Lecture 8: Large Cache Design I

- Topics: Shared vs. private, centralized vs. decentralized, UCA vs. NUCA, recent papers
Multi-Core Cache Organizations

- Private L1 caches
- Shared L2 cache
- Bus between L1s and single L2 cache controller
- Snooping-based coherence between L1s
Multi-Core Cache Organizations

Private L1 caches
Shared L2 cache, but physically distributed
Bus connecting the four L1s and four L2 banks
Snooping-based coherence between L1s
Multi-Core Cache Organizations

Private L1 caches
Shared L2 cache, but physically distributed
Scalable network
Directory-based coherence between L1s
Multi-Core Cache Organizations

Private L1 caches
Private L2 caches
Scalable network
Directory-based coherence between L2s
(through a separate directory)
Shared Vs. Private

• SHR: No replication of blocks
• SHR: Dynamic allocation of space among cores
• SHR: Low latency for shared data in LLC (no indirection thru directory)
• SHR: No interconnect traffic or tag replication to maintain directories
• PVT: More isolation and better quality-of-service
• PVT: Lower wire traversal when accessing LLC hits, on average
• PVT: Lower contention when accessing some shared data
• PVT: No need for software support to maintain data proximity
Innovations for Private Caches: Cooperation

- Cooperative Caching, Chang and Sohi, ISCA’06

- Prioritize replicated blocks for eviction with a given probability; directory must track and communicate a block’s “replica” status

- “Singlet” blocks are sent to sibling caches upon eviction (probabilistic one-chance forwarding); blocks are placed in LRU position of sibling
Dynamic Spill-Receive

- Dynamic Spill-Receive, Qureshi, HPCA’09

- Instead of forcing a block upon a sibling, designate caches as Spillers and Receivers and all cooperation is between Spillers and Receivers

- Every cache designates a few of its sets as being Spillers and a few of its sets as being Receivers (each cache picks different sets for this profiling)

- Each private cache independently tracks the global miss rate on its S/R sets (either by watching the bus or at the directory)

- The sets with the winning policy determine the policy for the rest of that private cache – referred to as set-dueling
Innovations for Shared Caches: NUCA

Issues to be addressed for Non-Uniform Cache Access:

- Mapping
- Migration
- Search
- Replication
Static and Dynamic NUCA

• Static NUCA (S-NUCA)
  ▪ The address index bits determine where the block is placed; sets are distributed across banks
  ▪ Page coloring can help here to improve locality

• Dynamic NUCA (D-NUCA)
  ▪ Ways are distributed across banks
  ▪ Blocks are allowed to move between banks: need some search mechanism
  ▪ Each core can maintain a partial tag structure so they have an idea of where the data might be (complex!)
  ▪ Every possible bank is looked up and the search propagates (either in series or in parallel) (complex!)
Latency 13-17cyc

Latency 65 cyc

Data must be placed close to the center-of-gravity of requests.
Alternative Layout

From Huh et al., ICS’05:

- Paper also introduces the notion of sharing degree

- A bank can be shared by any number of cores between N=1 and 16.

- Will need support for L2 coherence as well
Victim Replication, Zhang & Asanovic, ISCA’05

- Large shared L2 cache (each core has a local slice)
- On an L1 eviction, place the victim in local L2 slice (if there are unused lines)
- The replication does not impact correctness as this core is still in the sharer list and will receive invalidations
- On an L1 miss, the local L2 slice is checked before forwarding the request to the correct slice
Page Coloring

CACHE VIEW

OS VIEW

Bank number with Page-to-Bank

Bank number with Set-interleaving

Tag

Set Index

Block offset

Physical page number

Page offset
Cho and Jin, MICRO’06

- Page coloring to improve proximity of data and computation
- Flexible software policies
- Has the benefits of S-NUCA (each address has a unique location and no search is required)
- Has the benefits of D-NUCA (page re-mapping can help migrate data, although at a page granularity)
- Easily extends to multi-core and can easily mimic the behavior of private caches
• Awasthi et al., HPCA’09 propose a mechanism for hardware-based re-coloring of pages without requiring copies in DRAM memory

• They also formalize the cost functions that determine the optimal home for a page
A page is categorized as “shared instruction”, “private data”, or “shared data”; the TLB tracks this and prevents access of a different kind.

Depending on the page type, the indexing function into the shared cache is different:

- “Private data” only looks up the local bank
- “Shared instruction” looks up a region of 4 banks
- “Shared data” looks up all the banks
Rotational Interleaving

- Can allow for arbitrary group sizes and a numbering that distributes load
Title

- Bullet