Smallest font

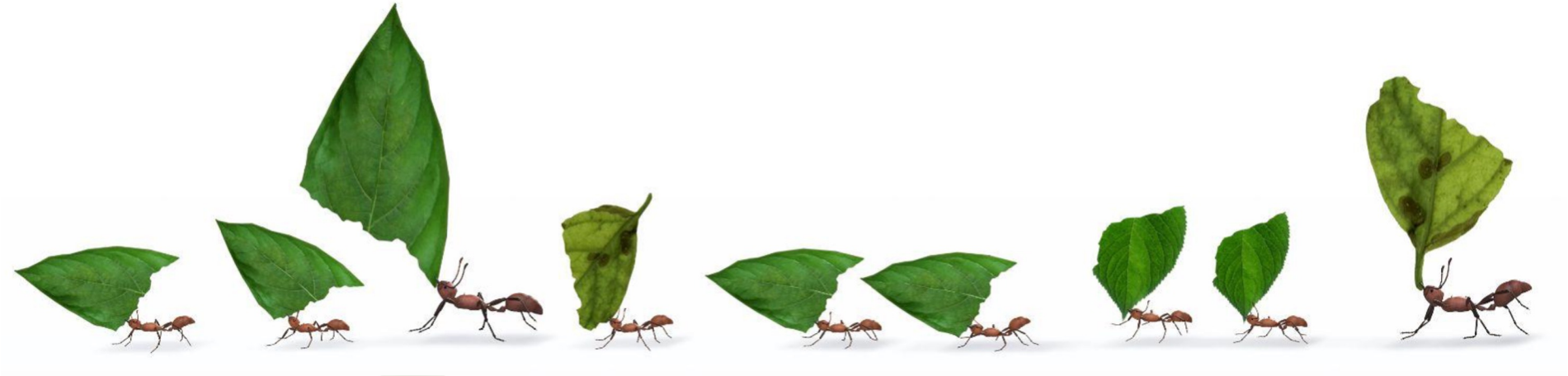
Please turn off and put away your cell phone

Calibration slide

These slides are meant to help with note-taking
They are no substitute for lecture attendance

Smallest font

Big Data



For context:
Colocasia gigantea
(Giant elephant ear)

NYU | Center for Data Science

Week 01: Welcome to Big Data

DS-GA 1004: Big Data

Instructor: Pascal Wallisch, PhD

Lesson plan

- What exactly *is* Big Data?
- Class logistics
- Philosophy and CS context

What is Big Data?

When I took this class, it seemed to be about cutting-edge tools and frameworks

In today's hyper-scaling data ecosystem, we seamlessly integrate **SQL** queries through **Postgres** alongside unstructured storage in **MongoDB**, while **Hadoop**'s distributed **HDFS** layer—augmented by **Parquet** for efficient columnar storage—and **MapReduce** functions coordinate massive parallel workloads. Simultaneously, **Docker**-based microservices orchestrated by **Kubernetes** simplify container deployment, and **Dask** manages in-memory computations. Meanwhile, **Kafka** handles real-time data pipelines for event-driven processing. Leveraging **Apache Pig** for higher-level scripting and **Spark** for iterative analytics, we adopt a **Dremel**-inspired architecture that unifies multi-source queries and aggregates. Critical data checkpoints rely on **hash functions** to ensure integrity and reliability throughout the pipeline. The entire environment is versioned and managed on **GitHub**, deployed on **AWS** for limitless scalability, and accelerated by **GPU** resources to ensure near-instantaneous insights across all touchpoints.

What this class is actually about: **Scaling**

In many fields, scale matters (critically):

You can drop a mouse down a thousand-yard mine shaft and, on arriving at the bottom, it gets a slight shock and walks away.	~1 oz (28.35 g)
A rat is killed.	~1 lb (16 oz)
A man is broken.	~180 lbs (2,880 oz)
A horse splashes.	~1000 lbs (16,000 oz)

--J.B.S. Haldane, biologist

In brief:
Quantity has a quality of its own

What about scaling of **data**?

Generally speaking, more data, i.e. a larger sample size (more rows) and a richer set of features (more columns) makes everything better in data science:

Specifically:

-Higher power / better recall

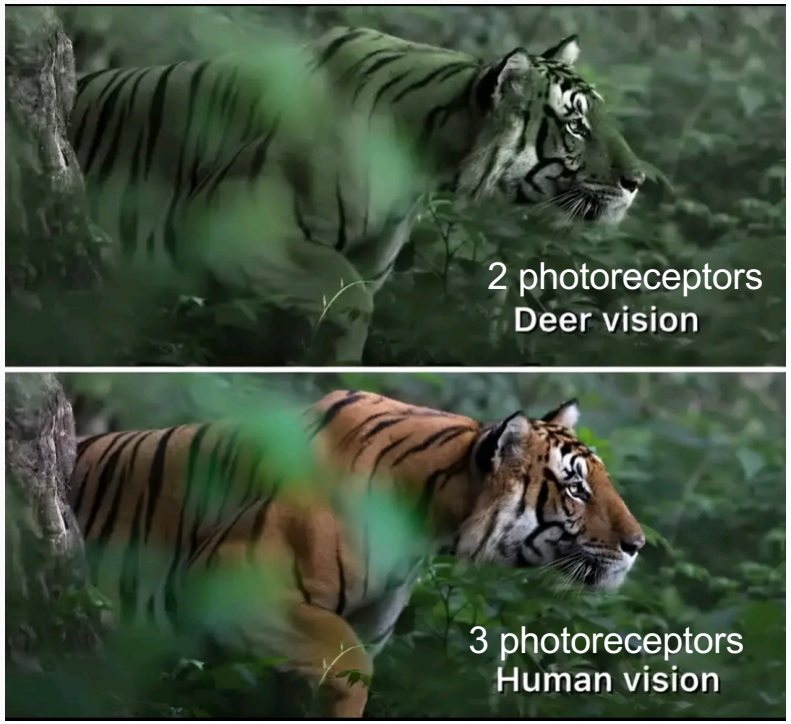
-Better parameter estimates

-Better recommendations

-Enables subgroup analysis

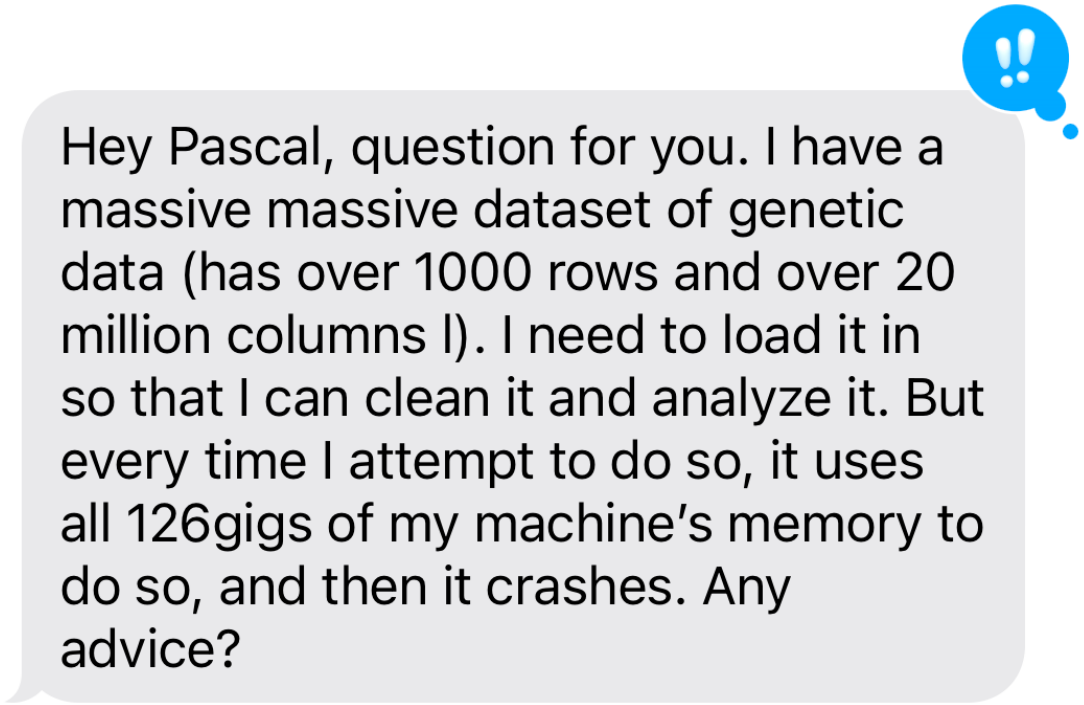
-The blessing of dimensionality

...



Can one have too much of a good thing?

What this looks like in the context of data:



Many such cases…

```
Error using zeros
Requested 702x120x1000000 (627.6GB) array exceeds maximum array
size preference (64.0GB). This might cause MATLAB to become
unresponsive.
```

A showstopper, if unhandled

Now what?

Operational definition

big data, *n.*:

Whatever doesn’t fit on your laptop…

More seriously...

- The definition of “big” depends on how the data is used and stored
- In practical terms, “**big data**” is often differentiated by requiring ***coordinated* processing by *multiple* computers (“machines”).**
- Much of this class will focus on **distributed storage and computation.**

Note: We will be using the CS concept of data in this class, not the DS conception.

- DS concept of data: The “givens” (latin). Immutable. Sacred. What we use to gain insights from, do statistics with and use to train machine learning models.

- CS concept of data: Bits and bytes. Digital information that has to be loaded and stored (to/in memory), fed to a processor, changed (by processing it), and sent around.

Big data is about more than just size: The five “V”s of big data

Volume	The quantity of data
Velocity	Speed at which new data is recorded
Variety	Data may be structured or heterogeneous
Veracity	Data can be noisy, incomplete, or wrong
Value	The benefits and actionable insights within

Laney, 2001;
Hurwitz et al., 2013

Why is there a whole class on big data?

Yann's keynote:

Y. LeCun

Machine Learning sucks! (compared to humans and animals)

- **Supervised learning (SL) requires large numbers of labeled samples.**
- **Reinforcement learning (RL) requires insane amounts of trials.**
- **Self-Supervised Learning (SSL) works great but...**
  - Generative prediction only works for text and other discrete modalities

So in essence:

- We need data to make machine learning and statistics work well.
- The more the better.
- But: At some point, handling all of this data becomes a challenge in itself
- More data = new problems…
- We’ll start discussing later today, which specific problems are introduced by an overwhelming quantity of data

What did we have in mind when designing this class?

- The tools are constantly evolving.

  We'll cover about 40 years worth of technology, skewing toward 2010+

- Current software will probably be obsolete in a few years

- The underlying PRINCIPLES don't change quite so rapidly!

  ⇒ Get proficient with concepts and current tools, and **learn to adapt**!

This class in the curricular context

- Introduction to Data Science (DS-GA 1001): Introducing the big concepts, but implemented with relatively small datasets.
- Big Data (DS-GA 1004): Revisiting these concepts, but scaled up with large datasets, and with a focus on implementation to address the specific problems that come with scale.
- So 1001 → 1004 is basically a sequence.

What should you get out of this class?

- Familiarity with distributed storage and computation
- Appreciation for the technical challenges of big data
- **Understanding of when to use which methods and tools**

Git

SQL

Hadoop

MapReduce

HDFS

Spark

Dremel

Parquet

Dask

CUDA…

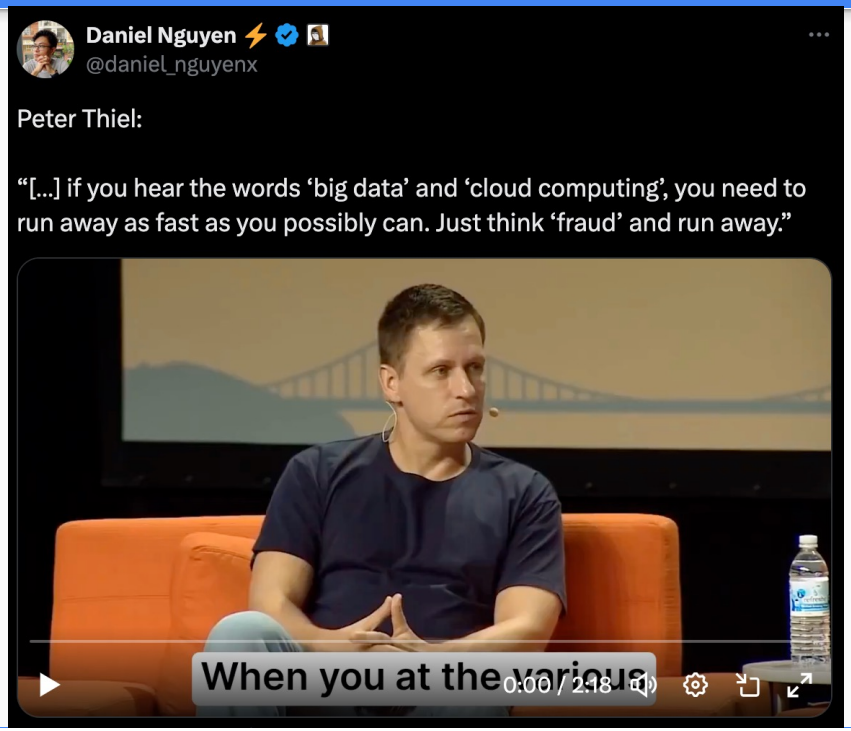
Again:
This subject matter involves a lot of obtuse terminology and buzzwords.
Don't worry.

If terms are ever unclear, stop and ask for clarification.

Relatedly: some of you undoubtedly have more experience than others.

Be mindful of others and the environment we create in the classroom!

We'll demystify a lot of this jargon, but beware:
A lot of shenanigans can be hidden with jargon, souring some on the integrity of the whole field:



HOW does this class work?

The teaching staff

Instructor

Pascal Wallisch, PhD

Section leaders

Tanvi Bansal

Iris Lu

Tom Zhou

Dennis Hu

Tutor

Jason Moon

Course coordinator

Adeet Patel

Graders

numerous & anonymous

We will use “Brightspace” as the learning management system (LMS) for this class

It will feature

- Announcements
- Lecture slides
- Datasets
- Code
- Assignments
- Assorted class materials (readings, videos, etc.)

