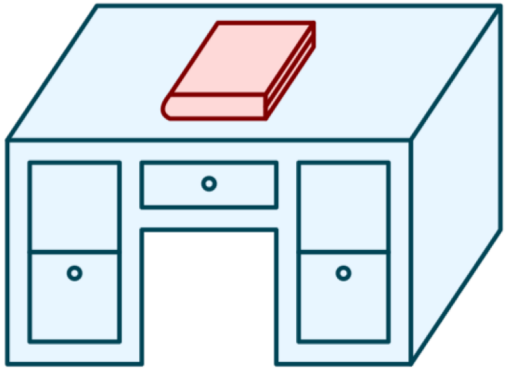
Lecture 10: Caches

CS 105

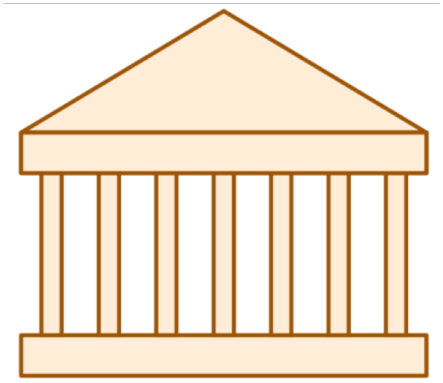
October 8, 2019

Life without caches

- You decide that you want to learn more about computer systems than is covered in this course
- The library contains all the books you could possibly want, but you don't like to study in libraries, you prefer to study at home.
- You have the following constraints:

