[544] Intro to Big Data Systems

Meenakshi Syamkumar

Who am I?

Meenakshi (Meena) Syamkumar
- Email: ms@cs.wisc.edu
- Please call me "Meena"

Industry and Teaching experience
- Citrix, Cisco, and Microsoft
- CS220, CS300, CS320, CS367, guest lectures in CS640, CS740

Research
- Network measurements
- CS education

Who are You?

Year in school? Major?

What CS courses have people taken before?

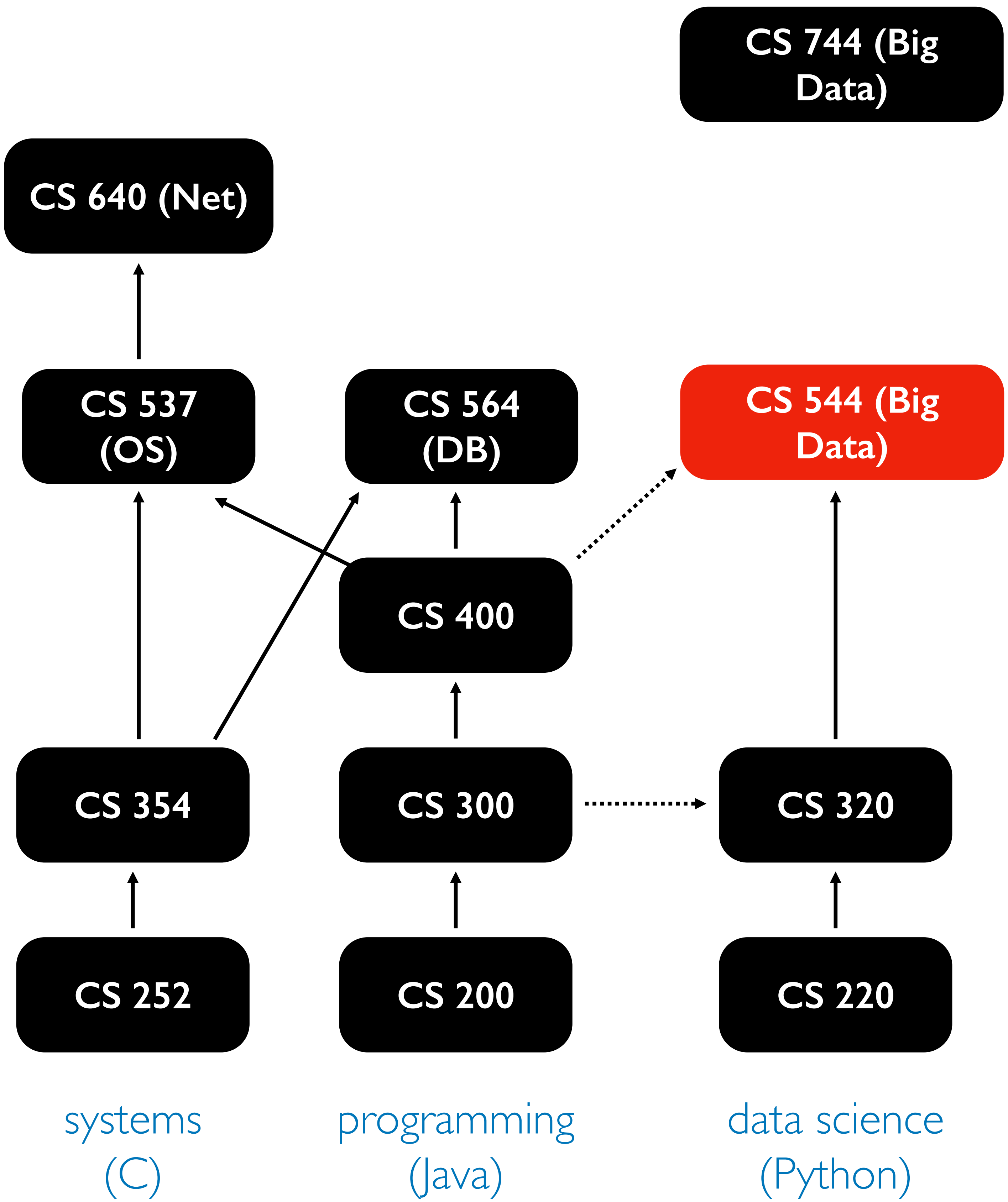
- 320? 400? 537/564/640?

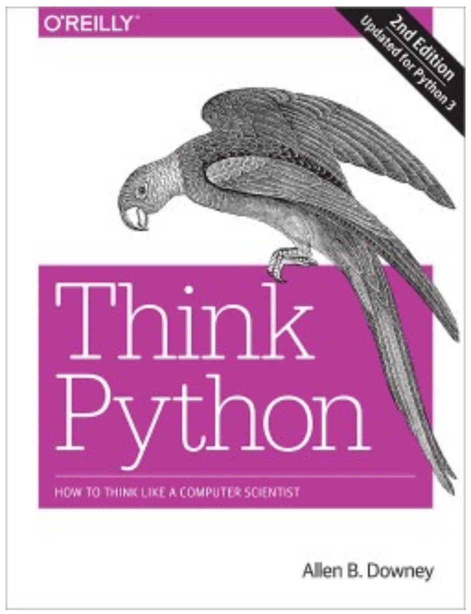
Please fill this form (**due next Wednesday, Jan 31st**):
https://forms.gle/2fCoJuHpccENdj5f6

Why?

- Help me get to know you
- Get participation credit

Related courses





- most coding will be in Python (400 folks will need to pick this up)
- first third of course will cover some foundations from operating systems, networking, and databases
- 744 will cover some similar systems, but from the research perspective (544 is hands on)

What are "*systems*"?

Some major categories of software

- analysis code (run once, get results)
- applications (long running, maybe a website)
- systems (manage resources, like storage space)

Other code uses systems. For example, without an operating system, your analyis code couldn't read files.

Whatever kind of programming you doing,
knowing how systems work with help you **use resources better**!

What are "*big data systems*"?

Some major categories of software

- analysis code (run once, get results)
- applications (long running, maybe a website)
- systems (manage resources, like storage space)

Other code uses systems. For example, without an operating system, your analyis code couldn't read files.

As data grows, we need to optimize our code and/or use more resources

Big data systems manage resources that are:

- specialized (e.g., GPUs)
- distributed (cluster of machines)

What will you learn in 544?

Learning objectives

- Deploy distributed systems for data storage and analytics
- Demonstrate competencies with tools and processes necessary for loading data into distributed storage systems
- Write programs that use distributed platforms to efficiently analyze large datasets
- Produce meaning from large datasets by training machine learning models in parallel or on distributed systems
- Measure resource usage and overall cost of running distributed programs
- Optimize distributed analytics programs to reduce resource consumption and program runtime
- Demonstrate competencies with cloud services designed to store or analyze large datasets

What will you learn today?

Learning objectives

- recall course logistics and policies
- describe different kinds of hardware resources
- compare scale up to scale out approaches
- compare different approaches for running code on a CPU

Outline

Course Overview

- Introductions
- Websites
- Other tools: Email, TopHat, Piazza, GitHub classroom, Anki

Resources

- Overview
- Compute
- Memory
- Storage
- Network

Deployment

Main Websites

1. https://ms.sites.cs.wisc.edu/cs544/s24/schedule.html
    - schedule, course content, how to get help
    - links to all other resources/tools
    - some lecture recordings (review only)

2. https://github.com/cs544-wisc/s24
    - project specifications
    - lecture demo code

3. Canvas
    - announcements (class-wide & personal)
    - quizzes
    - grade summaries
    - zoom office hours

Outline

Course Overview

- Introductions
- Websites
- Other tools: Email, TopHat, Piazza, GitHub classroom, Anki

Resources

- Overview
- Compute
- Memory
- Storage
- Network

Deployment

1. TopHat (me asking you questions during lecture)
   - Also one factor in participation
2. Piazza (asking questions of general interest)
   - our goal: responses <24 hours
   - don't post >5 lines of project code
3. Office hours (asking questions of individual interest)
   - Email: least-preferred (try to utilize office hours)
   - our goal: responses <2 business days
   - feel free to escalate by CC'ing instructor on same thread after 2 days
   - if contacting multiple staff members about same issue, please keep all on the same thread (don't start multiple threads)
4. GitHub classroom
   - you'll be given a **private** repo for each project
   - we'll provide feedback on GitHub
5. Anki Flash Cards
   - memory terms, basic ideas using flash cards and spaced repetition

Outline

Course Overview

- Introductions
- Websites
- Other tools: Email, TopHat, Piazza, GitHub classroom

Resources

- Overview
- Compute
- Memory
- Storage
- Network

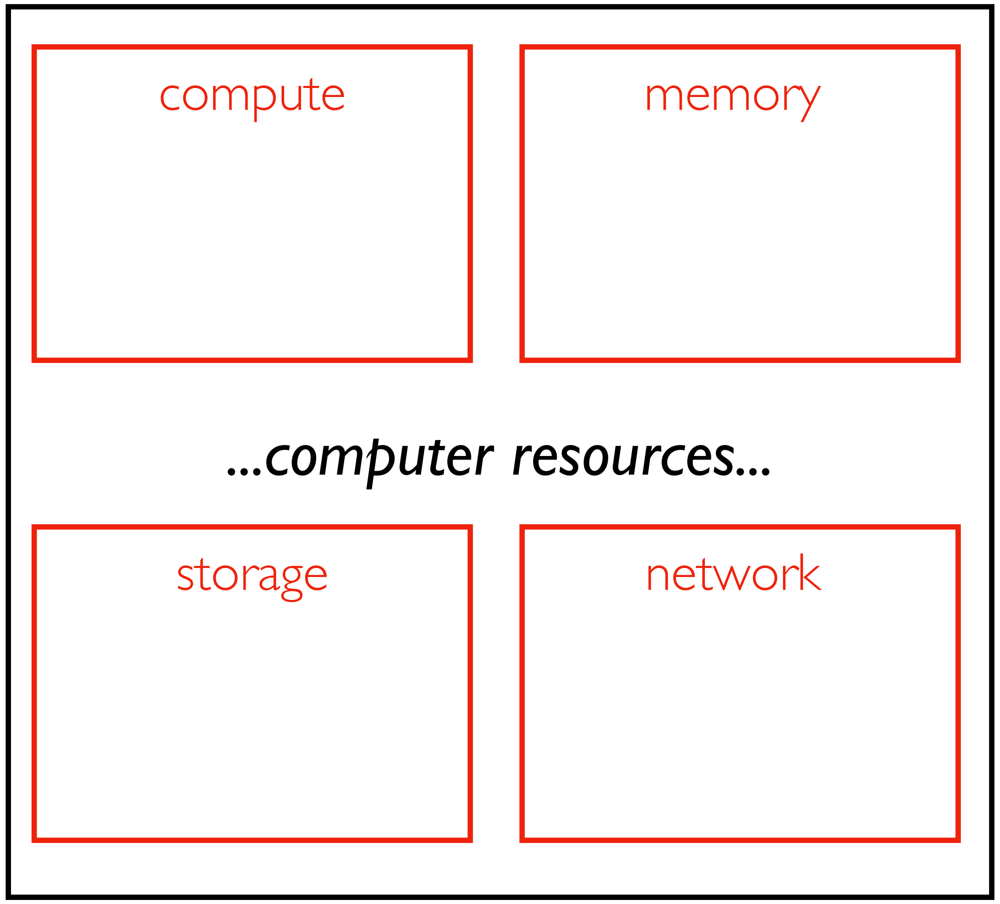
Deployment

Categories of resources

Systems: software for managing computer resources

Other kinds of software (analysis code, applications) rely on systems.

a computer:

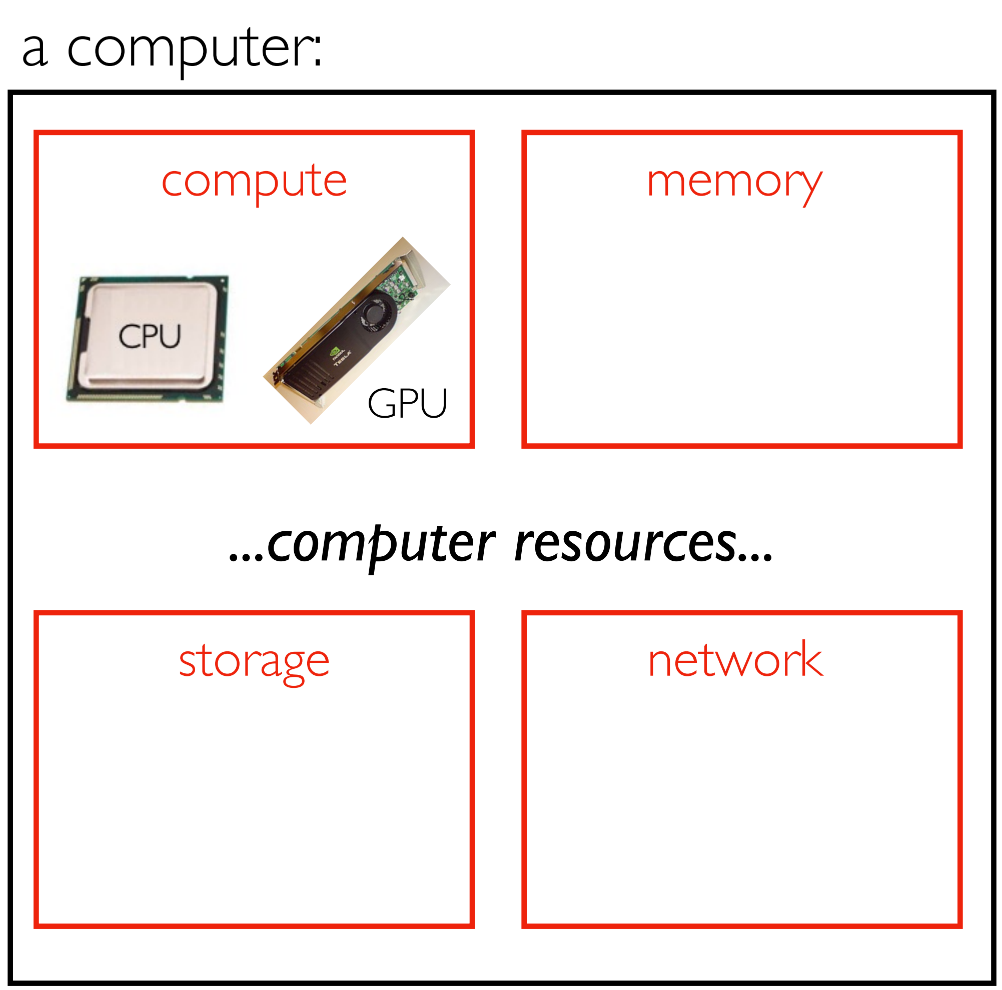


Categories of resources

Systems: software for managing computer resources

Other kinds of software (analysis code, applications) rely on systems.

