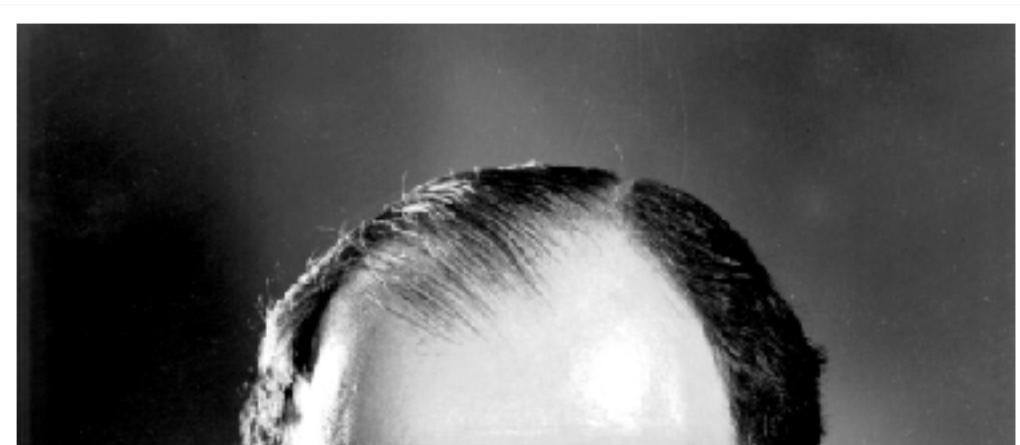


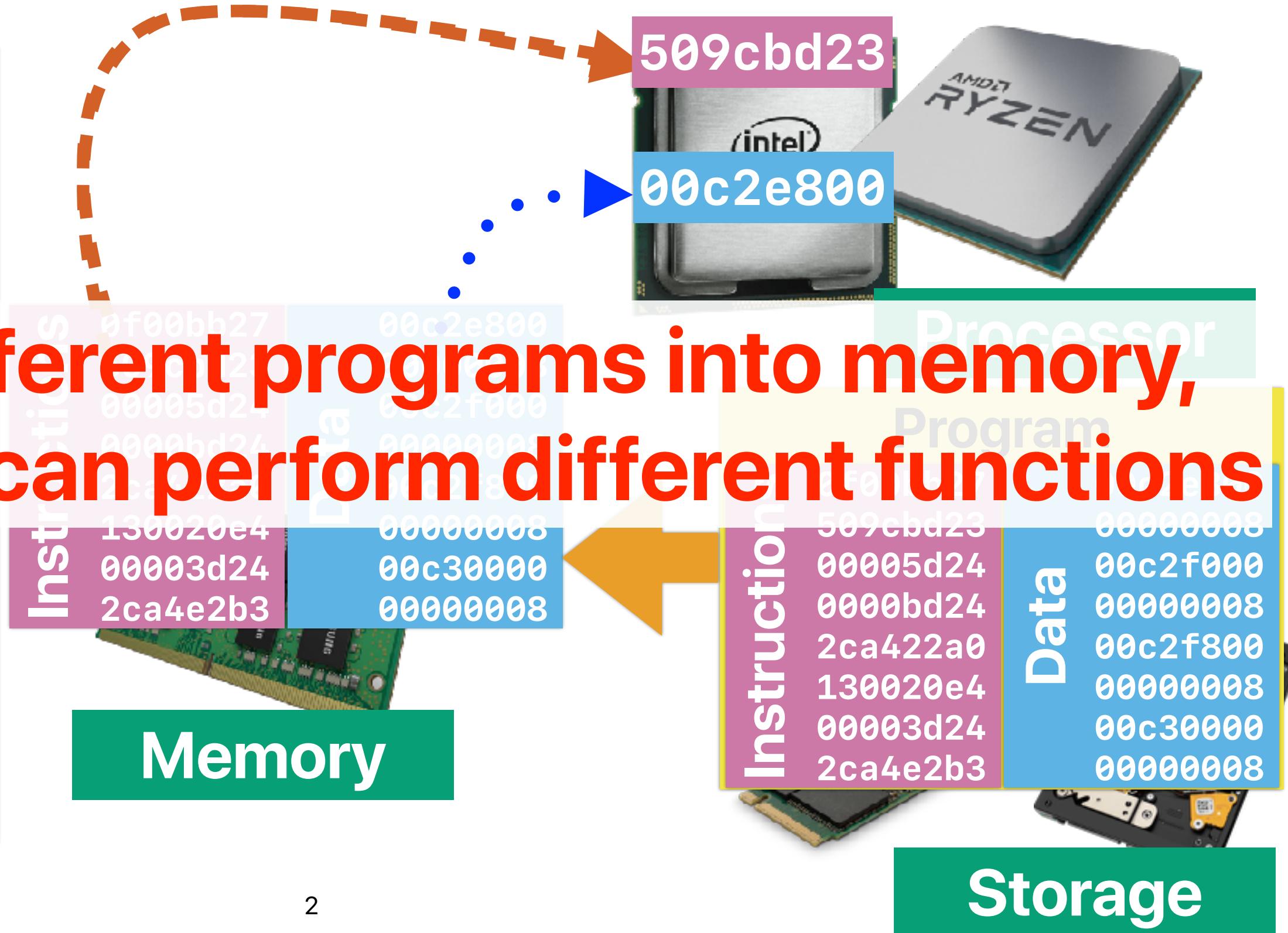
Performance (2): Make good decisions

Hung-Wei Tseng

Recap: von Neumann Architecture



By loading different programs into memory,
your computer can perform different functions



Recap: Performance & Speedup

$$\text{Performance} = \frac{1}{\text{Execution Time}}$$

$$\text{Execution Time} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Cycle}}$$

$$ET = IC \times CPI \times CT$$

$$\text{Speedup} = \frac{\text{Execution Time}_X}{\text{Execution Time}_Y}$$

Demo — programmer & performance

A

```
for(i = 0; i < ARRAY_SIZE; i++)  
{  
    for(j = 0; j < ARRAY_SIZE; j++)  
    {  
        c[i][j] = a[i][j]+b[i][j];  
    }  
}
```

B

```
for(j = 0; j < ARRAY_SIZE; j++)  
{  
    for(i = 0; i < ARRAY_SIZE; i++)  
    {  
        c[i][j] = a[i][j]+b[i][j];  
    }  
}
```

$O(n^2)$

Same

Same

Better

Complexity

Instruction Count?