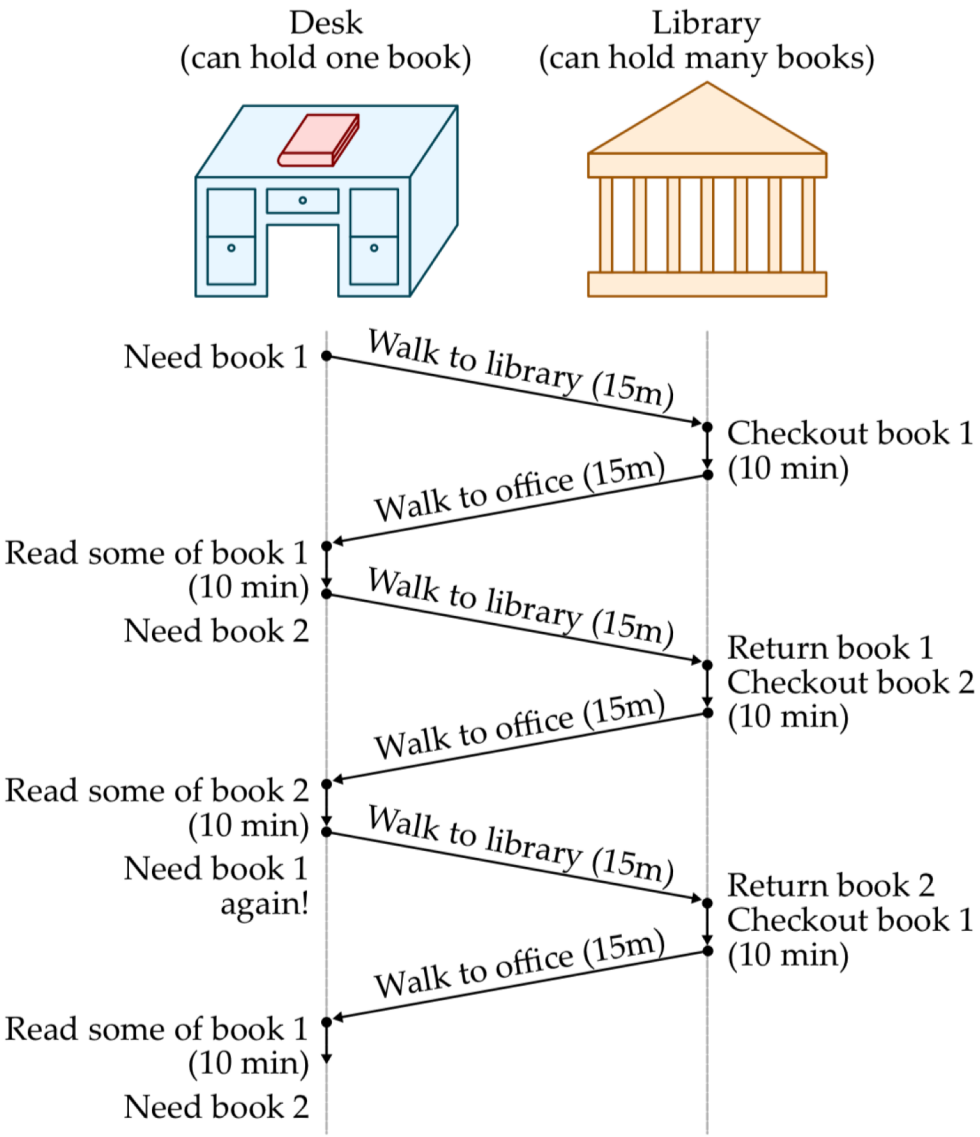


Desk
(can hold one book)



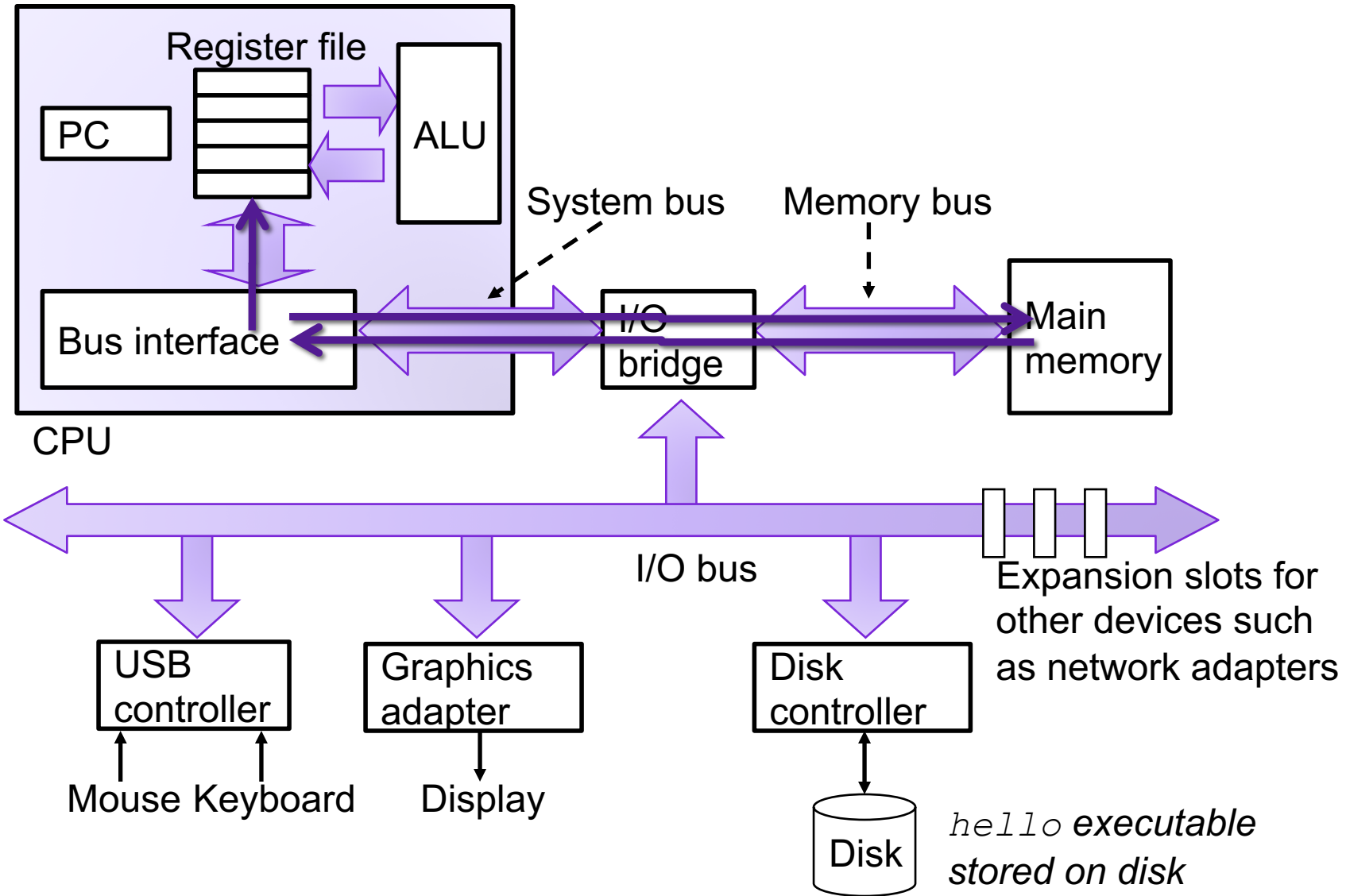
Library
(can hold many books)