computational resources
execute code



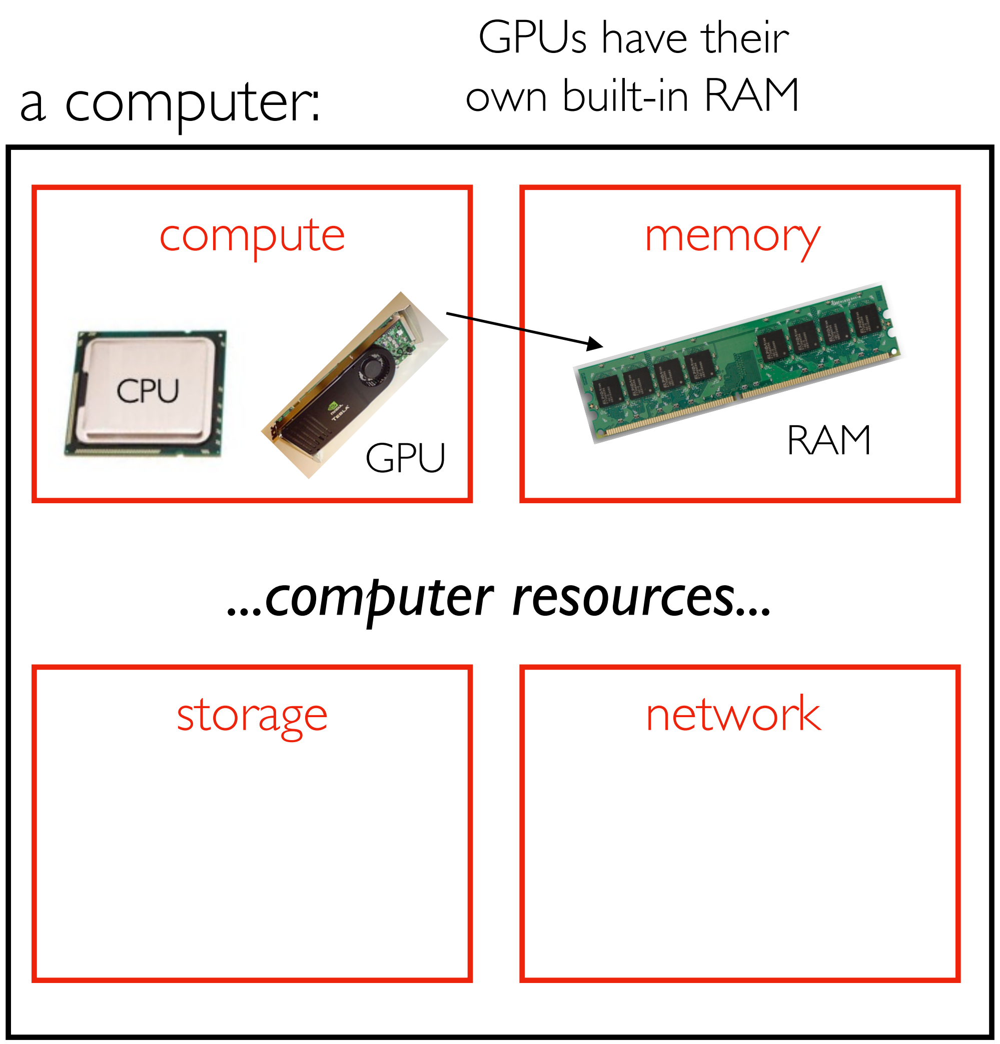
central processing unit (CPU), graphics processing unit (GPU)

Categories of resources

Systems: software for managing computer resources

Other kinds of software (analysis code, applications) rely on systems.

computational resources
execute code



memory holds data
for active usage

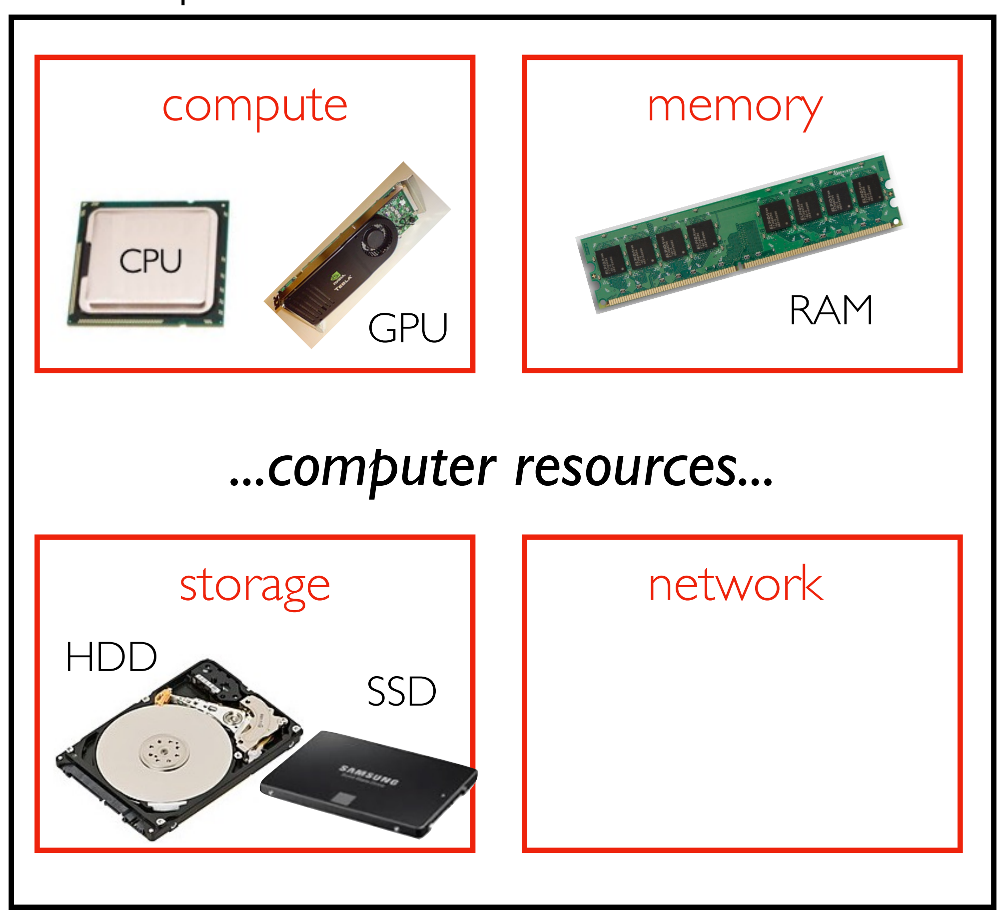
random-access memory (RAM)

Categories of resources

Systems: software for managing computer resources

Other kinds of software (analysis code, applications) rely on systems.

a computer:



computational resources execute code

memory holds data for active usage

storage holds long-term data

hard disk drive (HDD), solid-state disk (SSD)

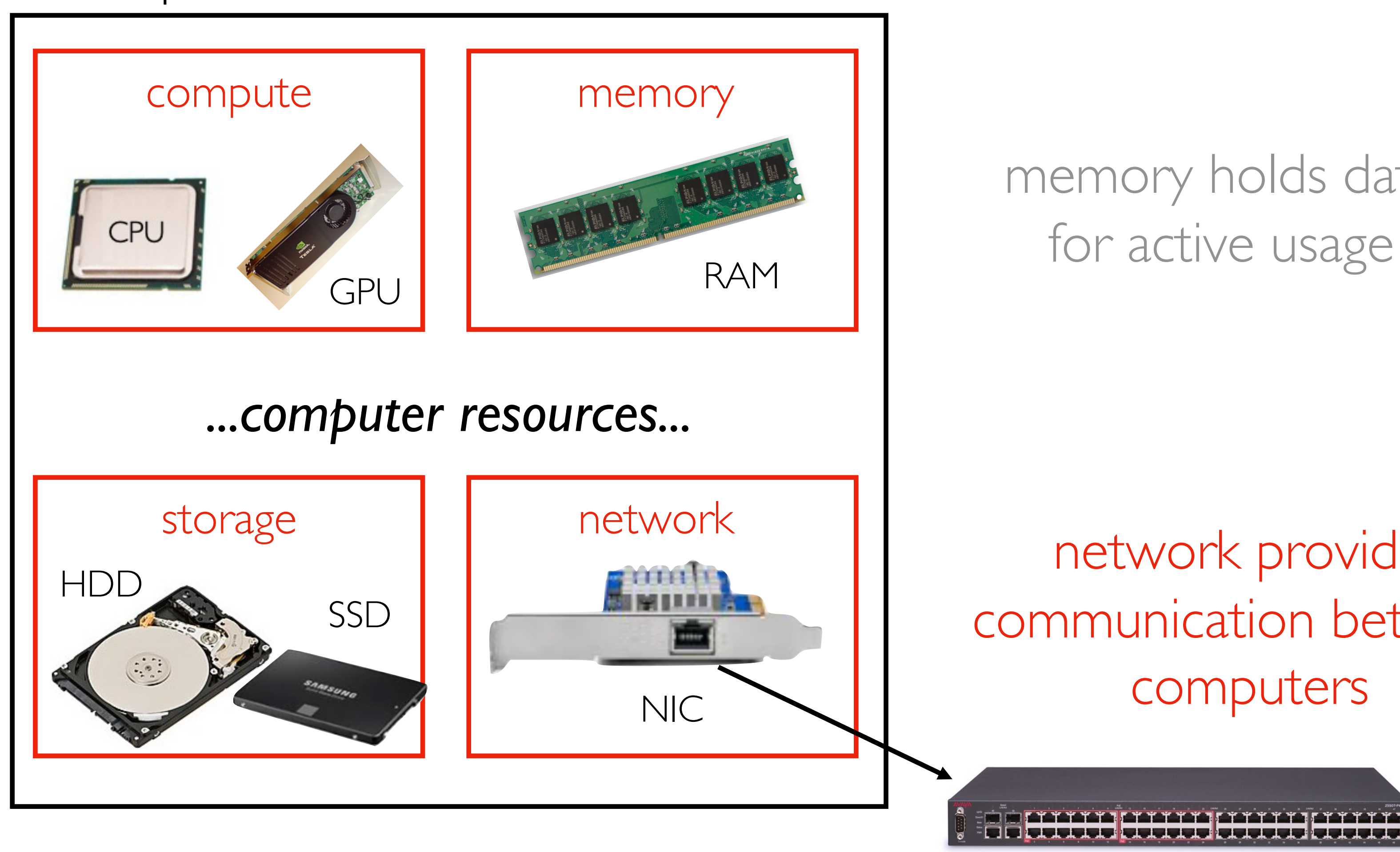
https://en.wikipedia.org/wiki/Hard_disk_drive

Categories of resources

Systems: software for managing computer resources

Other kinds of software (analysis code, applications) rely on systems.

a computer:



computational resources execute code

memory holds data for active usage

storage holds long-term data

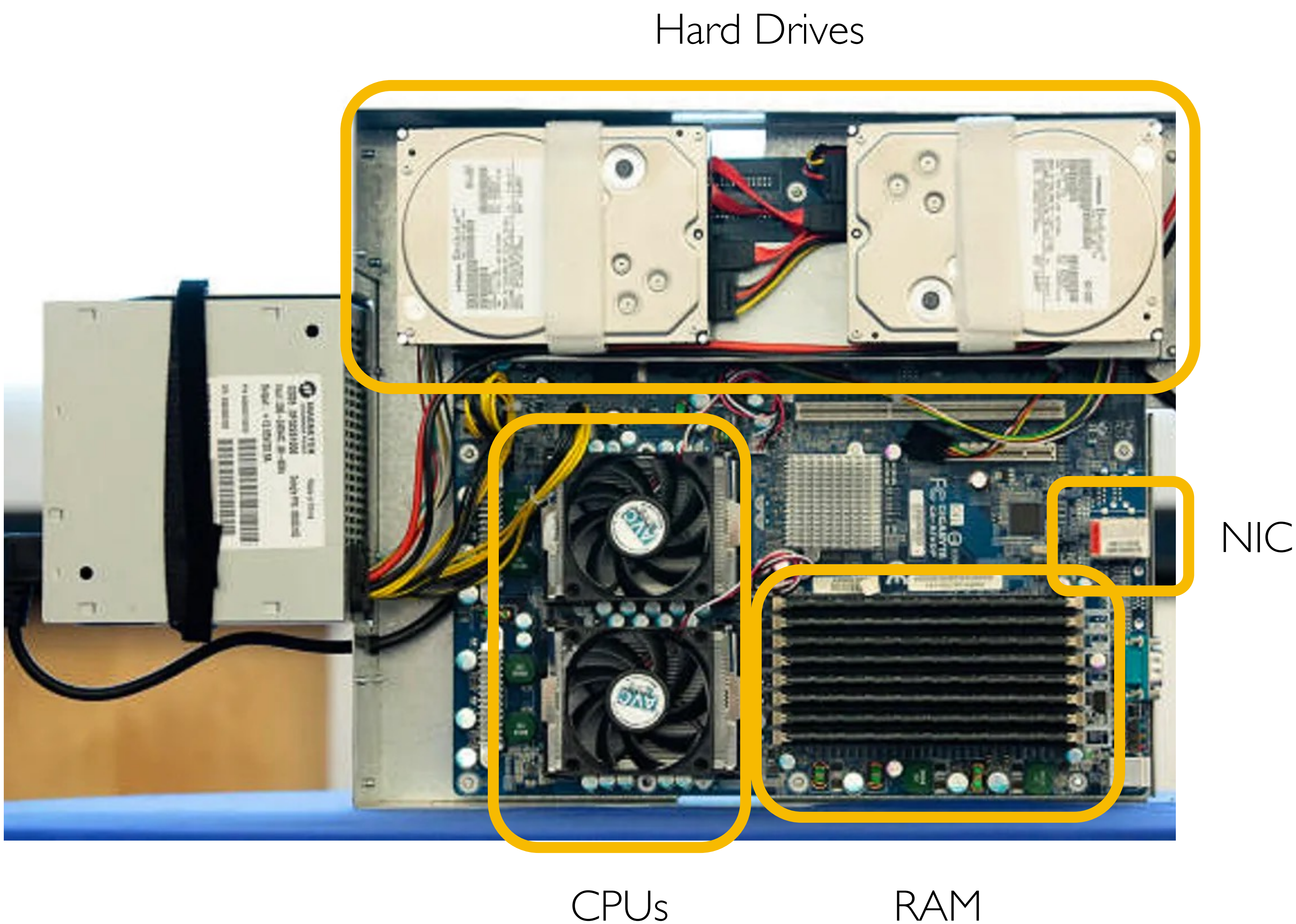
network provides communication between computers

switches/routers forward packets between computers

network interface card (NIC)

https://en.wikipedia.org/wiki/Network_switch#/media/File:2550T-PWR-Front.jpg

A real server



https://www.cnet.com/tech/tech-industry/google-uncloaks-once-secret-server-10209580/

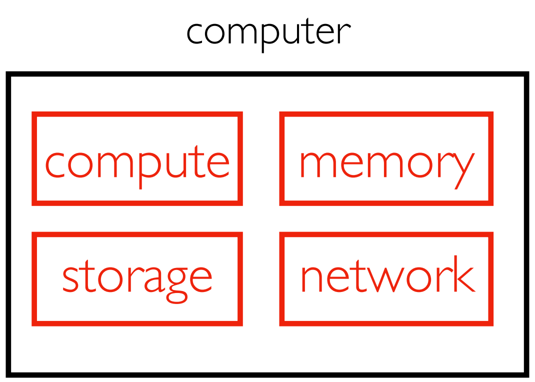
Big Data

Potential problems as datasets grow

- might run too slowly
- might not be able to run at all (for example, not enough memory)

Solutions:

- more efficient code
- use more resources



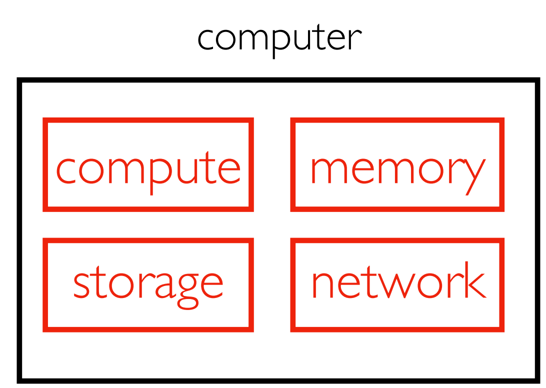
Big Data

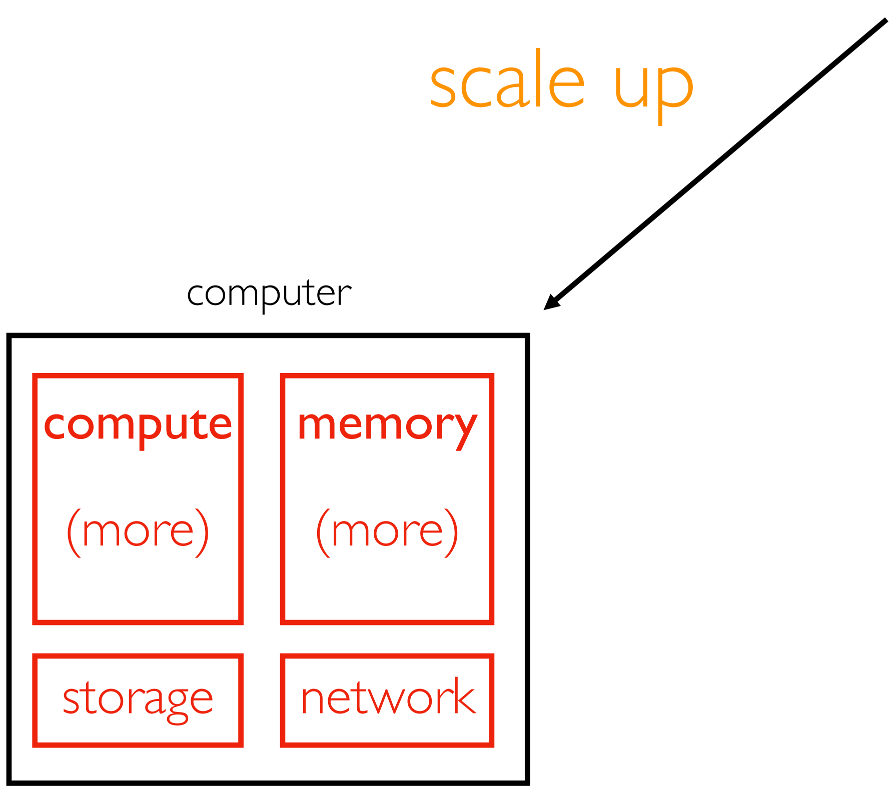
Potential problems as datasets grow

- might run too slowly
- might not be able to run at all (for example, not enough memory)