You can access it at

https://brightspace.nyu.edu/

The *sittyba*

Some highlights worth emphasizing…

Technology and resources

- All resources are available through **brightspace.nyu.edu**
  - Course schedule, assigned reading, etc.
  - Slides will be posted (usually) immediately before class
- HW assignments will be available via GitHub Classroom
  - If that's a problem, we can make other arrangements
  - Lab section will cover how exactly this works (starting *this* week)

Readings

- Each week will have assigned reading, listed in the sittyba
  - Expect a book chapter, or 1-2 papers each week
- Posted under “content” on **brightspace.nyu.edu**
- You’re expected to do the reading **before class meets**
  - Learning is most effective when you first encounter new ideas on your own.
  - We can use the class time to clarify difficult or confusing concepts.

Grading

- 25% Homework assignments
- 25% Capstone project
- 25% Final
- 9% Quizzes
- 16% Low stakes assignments

Homework assignments (25%)

- 5 ~bi-weekly programming assignments to be completed **in small groups**
  - Lab sections = get help with this assignment from the section leader
- You'll get access to NYU's high-performance computing (HPC) cluster
- You have **1 grace day** to use for each of these homework assignments
  - After that, 1% penalty per hour for late submissions.
  - No assignments will be accepted more than 5 days late.
  - Grading these assignments is not easy, please be mindful of the graders' time!
  - Do not plan on using the grace day – this is intended for emergencies (e.g. cluster crash)

Capstone project (25%)

- This will be an extended homework / programming assignment over 3-4 weeks, integrating several of the tools and methods that we'll cover.
- Due at the end of the semester
- Same rules re late assignments (we are being overly lenient)
- Details will be posted in April

Final Interview & Skills Test (25%)

- Cumulative, in-class, 64 T/F questions only
- Modest A&I penalty (so EV is still positive, and corresponds to 4 option MC exam)
- Beyond this class, you'll be expected to be conversant in these topics
  - (Think: job interviews!)
- All straight from the lecture. Quizzes will give you a good idea of what to expect
- Date / room and details will be posted on Albert by mid-semester (once the registrar tells us), but it will be during finals week. Make travel plans accordingly.

Quizzes (9%)

- In-person quizzes, during the labs. We count the highest 9 for credit.
- Quizzes are closed book and must be **completed independently**.
- Quiz will be paced during the lab. Don’t be latest
- Your lowest quiz scores are automatically dropped.
  - There will not be “make-up” quizzes, but this avoids penalties if you miss one (or 2 or 3).

Glossary (from my machine learning class, lecture 1)

- Vector
- Scalar
- Column vector
- Row vector
- Transpose
- Vector addition
- Scalar multiplication
- Vector space
- Neutral element
- Zero vector
- Vector subspace
- Linear combination
- Span
- Linear independence
- Basis vectors
- Basis
- Standard basis
- Matrix
- Matrix Dimensionality
- Square matrix
- Symmetric matrix
- Diagonal matrix
- Identity matrix
- Linear transformations
- Linear mappings
- Injective mapping
- One-to-one mapping
- Surjective mapping
- Onto mapping
- Bijective mapping
- Invertible mapping
- Isomorphic mapping
- Automorphic mapping
- Identity mapping
- Kernel
- Null space
- Image
- Range
- Column space
- Domain
- Codomain
- Dot product
- Scalar product
- Inner product
- Hadamard product
- Magnitude
- Vector length
- Euclidian norm
- Unit vector
- Norms
- L_1 norm
- L_2 norm
- Collinear vectors
- Orthogonal vectors
- Projection
- Hyperplane
- Matrix multiplication
- Orthogonal matrices
- Determinant
- Singular matrices
- Tensor
- Cosine similarity
- Affine subspace
- Matrix inverse
- Linear coordinates
- Pseudoinverse

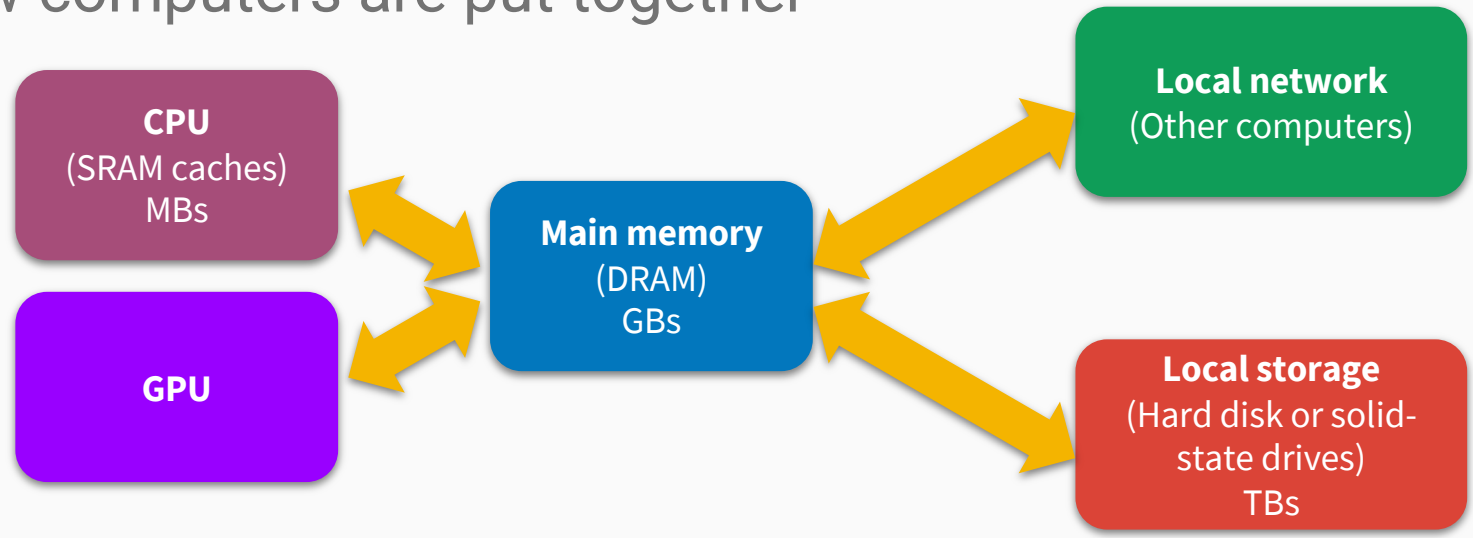
Academic integrity

- See sittyba for the detailed statement.
- You are responsible for all submitted materials.
- Yes, bots exist, but you still need to understand the material, so will not be allowed to use them during the final.
- **We take academic integrity seriously in this class.**
- Number of people reported for academic integrity violations in this class:
- 2021: 28
- 2022: 14
- 2023: 32 (2 of them twice, some expelled)
- 2024: 0

Big Data in the broader CS context

Some basic computer organization

- To know what counts as “too big”, we’ll need to understand how computers are put together



So: What are our resources?

- **Storage**
  - Where and how is data kept?
  - How much data can we keep?
- **Communication**
  - How quickly can we move data between locations?
    (Eg: over network, disk→RAM, RAM→CPU, RAM→GPU)
- **Computation**
  - How quickly can we process data? (Convert *inputs* into *outputs* via *instructions*)

1) Storage: The cost of storage over time...

$3398 10MB

THE HARD DISK YOU'VE BEEN WAITING FOR

XCOMP introduces a complete micro-size disk subsystem with more...