Life without caches

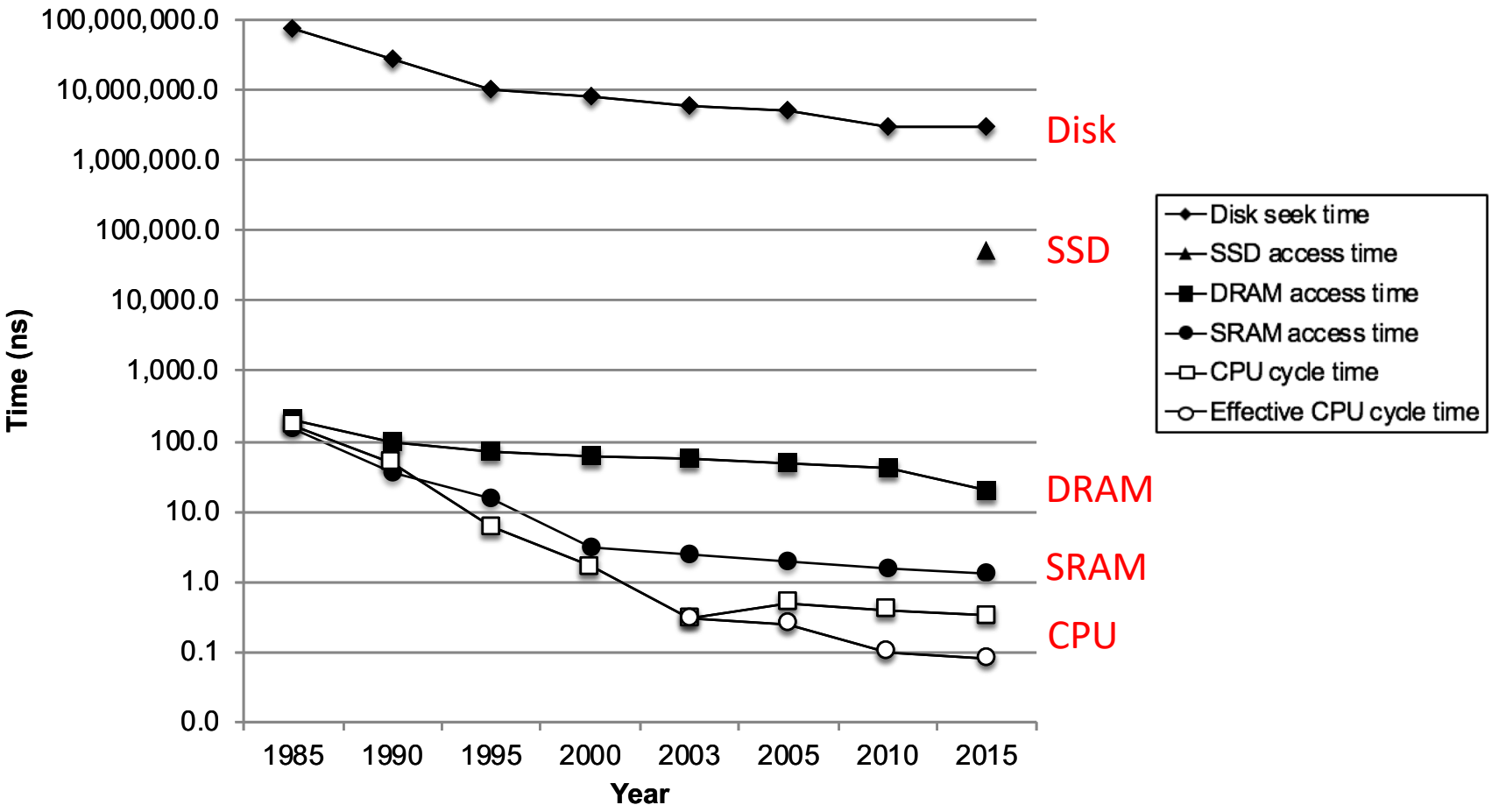


- Average latency to access a book: 40mins
- Average throughput (incl. reading time): 1.2 books/hr