Clock Rate

CPI

$O(n^2)$

Same

Same

Worse

Programmer's impact

- By adding the “sort” in the following code snippet, what changes in the performance equation to achieve **better** performance?

```
std::sort(data, data + arraySize);
```

```
for (unsigned c = 0; c < arraySize*1000; ++c) {
    if (data[c%arraySize] >= INT_MAX/2)
        sum ++;
```

programmer changes
IC as well, but not in
the positive direction

- A. CPI
- B. IC
- C. CT
- D. IC & CPI
- E. CPI & CT

option	0	1
ET	Slower	Faster
IC	Smaller (Better)	Larger (Worse)
CPI	Larger (A lot worse)	Smaller (A lot better)
CT	Same	Same

Outline

- What affects each factor in “Performance Equation”
 - Programmer
 - Programming languages
 - Compilers
- Amdahl’s Law

Programmers can also set the cycle time

<https://software.intel.com/sites/default/files/comment/1716807/how-to-change-frequency-on-linux-pub.txt>

```
=====
Subject: setting CPU speed on running linux system
```

If the OS is Linux, you can manually control the CPU speed by reading and writing some virtual files in the "/proc"

1.) Is the system capable of software CPU speed control?

If the "directory" /sys/devices/system/cpu/cpu0/cpufreq exists, speed is controllable.

-- If it does not exist, you may need to go to the BIOS and turn on EIST and any other C and P state control and vi:

2.) What speed is the box set to now?

Do the following:

```
$ cd /sys/devices/system/cpu
$ cat ./cpu0/cpufreq/cpuinfo_max_freq
3193000
$ cat ./cpu0/cpufreq/cpuinfo_min_freq
1596000
```

3.) What speeds can I set to?

Do

```
$ cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_available_frequencies
```

It will list highest settable to lowest; example from my NHM "Smackover" DX58SO HEDT board, I see:

```
3193000 3192000 3059000 2926000 2793000 2660000 2527000 2394000 2261000 2128000 1995000 1862000 1729000 1596000
```

You can choose from among those numbers to set the "high water" mark and "low water" mark for speed. If you set "h

4.) Show me how to set all to highest settable speed!

Use the following little sh/ksh/bash script:

```
$ cd /sys/devices/system/cpu # a virtual directory made visible by device drivers
$ newSpeedTop=`awk '{print $1}' ./cpu0/cpufreq/scaling_available_frequencies`
$ newSpeedLcw=$newSpeedTop # make them the same in this example
$ for c in ./cpu[0-9]* ; do
>   echo $newSpeedTop > ${c}/cpufreq/scaling_max_freq
>   echo $newSpeedLow > ${c}/cpufreq/scaling_min_freq
> done
$
```

5.) How do I return to the default - i.e. allow machine to vary from highest to lowest?

Edit line # 3 of the script above, and re-run it. Change the line:

```
$ newSpeedLcw=$newSpeedTop # make them the same in this example
```

To read

How programmer affects performance?

- Performance equation consists of the following three factors

① IC

② CPI

③ CT

How many can a **programmer** affect?

- A. 0
- B. 1
- C. 2
- D. 3



How programming languages affect performance

- Performance equation consists of the following three factors
 - ① IC
 - ② CPI
 - ③ CT

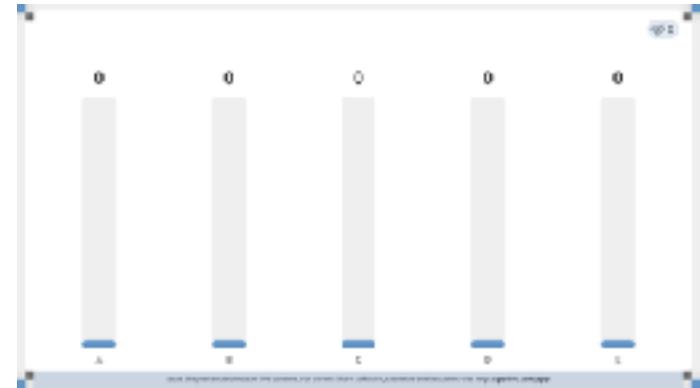
How many can the **programming language** affect?

- A. 0
- B. 1
- C. 2
- D. 3



Programming languages

- Which of the following programming language needs to highest instruction count to print “Hello, world!” on screen?
 - C
 - C++
 - Java
 - Perl
 - Python