- MORE STORAGE
- MORE SPEED
- MORE VALUE
- MORE SUPPORT

S100 users . . . The XCOMP subsystem is now available with 10 megabytes of storage; 5 megabytes also available at $2,898.00. Compare the price and features of any other 5¼-inch — or even 8-inch system, and you'll agree that XCOMP's value is unbeatable.

OUTPERFORMS OTHER HARD DISKS

Floppy disk and larger, more expensive hard disks are no match for this powerful little system. More data is available on every seek: 64K on 10MB and 32K on 5MB. Faster seek time too — an average of 70MS. It provides solid performance anywhere with only 20 watts of power. Data is protected in the sealed enclosure, and the landing zone for heads provides another margin of safety. The optional power board plugs directly into the S100 bus and provides power for the drive.

FAST CONTROLLER

The XCOMP controller is the key to this system's high efficiency operation. Speed-up features include interleave without table lookup, block-deblock with controller buffer, and read lookahead. OEMs world-wide have already proven the outstanding performance of the XCOMP controller.

MORE SOFTWARE

Included with the system is software for testing, formatting, I/O drivers for CP/M®, plus an automatic CP/M driver attach program. Support software and drivers for MP/M® and Oasis® are also available. The sophisticated formatting program assigns alternate sectors for any weak sectors detected during formatting, assuring the lowest possible error rate — at least ten times better than floppies.

WARRANTY

The system has a full one-year warranty on parts and workmanship.

ALSO AVAILABLE FROM XCOMP

- General Purpose controllers (8 bit interface), with easy interface to microprocessor-based systems.
- GP controller adapter that plugs directly into most Z80 computers.
- ST/R GP controller for the 5MB and 10MB drive above, with ST506 type interface.
- SG/R GP controller for SA1000 interface.
- SM/R GP controller for storage module drives.
- ST/S, SG/S, and SM/S, same as above, for the S100 bus.

Quantity discounts available. Distributor, Dealer, and OEM inquiries invited.

See your local Dealer, or call:

XCOMP

XCOMP, Inc.
7566 Trade Street
San Diego, CA 92121
Tel: (714) 271-8730
Telex: 182766

Circle 405 on inquiry card.

$3398 = $11,292.77 in 01/2024

Byte Magazine, Vol. 6, No. 8 (1981/08)

1) Storage: The cost of storage over time...

$3398 = $11,292.77 in 01/2024

x 2 million

x27

= 540TB

WD Ultrastar HC560 WUH SE SATA HDD 0F38785

★★★★½ 12

$409.15

FREE delivery **Jan 25 - 29**

Add to Cart

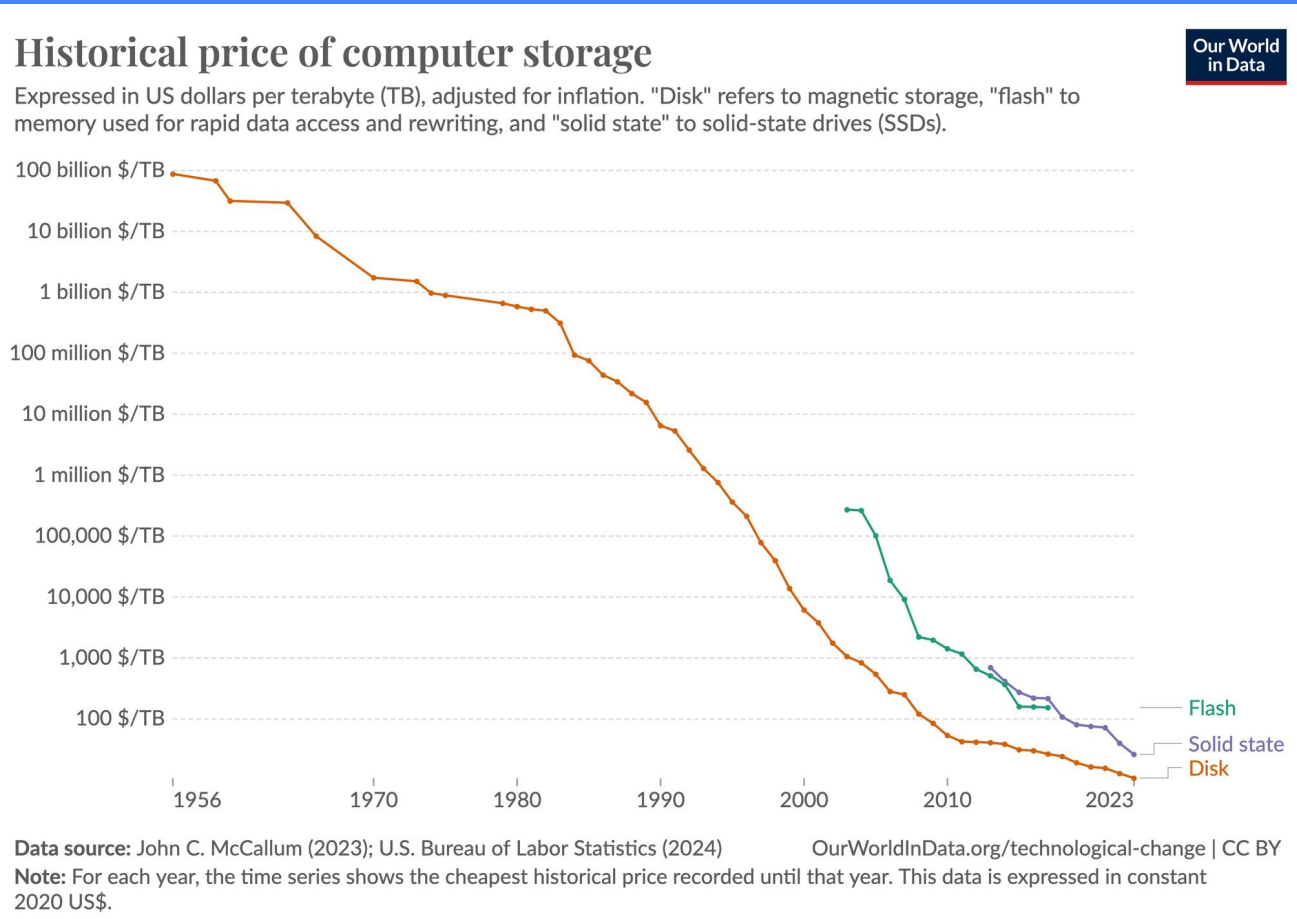
More Buying Choices

$359.99 (25 used & new offers)

Byte Magazine, Vol. 6, No. 8 (1981/08)

