Caches and Virtual Memory

Ryan Lee

6.823 Spring 2021

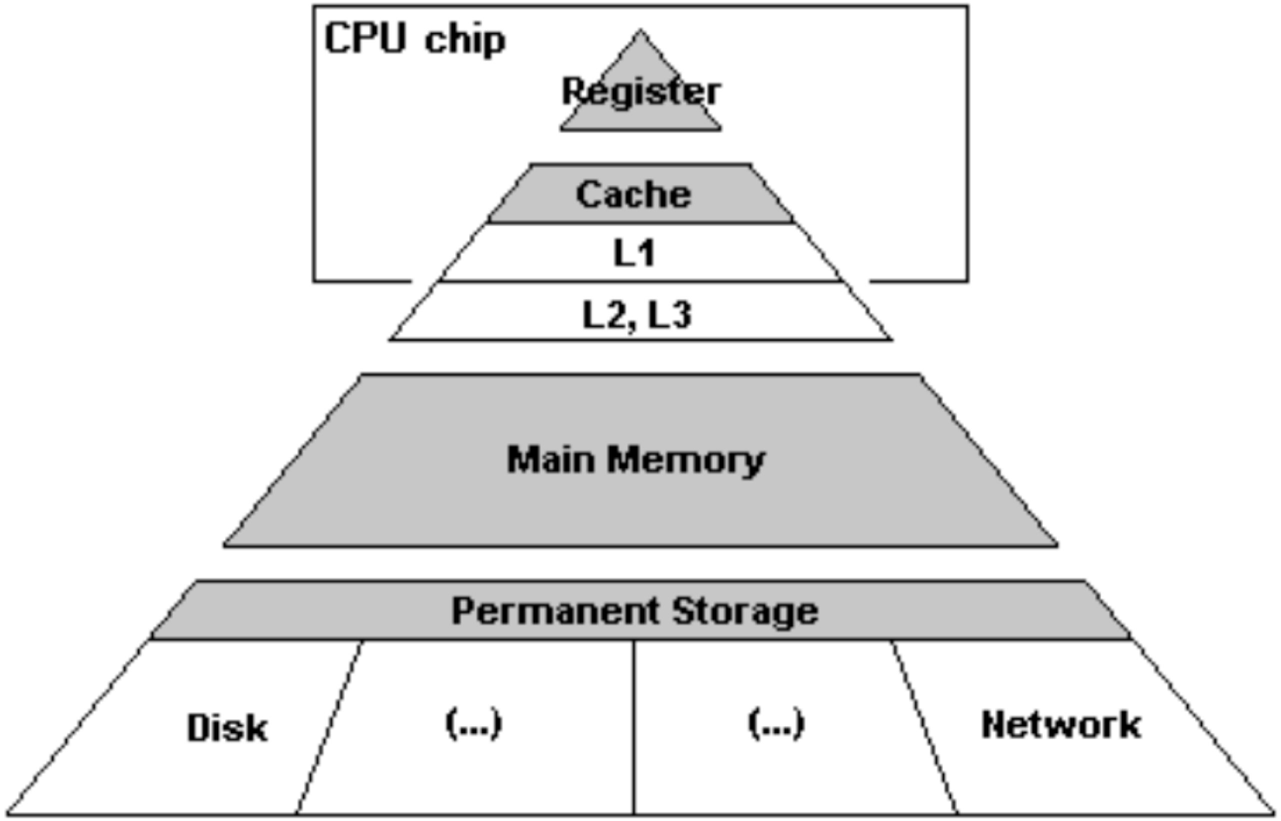
Recap: Caches

- A small, fast storage holding **frequently used data**

- Memory reference patterns exhibit locality
  - Spatial locality
  - Temporal locality

- Can have multiple levels with larger, slower caches

Recap: Caches



Cache organization

- Need to find a piece of data in the cache:
  - **Index**: determined by number of cache lines(sets)
  - **Block offset**: determined by number of bytes in a block
  - **Tag**: enough to uniquely identify blocks (i.e. rest of the bits of the address)

Address:	Tag	Index	Block Offset

Cache organization

- Direct-mapped vs. Set-associative
  - Flexibility of placement in the cache

- Common replacement policies
  - Random, Least Recently Used (LRU), FIFO, etc...