Programming languages

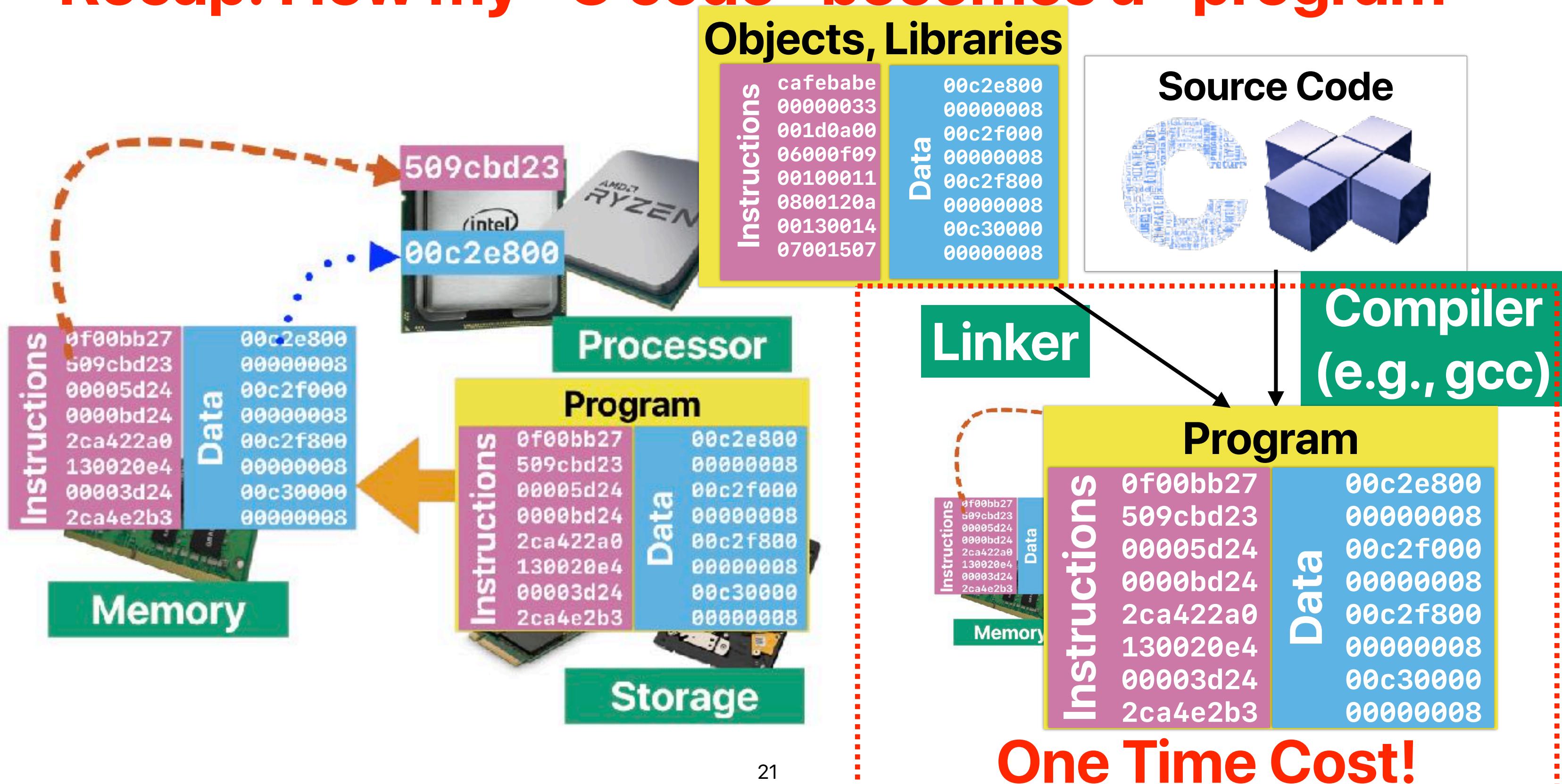
- How many instructions are there in “Hello, world!”

	Instruction count	LOC	Ranking
C	600k	6	1
C++	3M	6	2
Java	~145M	8	5
Perl	~12M	4	3
Python	~33M	1	4
GO (Interpreter)	~1200M	1	6
GO (Compiled)	~1.7M	1	
Rust	~1.4M	1	

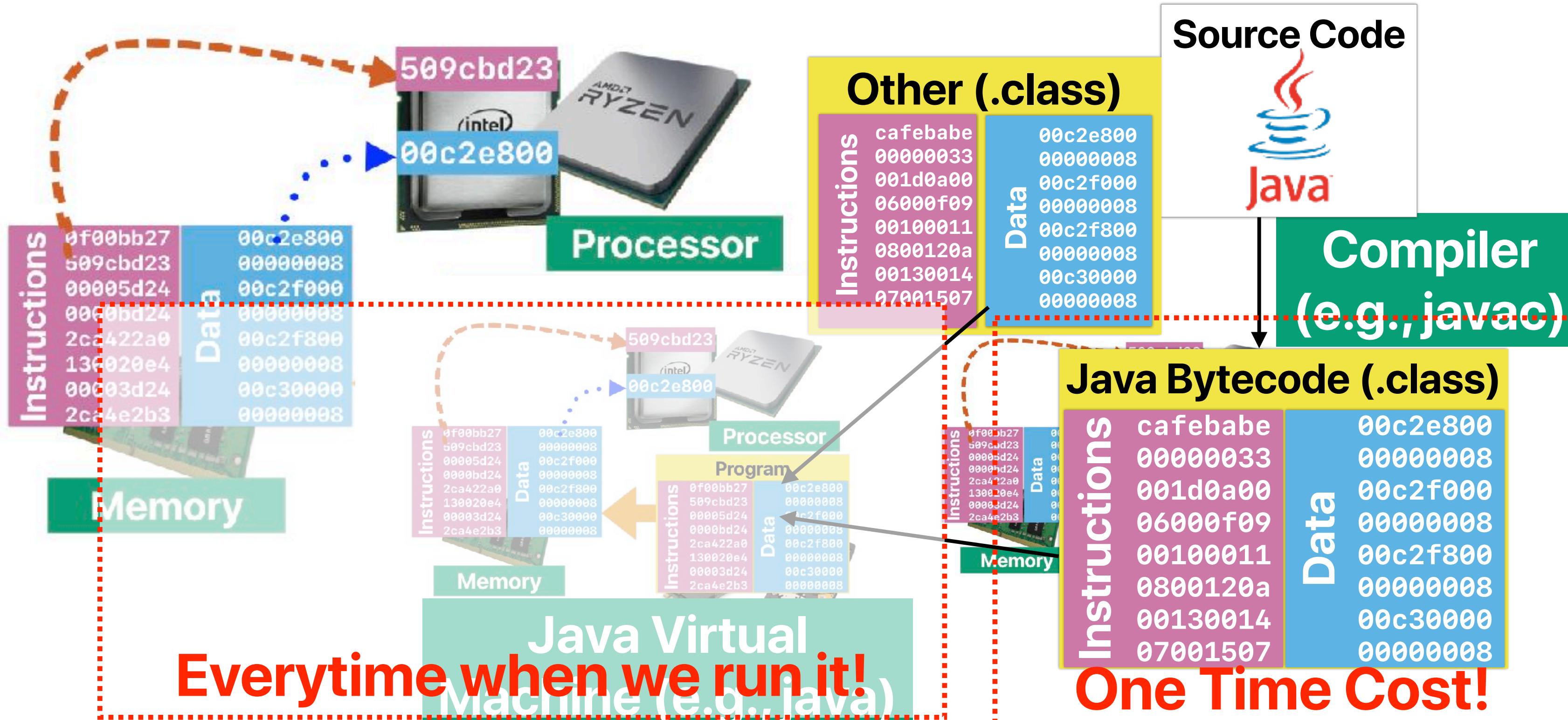
Programming languages

- Which of the following programming language needs to highest instruction count to print “Hello, world!” on screen?
 - A. C
 - B. C++
 - C. Java
 - D. Perl
 - E. Python