Solutions:

- more efficient code
- use more resources





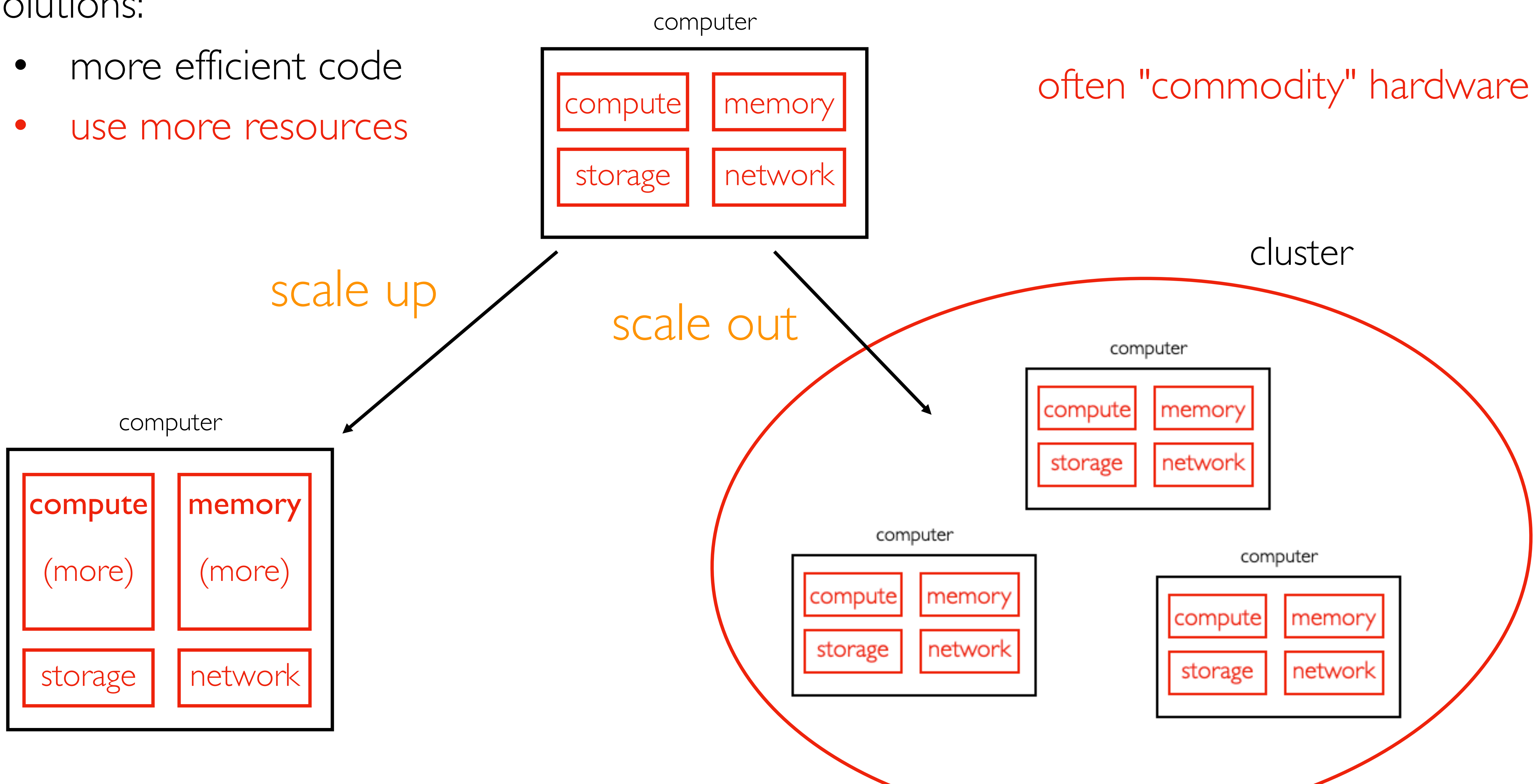
Big Data

Potential problems as datasets grow

- might run too slowly
- might not be able to run at all (for example, not enough memory)

Solutions:

- more efficient code
- use more resources



Outline

Course Overview

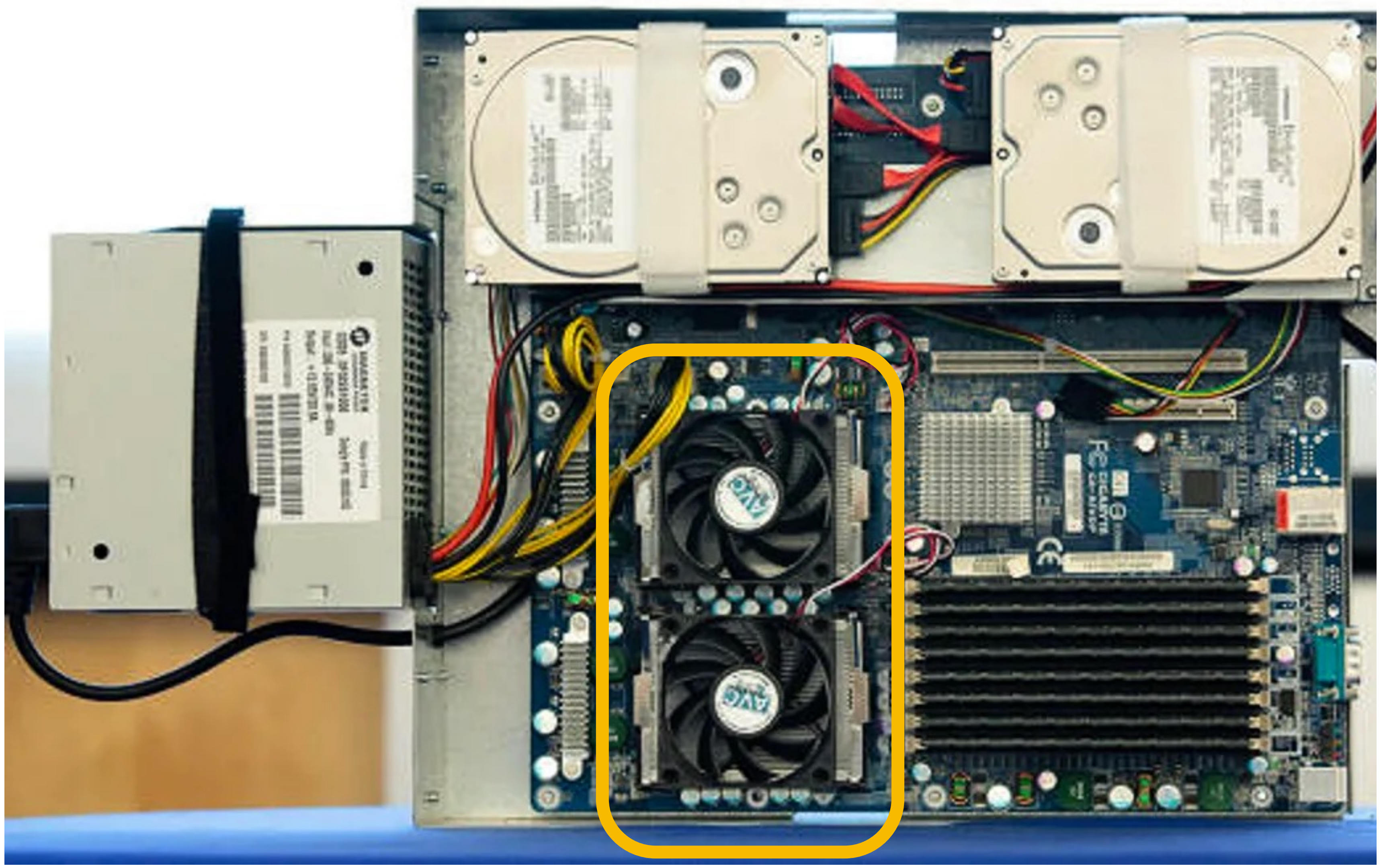
- Introductions
- Websites
- Other tools: Email, TopHat, Piazza, GitHub classroom

Resources

- Overview
- Compute
- Memory
- Storage
- Network

Deployment

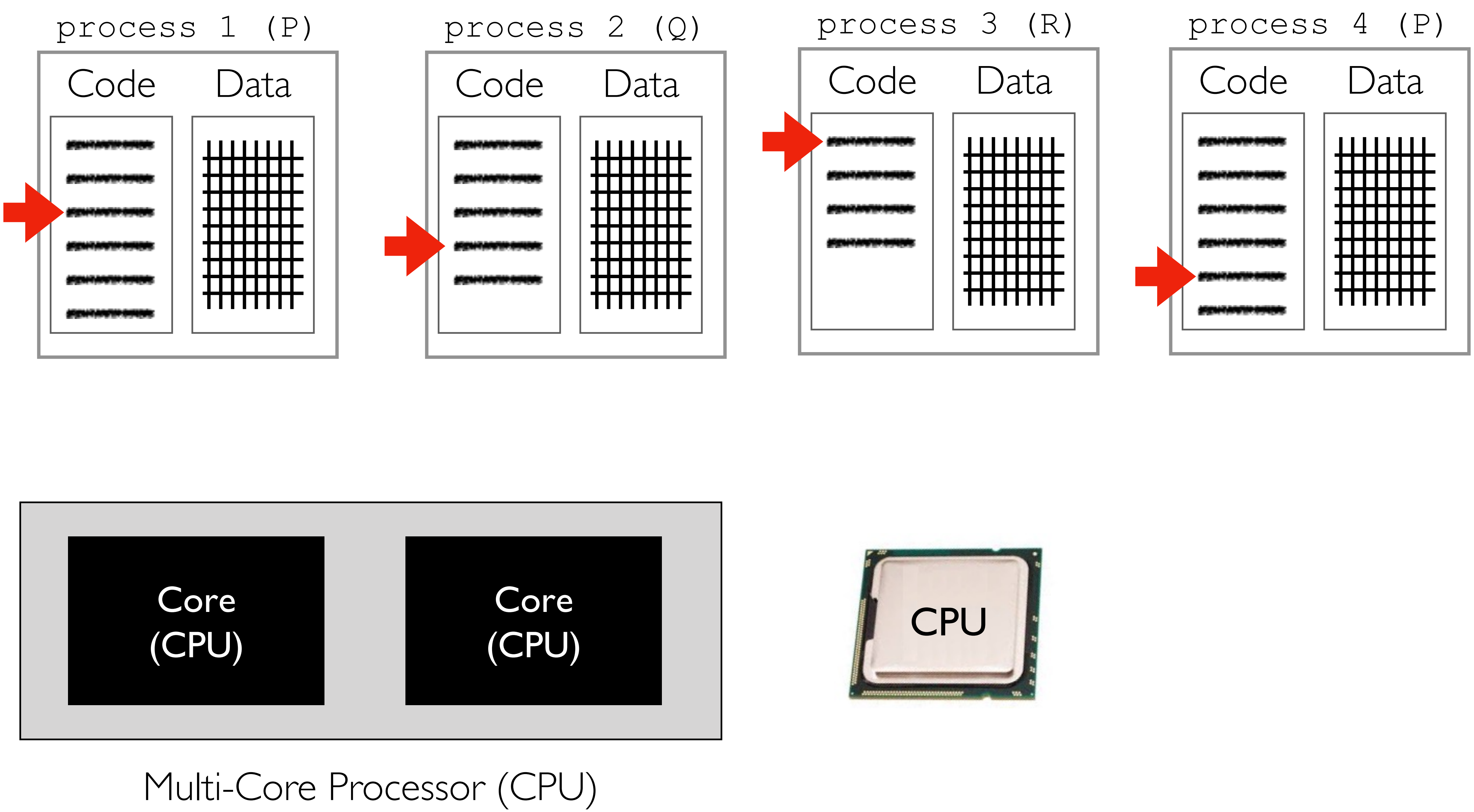
Compute

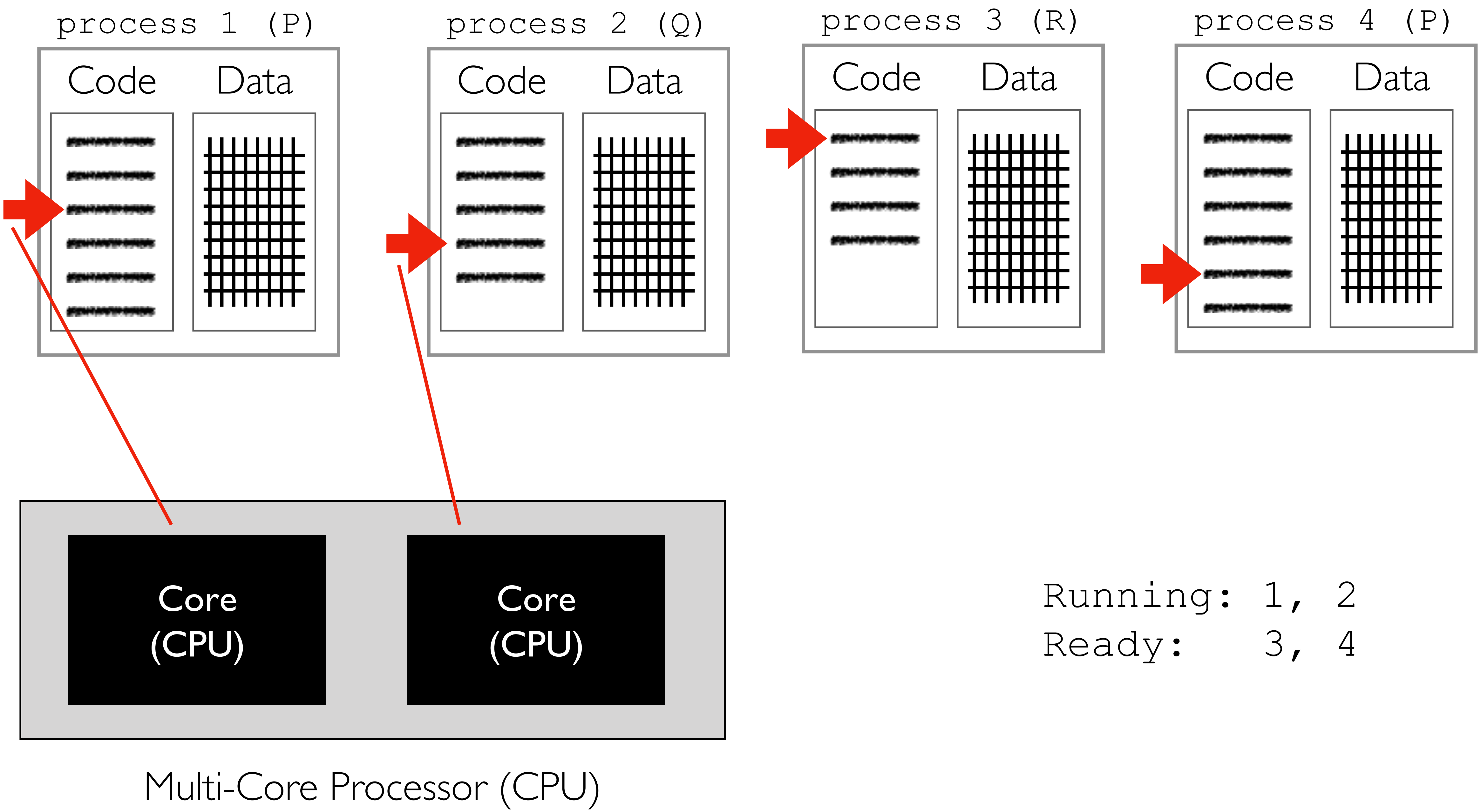


Some computers have multiple CPUs. Modern CPUs typically have multiple cores. Each core works like a CPU and runs programs by executing instructions.