A Computer System



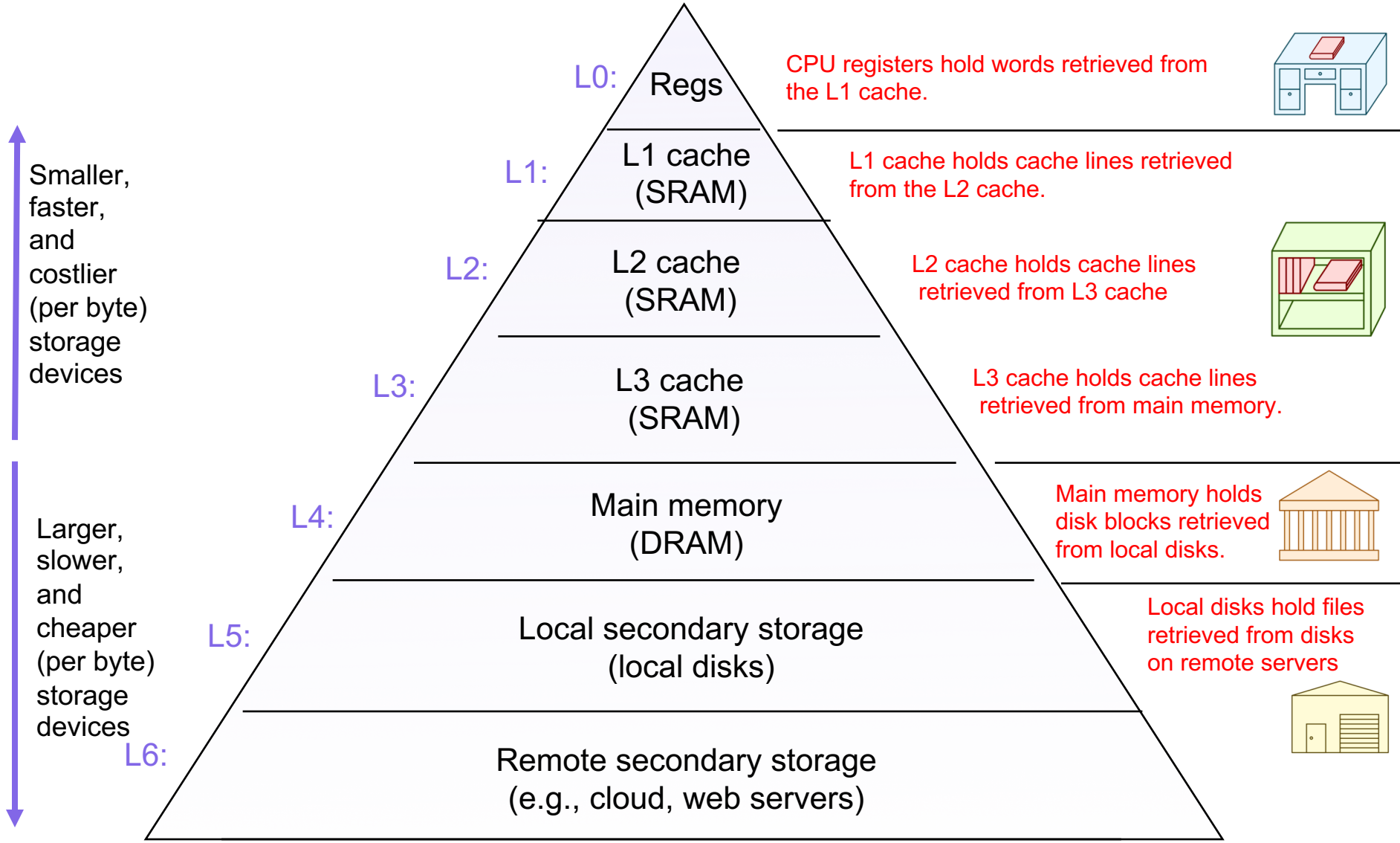
The CPU-Memory Gap



Caching—The Very Idea

- Keep some memory values nearby in fast memory
- Modern systems have 3 or even 4 levels of caches
- Cache idea is widely used:
  - Disk controllers
  - Web
  - (Virtual memory: main memory is a “cache” for the disk)