- Categorizing cache misses
  - Compulsory, Capacity, Conflict
- Average memory access time (AMAT)

Virtual Memory

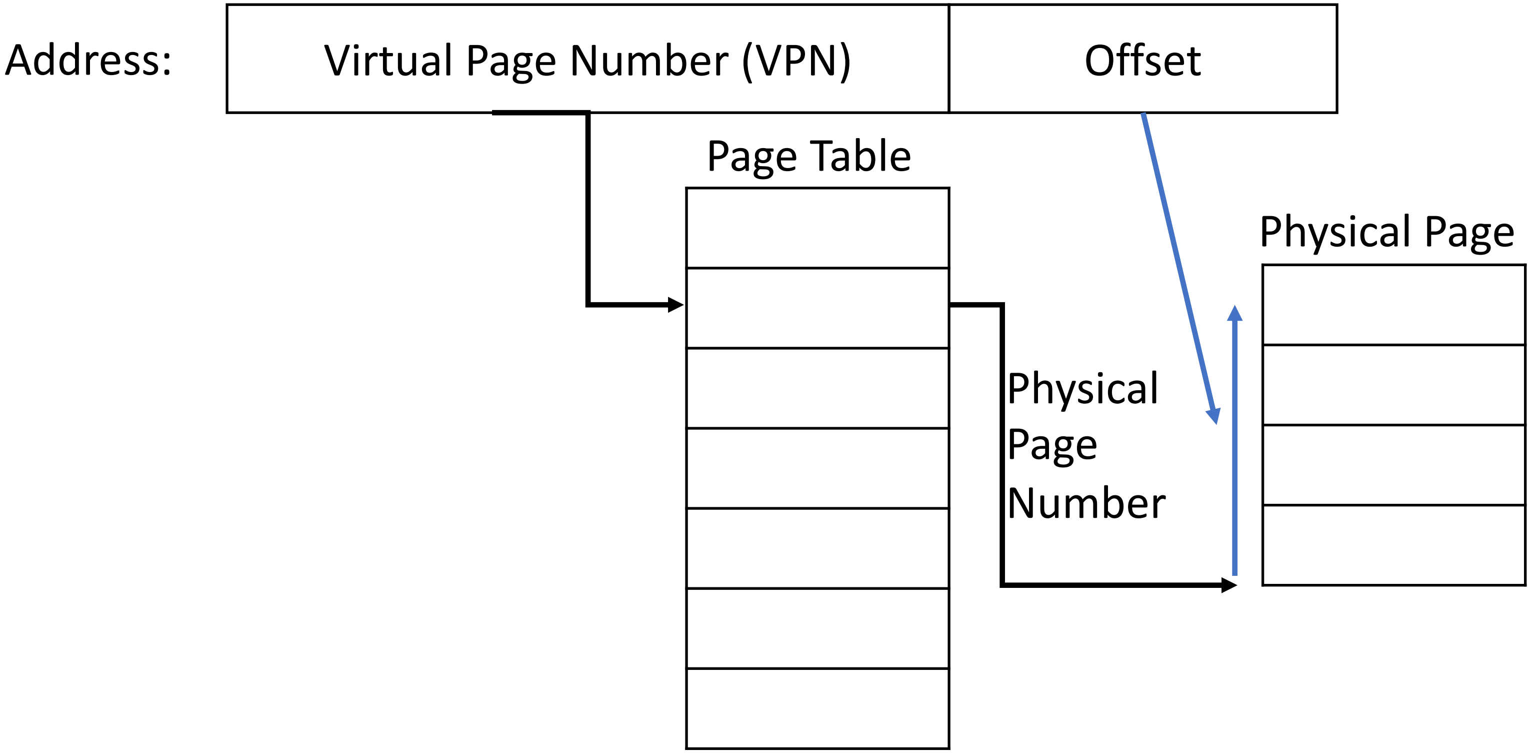
- Provide an illusion of large, private memory for each program
  - Protection
  - Privacy
  - Demand Paging

- Base & Bound

- Paged Memory Systems

Virtual Memory

- Processor-generated address is split into a virtual page number and offset



Virtual Memory

- Walking the page table is expensive
  - Translation Lookaside Buffer (TLB)

- Page tables are prohibitively large
  - Hierarchical page tables

- Caches can use either the virtual address or physical address
  - VIVT (Virtually Indexed Virtually Tagged)
  - PIPT (Physically Indexed Physically Tagged)
  - VIPT (Virtually Indexed Physically Tagged)