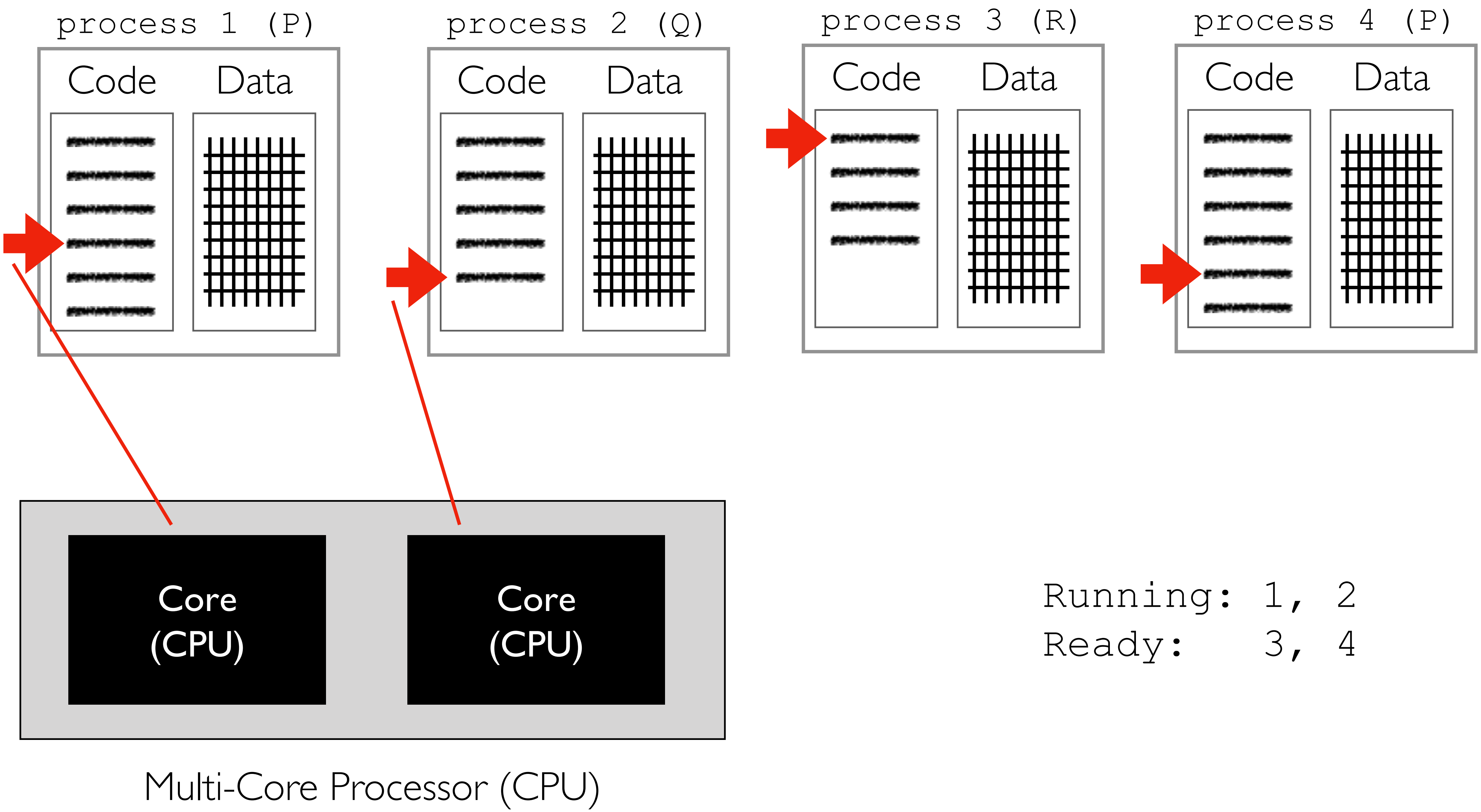
How do cores run machine code?

the operating system "schedules" processes on cores
(decides when they get to run)

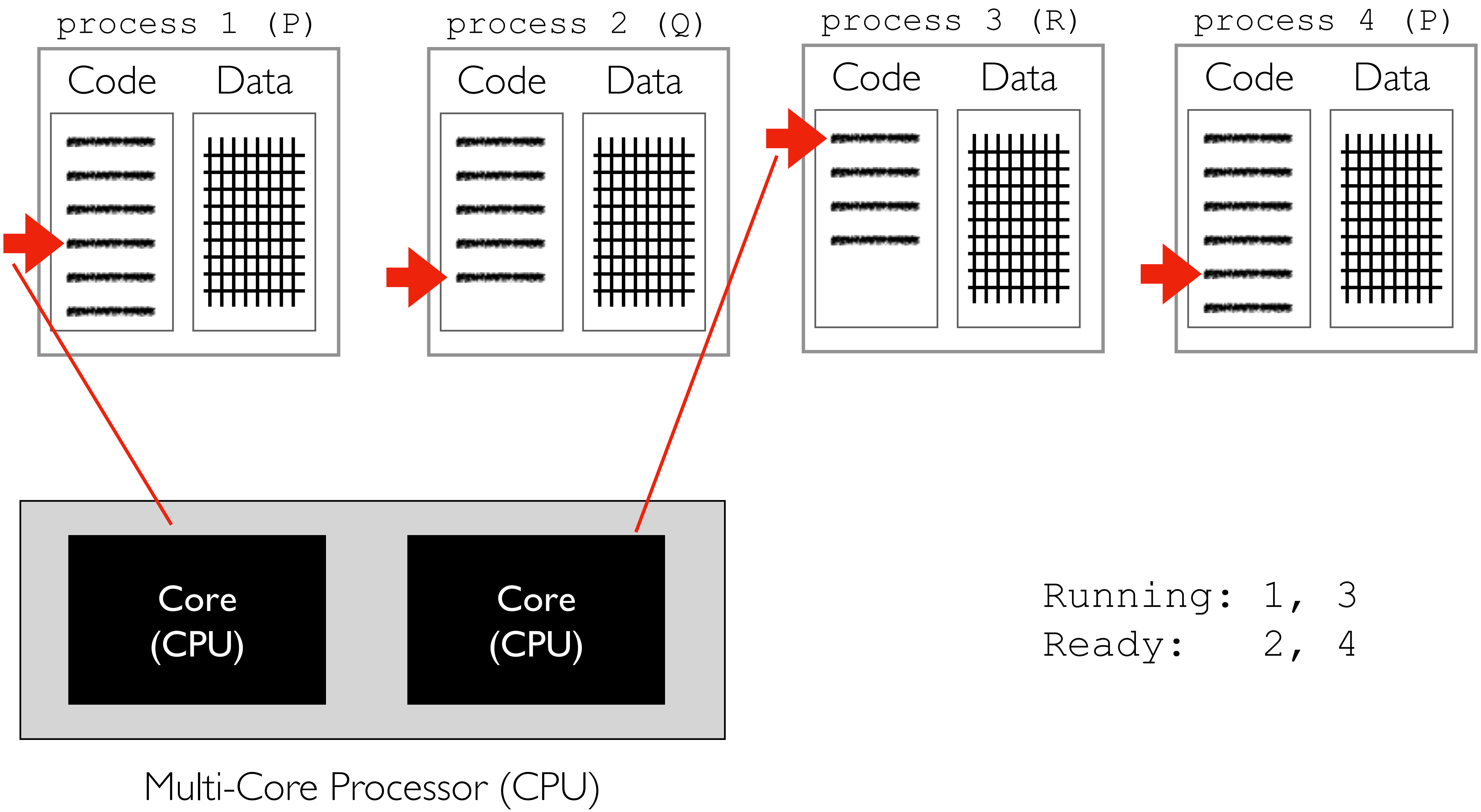




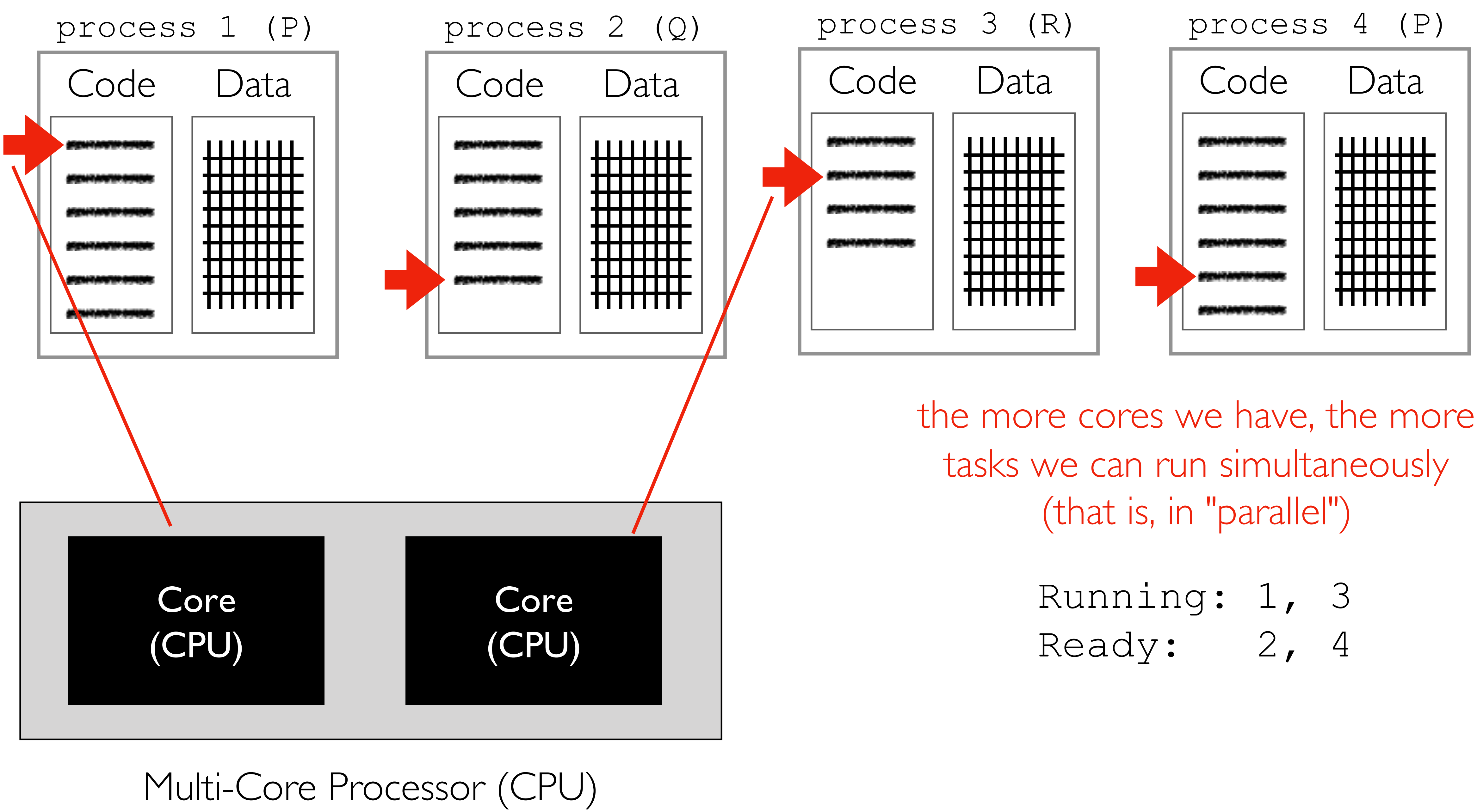

```
Running: 1, 2
Ready:   3, 4
```




```
Running: 1, 2
Ready:   3, 4
```



process 1 (P)
Code
Data
process 2 (Q)
Code
Data
process 3 (R)
Code
Data
process 4 (P)
Code
Data
Core
(CPU)
Core
(CPU)
Multi-Core Processor (CPU)
Running: 1, 3
Ready: 2, 4

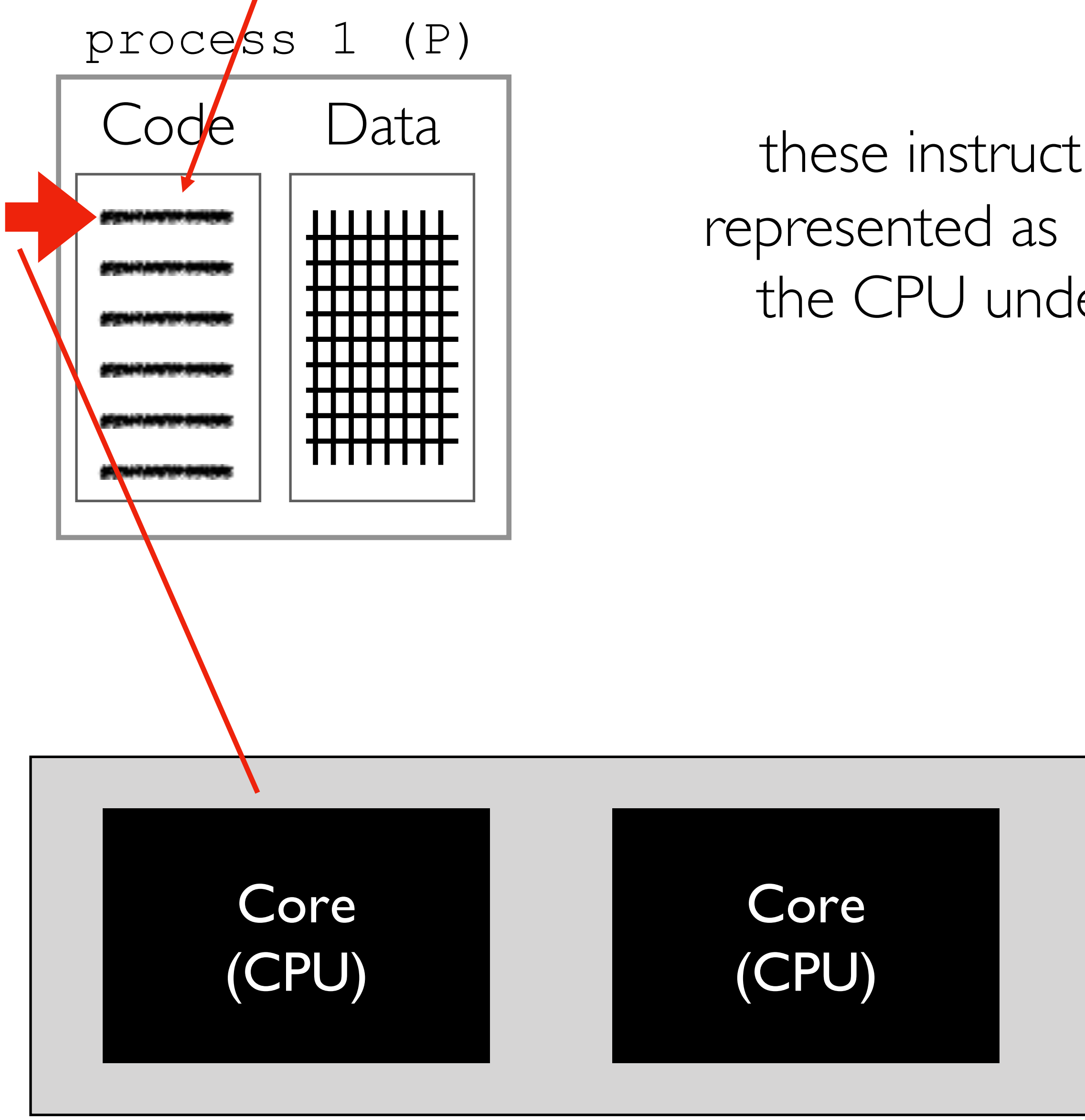


the more cores we have, the more tasks we can run simultaneously (that is, in "parallel")

```
Running: 1, 3
Ready:   2, 4
```