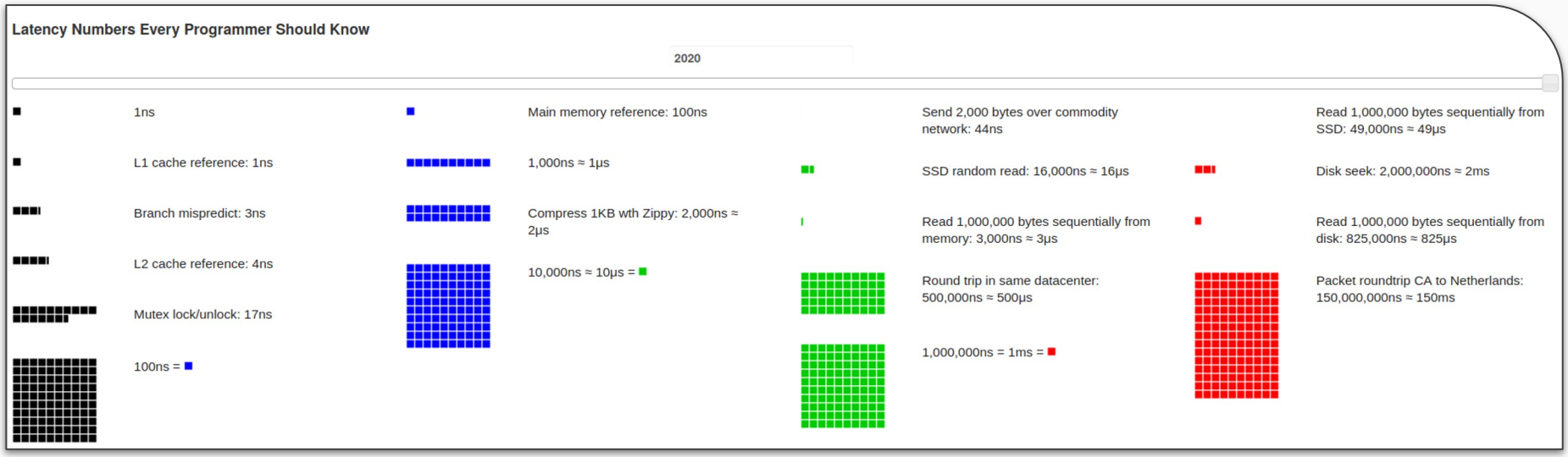
westerndigital.com (2024/01)

This just keeps going! (The price of storage is usually NOT the bottleneck)



2) Data volume and velocity

- (2023) YouTube adds 500 hours of HD video every minute
  http://www.businessofapps.com/data/youtube-statistics/
- (2019) Facebook generates 4 Petabytes (4e15 bytes) of new data per day
  https://research.fb.com/facebook-s-top-open-data-problems/
- (2017) Twitter stores >500PB of data
  https://blog.twitter.com/engineering/en_us/topics/infrastructure/2017/the-infrastructure-behind-twitter-scale.html
- (2017) CERN data center: > 200PB
  https://home.cern/news/news/computing/cern-data-centre-passes-200-petabyte-milestone

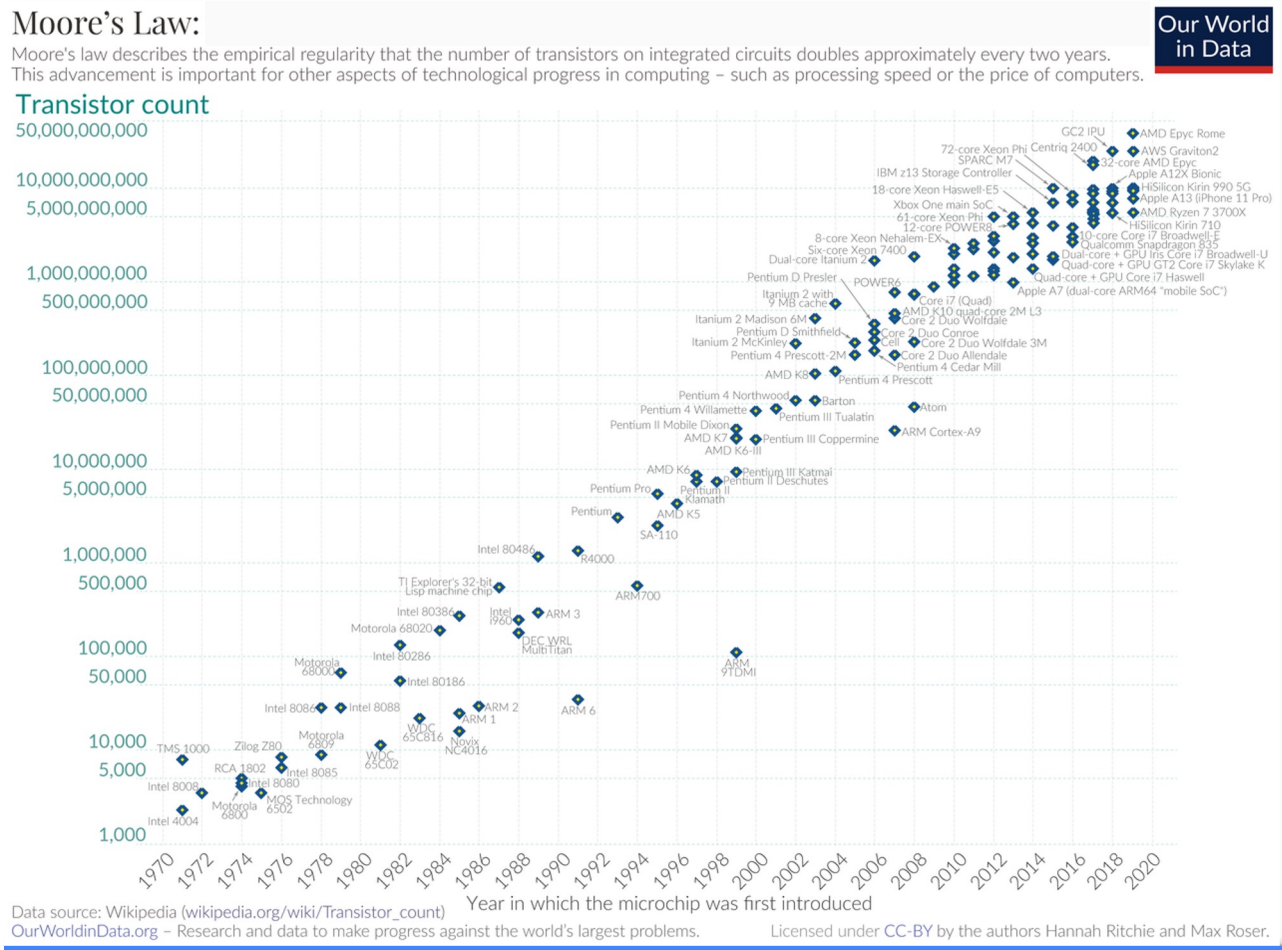


L1 cache read	1 ns
L2 cache read	4 ns
Main memory read	100 ns
SSD random read	16,000 ns
HDD random read	2,000,000 ns