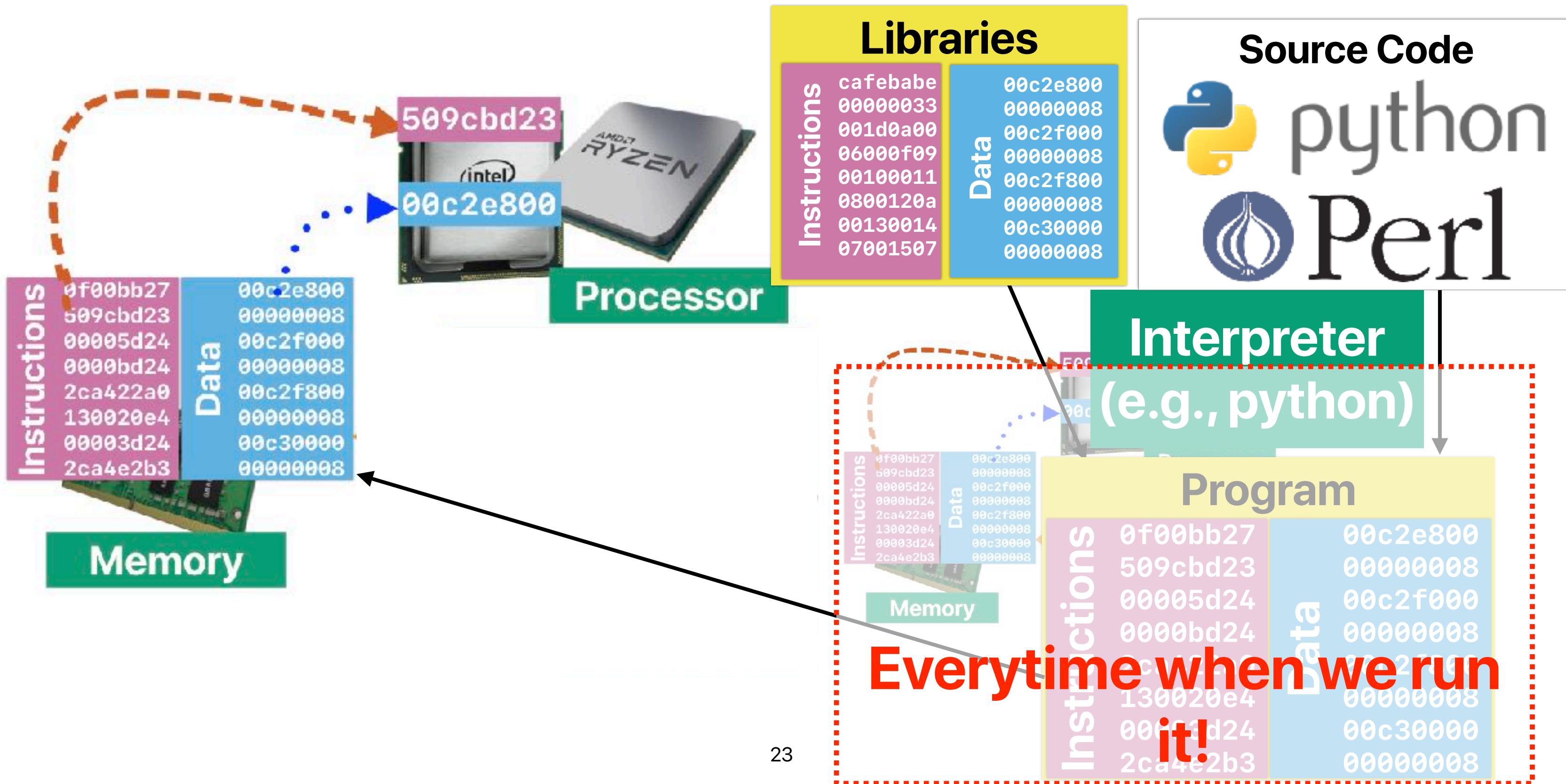
Recap: How my “C code” becomes a “program”



Recap: How my “Java code” becomes a “program”



Recap: How my “Python code” becomes a “program”



How programming languages affect performance

- Performance equation consists of the following three factors

① IC

② CPI

③ CT

Programmer uses programming languages to create library/programs that changes the CT, not the programming language itself makes the change



How many can the **programming language** affect?

A. 0

B. 1

C. 2

D. 3



How compilers affect performance

- Performance equation consists of the following three factors
 - ① IC
 - ② CPI
 - ③ CT

How many can the **compiler** affect?

- A. 0
- B. 1
- C. 2
- D. 3

How compilers affect performance

- Performance equation consists of the following three factors

- ① IC
- ② CPI
- ③ CT

Compiler is less effective than programmer if the programmer wisely tweaked the code based on architecture!

How many can the **compiler** affect?

- A. 0
- B. 1
- C. 2
- D. 3

Revisited the demo with compiler optimizations!

- gcc has different optimization levels.
 - -O0 — no optimizations
 - -O3 — typically the best-performing optimization

A

```
for(i = 0; i < ARRAY_SIZE; i++)
{
    for(j = 0; j < ARRAY_SIZE; j++)
    {
        c[i][j] = a[i][j]+b[i][j];
    }
}
```

B

```
for(j = 0; j < ARRAY_SIZE; j++)
{
    for(i = 0; i < ARRAY_SIZE; i++)
    {
        c[i][j] = a[i][j]+b[i][j];
    }
}
```

How about complexity?

