Caching

1/31/25

Administrivia

- HW 4 (histogram of letters) due tonight

Memory so far

- Big “array” indexed by the memory address
  - accessed by load and store of various types

- Implicitly assumed:
  - One memory, all of which is available to every program
  - Every cell equally fast to access

It’s all lies!!!

- Several different types of memory
- Some parts are much faster to access than others
- Memory addresses as used by the program are not the same as physical addresses
- Every program lives in its own address space (possibly with shared regions)

Types of memory

	access time (ns)	$ / GiB
SRAM	0.5 – 2.5	500 – 1,000
DRAM	50 – 70	10 – 20
Flash	5,000 – 50,000	0.75 – 1
magnetic disk	5 million – 20 million	0.05 – 0.10

Source: Patterson & Hennessy, “Computer organization and design”, 2014

Numbers from 2012

Types of memory

	access time (ns)	$ / GiB
SRAM	0.5 – 2.5	500 – 1,000
DRAM	50 – 70	10 – 20
Flash	5,000 – 50,000	0.75 – 1
magnetic disk	5 million – 20 million	0.05 – 0.10

Source: Patterson & Hennessy, "Computer organization and design", 2014
Numbers from 2012

Idea of caching:

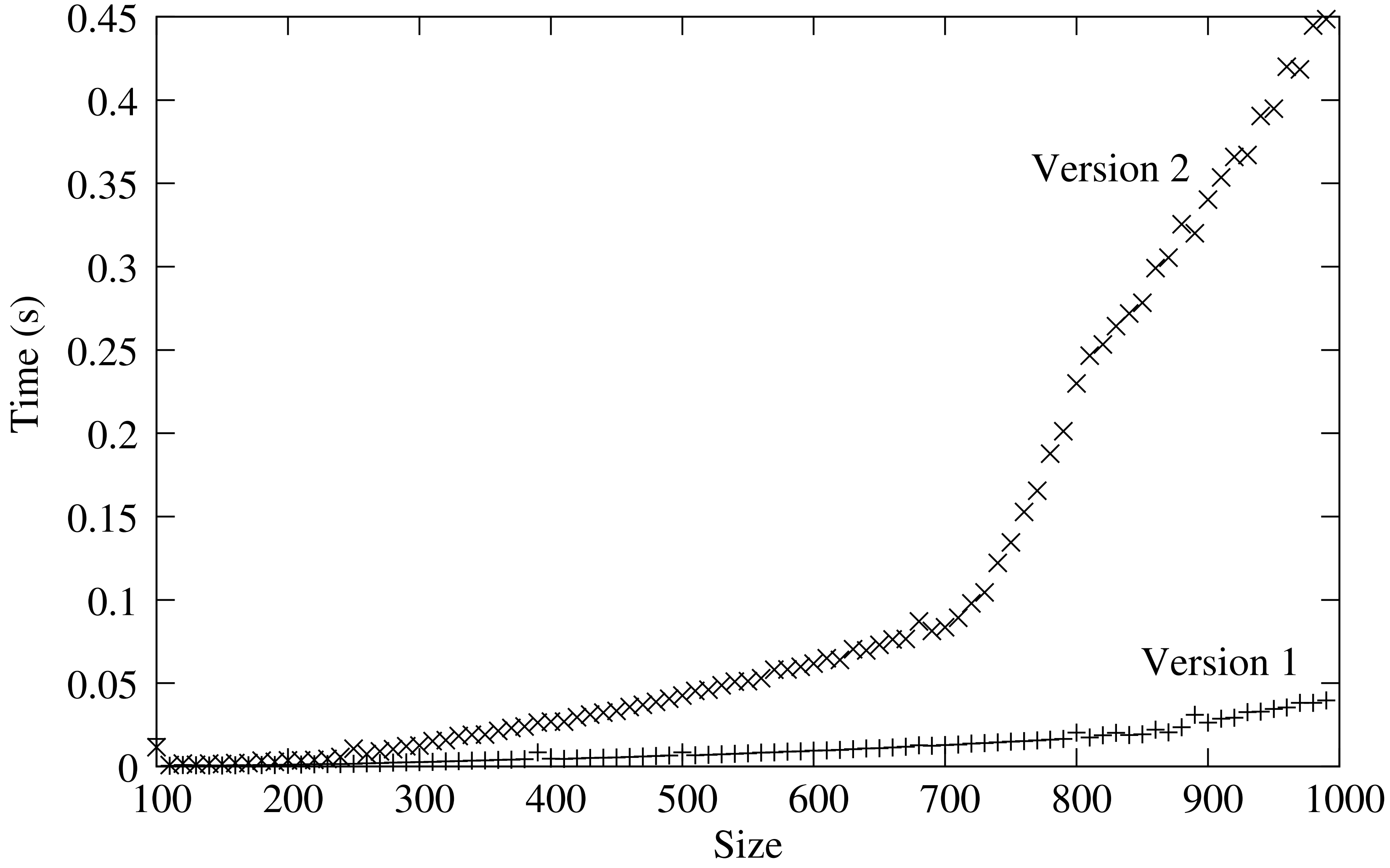
Use different kinds of memory in layers of increasing size and decreasing speed so that some of memory is fast

