banshee reduces **3%** off-package DRAM traffic over the best-previous design