How about “computational complexity”

- Algorithm complexity provides a good estimate on the performance if —
 - Every instruction takes exactly the same amount of time
 - Every operation takes exactly the same amount of instructions

These are unlikely to be true

Summary of CPU Performance Equation

$$\text{Performance} = \frac{1}{\text{Execution Time}}$$

$$\text{Execution Time} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Cycle}}$$

$$ET = IC \times CPI \times CT$$

- IC (Instruction Count)
 - ISA, Compiler, algorithm, programming language, **programmer**
- CPI (Cycles Per Instruction)
 - Machine Implementation, microarchitecture, compiler, application, algorithm, programming language, **programmer**
- Cycle Time (Seconds Per Cycle)
 - Process Technology, microarchitecture, **programmer**

Amdahl's Law — and It's Implication in the Multicore Era

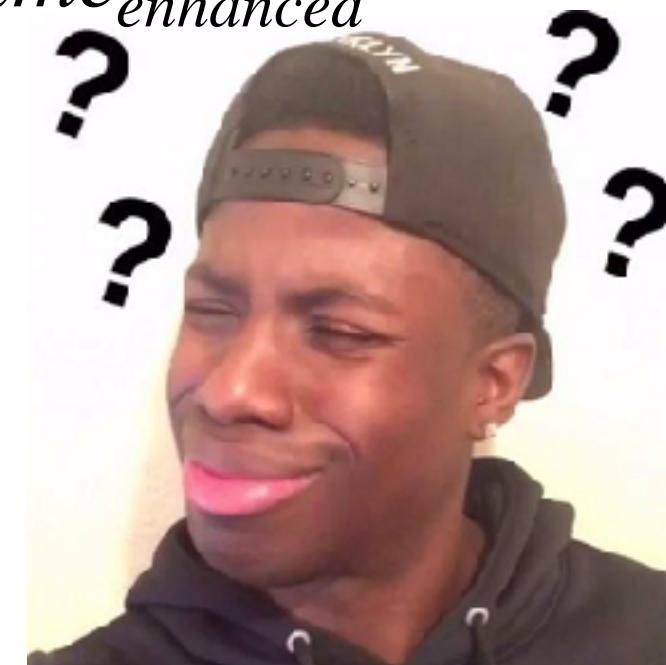
Amdahl's Law



$$\text{Speedup}_{\text{enhanced}}(f, s) = \frac{1}{(1-f) + \frac{f}{s}}$$

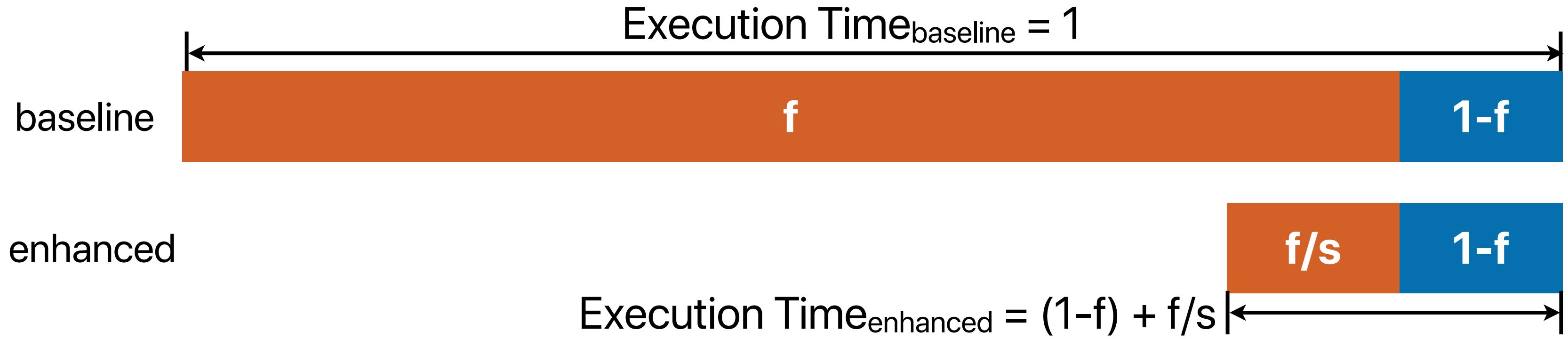
- f — The fraction of time in the original program
 s — The speedup we can achieve on f

$$\text{Speedup}_{\text{enhanced}} = \frac{\text{Execution Time}_{\text{baseline}}}{\text{Execution Time}_{\text{enhanced}}}$$



Amdahl's Law

$$Speedup_{enhanced}(f, s) = \frac{1}{(1 - f) + \frac{f}{s}}$$



$$Speedup_{enhanced} = \frac{Execution\ Time_{baseline}}{Execution\ Time_{enhanced}} = \frac{1}{(1 - f) + \frac{f}{s}}$$

