Discussion 11: Cache

PEIFAN LI

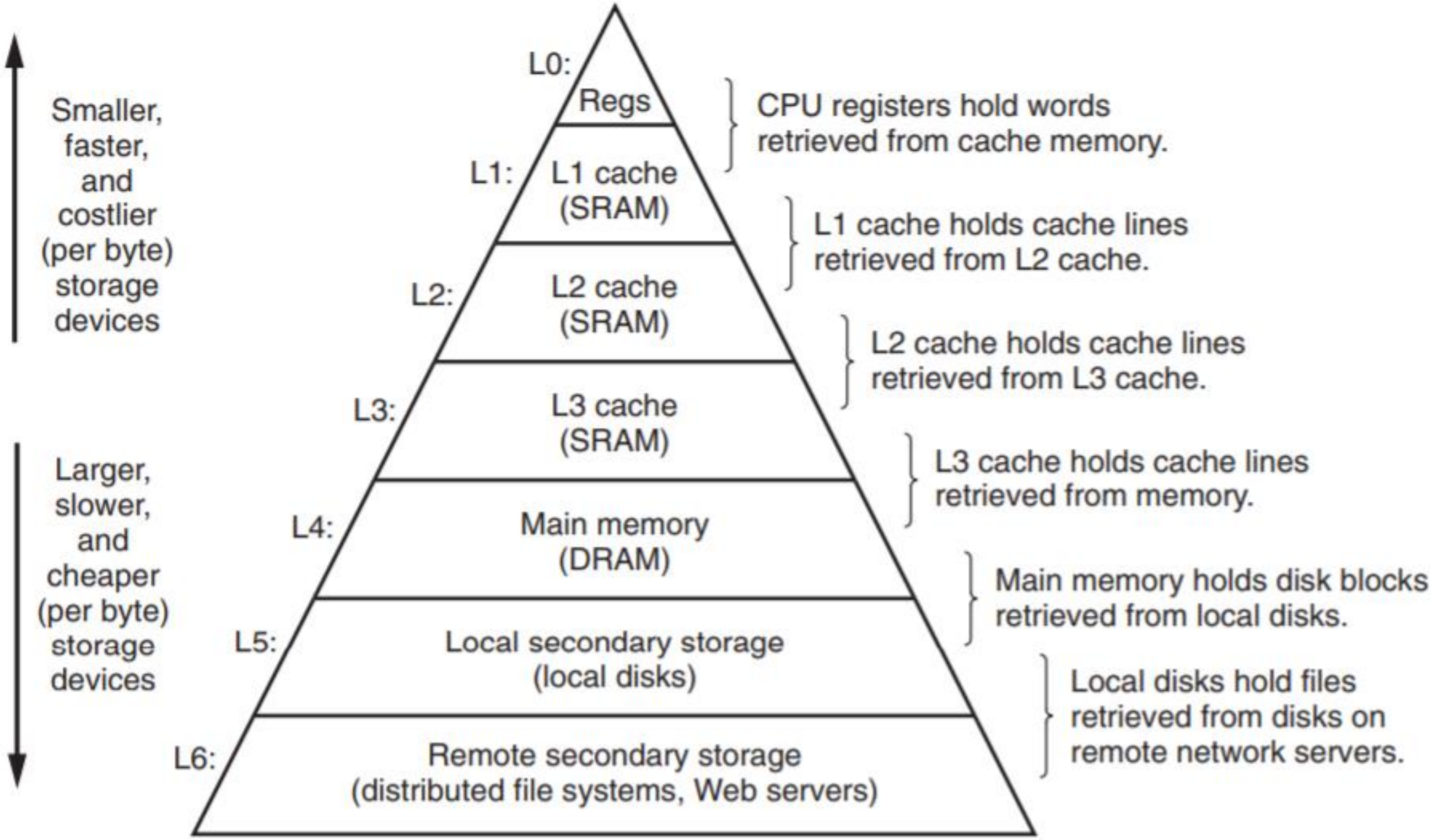
Memory Hierarchy

- Cache
  - SRAM: Static Random Access Memory
  - On-chip Component
  - Including L1, L2, L3 cache
- Main memory
  - DRAM: Dynamic Random Access Memory
  - E.g. DDR3(L): Double Data Rate 3 (Low Voltage) Synchronous Dynamic Random-Access Memory
- Disk
  - HDD: Hard Disk Drive
  - SSD: Solid State Drive