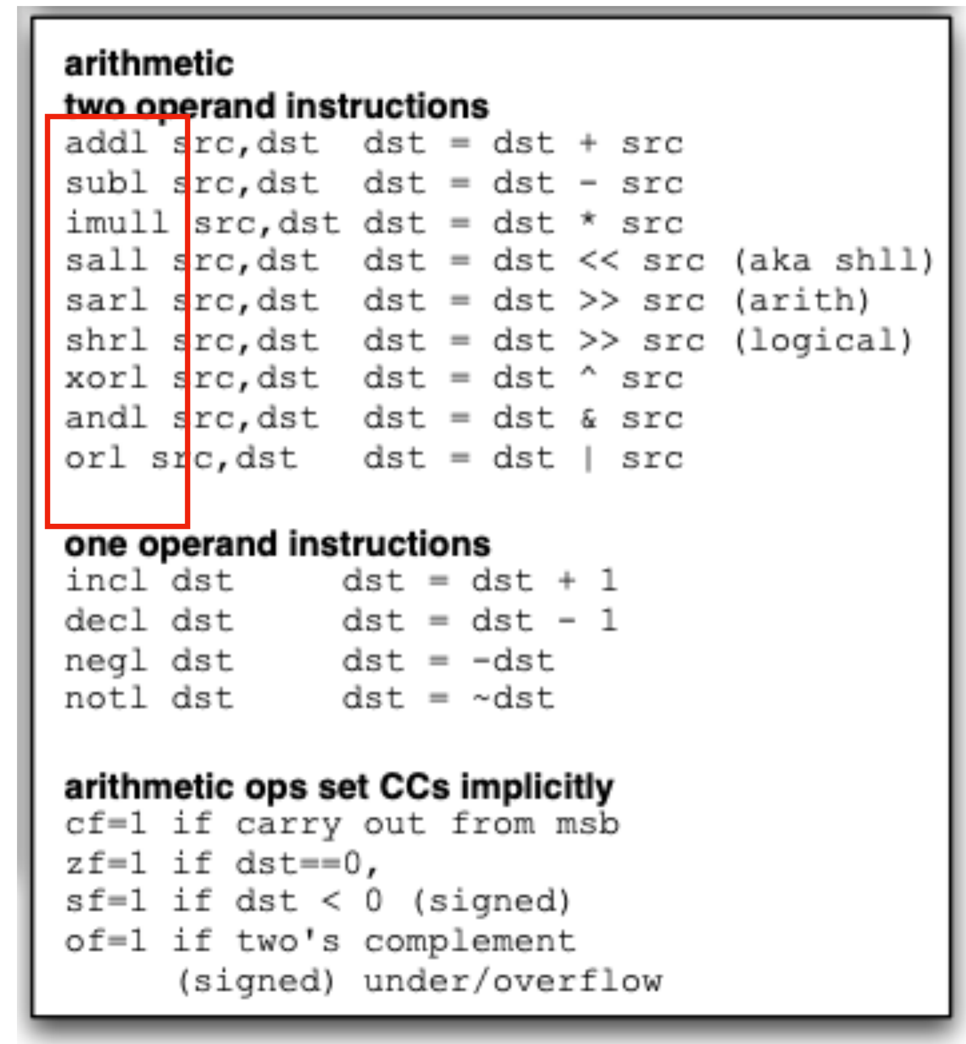
these instructions are in "machine code"
that the CPU can understand

https://pages.cs.wisc.edu/~deppeler/cs354/reference/x86-cheat-sheet.pdf



Multi-Core Processor (CPU)

these instructions are
represented as 1's and 0's
the CPU understands



arithmetic
two operand instructions

```
addl  src,dst   dst = dst + src
subl  src,dst   dst = dst - src
imull src,dst   dst = dst * src
sall  src,dst   dst = dst << src (aka shll)
sarl  src,dst   dst = dst >> src (arith)
shrl  src,dst   dst = dst >> src (logical)
xorl  src,dst   dst = dst ^ src
andl  src,dst   dst = dst & src
orl   src,dst   dst = dst | src
```

one operand instructions

```
incl dst        dst = dst + 1
decl dst        dst = dst - 1
negl dst        dst = -dst
notl dst        dst = ~dst
```

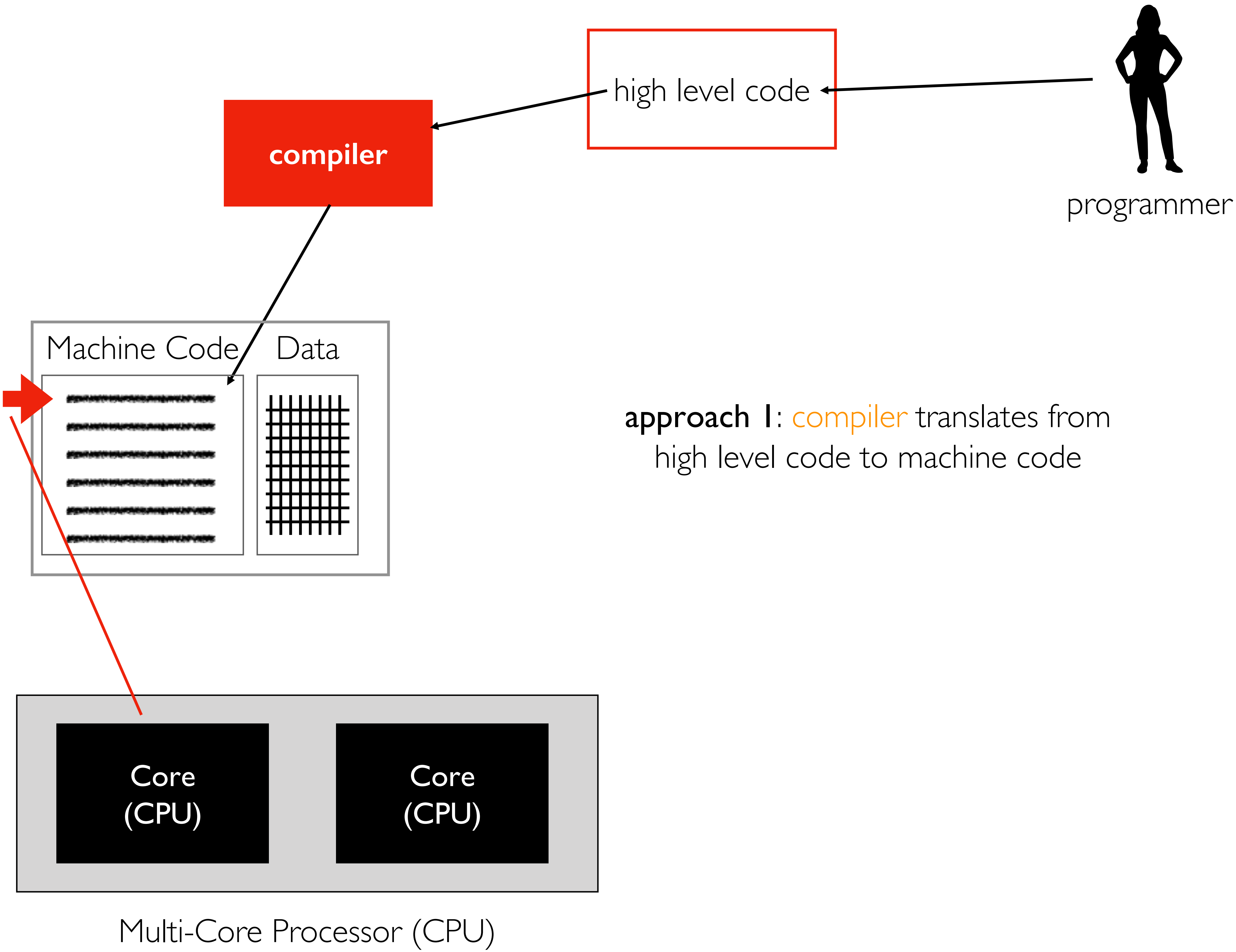
arithmetic ops set CCs implicitly

```
cf=1 if carry out from msb
zf=1 if dst==0,
sf=1 if dst < 0 (signed)
of=1 if two's complement
     (signed) under/overflow
```

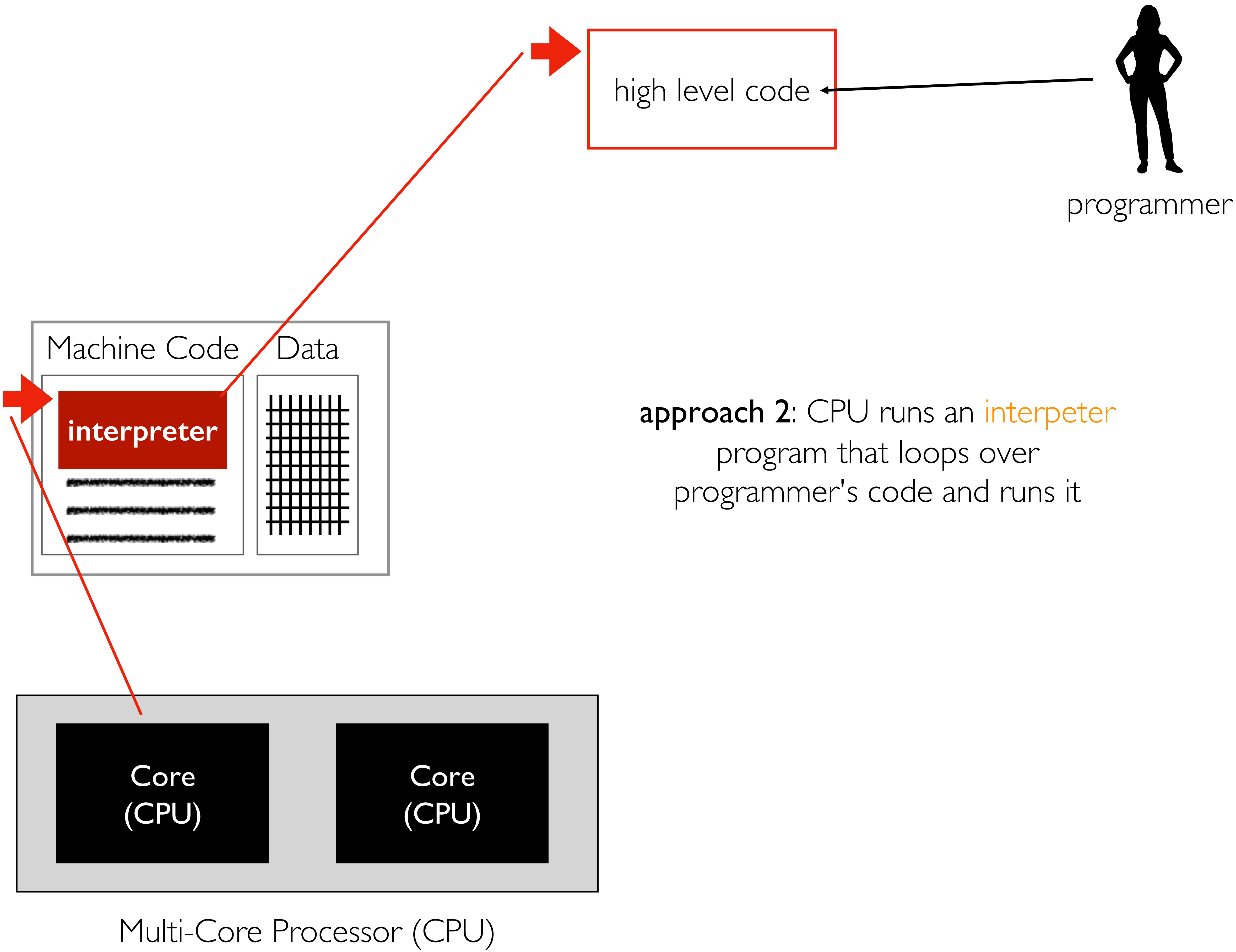
how do we bridge the gap between "high level"
code (Python/Java/etc) and machine code?

how do we bridge the gap between "high level" code (Python/Java/etc) and machine code?

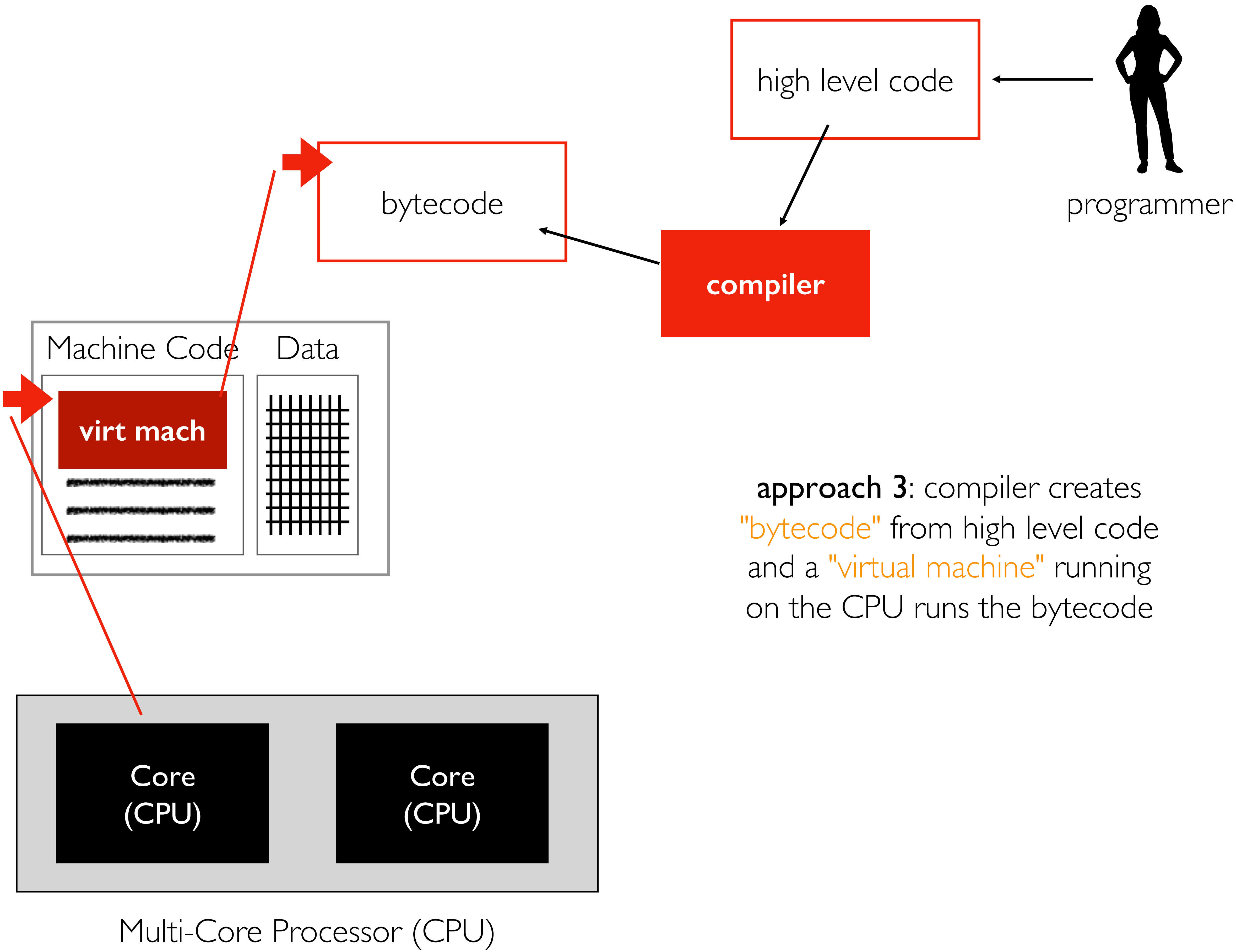
Note: we'll primarily write Python this semester, but it helps to explore this in general to understand how systems like Spark work (which is written in Scala and uses the Java Virtual Machine)