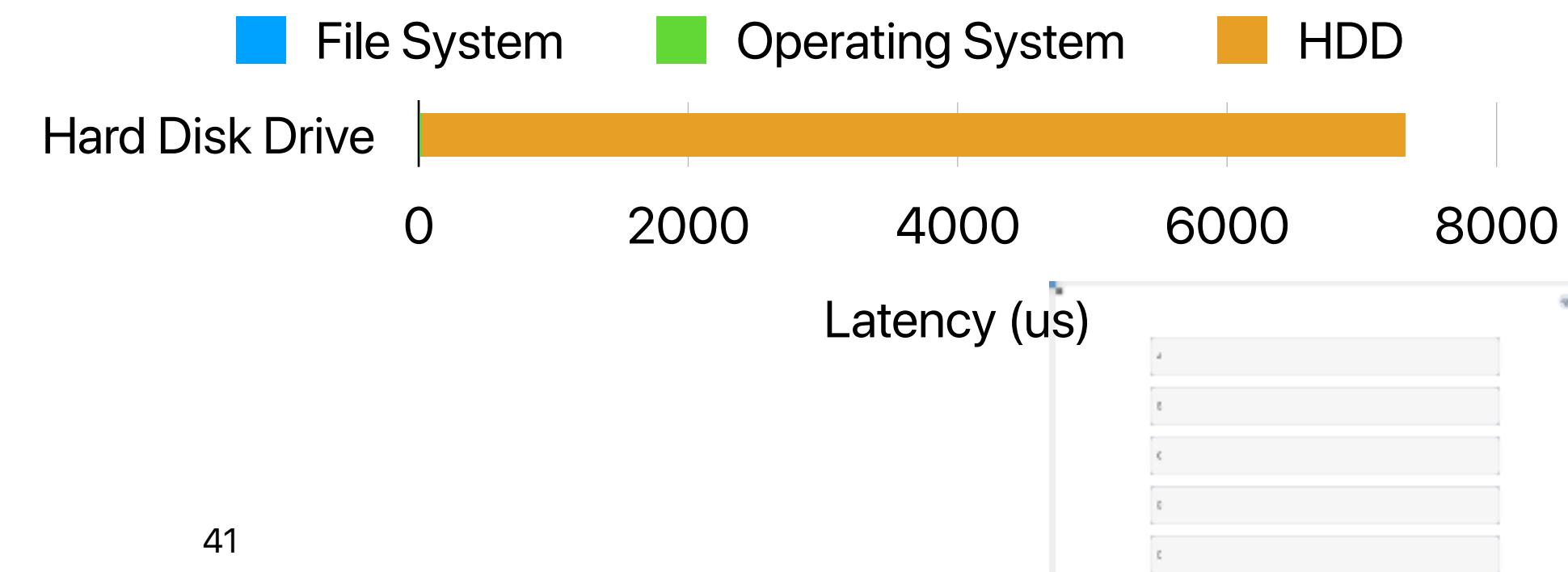




Practicing Amdahl's Law

- Final Fantasy XV spends lots of time loading a map — within which period that 95% of the time on the accessing the H.D.D., the rest in the operating system, file system and the I/O protocol. If we replace the H.D.D. with a flash drive, which provides 100x faster access time. By how much can we speed up the map loading process?

- A. ~7x
- B. ~10x
- C. ~17x
- D. ~29x
- E. ~100x

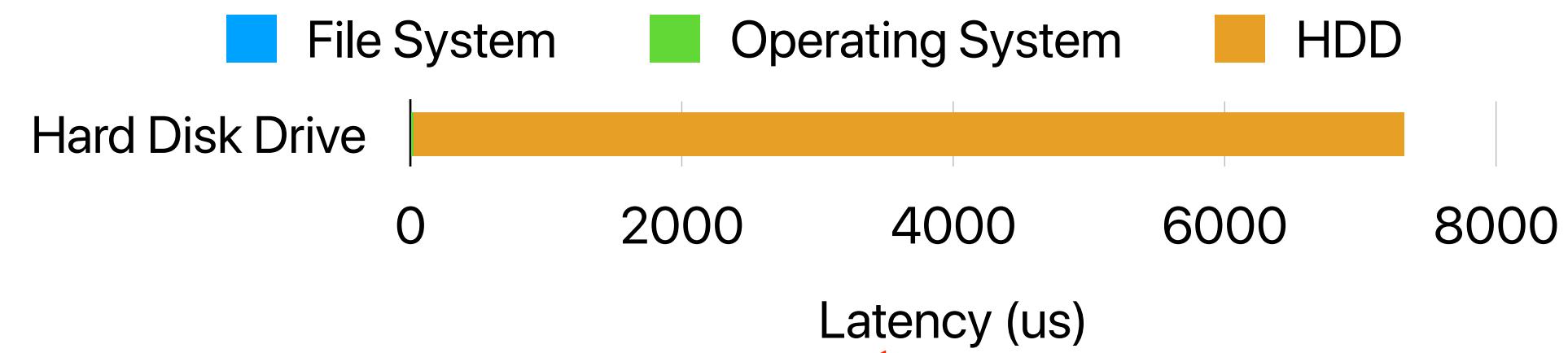


Practicing Amdahl's Law

- Final Fantasy XV spends lots of time loading a map — within which period that 95% of the time on the accessing the H.D.D., the rest in the operating system, file system and the I/O protocol. If we replace the H.D.D. with a flash drive, which provides 100x faster access time. By how much can we speed up the map loading process?

- A. ~7x
- B. ~10x
- C. ~17x
- D. ~29x
- E. ~100x

$$Speedup_{enhanced}(95\%, 100) = \frac{1}{(1 - 95\%) + \frac{95\%}{100}} = 16.81 \times$$



Amdahl's Law on Multiple Optimizations

- We can apply Amdahl's law for multiple optimizations
- These optimizations must be dis-joint!
 - If optimization #1 and optimization #2 are dis-joint:



$$Speedup_{enhanced}(f_{Opt1}, f_{Opt2}, s_{Opt1}, s_{Opt2}) = \frac{1}{(1 - f_{Opt1} - f_{Opt2}) + \frac{f_{Opt1}}{s_{Opt1}} + \frac{f_{Opt2}}{s_{Opt2}}}$$

- If optimization #1 and optimization #2 are not dis-joint:



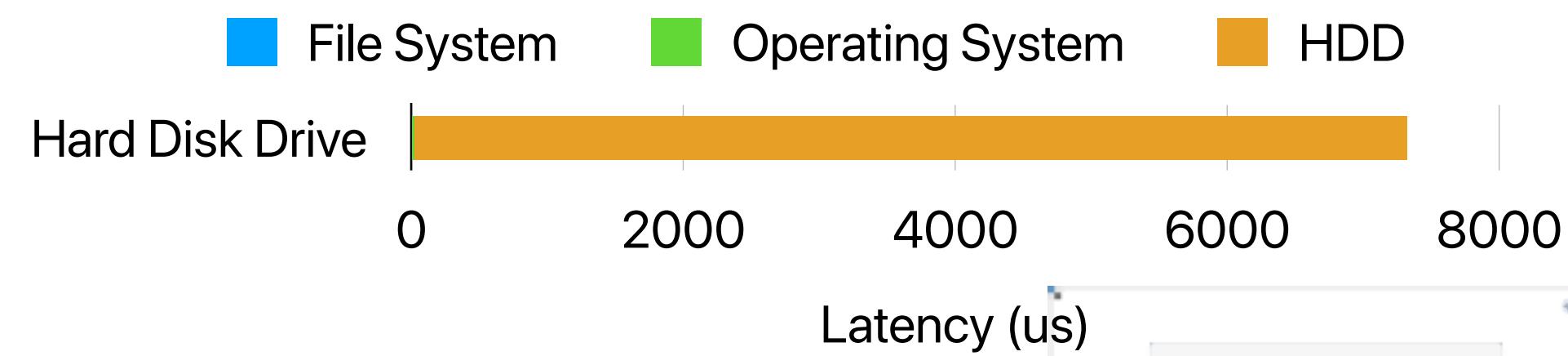
$$Speedup_{enhanced}(f_{OnlyOpt1}, f_{OnlyOpt2}, f_{BothOpt1Opt2}, s_{OnlyOpt1}, s_{OnlyOpt2}, s_{BothOpt1Opt2}) = \frac{1}{(1 - f_{OnlyOpt1} - f_{OnlyOpt2} - f_{BothOpt1Opt2}) + \frac{f_{BothOpt1Opt2}}{s_{BothOpt1Opt2}} + \frac{f_{OnlyOpt1}}{s_{OnlyOpt1}} + \frac{f_{OnlyOpt2}}{s_{OnlyOpt2}}}$$