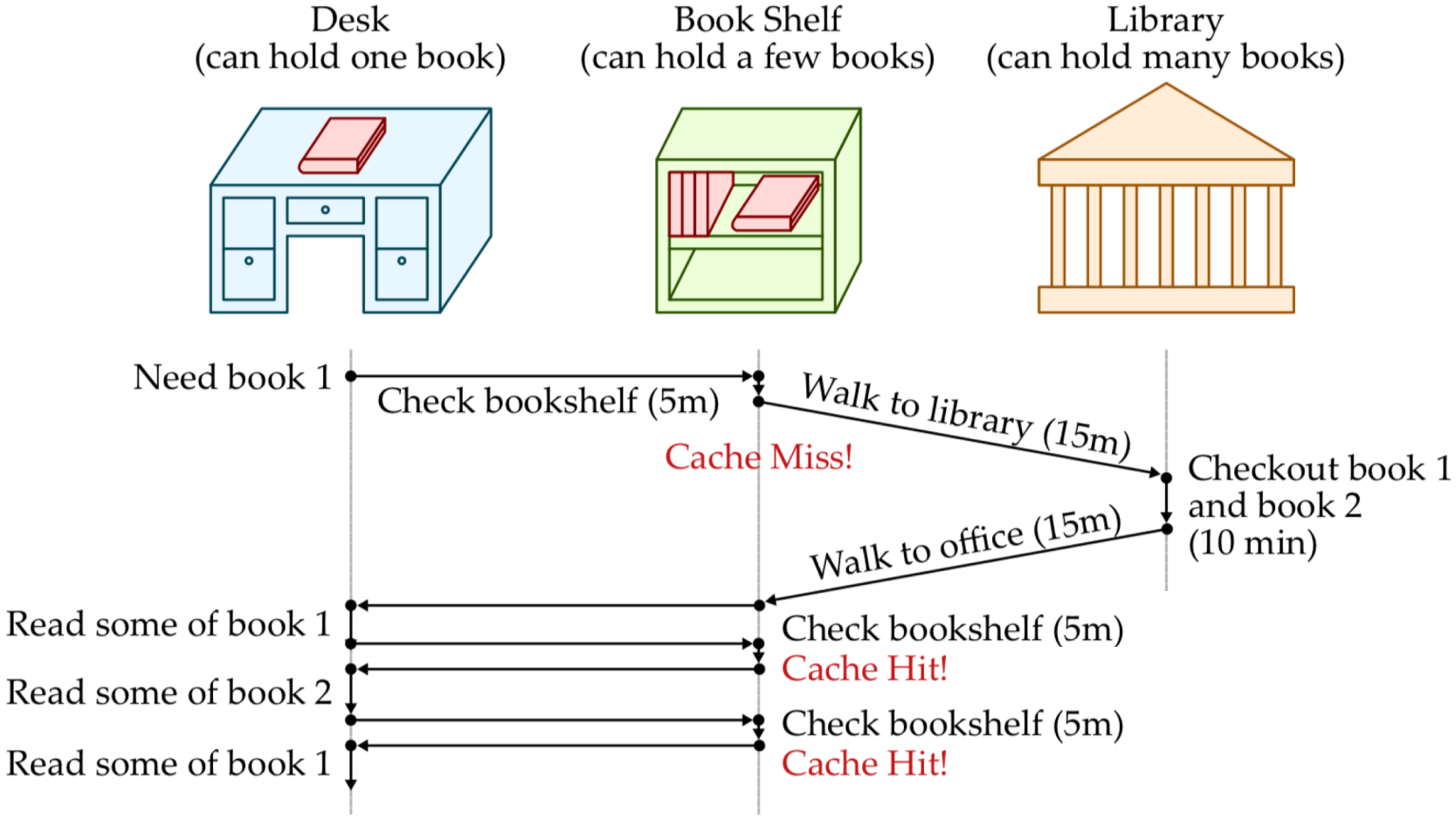
Memory Hierarchy



Latency numbers every programmer should know (2019)

L1 cache reference	1 ns	
Branch mispredict	3 ns	
L2 cache reference	4 ns	
Main memory reference	100 ns	
memory 1MB sequential read	4,000 ns	4 μs
SSD random read	16,000 ns	16 μs
SSD 1MB sequential read	62,000 ns	62 μs
Disk random read	3,000,000 ns	3 ms
Disk 1MB sequential read	947,000 ns	< 1 ms
Round trip in Datacenter	500,000 ns	500 μs
Round trip CA<->Europe	150,000,000 ns	150 ms

Life with caching



- Average latency to access a book: <20mins
- Average throughput (incl. reading time): ~2 books/hr