- Pre-fetching and pipelining can **accelerate sequential reads**
- **Main memory** is typically the first bottleneck
- The less communication (read/write) we have to do, the better

https://people.eecs.berkeley.edu/~rcs/research/interactive_latency.html

3) Communication



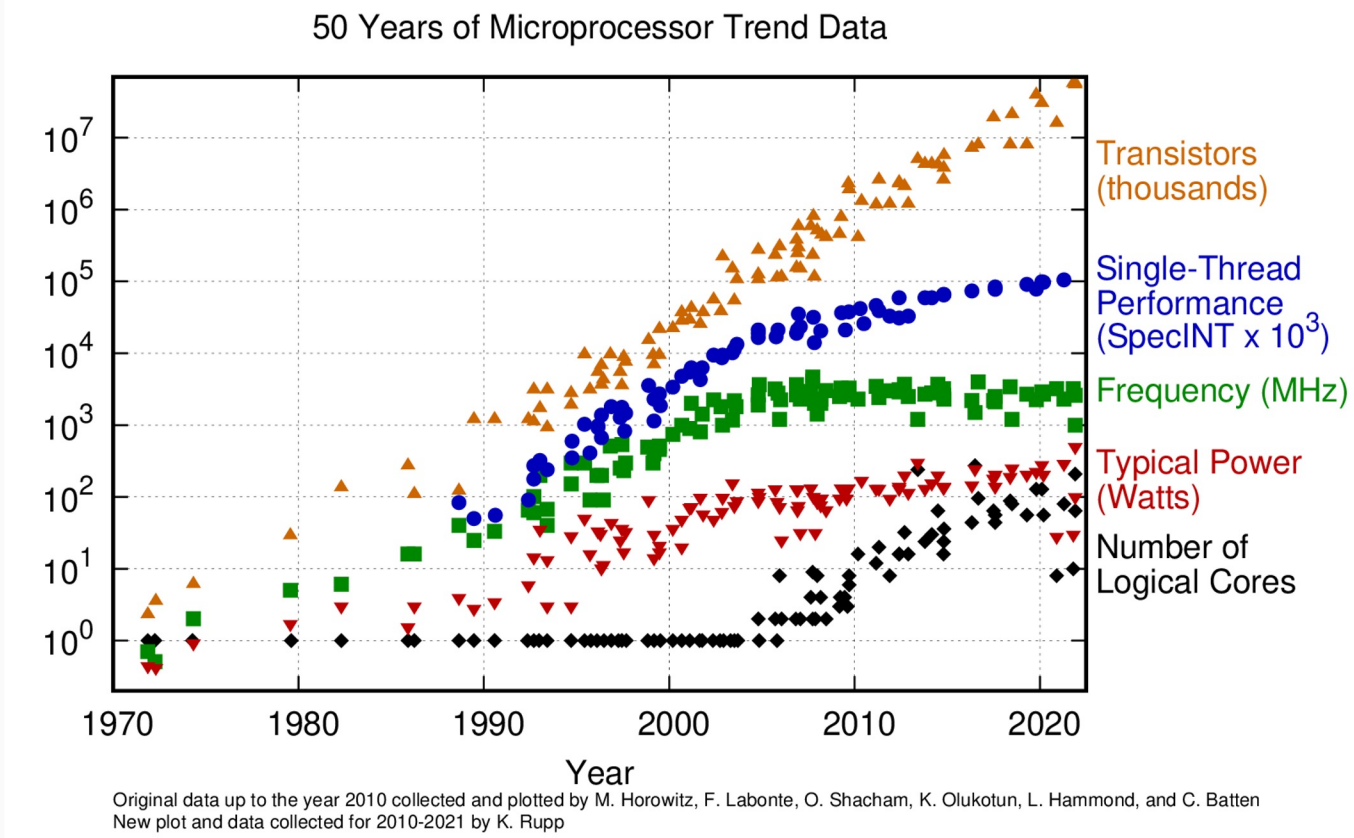
4) Computation

Note: transistor count ≠ speed!

Moore in 2015:

"I see Moore's law dying here in the next decade or so."

Gordon Moore (1965, 1975)



Gains now come from **parallelism** (multi-core), not **clock speed!**

https://github.com/karlrupp/microprocessor-trend-data

To summarize - what is the upshot?

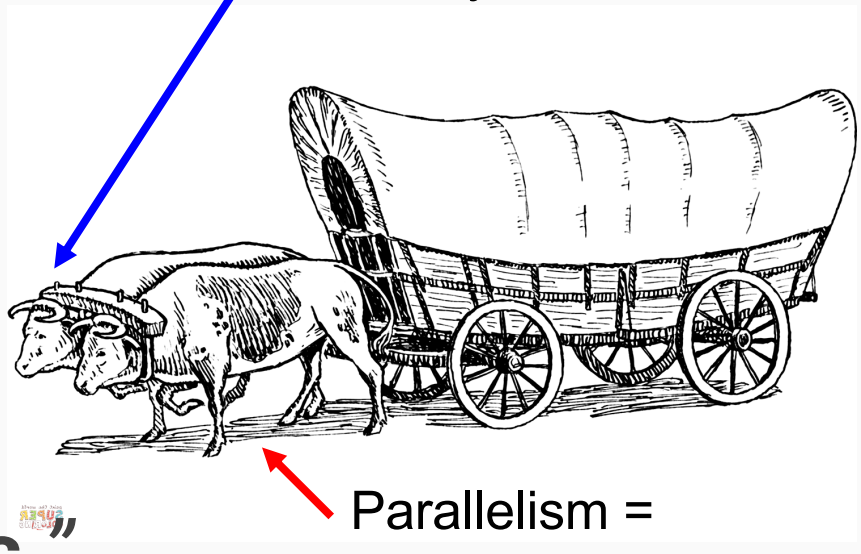
- Storage capacity (per $) continues to increase
- However, data velocity is also ever increasing
- But CPU speed is *not* keeping up
- Communication is even worse (slower), avoid it

What to do?

What is the solution?

"In pioneer days, they used oxen for heavy pulling, and when one ox couldn't budge a log, they didn't try to grow a bigger ox.