Practicing Amdahl's Law (2)

- Final Fantasy XV spends lots of time loading a map — within which period that 95% of the time on the accessing the H.D.D., the rest in the operating system, file system and the I/O protocol. If we replace the H.D.D. with a flash drive, which provides 100x faster access time and a better processor to accelerate the software overhead by 2x. By how much can we speed up the map loading process?

- A. ~7x
- B. ~10x
- C. ~17x
- D. ~29x
- E. ~100x



Practicing Amdahl's Law (2)

- Final Fantasy XV spends lots of time loading a map — within which period that 95% of the time on the accessing the H.D.D., the rest in the operating system, file system and the I/O protocol. If we replace the H.D.D. with a flash drive, which provides 100x faster access time and a better processor to accelerate the software overhead by 2x. By how much can we speed up the map loading process?

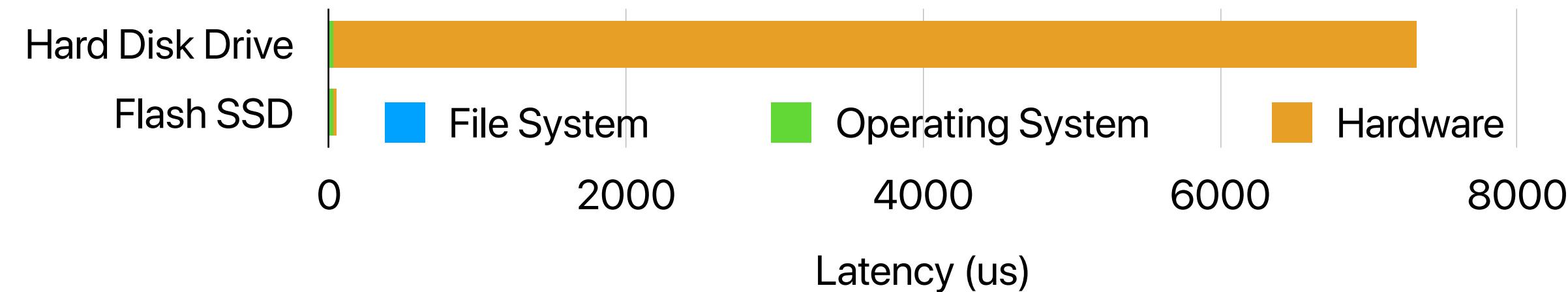
A. ~7x

B. ~10x

C. ~17x

D. ~29x

E. ~100x



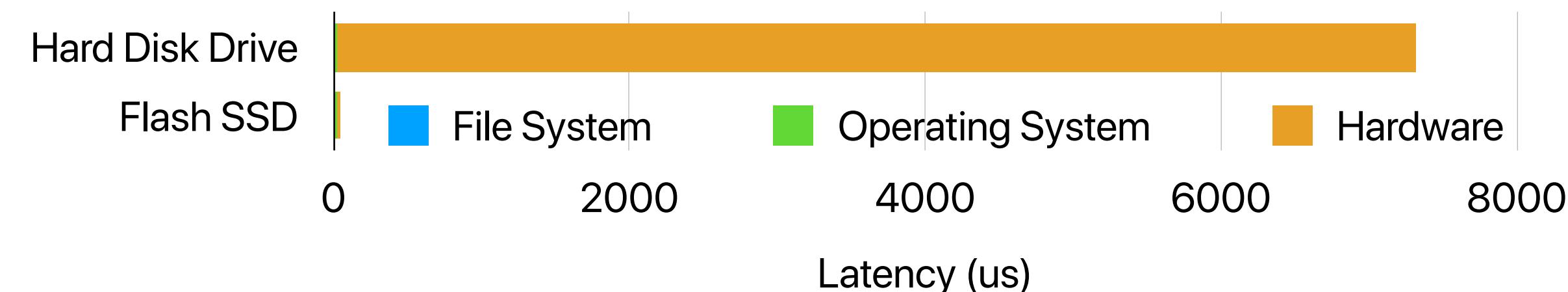
$$Speedup_{enhanced}(95\%, 5\%, 100, 2) = \frac{1}{(1 - 95\% - 5\%) + \frac{95\%}{100} + \frac{5\%}{2}} = 28.98 \times$$



Speedup further!

- With the latest flash memory technologies, the system spends 16% of time on accessing the flash, and the software overhead is now 84%. If your company ask you and your team to invent a new memory technology that replaces flash to achieve 2x speedup on loading maps, how much faster the new technology needs to be?