Caching—The Vocabulary

- **Size:** the total number of bytes that can be stored in the cache

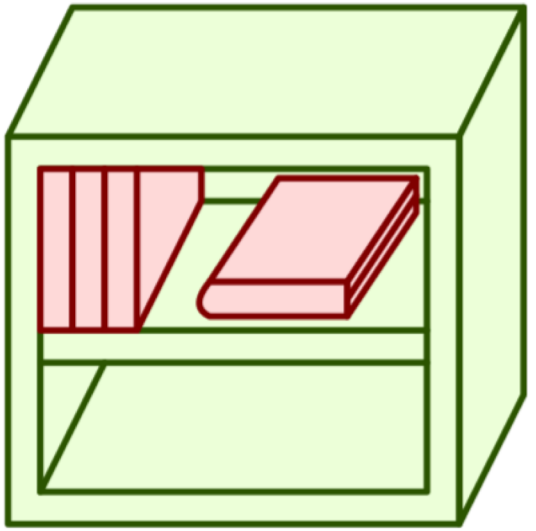
- **Cache Hit:** the desired value is in the cache and returned quickly
- **Cache Miss:** the desired value is not in the cache and must be fetched from a more distant cache (or ultimately from main memory)

- **Miss rate:** the fraction of accesses that are misses

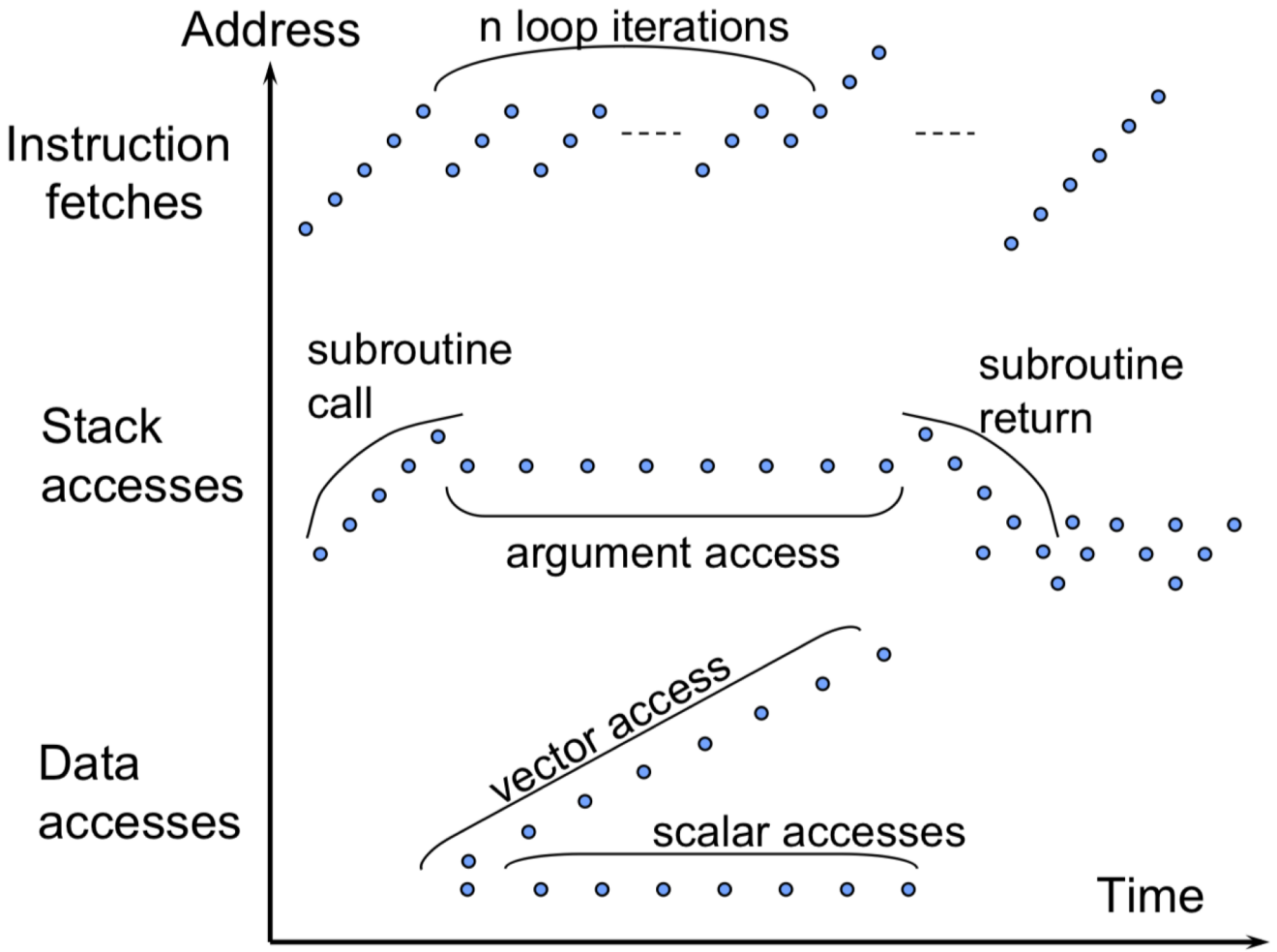
- **Hit time:** the time to process a hit
- **Miss penalty:** the *additional* time to process a miss

