Caching 50.5*

COS 518: Advanced Computer Systems
Lecture 9
Michael Freedman

* Half of 101

Basic caching rule

- Tradeoff
  - Fast: Costly, small, close
  - Slow: Cheap, large, far

- Based on two assumptions
  - Temporal location: Will be accessed again soon
  - Spatial location: Nearby data will be accessed soon

Multi-level caching in hardware

Caching in distributed systems

Web Caching and Zipf-like Distributions: Evidence and Implications
Lee Breslau, Pei Cao, Li Fan, Graham Phillips, Scott Shenker
Caching common in distributed systems

- Web
  - Web proxies at edge of enterprise networks
  - “Server surrogates” in CDNs downstream of origin
- DNS
  - Caching popular NS, A records
- File sharing
  - Gnutella & flooding-based p2p networks

Caching within datacenter systems

load balancers  front-end web servers  DB / backend
identical  identical  partitioned
Caching within datacenter systems

- Load balancers
  - identical
- Front-end web servers
  - identical
  - partitioned
- DB/Backend
  - identical
  - partitioned

• Look aside cache

Cache management

- Write-through
  - Data written simultaneously to cache and storage
- Write-back
  - Data updated only in cache
  - On cache eviction, written “back” to storage
Caching within datacenter systems

function get_foo(foo_id)
    foo = memcached_get("foo:" + foo_id)
    return foo if defined foo

    foo = fetch_foo_from_database(foo_id)
    memcached_set("foo:" + foo_id, foo)
    return foo
end