Memory Hierarchy

- Volatile Memory
  - DRAM
  - SRAM
- Non-volatile Memory
  - ROM: Read-Only Memory
  - Flash memory

Memory Hierarchy


Smaller,
faster,
and
costlier
(per byte)
storage
devices
Larger,
slower,
and
cheaper
(per byte)
storage
devices
L0:
Regs
L1:
L1 cache
(SRAM)
L2:
L2 cache
(SRAM)
L3:
L3 cache
(SRAM)
L4:
Main memory
(DRAM)
L5:
Local secondary storage
(local disks)
L6:
Remote secondary storage
(distributed file systems, Web servers)
CPU registers hold words retrieved from cache memory.
L1 cache holds cache lines retrieved from L2 cache.
L2 cache holds cache lines retrieved from L3 cache.
L3 cache holds cache lines retrieved from memory.
Main memory holds disk blocks retrieved from local disks.
Local disks hold files retrieved from disks on remote network servers.

Locality

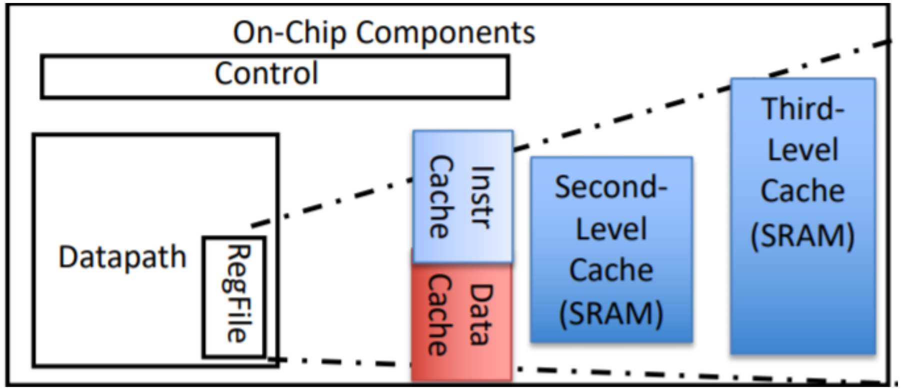
Principle of Locality

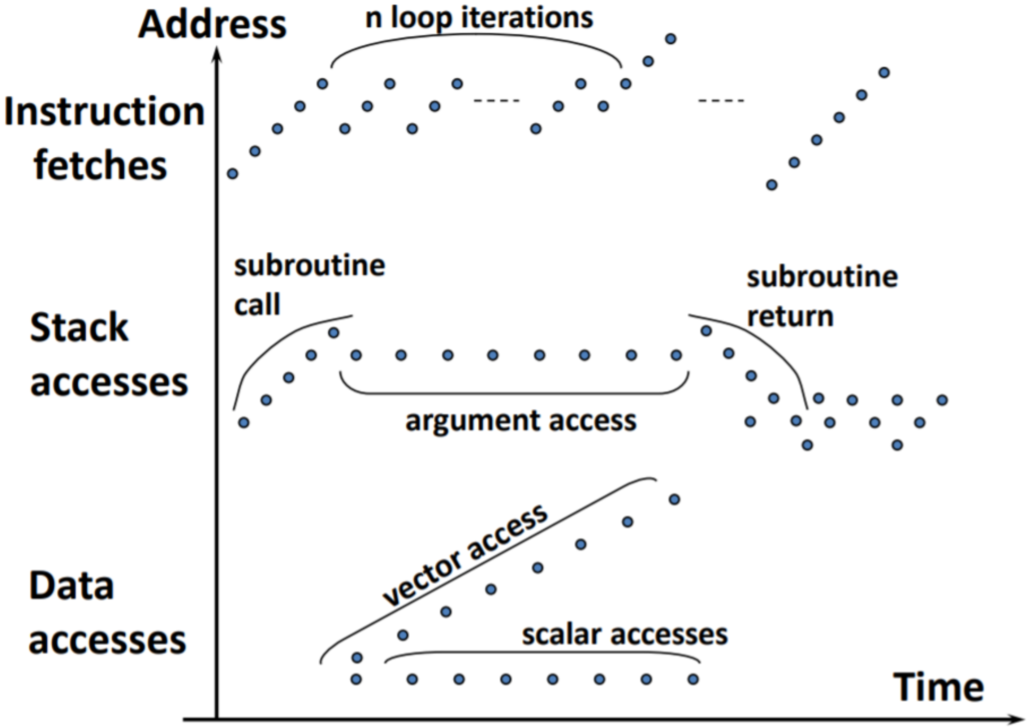
Programs access small portion of address space at any instant of time (spatial locality) and repeatedly access that portion (temporal locality)