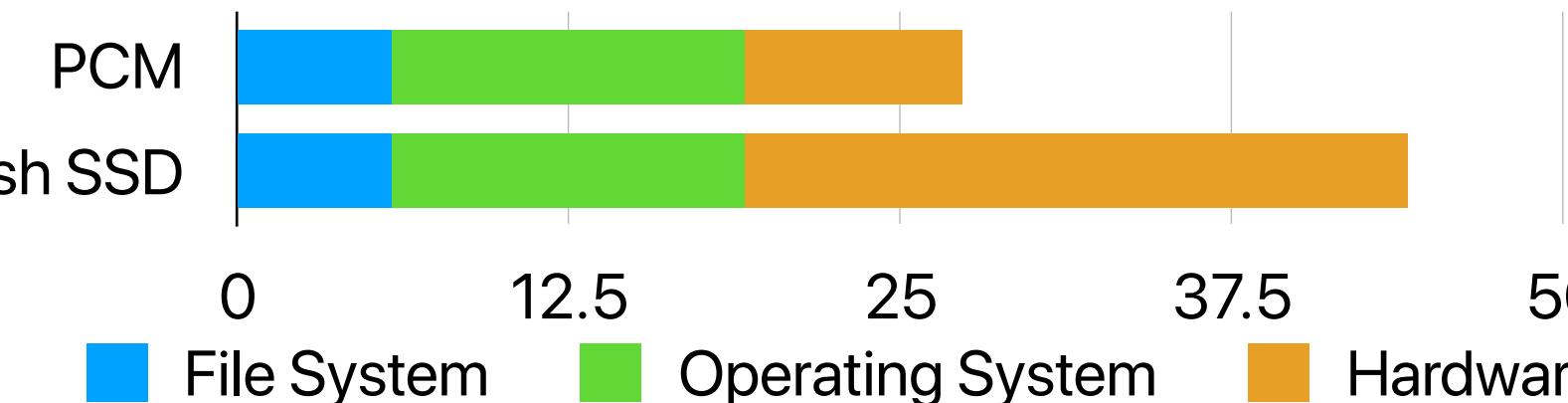
- A. ~5x
- B. ~10x
- C. ~20x
- D. ~100x
- E. None of the above



Speedup further!

- With the latest flash memory technologies, the system spends 16% of time on accessing the flash, and the software overhead is now 84%. If your company ask you and your team to invent a new memory technology that replaces flash to achieve 2x speedup on loading maps, how much faster the new technology needs to be?

A. ~5x



C. ~20x

D. ~100x

E. None of the above

$$\text{Speedup}_{\text{enhanced}}(16\%, x) = \frac{1}{(1 - 16\%) + \frac{16\%}{x}} = 2$$
$$x = 0.47$$

Amdahl's Law Corollary #1

- The maximum speedup is bounded by

$$\text{Speedup}_{max}(f, \infty) = \frac{1}{(1-f) + \frac{f}{\infty}}$$

$$\text{Speedup}_{max}(f, \infty) = \frac{1}{(1-f)}$$

Speedup further!

- With the latest flash memory technologies, the system spends 16% of time on accessing the flash, and the software overhead is now 84%. If your company ask you and your team to invent a new memory technology that replaces flash to achieve 2x speedup on loading maps, how much faster the new technology needs to be?

A. ~5x



B. ~10x Flash SSD

C. ~20x

D. ~100x

E. None of the above

$$\text{Speedup}_{max}(16\%, \infty) = \frac{1}{(1 - 16\%)} = 1.19$$

2x is not possible

Intel kills the remnants of Optane memory

The speed-boosting storage tech was already on the ropes.



By [Michael Crider](#)

Staff Writer, PCWorld | JUL 29, 2022 6:59 AM PDT



Image: Intel

MILLIPORE
SIGMA



MISSION® es

targeting mol
2010204k13r

Corollary #1 on Multiple Optimizations

- If we can pick just one thing to work on/optimize



$$Speedup_{max}(f_1, \infty) = \frac{1}{(1 - f_1)}$$

$$Speedup_{max}(f_2, \infty) = \frac{1}{(1 - f_2)}$$

$$Speedup_{max}(f_3, \infty) = \frac{1}{(1 - f_3)}$$

$$Speedup_{max}(f_4, \infty) = \frac{1}{(1 - f_4)}$$

The biggest f_x would lead to the largest $Speedup_{max}$!

Corollary #2 — make the common case fast!

- When f is small, optimizations will have little effect.
- Common == **most time consuming** not necessarily the most frequent
 - $ET = IC \times CPI \times CT$
- The uncommon case doesn't make much difference
- The common case can change based on inputs, compiler options, optimizations you've applied, etc.

Identify the most time consuming part

- Compile your program with -pg flag
- Run the program
 - It will generate a gmon.out
 - `gprof your_program gmon.out > your_program.prof`
- It will give you the profiled result in `your_program.prof`

Please read carefully before you post

Getting Help

You might run into trouble while doing this assignment. Here's how to get help:

1. Re-read the instructions and make sure you've followed them.
2. Try saving and reloading the notebook.
3. If it says you are not authenticated, go to the [the login section of the assignment](#) and (re)authenticate.
4. If you get a `FileNotFoundException` make sure you've run all the code cells above your current point in the assignment.
5. If you get an exception or stack dump, check that you didn't accidentally modify the contents of one of the python cells.
6. If all else fails, post a question to edstem/piazza.

Posting Answerable Questions on Piazza

If you want useful answers on piazza, you need to provide information that is specific enough for us to provide a useful answer. Here's what we need:

1. Which part of which assignment are you working on (use the section numbers)?
2. Which problem (copy and paste the text of the question along with the number).

If it's a question about instructions:

1. Try to be as specific as you can about what is confusing or what you don't understand (e.g., 'I'm not sure if I should do X or Y.')

If it's a question about an error while running code, then we need:

1. If you've committed anything, your github repo url.
2. If you've submitted a job with `cs203` you *must* provide the job id. With the id, we can figure out just about anything about your job. Without it, we know nothing.
3. The *entire* output you received. There's no limit on how long an piazza post can be. Give us all the information, not just the last few lines. We like to scroll!

For all of the above **paste the text** into the edstem/piazza question. Please **do not provide screen captures**. The course staff refuses to type in job ids found in screen shots.

We Can't Answer Unanswerable Questions If you don't follow these guidelines (especially about the github repo and the job id), we will probably not be able to answer your question on edstem/piazza. We will archive it and ask¹⁰⁹ to re-post your question with the information we need.

Announcement

- We will be **back in person next Tuesday**
- Reading quiz due next Tuesday before the lecture
 - We will drop one of your least performing reading quizzes
 - You have unlimited time
- Assignment #1 due **this evening**
 - You need to agree the course agreement to start — if you have done it late but haven't send us an e-mail explaining why, you're not allowed to start
 - Make sure your code is **committed** to GitHub before submission
- Assignment #2 is up and due **next Thursday**
 - Please act quickly
- Check our website for slides, gradescope for quizzes/assignments, piazza for discussions
- Youtube channel for lecture recordings:
<https://www.youtube.com/c/ProfUsagi/playlists>

Computer Science & Engineering

203

つづく