approach 1: compiler translates from high level code to machine code

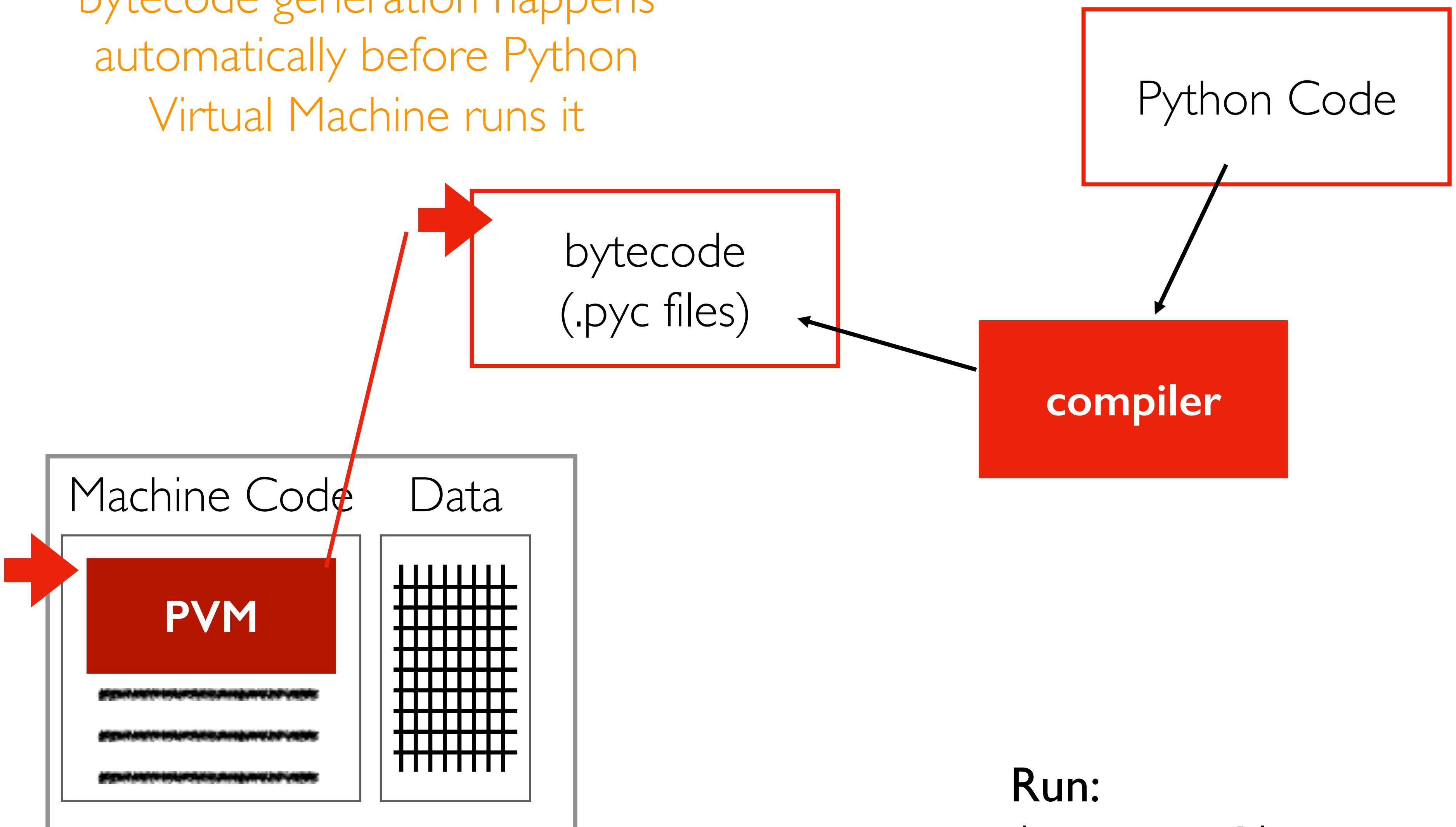


approach 2: CPU runs an interpeter program that loops over programmer's code and runs it



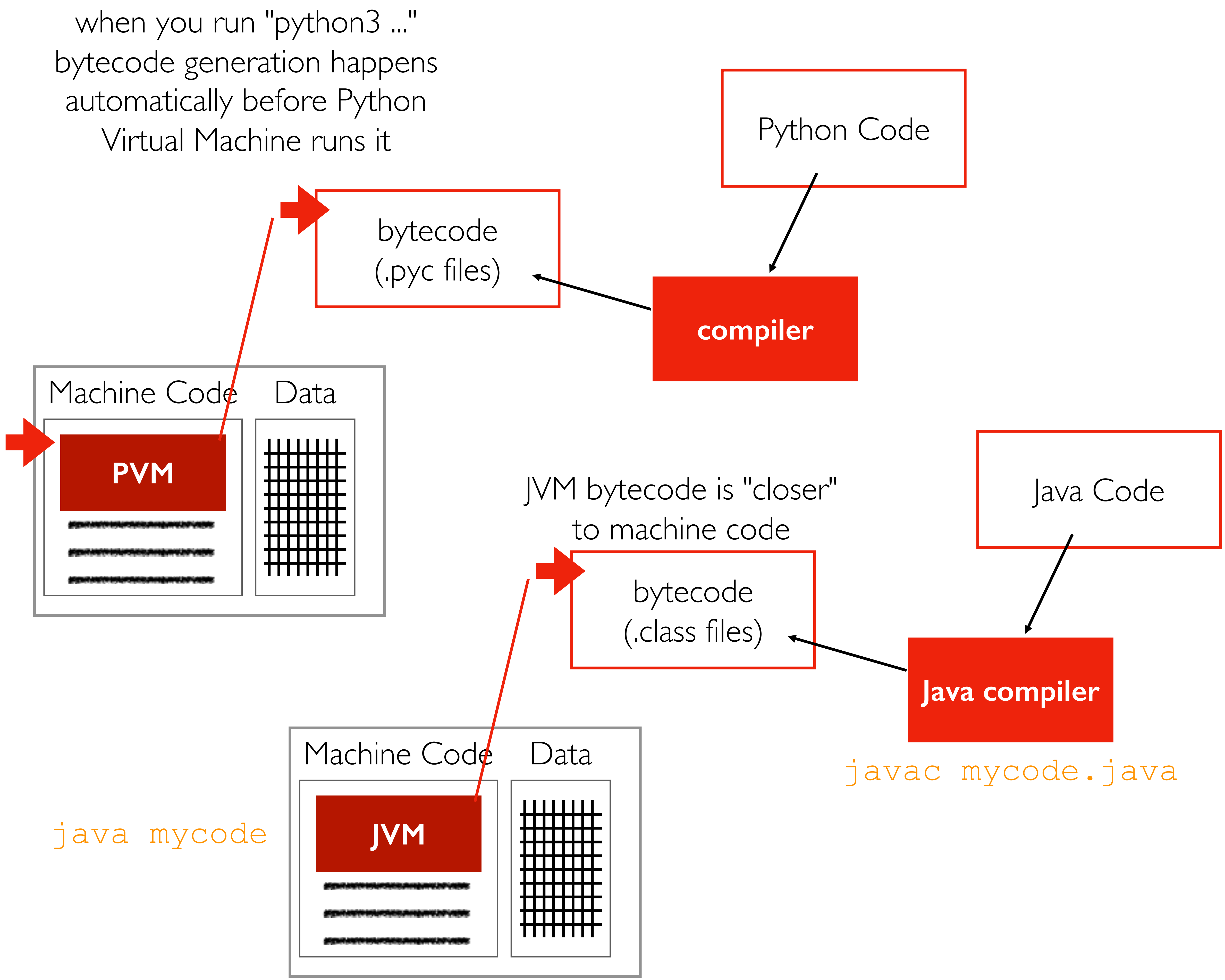
approach 3: compiler creates "bytecode" from high level code and a "virtual machine" running on the CPU runs the bytecode

when you run "python3 ..."
bytecode generation happens
automatically before Python
Virtual Machine runs it

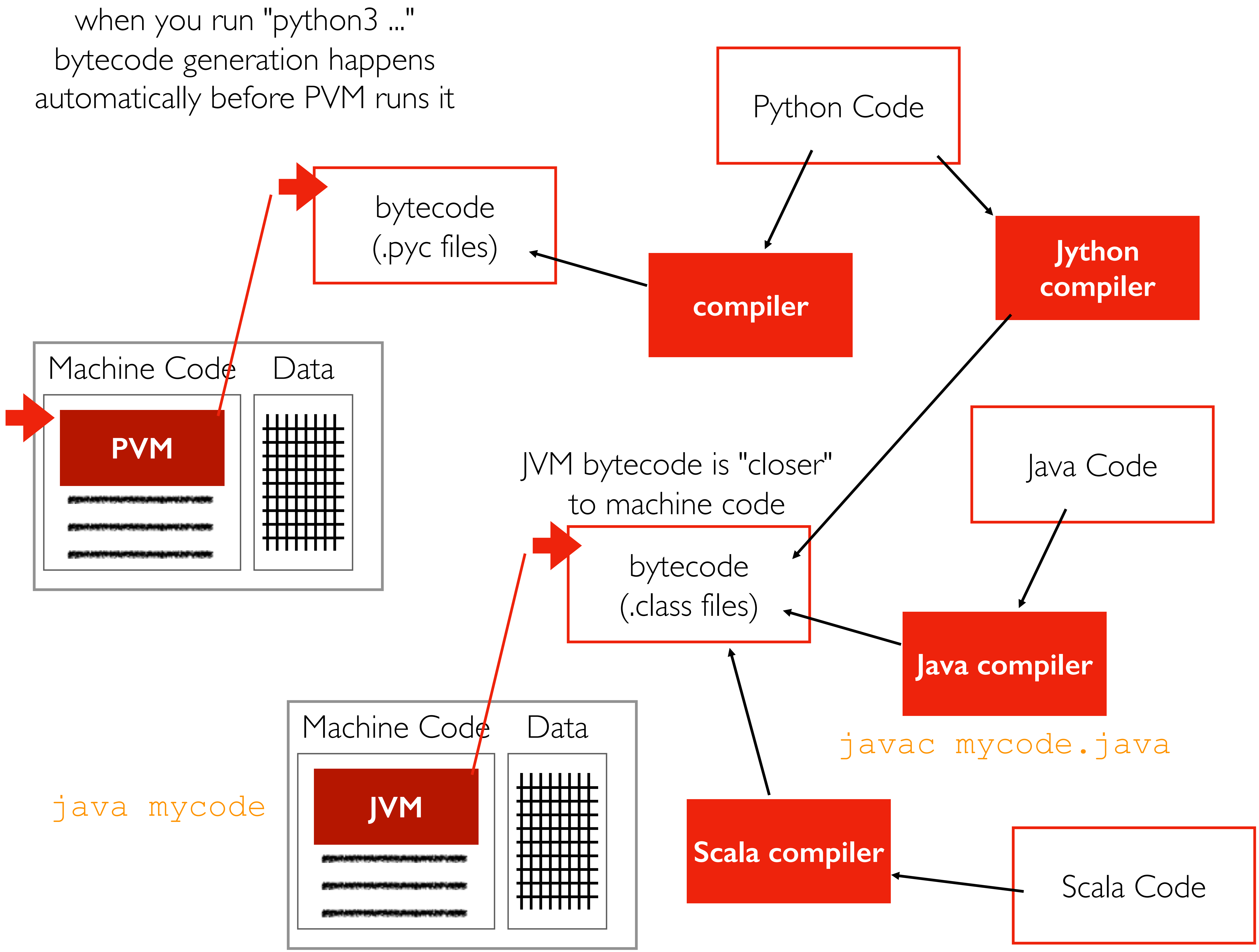


Run:

```
import dis
dis.dis("z = x + y * 2")
```

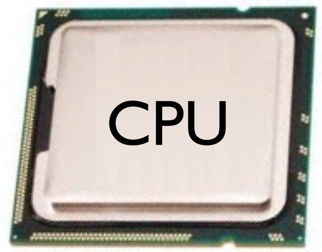


when you run "python3 ..."
bytecode generation happens
automatically before Python
Virtual Machine runs it
Python Code
bytecode
(.pyc files)
compiler
Machine Code
Data
PVM
JVM bytecode is "closer"
to machine code
Java Code
bytecode
(.class files)
Java compiler
javac mycode.java
java mycode
Machine Code
Data
JVM

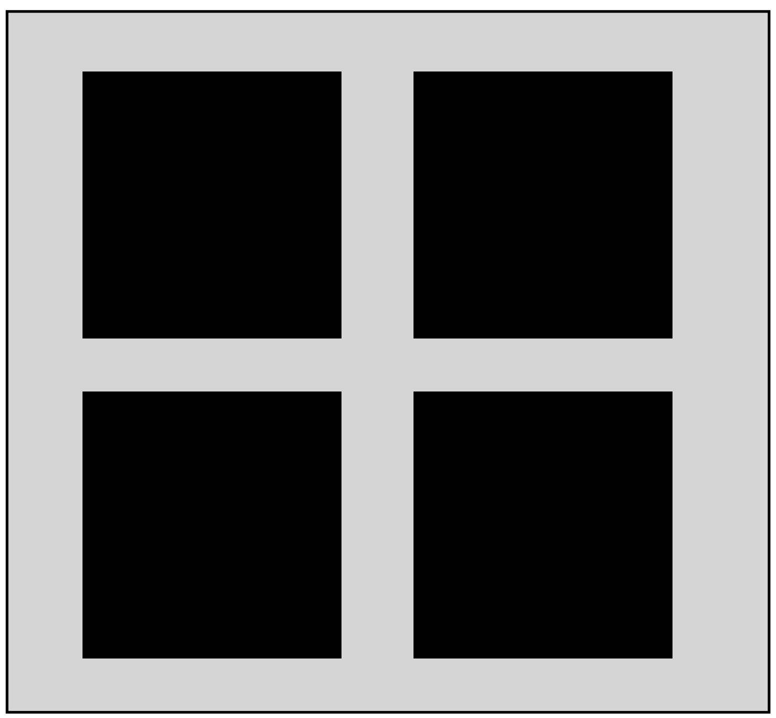

when you run "python3 ..."
bytecode generation happens
automatically before PVM runs it
Python Code
bytecode
(.pyc files)
compiler
Jython
compiler
Machine Code
Data
PVM
JVM bytecode is "closer"
to machine code
Java Code
bytecode
(.class files)
Java compiler
javac mycode.java
Machine Code
Data
java mycode
JVM
Scala compiler
Scala Code

What are alternatives to CPUs for compute?

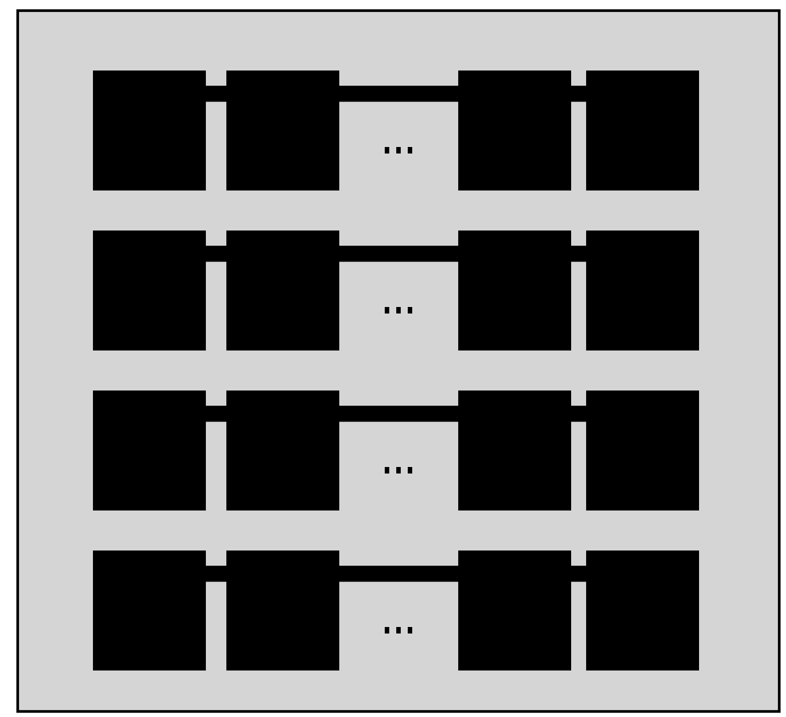
GPUs (graphical processing units) are an alternative compute resource.



https://en.wikipedia.org/wiki/Nvidia_Tesla



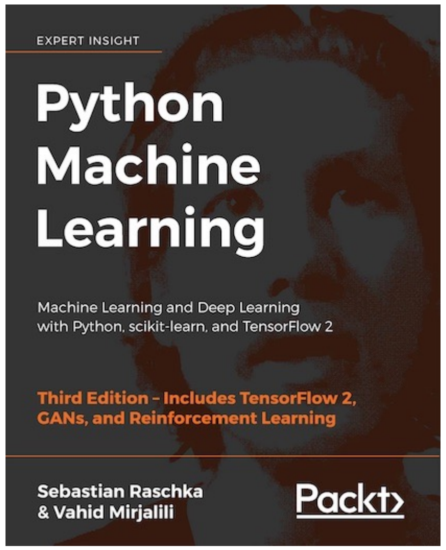
few cores that are fast,
flexible, independent



many cores that are slow,
float-optimized, coordinated

GPU vs. CPU: Cost Comparison

TOP HAT



The GPU is 30% cheaper but 28x faster at floating-point operations!