Memory Reference Patterns

Instruction accesses & Data accesses

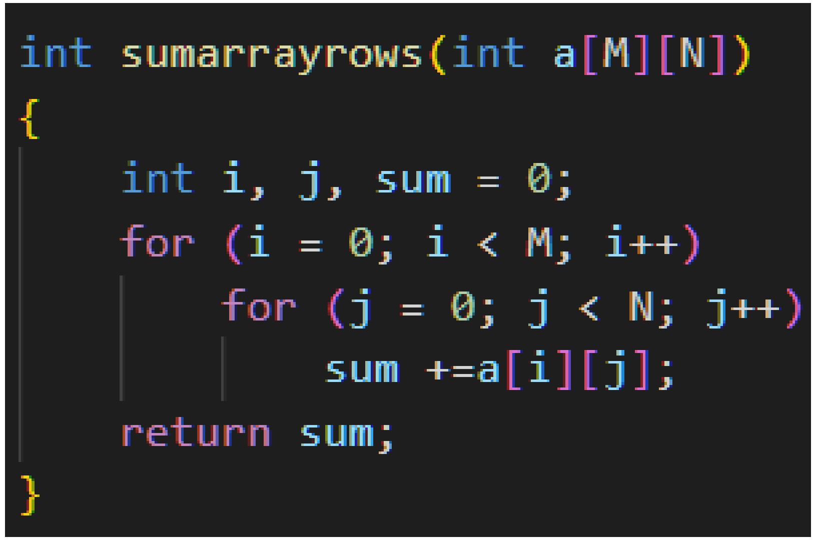
Separated L1 instruction/data cache design



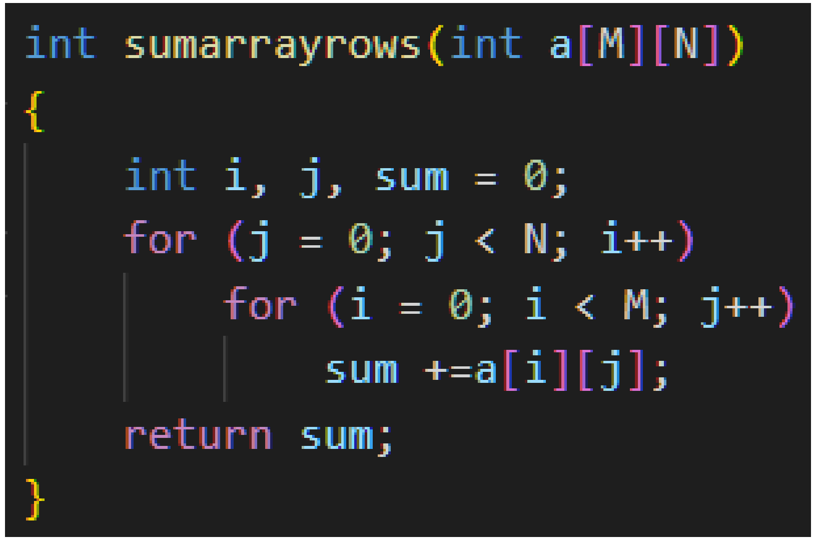


Locality

Which program has better temporal locality and spatial locality?