- **Average access time:** hit-time + miss-rate * miss-penalty

Question: how do we decide which books to put on the bookshelf?



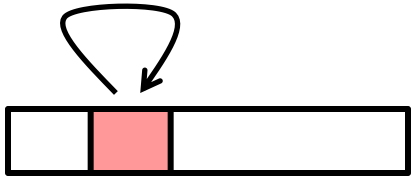
Example Access Patterns



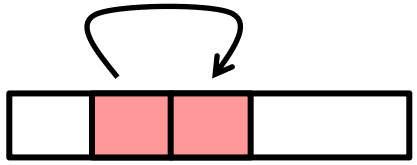
Principle of Locality

Programs tend to use data and instructions with addresses near or equal to those they have used recently

- Temporal locality:
  - Recently referenced items are likely to be referenced again in the near future



- Spatial locality:
  - Items with nearby addresses tend to be referenced close together in time



Locality Example

```
sum = 0;
for (i = 0; i < n; i++)
    sum += a[i];
return sum;
```

- Data references
  - Reference array elements in succession (stride-1 reference pattern).
  - Reference variable **`sum`** each iteration.
- Instruction references
  - Reference instructions in sequence.
  - Cycle through loop repeatedly.

Locality Example

- Which of the following functions is better in terms of locality with respect to array src?

```
void copyij(int src[2048][2048],
            int dst[2048][2048])
{
  int i,j;
  for (i = 0; i < 2048; i++)
    for (j = 0; j < 2048; j++)
      dst[i][j] = src[i][j];
}
```

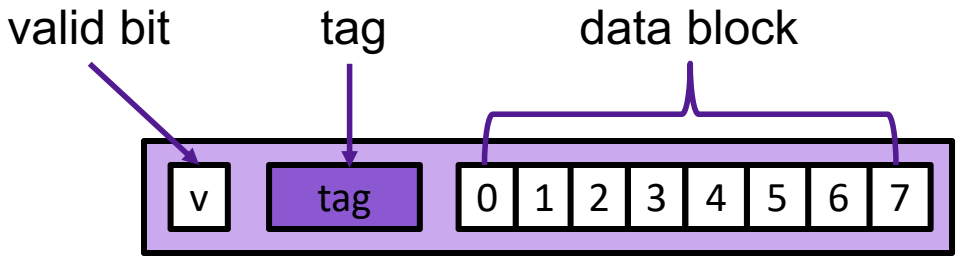
```
void copyji(int src[2048][2048],
            int dst[2048][2048])
{
  int i,j;
  for (j = 0; j < 2048; j++)
    for (i = 0; i < 2048; i++)
      dst[i][j] = src[i][j];
}
```