Specifications	Intel® Core™ i7-6900K Processor Extreme Ed.	NVIDIA GeForce® GTX™ 1080 Ti
Base Clock Frequency	3.2 GHz	< 1.5 GHz
Cores	8	3584
Memory Bandwidth	64 GB/s	484 GB/s
Floating-Point Calculations	409 GFLOPS	11300 GFLOPS
Cost	~ $1000.00	~ $700.00

https://sebastianraschka.com/books.html

Resource metric: FLOPS (floating-point operations per second)

- floating-point ops: add, mult, etc (how to weight?)
- prefixes: K (thousand), M (million), G (billion), T (trillion)
  how many TFLOPS does the above GPU provide?

PyTorch Python package

- lets you readily use GPUs; we'll use this for project 2

Outline

Course Overview

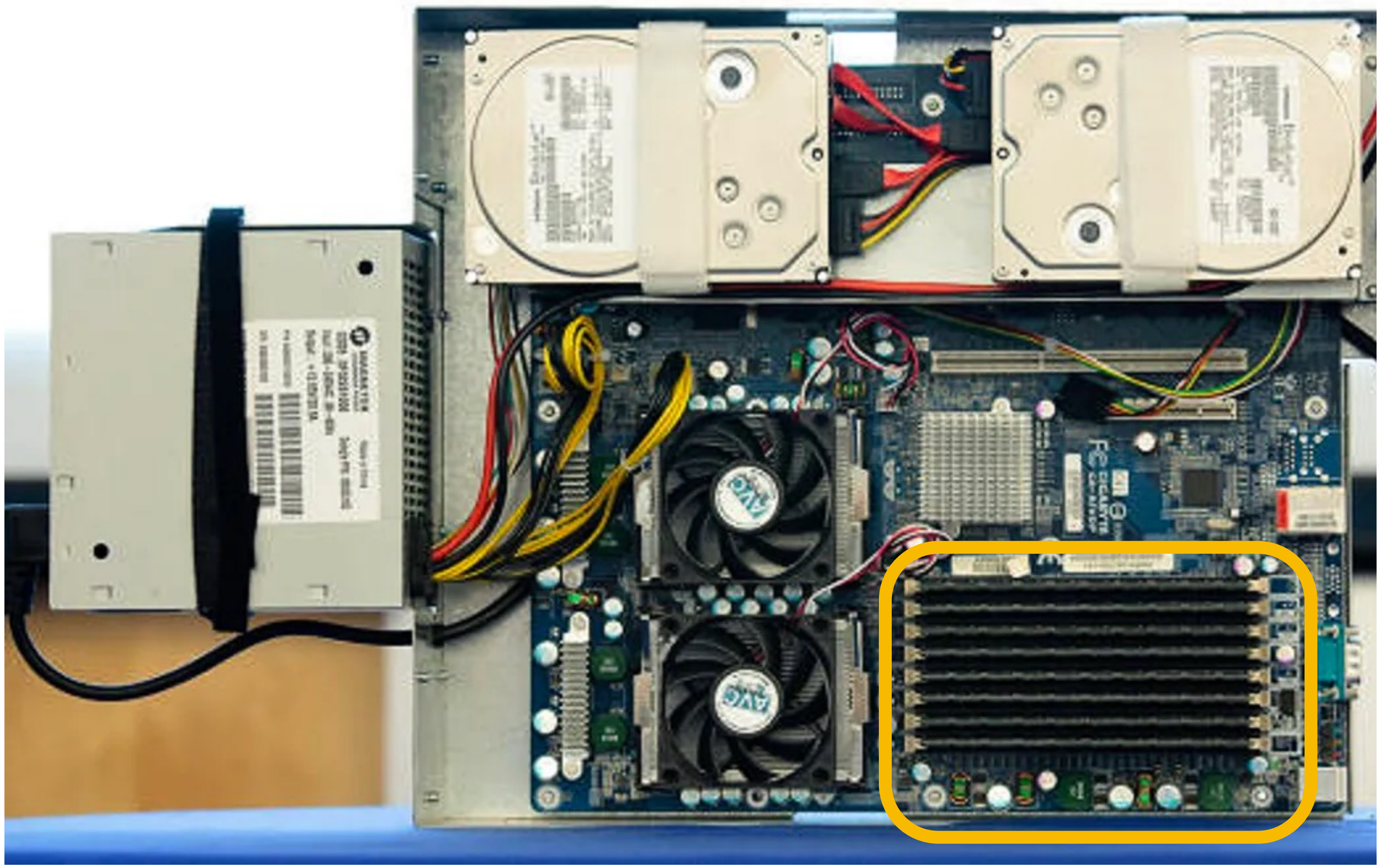
- Introductions
- Websites
- Other tools: Email, TopHat, Piazza, GitHub classroom

Resources

- Overview
- Compute
- Memory
- Storage
- Network

Deployment

RAM: Random Access Memory



What is "Random"?

"Random" means we can jump around and access data from different locations efficiently.

In contrast, some devices that hold data are only efficient sequentially:

https://en.wikipedia.org/wiki/Cassette_tape

Bits

RAM holds bits.

A "bit" holds a 0 or a 1 (two possible values).

2 bits can hold 00, 01, 10, 11 (four possible values).

N bits can hold 2^N possible values.

Representation

Different encodings/representations decide what a combination of bits mean.

bits				
000	A	0	-4	
001	B	1	-3	
010	C	2	-2	colors
011	D	3	-1	images
100	E	4	0	floats
101	F	5	1	etc.
110	G	6	2	
111	H	7	3	

Bytes

A byte is 8 bits, so can hold $2^8 = 256$ possible values.

RAM is "byte addressible"

- each byte of data has it's own address the CPU can use to access it
- extracting a single bit from a byte actually involves more steps than using the whole byte

Units:

- 1 KB = 1024 bytes (or sometimes 1000 bytes)
- 1 MB = 1024 KB (or sometimes 1000 KB)
- 1 GB = 1024 MB (or sometimes 1000 MB)
- 1 TB = 1024 GB (or sometimes 1000 GB)

RAM Characteristics

Characteristics

- small (for example, an e2-medium VM has 4 GB)
- volatile (contents lost upon reboot)
- fast (much faster that storage devices)

Some uses

- actively used data (e.g., Python list, program code, DataFrame)
- copies of "hot" data (frequently accessed) from storage

Outline

Course Overview

- Introductions
- Websites
- Other tools: Email, TopHat, Piazza, GitHub classroom

Resources

- Overview
- Compute
- Memory
- Storage
- Network

Deployment

Block devices: storing 0s and 1s

Hard Disk Drives (HDDs)

- 0s/1s stored on spinning magnatized platter
- moving head reads/writes data

Solid State Disks (SSDs)

- 0s/1s stored in charged cells
- no moving parts (faster)

Both are "block devices"

- data is read/written in blocks of many bytes (for example, 0.5 KB)
- reading 1 byte or 1 block takes same time

HDD and SSD Characteristics

Characteristics

- large (> 1 TB devices are affordable)
- nonvolatile (contents lost upon reboot)
- slow (much slower that memory)

Some uses

- large datasets
- data that needs to be preserved long term

Metrics

Capacity

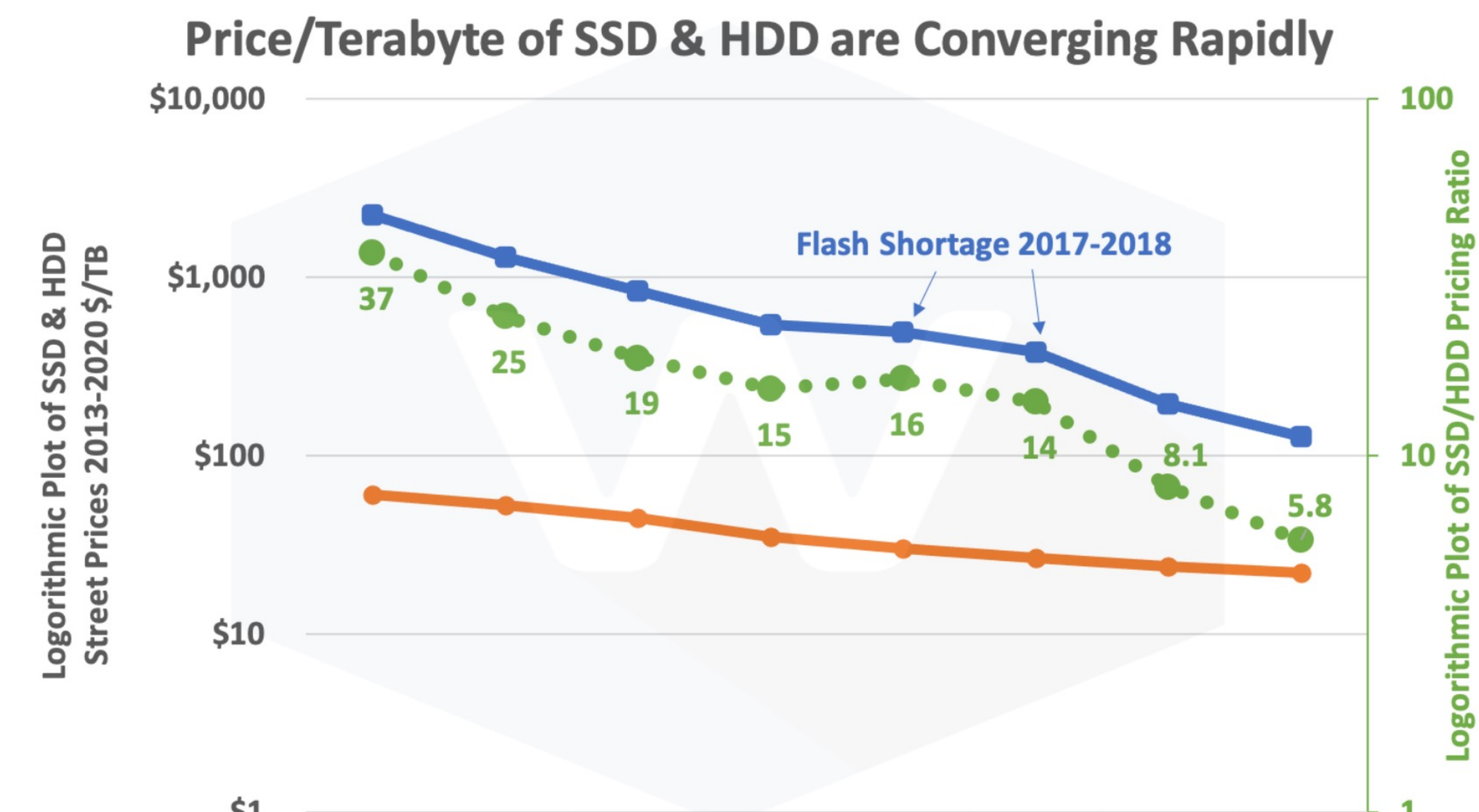
- how much data can be stored?
- measured in bytes (for example, 500 GB)

Throughput

- how fast can data be read/written?
- measure in bytes/second (for example, 200 MB/s)
- throughput will depend on access pattern (for example, spinning disks have low throughput for random accesses)

Latency

- how long does it take to do one I/O (e.g., 10 ms)



	2013	2014	2015	2016	2017	2018	2019	2020
SSD $/TB	$2,220	$1,293	$840	$538	$493	$378	$195	$128
HDD $/TB	$60	$53	$45	$35	$30	$26.8	$23.9	$22.2
Ratio SSD/HDD $/TB	37	25	19	15	16	14	8.1	5.8

SSD $/TB HDD $/TB Ratio SSD/HDD $/TB

Source: © Wikibon, 2021

Figure 3 – Flash & HDD Pricing 2013 – 2020

Source: © Wikibon, 2021. Wikibon uses historical data is from multiple sources, including IDC, Gartner, Kitguru, Nidec, Trendfocus, and Wells Fargo LLC

https://wikibon.com/qlc-flash-hamrs-hdd/

Outline

Course Overview

- Introductions
- Websites
- Other tools: Email, TopHat, Piazza, GitHub classroom