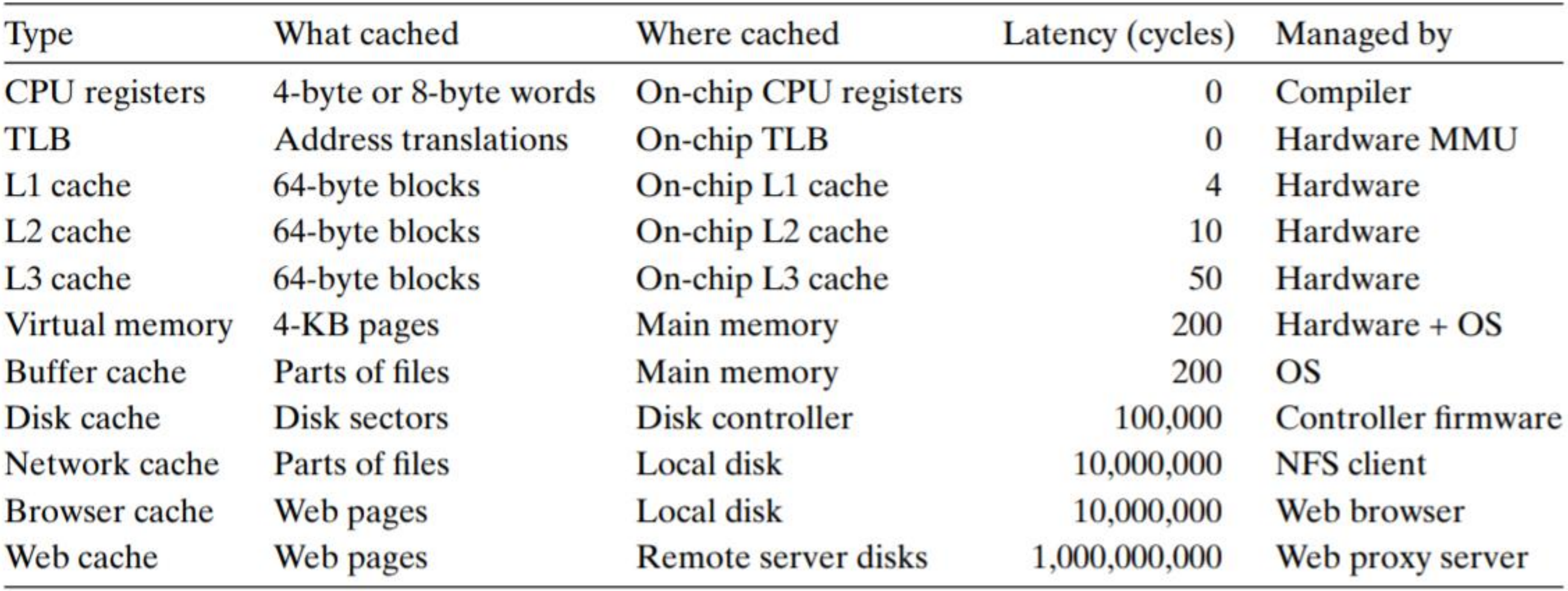


```
int sumarrayrows(int a[M][N])
{
    int i, j, sum = 0;
    for (i = 0; i < M; i++)
        for (j = 0; j < N; j++)
            sum +=a[i][j];
    return sum;
}
```



```
int sumarrayrows(int a[M][N])
{
    int i, j, sum = 0;
    for (j = 0; j < N; i++)
        for (i = 0; i < M; j++)
            sum +=a[i][j];
    return sum;
}
```