4.3ms 81.8ms

2.0 GHz Intel Core i7 Haswell

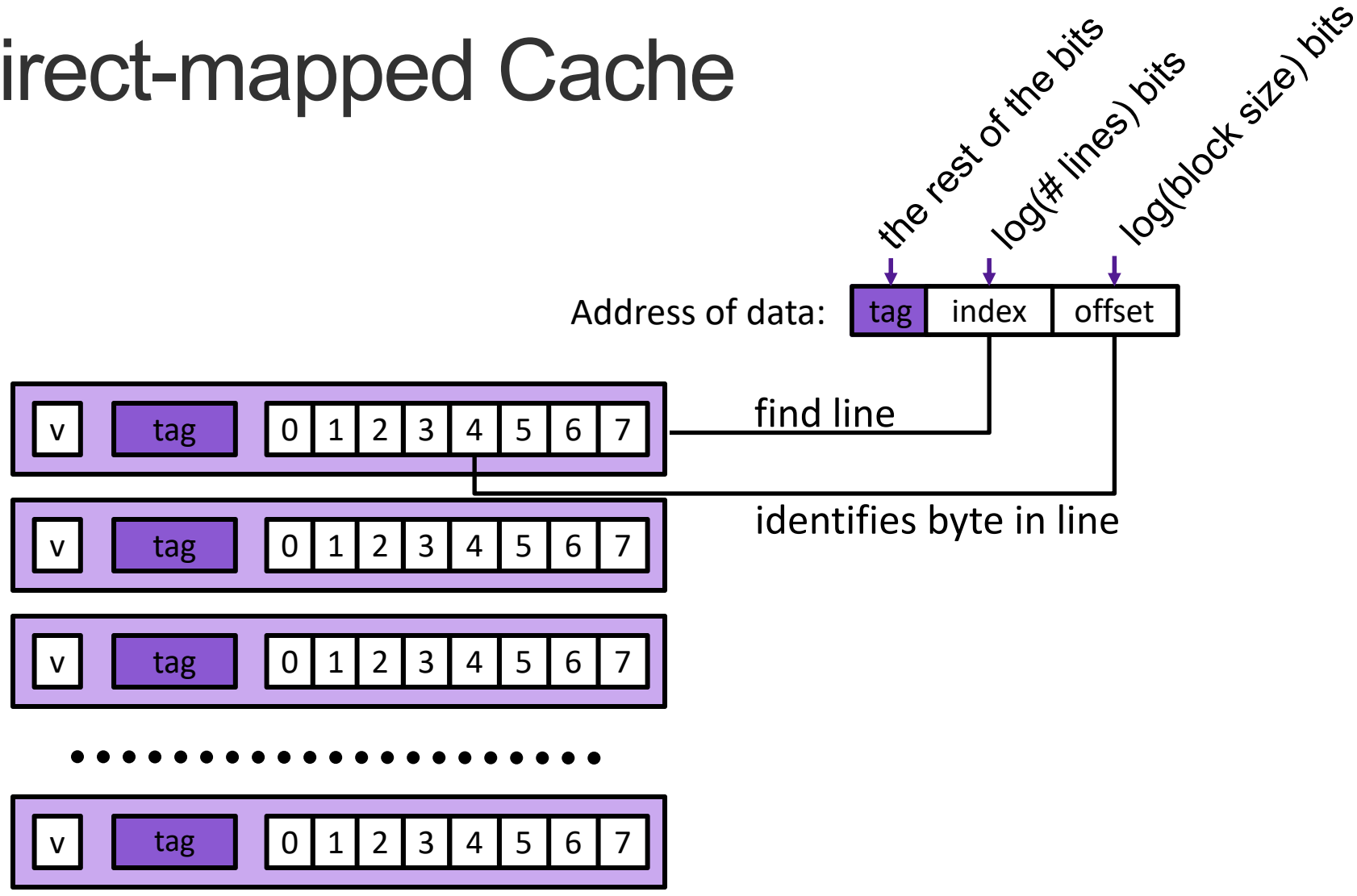
CACHE ORGANIZATION

Cache Lines



- **data block:** cached data
- **tag:** uniquely identifies which data is stored in the cache line
- **valid bit:** indicates whether or not the line contains meaningful information

Direct-mapped Cache



Example: Direct-mapped Cache

Assume: cache block size 8 bytes
Assume: assume 8-bit machine

Exercise: Direct-mapped Cache

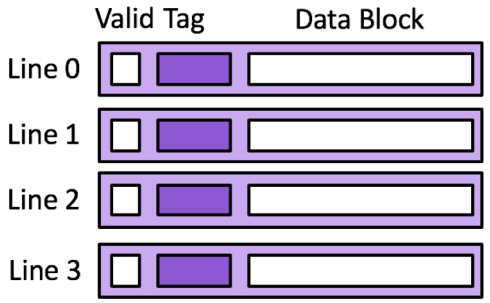
Transaction Stream

```
rd 0x00
rd 0x04
rd 0x10
rd 0x00
rd 0x04
rd 0x20
```

Memory

rd 0x14	18
rd 0x10	17
rd 0x0c	16
rd 0x08	15
rd 0x04	14
rd 0x00	13

Cache



Transaction	tag	index	h/m	Line 0	Line 1	Line 2	Line 3
rd 0x00				?	?	?	?
rd 0x04							
rd 0x10							
rd 0x00							
rd 0x04							
rd 0x10							

How well does this take advantage of spacial locality?
How well does this take advantage of temporal locality?