Effects actually visible to user

```
int[][] array = new int[size][size];
for(int i=0; i < size; i++)
    for(int j=0; j < size; j++)
        /* DO SOMETHING */
```

Version 1: `array[i][j] *= 2;`

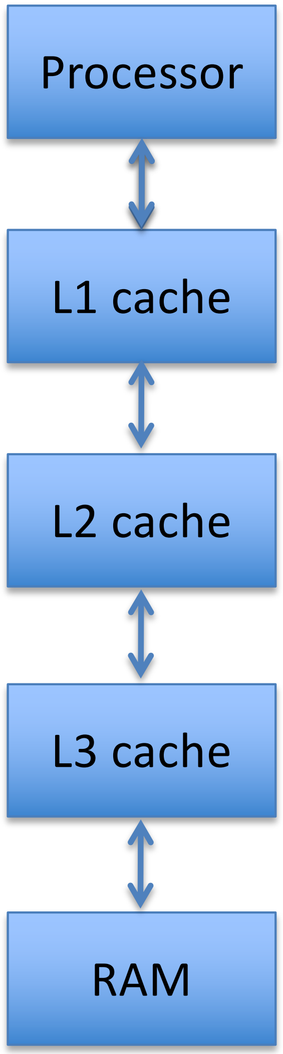
Version 2: `array[j][i] *= 2;`



Time (s)
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
100
200
300
400
500
600
700
800
900
1000
Size
Version 2
Version 1

Idea of caching

- Have small, fast memory near the processor and bigger, slower memory far way
- Each level has a subset of the data from the level below
- Goal: Average speed close to that of fast memory, but cost and size close to that of cheap memory
- Locality:
  - temporal: Likely to reuse memory addresses soon
  - spatial: Likely to use memory addresses near those we use now



Cache terminology

- Hit: Cache delivers the data and saves trip to to memory

- Miss: Cache doesn’t have the data

Cache organization

- Line: unit of storage in the cache
  - Able to store a contiguous chunk of memory

- Data block: Chunk of data that goes into a line

Direct-mapped cache

- Memory address determines line for each data block

Direct-mapped cache

- Memory address determines line for each data block

Which address bits should determine the line?

A. High-order bits (left part of #)

B. Low-order bits (right part of #)

C. Doesn’t matter

Direct-mapped cache

- Memory address determines line for each data block

Which address bits should determine the line?

A. High-order bits (left part of #)

B. Low-order bits (right part of #)

C. Doesn’t matter

Caveat: Data blocks contain data for multiple addresses

Caveat: Data blocks contain data for multiple addresses

What does this allow the cache to exploit?

A. Temporal locality

B. Spatial locality

C. Both of these

D. Neither of these

E. The proletariat

Caveat: Data blocks contain data for multiple addresses

What does this allow the cache to exploit?