Cache



Type	What cached	Where cached	Latency (cycles)	Managed by
CPU registers	4-byte or 8-byte words	On-chip CPU registers	0	Compiler
TLB	Address translations	On-chip TLB	0	Hardware MMU
L1 cache	64-byte blocks	On-chip L1 cache	4	Hardware
L2 cache	64-byte blocks	On-chip L2 cache	10	Hardware
L3 cache	64-byte blocks	On-chip L3 cache	50	Hardware
Virtual memory	4-KB pages	Main memory	200	Hardware + OS
Buffer cache	Parts of files	Main memory	200	OS
Disk cache	Disk sectors	Disk controller	100,000	Controller firmware
Network cache	Parts of files	Local disk	10,000,000	NFS client
Browser cache	Web pages	Local disk	10,000,000	Web browser
Web cache	Web pages	Remote server disks	1,000,000,000	Web proxy server

TLB: translation lookaside buffer; MMU: memory management unit; OS: operating system; NFS: network file system.

Types of Cache

Direct Mapping cache

↑ size of set = 1

N-Way Set Associative

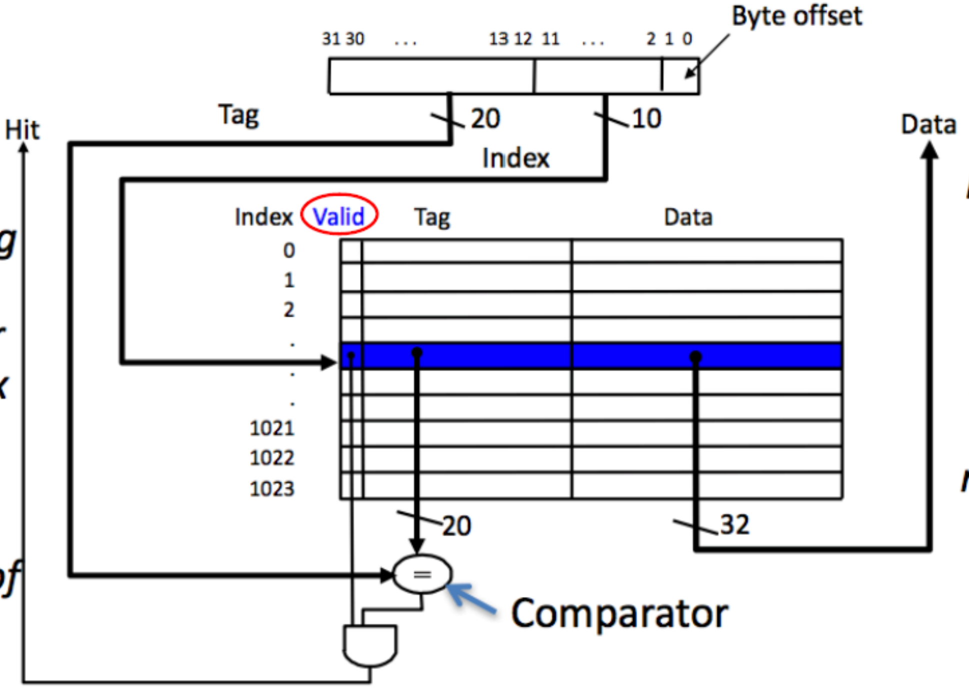
↓ # of set = 1

Fully Associative

Extra bits in a cache entry
What's Valid flag used for?

Valid bit ensures something useful in cache for this index

Compare Tag with upper part of Address to see if a Hit



Read data from cache instead of memory if a Hit

Cache Write Policy

Write-Through Policy

Immediately write data to the memory

Involve Write Buffer to reduce the overhead

Reliable

Write-Back Policy

Write data until it is evited from the cache

Involve the Dirty flag

Less reliable – why? inconsistency

What are the cache replacement policies? RAND, FIFO, LRU, LFU

Cache Miss

- Compulsory
- Capacity
- Conflict