NATIONAL RESEARCH UNIVERSITY

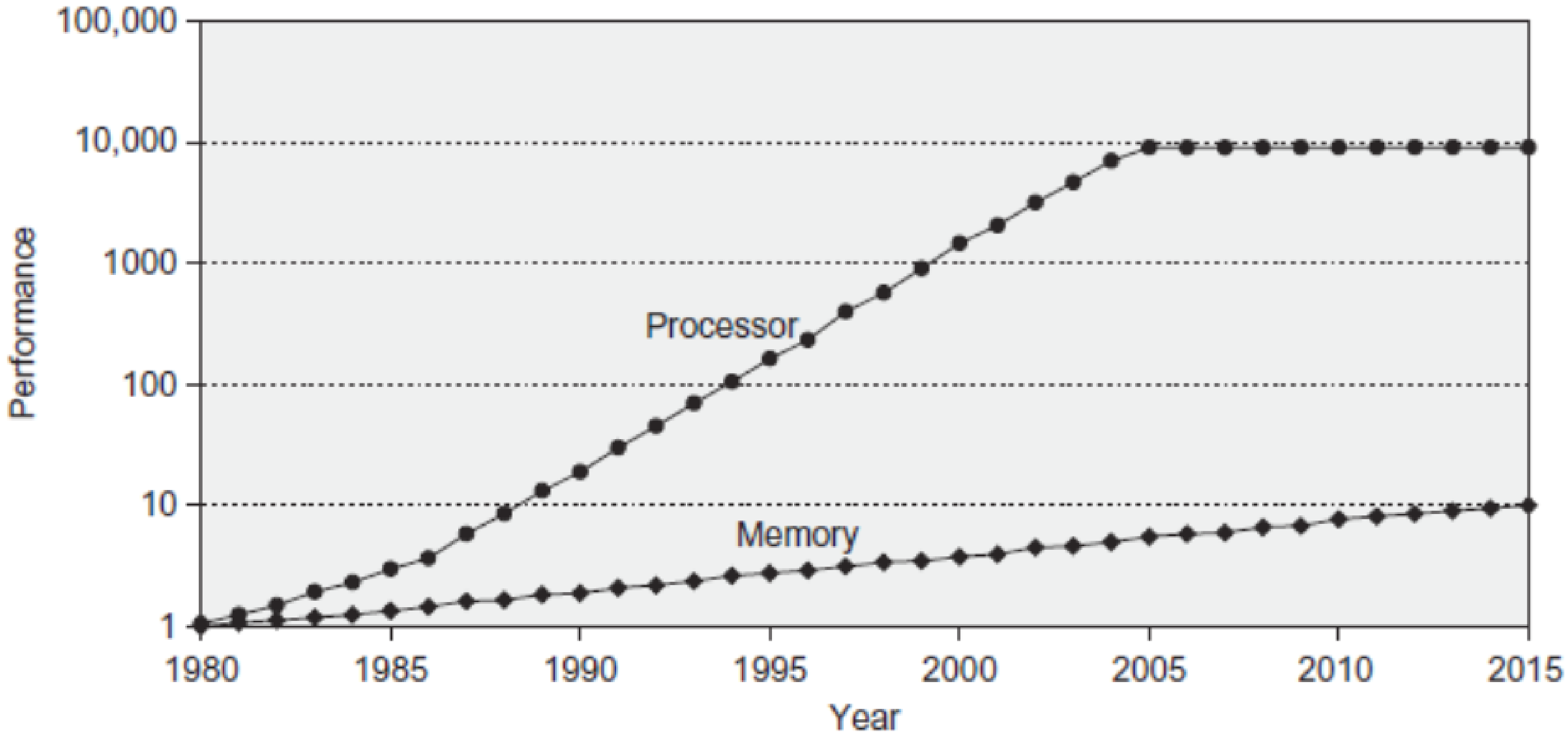
Computer Architecture and Operating Systems
Lecture 11: Memory and Caches

Andrei Tatarnikov

atatarnikov@hse.ru
@andrewt0301

Processor-Memory Performance Gap

- Computer performance depends on:
  - Processor performance
  - Memory performance



Memory Challenge

- Make memory appear as fast as processor
- Ideal memory:
  - Fast
  - Cheap (inexpensive)
  - Large (capacity)

But can only choose two!