A. Temporal locality

B. Spatial locality

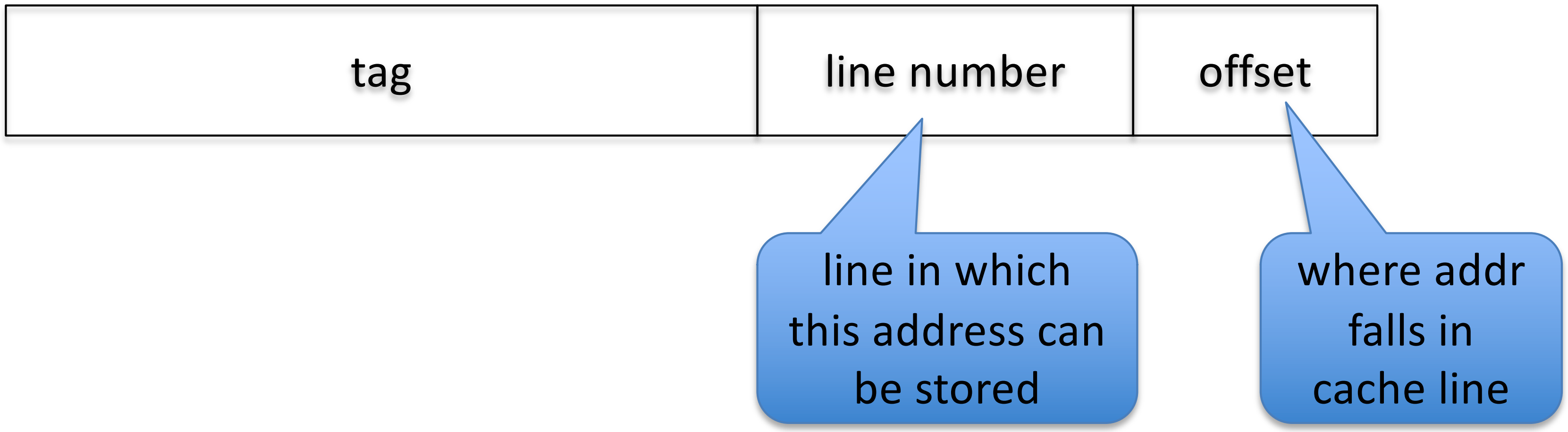
C. Both of these

D. Neither of these

E. The proletariat

Direct-mapped cache

- Each address has one possible position in the cache, determined by its "middle" bits


tag
line number
offset
line in which this address can be stored
where addr falls in cache line

How many bits would the tag occupy in a system with a 32-bit address space where the cache has 1024=2^{10} lines, each holding 4 bytes?

A. 10
B. 14
C. 18
D. 22
E. None of the above or cannot be determined

How many bits would the tag occupy in a system with a 32-bit address space where the cache has $1024=2^{10}$ lines, each holding 4 bytes?

A. 10
B. 14
C. 18
D. 22
E. None of the above or cannot be determined
(20 = 32 – 10 – 2)

How many bits would the tag occupy in a system with a 32-bit address space where the cache has 2048=2^{11} lines, each holding 16=2^4 bytes?

A. 15
B. 17
C. 18
D. 20
E. None of the above or cannot be determined

How many bits would the tag occupy in a system with a 32-bit address space where the cache has $2048=2^{11}$ lines, each holding $16=2^{4}$ bytes?

A. 15
B. 17 (= 32 – 11 – 4)
C. 18
D. 20
E. None of the above or cannot be determined

Which cache line would store address 1011001010100111 in a cache with $64=2^6$ lines, each holding $8=2^3$ bytes?

A. 101100

B. 100101

C. 010100

D. 100111

E. None of the above

Which cache line would store address 1011001010100111 in a cache with 64=2^6 lines, each holding 8=2^3 bytes?

A. 101100
B. 100101
C. <u>010100</u>
D. 100111
E. None of the above

Valid bit

- Each line also has a “valid bit” to indicate whether it contains useful data (remember no part of memory is ever “empty”)
- The cache successfully delivers the data (“hits”) if the appropriate line is valid and its tag matches that of the desired memory addr
- Otherwise, it “misses” and the system proceeds to the next lower level

#	valid	tag	data
0	T	0110	???
1	F	0110	???
2	T	1011	???
3	T	1000	???
4	F	1011	???
5	T	0110	???
6	F	1001	???
7	F	1000	???

What happens on a read to address 1011 100 1000?

#	valid	tag	data
0	T	0110	???
1	F	0110	???
2	T	1011	???
3	T	1000	???
4	F	1011	???
5	T	0110	???
6	F	1001	???
7	F	1000	???

A. Hit
B. Miss – that line is occupied by another data block
C. Miss – that line is not valid
D. Can’t be determined

What happens on a read to address 1011 100 1000?

#	valid	tag	data
0	T	0110	???
1	F	0110	???
2	T	1011	???
3	T	1000	???
4	F	1011	???
5	T	0110	???
6	F	1001	???
7	F	1000	???

A. Hit
B. Miss – that line is occupied by another data block
C. Miss – that line is not valid
D. Can’t be determined

What happens on a read to address 1011 010 0110?

#	valid	tag	data
0	T	0110	???
1	F	0110	???
2	T	1011	???
3	T	1000	???
4	F	1011	???
5	T	0110	???
6	F	1001	???
7	F	1000	???

A. Hit
B. Miss – that line is occupied by another data block
C. Miss – that line is not valid
D. Can’t be determined

What happens on a read to address 1011 010 0110?

#	valid	tag	data
0	T	0110	???
1	F	0110	???
2	T	1011	???
3	T	1000	???
4	F	1011	???
5	T	0110	???
6	F	1001	???
7	F	1000	???

A. Hit (line 2)
B. Miss – that line is occupied by another data block
C. Miss – that line is not valid
D. Can't be determined