Memories

Chapter 12

Caches And Caching

Topics

- Introduction
- Definition
- Key idea
- Characteristics of a cache
- Importance of caching
- Examples of caching
- Terminology
- Locality of reference
- Best and worst case cache performance

Topics

- Hit and miss ratio
- Cache replacement policy
- LRU
- Hierarchy of Caches
- Improving Cache Performance
- Caching in Virtual and Physical Memory
- Improving performance through parallelism
- Write Through and Write Back

Topics

- Cache coherence
- L1, L2, and L3 caches
- Cache Flush
- Summary

Introduction

Several slides missing

TLB and Demand Paging

- TLB
  - TLB is a cache, it holds selected page entries
  - Whenever MMU looks up a page entry, it stores it in the TLB
  - TLB uses LRU
- Demand Paging
  - is a form for caching, view physical memory as cache and external storage as large data storage
  - physical memory only holds fraction of total pages

Demand Paging

Cache analysis shows that using demand paging on a computer system with a small physical memory can perform almost as well as the if the computer had a physical memory large enough for the entire virtual address space

- Improving performance through parallelism

  To achieve high performance, a memory cache is designed to simultaneously search the local cache and access the underlying memory. Parallelism complicates the hardware.

Write Through and Write Back

- Performance and Coherence Issues
- Caching improves read performance, not write performance

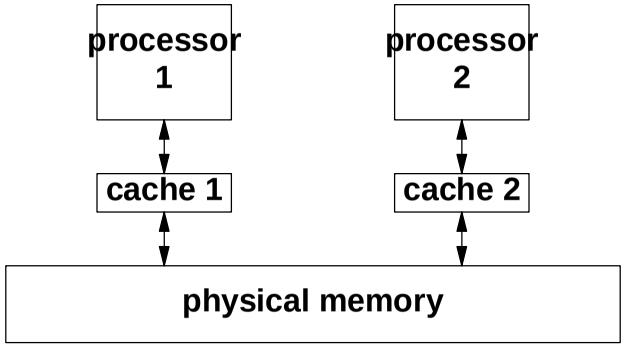
- Write through
  - when cache is written to, update underlying memory immediately of the change

- Write-back
  - cache keeps data item locally, and updates memory when it needs to be replaced.
  - uses dirty bit to keep track

Write-back improves performance

- What if data item in cache updated several times before being replaced? write-back avoids multiple updates to memory
- Will old value of data in memory be accessed? Not with single processor. Data values are accessed from the closest storage, here cache values will be retrieved

Cache coherence



- Two processors sharing an underlying memory. Because each processor has a seperate cache, conflicts can occur if both processors reference the same memory address.

- Solution: need cache coherence protocol

L1, L2, and L3 caches

Computer systems use a multilevel cache hierarchy in which an L1 cache is embedded on the processor chip, and an L2 is external to the processor. In the best case, the two-level cache makes the cost of accessing memory approximately the same as the cost of accessing a register

Processor	L1 Cache	L2 Cache	L3 Cache
Itanium 2	32KB	256KB	3MB, 4MB, or 6MB
Itanium	32KB	96KB	2MB or 4MB
Xeon MP	8KB	256KB or 512KB	512KB, 1MB or 2MB
P4	8KB	512KB	-

Example cache sizes.

Cache Flush

- How can a cache resolve the ambiguity that occurs because multiple applications the same range of addresses?
  - cache flush operation
  - disambiguating identifier
- Cache flushing
  - removing all values from the cache
  - cache must be flushed whenever OS changes to new virtual address space
- Disambiguation
  - use extra bits that identify the address space

Implementation of Memory Caching

Slide Missing

Summary

- Caching is a fundamental optimization technique
- TLB and demand paging are also forms of caching
- Cache intercepts requests, automatically stores values, and answers requests
  whenever possible
- Caches can be organized in a multilevel hierarchy