Memory Technology

- Static RAM (SRAM)
  - 0.5ns – 2.5ns, $2000 – $5000 per GB
- Dynamic RAM (DRAM)
  - 50ns – 70ns, $20 – $75 per GB
- Magnetic disk
  - 5ms – 20ms, $0.20 – $2 per GB
- Ideal memory
  - Access time of SRAM
  - Capacity and cost/GB of disk

Locality

No need for large memory to access it fast

Just exploit locality

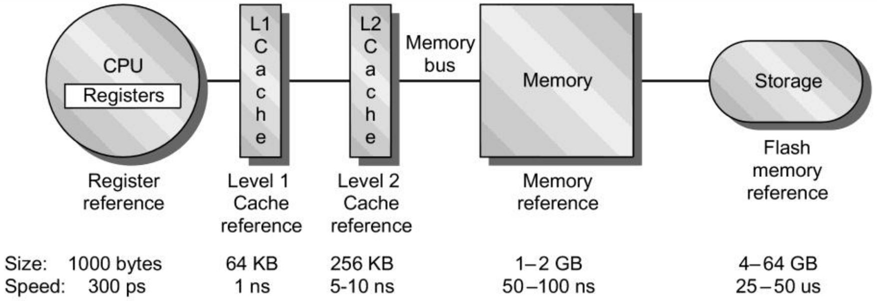
- Temporal Locality:
  - Locality in time
  - If data used recently, likely to use it again soon
  - How to exploit: keep recently accessed data in higher levels of memory hierarchy
- Spatial Locality:
  - Locality in space
  - If data used recently, likely to use nearby data soon
  - How to exploit: when access data, bring nearby data into higher levels of memory hierarchy too

Taking Advantage of Locality

- Memory hierarchy
- Store everything on disk
- Copy recently accessed (and nearby) items from disk to smaller DRAM memory
  - Main memory
- Copy more recently accessed (and nearby) items from DRAM to smaller SRAM memory
  - Cache memory attached to CPU

Memory Hierarchy

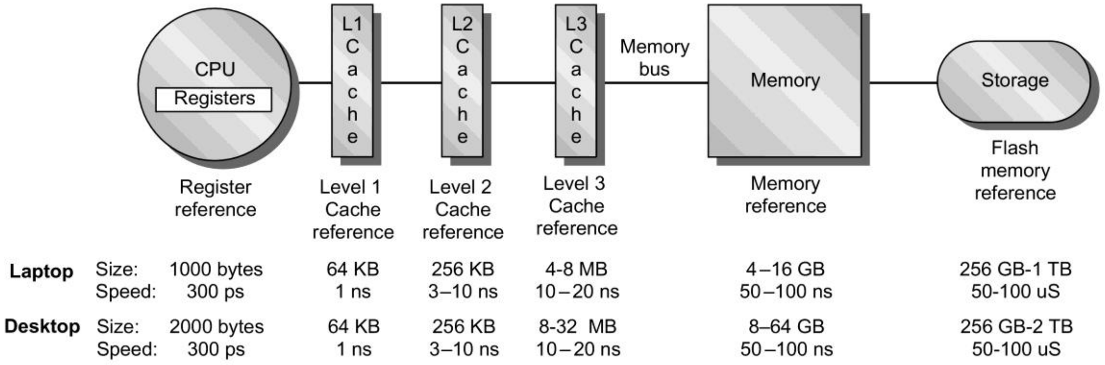
- Personal mobile device



	Register reference	Level 1 Cache reference	Level 2 Cache reference	Memory reference	Flash memory reference
Size:	1000 bytes	64 KB	256 KB	1–2 GB	4–64 GB
Speed:	300 ps	1 ns	5-10 ns	50–100 ns	25–50 us