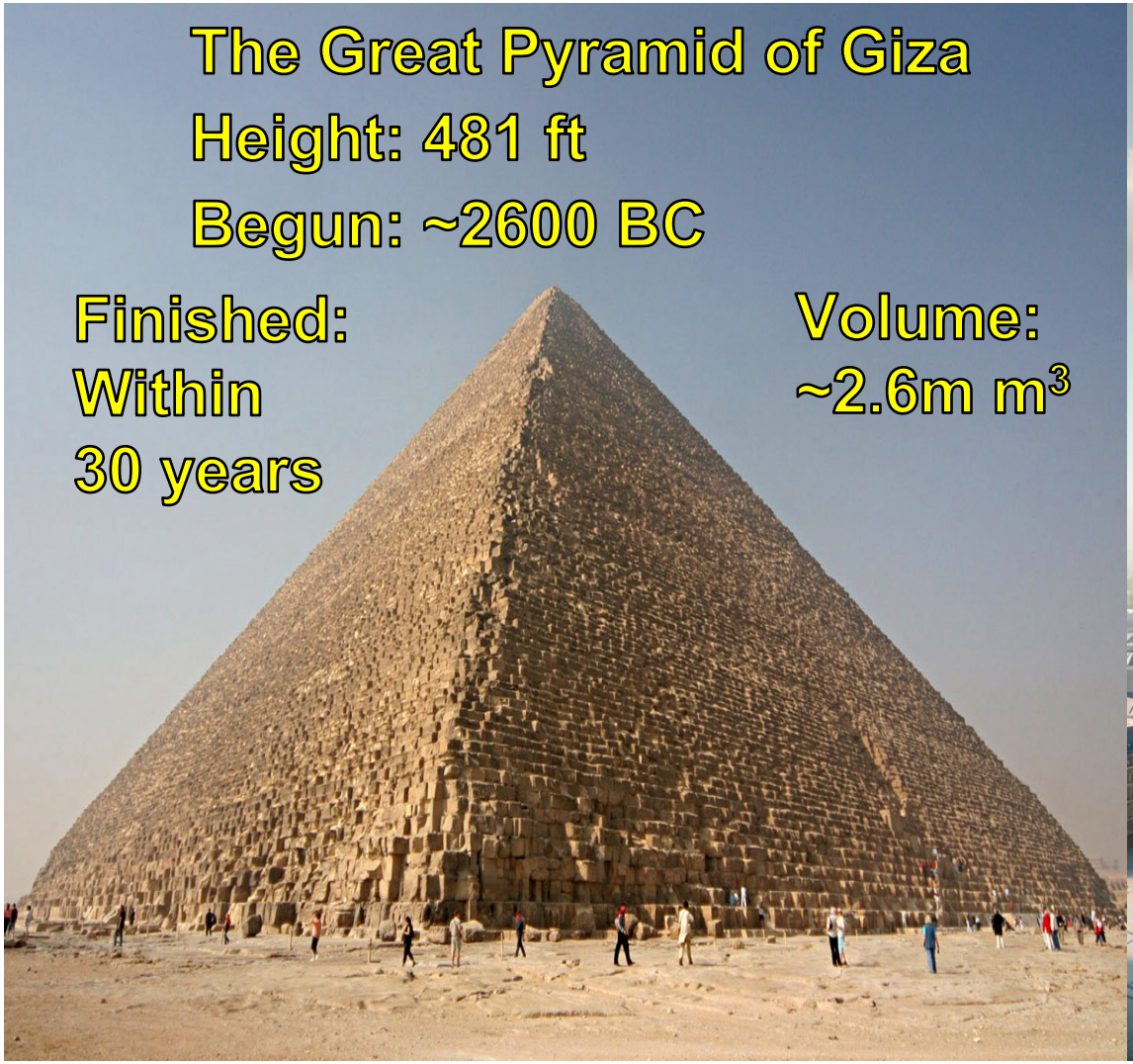
We shouldn't be trying for bigger computers, but for more ***systems of computers****."*



Rear Admiral Grace Hopper (1906-1992)

Parallelism / load distribution can also help with speed

WHY ”communication” is the common enemy of all mankind (in the context of Big Data)


The Great Pyramid of Giza
Height: 481 ft
Begun: ~2600 BC
Finished:
Within
30 years
Volume:
~2.6m m³


Cologne Cathedral
Height: 515 ft
Begun: 1248
V: ~0.5m m³
Finished:
1880

Principles of scaling: At the seashore

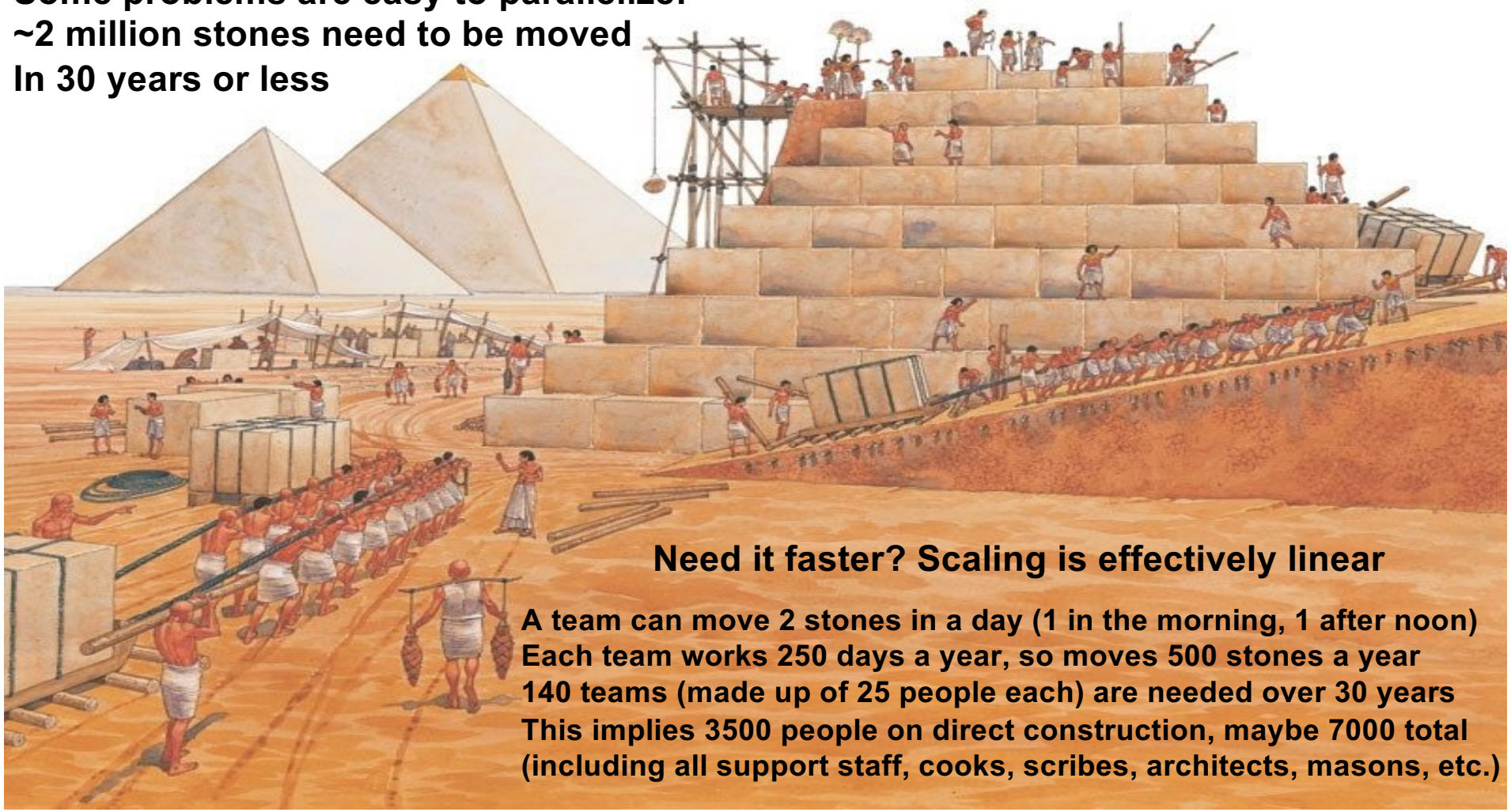
On average, Suzy collects 100 seashells per hour

Valid

?

Together, Suzy and Lucy collect 200 seashells per hour

Some problems are easy to parallelize:
~2 million stones need to be moved
In 30 years or less



Need it faster? Scaling is effectively linear

A team can move 2 stones in a day (1 in the morning, 1 after noon)
Each team works 250 days a year, so moves 500 stones a year
140 teams (made up of 25 people each) are needed over 30 years
This implies 3500 people on direct construction, maybe 7000 total
(including all support staff, cooks, scribes, architects, masons, etc.)

In other words, the Pyramid of Giza took about 2.5 million “man-months” of human effort to build (all people on all teams over all years of construction).

> A modern operating system typically consists of about 100 million lines of code

A professional software engineer can write about 10,000 production level (unit tested, etc.) lines of code a year (40/day)

Can a major corporation simply hire 10,000 software engineers and finish the project within a year?