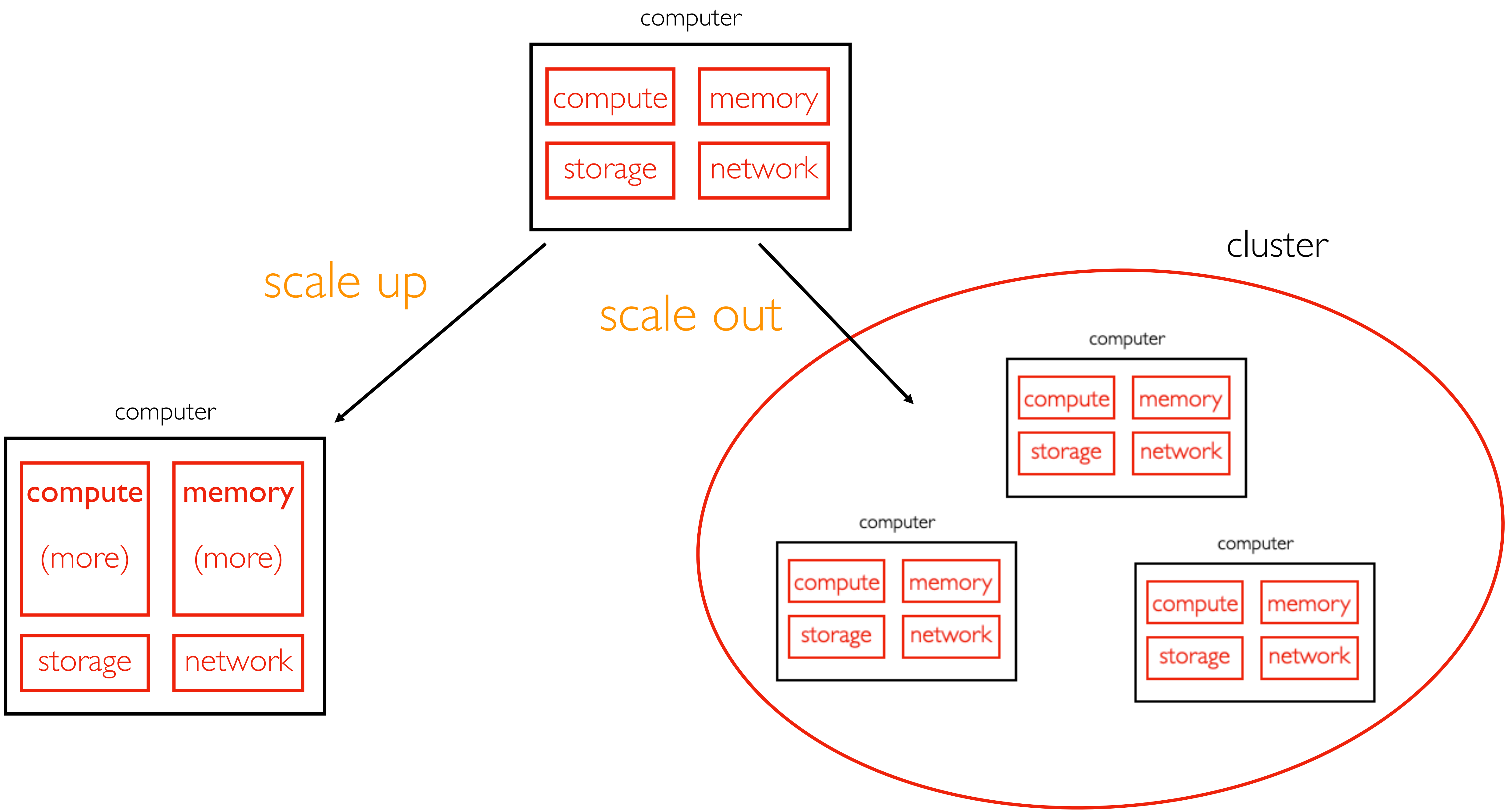
Resources

- Overview
- Compute
- Memory
- Storage
- Network

Deployment

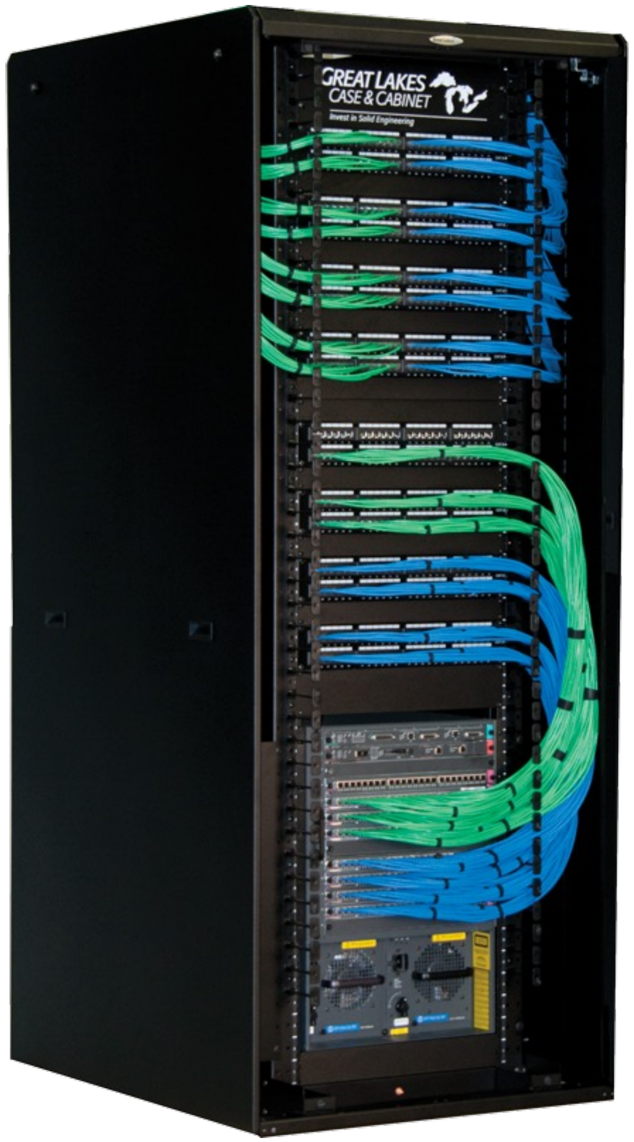
Network

When scaling out, many nodes (computers) will be communicating via a network.



Network

Server



Rack

https://www.dotmagazine.online/issues/digital-infrastructure-and-transforming-markets/data-center-models

10-plus-e-2336-2-9ghz-6-core-1p-16gb-u-4sff-500w-rps-server/p/p44115-b21?ef_id=Cj0KCQiAt66eBhCnARIsAKf3ZNFJsg49UV6Zm33R7lkRqi-XOd_JECmdyqNMAm2CKLSm_F-z6JTYDTQaAgMTEALw_wcB:G:s&s_kwcid=AL!13472!3!331628972784!!!g!318267171339!!1707918369!67076417419&gclsrc=aw.ds&gclid=Cj0KCC

erver-rack-online.com/gl910ent-4048sss.html?utm_medium=shoppingengine&utm_source=googlebase&utm_source=google&utm_medium=cpc&adpos=&scid=scplpgl910ent-4048sss&sc_intid=gl910ent-4048sss&gclid=Cj0KCQiAt66eBhCnARIsAKf3ZNEMYINPAA0RFGQIF0DsieCM6oh7i3kuJvJlpnmJAIOpAJ3RWT11QMAaAq

Network

Server

Data Center

https://www.dotmagazine.online/issues/digital-infrastructure-and-transforming-markets/data-center-models

10-plus-e-2336-2-9ghz-6-core-1p-16gb-u-4sff-500w-rps-server/p/p44115-b21?ef_id=Cj0KCQiAt66eBhCnARIsAKf3ZNFJsg49UV6Zm33R7lkRqi-XOd_JECmdyqNMAm2CKLSm_F-z6JTYDTQaAgMTEALw_wcB:G:s&s_kwcid=AL!13472!3!331628972784!!!g!318267171339!!1707918369!67076417419&gclsrc=aw.ds&gclid=Cj0KCQ

erver-rack-online.com/gl910ent-4048sss.html?utm_medium=shoppingengine&utm_source=googlebase&utm_source=google&utm_medium=cpc&adpos=&scid=scplpgl910ent-4048sss&sc_intid=gl910ent-4048sss&gclid=Cj0KCQiAt66eBhCnARIsAKf3ZNEMYINPAA0RFGQIF0DsieCM6oh7i3kuJvJlpnmJAIOpAJ3RWT11QMAaAq

Topology

Example configuring Hadoop File System (HDFS) to store data based on network topology:

python Example

```
#!/usr/bin/python3
# this script makes assumptions about the physical environment.
#  1) each rack is its own layer 3 network with a /24 subnet, which
# could be typical where each rack has its own
#     switch with uplinks to a central core router.
#
#             +-----------+
#             |core router|
#             +-----------+
#            /             \
#   +-----------+        +-----------+
#   |rack switch|        |rack switch|
#   +-----------+        +-----------+
#   | data node |        | data node |
#   +-----------+        +-----------+
#   | data node |        | data node |
#   +-----------+        +-----------+
```

https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/RackAwareness.html

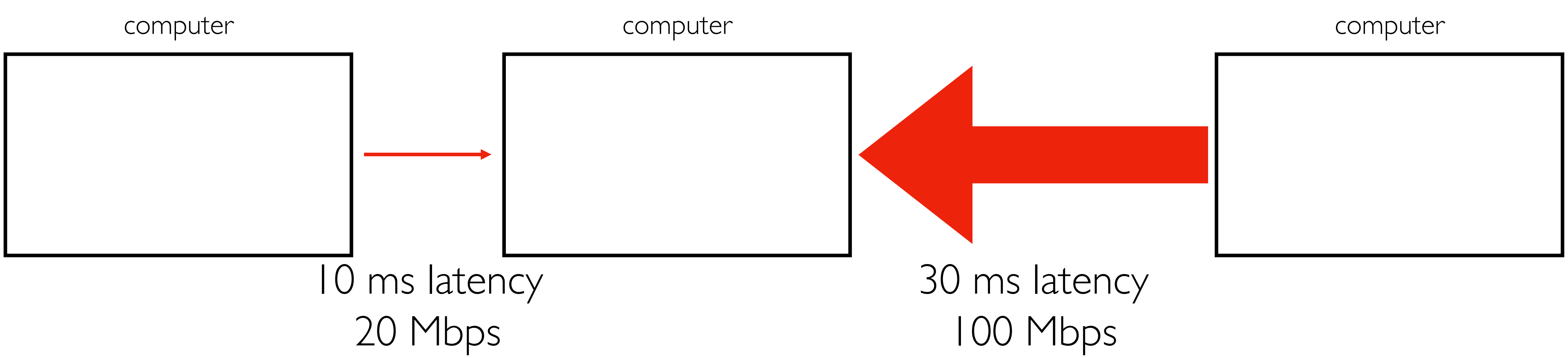
Metrics

Latency

- how long does it take to send messages between two points
- seconds, milliseconds (ms), etc

Bandwidth/Throughput

- how many **bits** can be sent per second?
- Mbps (mega bits per second -- note lower case "b")
- What is faster, 10 Mbps or 10 MB/s?



Outline

Course Overview

- Introductions
- Websites
- Other tools: Email, TopHat, Piazza, GitHub classroom

Resources

- Overview
- Compute
- Memory
- Storage
- Network

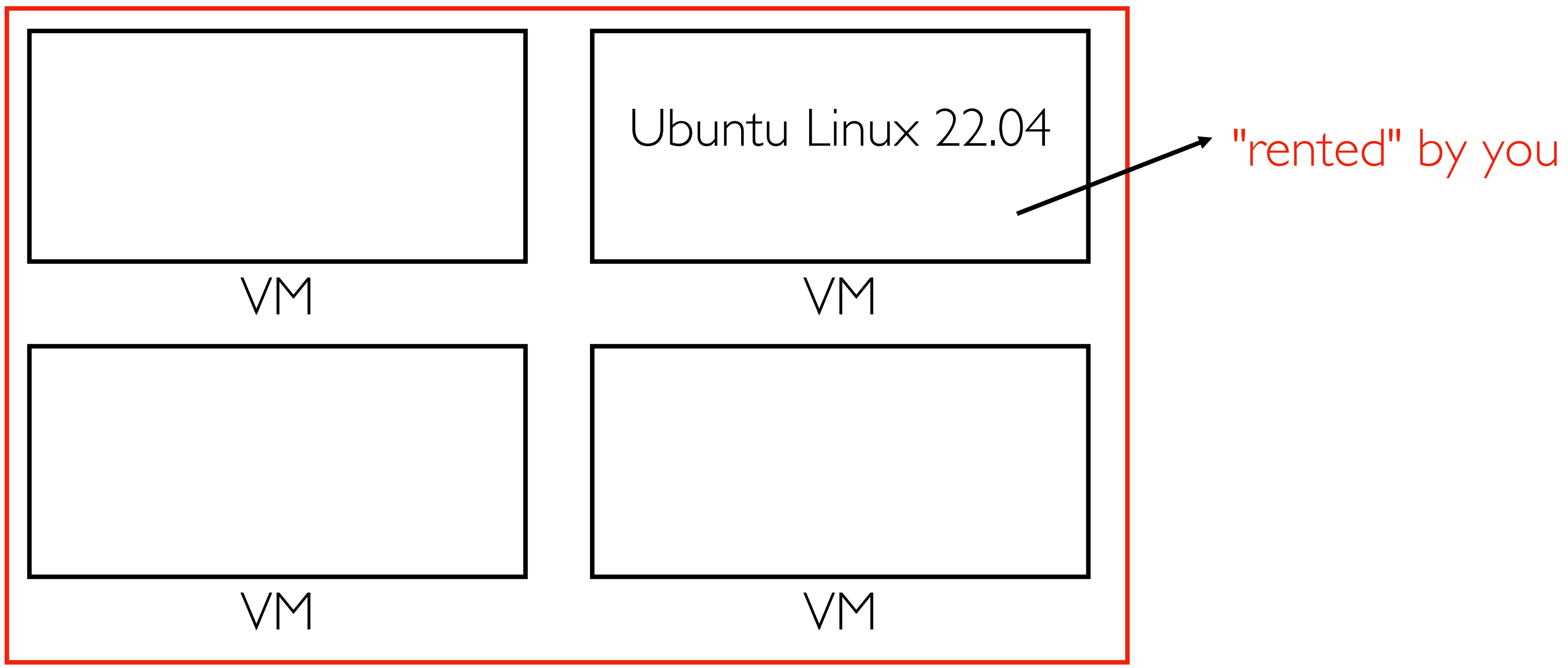
Deployment

Deployment

Deployment means running code somewhere

- often a major undertaking when working with clusters

We'll be deploying systems on Google Cloud Platform (GCP) this semester using virtual machines (VMs). We'll use Ubuntu Linux 22.04.



physical machine in a Google data center

Cloud Budget Plan

We'll be announcing how to get $100 of free credit for the semester.

You'll use different VMs/resources at different points. Follow the plan here to create the right types.

https://github.com/cs544-wisc/s24/blob/main/projects.md#compute-setup

Important: check your credits regularly to make sure you're not using too much too soon based on the "cumulative" column. If you accidentally configured to use too much, fix the issue, then find ways to catch up (like shutting down your VM overnight).

VM Maintenance Responsibilities

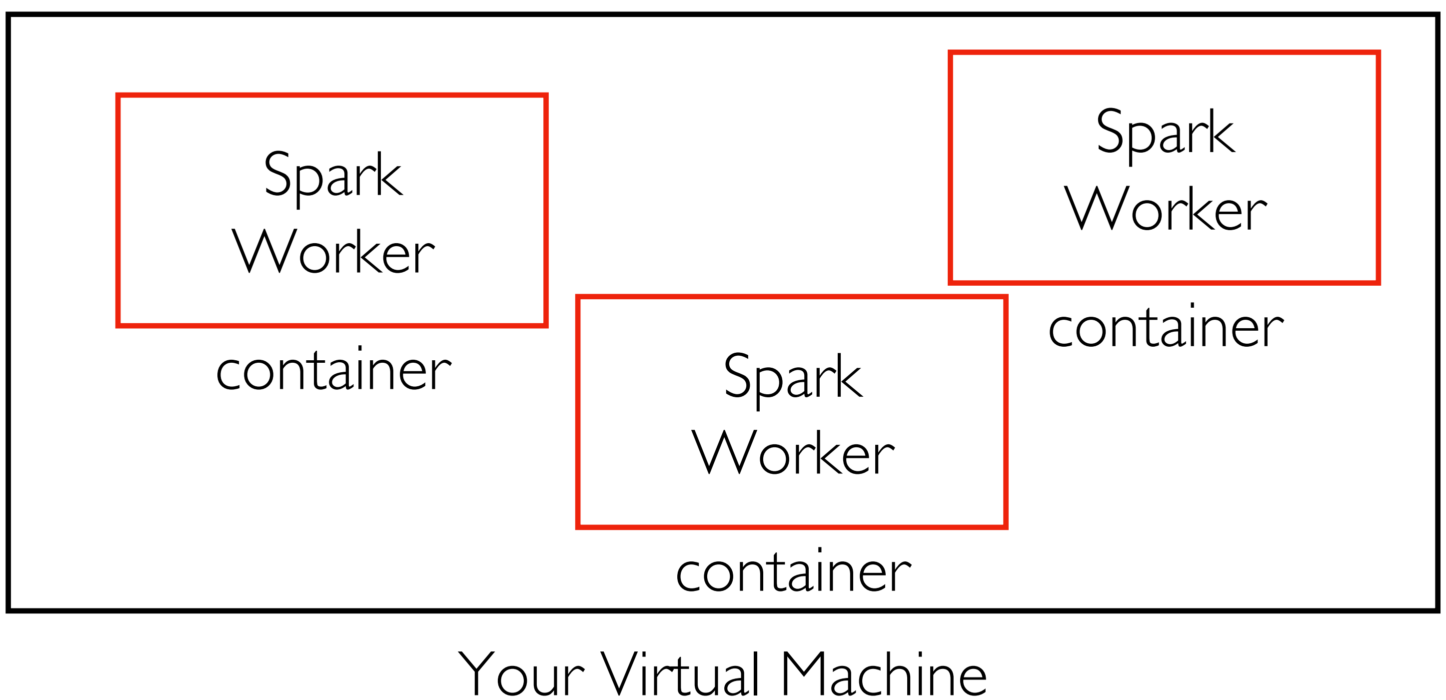
Some of your responsibities:

- creating/re-creating the VM (choosing a region, etc)
- setting up SSH
- installing software
- rebooting after updates
- backing up work on GitHub (private) or with scp
- resolving issues directly with Google (e.g., false positive crypto mining)

Docker containers

Containers are a lightweight alternative to virtual machines.

You'll run Docker containers this semester to have your own "mini cluster"



Resources of the "cluster" are limited to those of a single VM, so we'll scale projects accordingly. But the techniques will apply to large clusters and datasets.

Conclusion

Systems manage resources like compute, memory, storage, and networking.

Big data systems use specialized or distributed resources to make it faster to work with large datasets.

We'll deploy these systems using containers and VMs.

Tasks for next time:

- read syllabus, become familiar with course websites
- introduce yourself via the welcome form
- redeem Google credit and create first VM