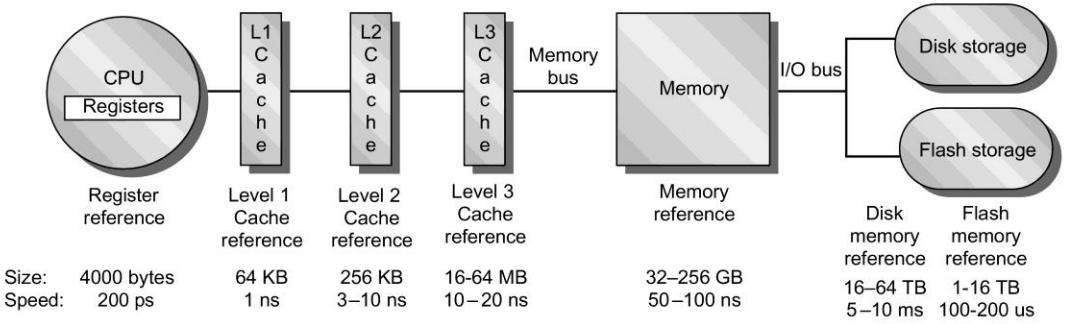
mobile device

- Laptop or desktop



		Register reference	Level 1 Cache reference	Level 2 Cache reference	Level 3 Cache reference	Memory reference	Flash memory reference
Laptop	Size:	1000 bytes	64 KB	256 KB	4-8 MB	4–16 GB	256 GB-1 TB
	Speed:	300 ps	1 ns	3–10 ns	10–20 ns	50–100 ns	50-100 uS
Desktop	Size:	2000 bytes	64 KB	256 KB	8-32 MB	8–64 GB	256 GB-2 TB
	Speed:	300 ps	1 ns	3–10 ns	10–20 ns	50–100 ns	50-100 uS

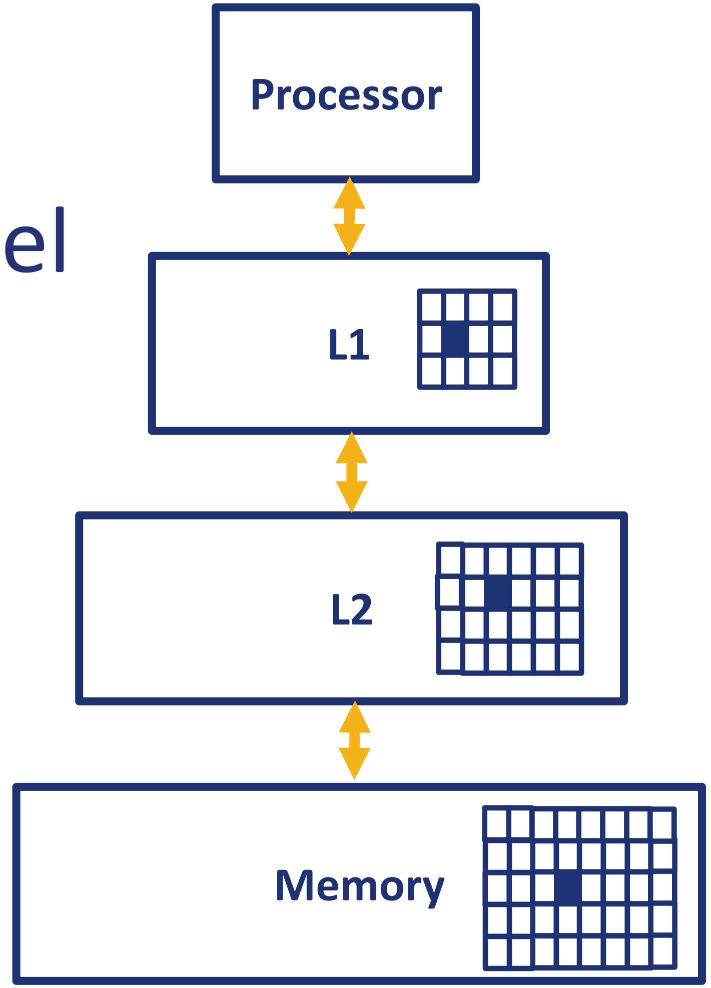
- Server



	Register reference	Level 1 Cache reference	Level 2 Cache reference	Level 3 Cache reference	Memory reference	Disk memory reference	Flash memory reference
Size:	4000 bytes	64 KB	256 KB	16-64 MB	32–256 GB	16–64 TB	1-16 TB
Speed:	200 ps	1 ns	3–10 ns	10–20 ns	50–100 ns	5–10 ms	100-200 us

How it works?