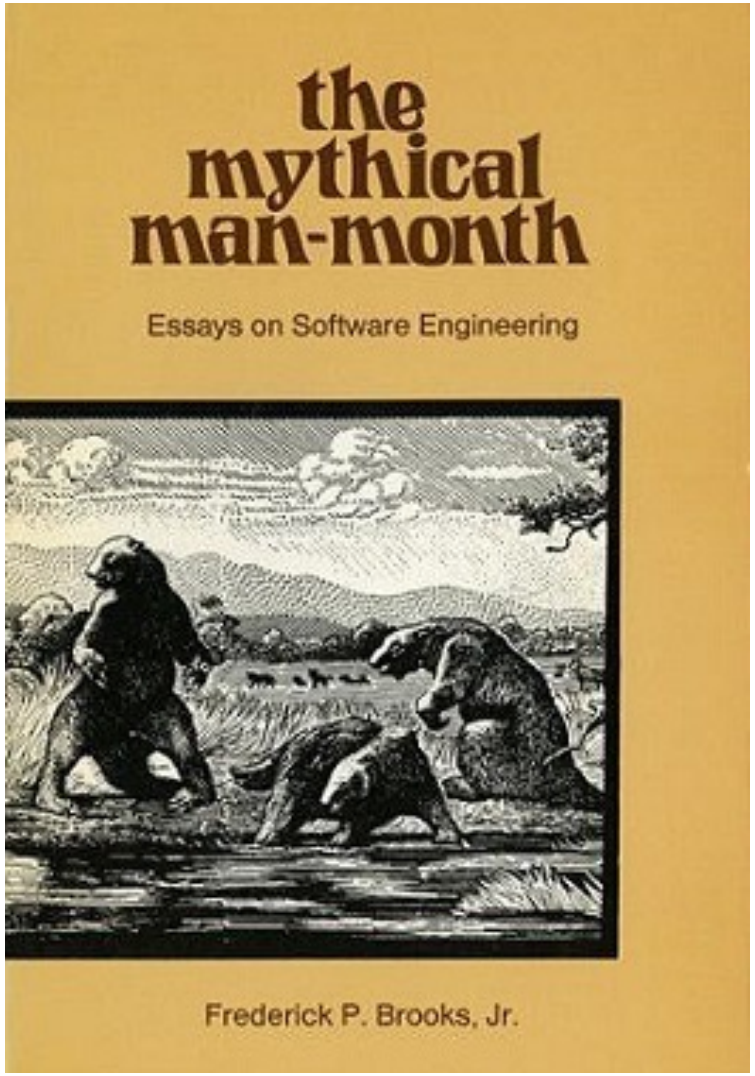
No!

But why not?

Assembling the blocks of code is not independent!

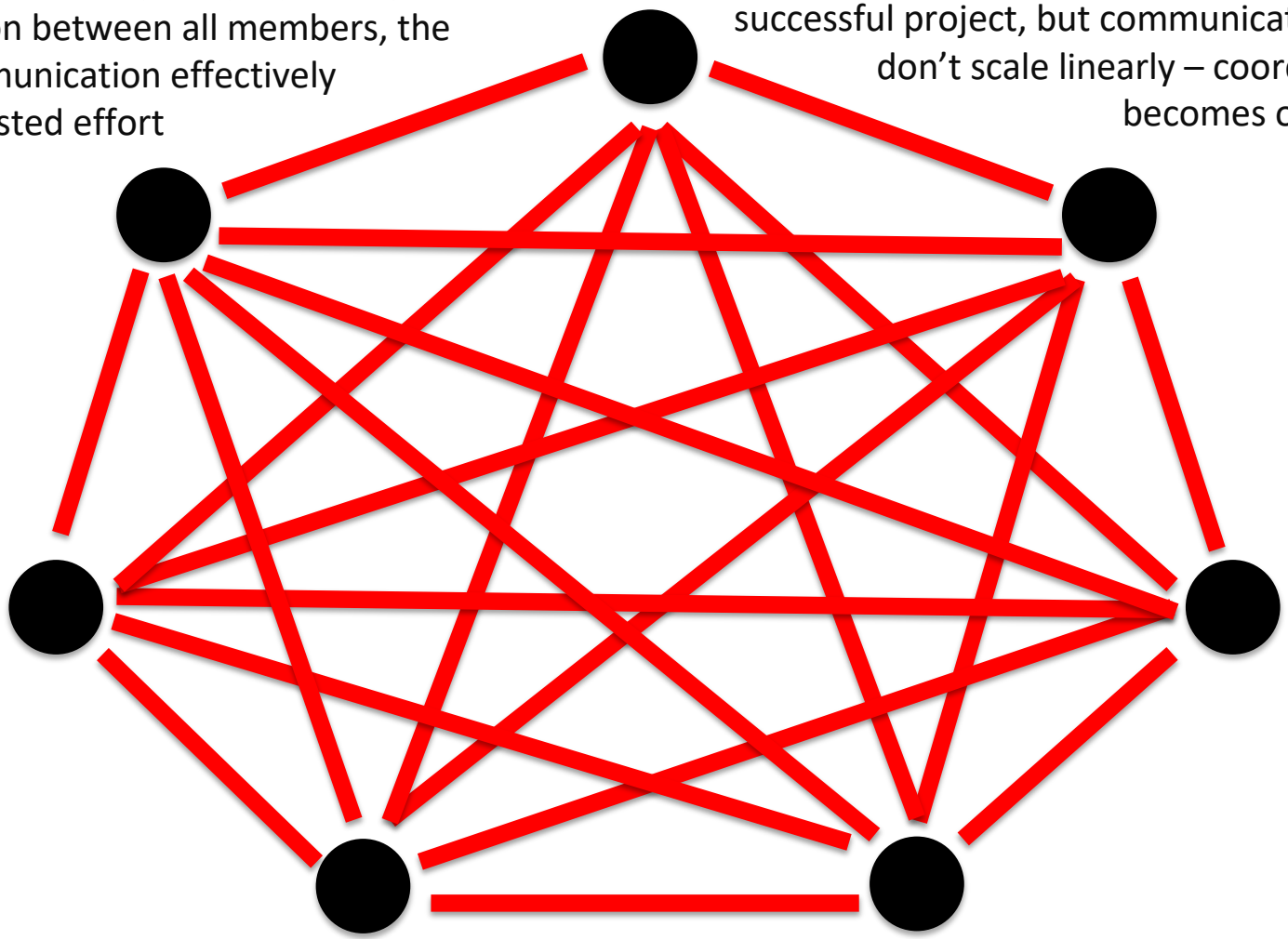
"Adding manpower to a late software project makes it later.”

But why?



If the success of a (complex) project depends on coordination between all members, the need for communication effectively represents wasted effort

Communication might be necessary for a successful project, but communication demands don't scale linearly – coordination soon becomes overwhelming



The upshot:
Parallelizing complex tasks like data analysis jobs is not trivial

- We will distribute processing
- We will distribute storage
- While minimizing communication and coordination, wherever possible
- We might have to impose processing hierarchies…

Next time…

Centralized systems:
File systems
The relational model
and databases

- Reading for next week

  [Garcia-Molina, Ullman, & Widom, 2009, ch2]