Memory Hierarchies

Vipin Vasu

February 18, 2019

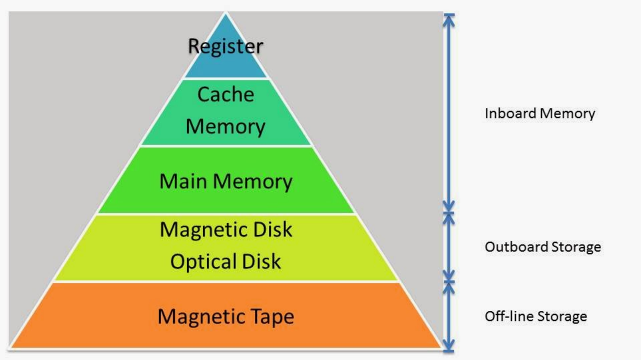
Outline

1. Introduction
2. Memory Hierarchy
3. Cache
4. Cache Mapping
5. Prefetching
6. Conclusion

What I'm gonna do right now

- Give a basic outline of Memory Hierarchy
- Get in depth with cache memory
- Tell a bit about cache mapping and cache mapping techniques
- Prefetching stuff

Hierarchy


Register
Cache Memory
Main Memory
Magnetic Disk
Optical Disk
Magnetic Tape
Inboard Memory
Outboard Storage
Off-line Storage

Why cache?

- Low Capacity,High Speed memory
- Helps Remove DRAM Gap
- and More..

How Cache Works?

- Split into Levels. Closer levels are split into Instruction and Data Caches while Outer Caches are unified.
- Checks of data are made from inner levels and progress out to outer level until memory.
- Hardware implemented algorithm to evict old items from cache

Performance Gain of Cache

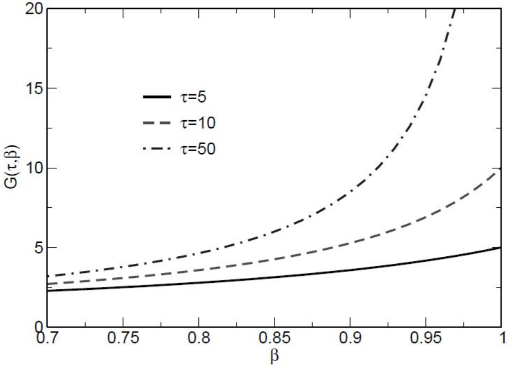


Figure 1.9: The performance gain from accessing data from cache versus the cache reuse ratio, with the speed advantage of cache versus main memory being parametrized by τ.

Locality of reference

Cache is advantageous only if application shows some locality of reference both spatial and temporal.
Consider example of a for loop.

Writing into Cache

If data to be written out already resides in cache, a write hit occurs. There are several possibilities for handling this case, but usually outermost caches work with a write-back strategy: The cache line is modified in cache and written to memory as a whole when evicted.

...aaand If its not there

- Nontemporal stores. These are special store instructions that bypass all cache levels and write directly to memory.
- Cache line zero. Special instructions zero out a cache line and mark it as modified without a prior read. The data is written to memory when evicted.

Cache Mapping

- Fully Associative Mapping(All for All)
- Direct Mapped Cache(Divide the memory equally)
- Set Associative

Why Prefetch

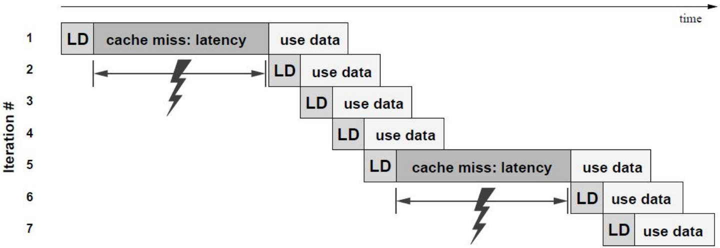
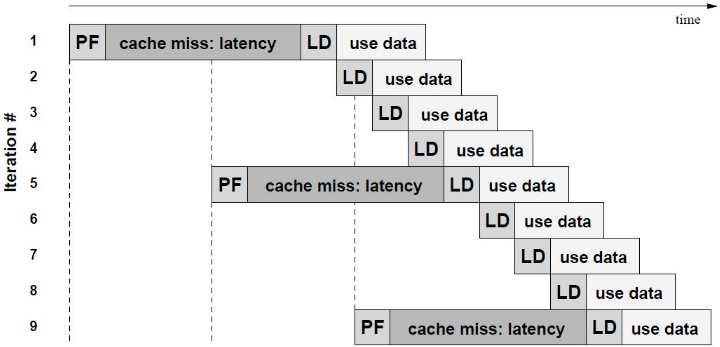


Figure 1.12: Timing diagram on the influence of cache misses and subsequent latency penalties for a vector norm loop. The penalty occurs on each new miss.

Prefetch

- Looks forward into the code to touch memory items which will be used in future
- Solves the issue of first miss
- Gets the data asynchronously so when the time comes the data is already available in cache.

The Change


time
Iteration #
1
2
3
4
5
6
7
8
9
PF
cache miss: latency
LD
use data
LD
use data
LD
use data
LD
use data
PF
cache miss: latency
LD
use data
LD
use data
LD
use data
LD
use data
PF
cache miss: latency
LD
use data

The End