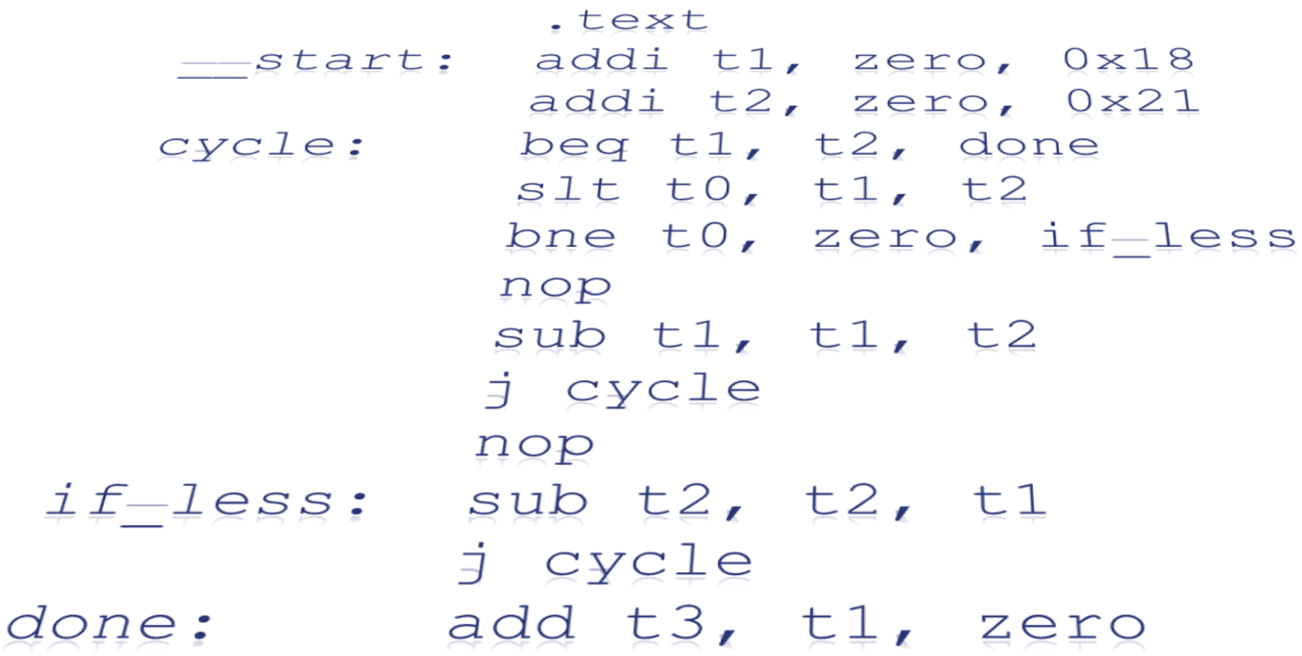
- Block (aka line): unit of copying
  - May be multiple words
- If accessed data is present in upper level
  - Hit: access satisfied by upper level
    - Hit ratio: hits/accesses
- If accessed data is absent
  - Miss: block copied from lower level
    - Time taken: miss penalty
    - Miss ratio: misses/accesses = 1 – hit ratio
  - Then accessed data supplied from upper level



Memory Performance

- **Hit**: data found in that level of memory hierarchy
- **Miss**: data not found (must go to next level)
  - **Hit Rate** = # hits / # memory accesses = 1 – Miss Rate
  - **Miss Rate** = # misses / # memory accesses = 1 – Hit Rate
- **Average memory access time (AMAT):** average time for processor to access data
  - **AMAT** = $t_{cache} + MR_{cache}[t_{MM} + MR_{MM}(t_{VM})]$

Any Questions?



```
            .text
__start:    addi t1, zero, 0x18
            addi t2, zero, 0x21
cycle:      beq t1, t2, done
            slt t0, t1, t2
            bne t0, zero, if_less
            nop
            sub t1, t1, t2
            j cycle
            nop
if_less:    sub t2, t2, t1
            j cycle
done:       add t3, t1, zero
```