Smallest font

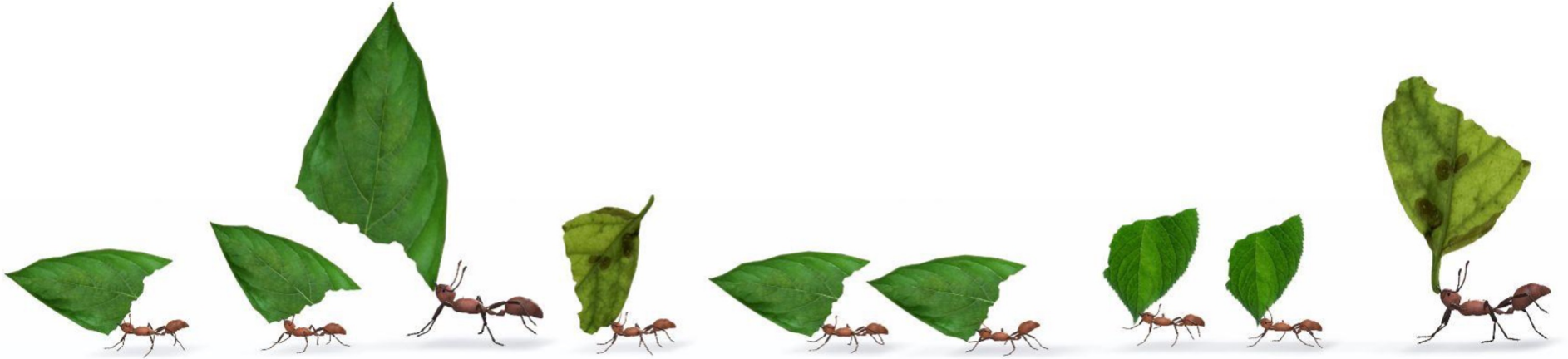
Please turn off and put away your cell phone

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

- Class logistics
- What exactly *is* Big Data?
- CS context
- Historical timeline

The teaching staff

Instructor

Pascal Wallisch, PhD

Section leaders

Aman Singhal
Sharad Dargan
Shreemayi Sonti
Andrew Deur

Tutor

Avinav Goel

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 *next* 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
- 12% Quizzes
- 13% 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
- Details will be posted in April

Final interview & skills test (25%)

- Cumulative, in-class, T/F questions only
- Beyond this class, you'll be expected to be conversant in these topics
  - (Think: job interviews!)
- 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 (12%)

- 5 online quizzes, ~biweekly on Fridays
- Quizzes are open book, open note, but must be **completed independently**.
- You will have 1 hour to complete a quiz once you start. Take your time!
- Your lowest quiz score is automatically dropped.
  - There will not be “make-up” quizzes, but this avoids penalties if you miss one.

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)
- 2024: ?

What is big data?

Operational definition

big data, *n.*:

Whatever doesn’t fit on your laptop…

Ever had this happen to you?

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

Now what?

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.**
- Much of this class will focus on **distributed storage and computation.**

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 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 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 concepts don't change 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...

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

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

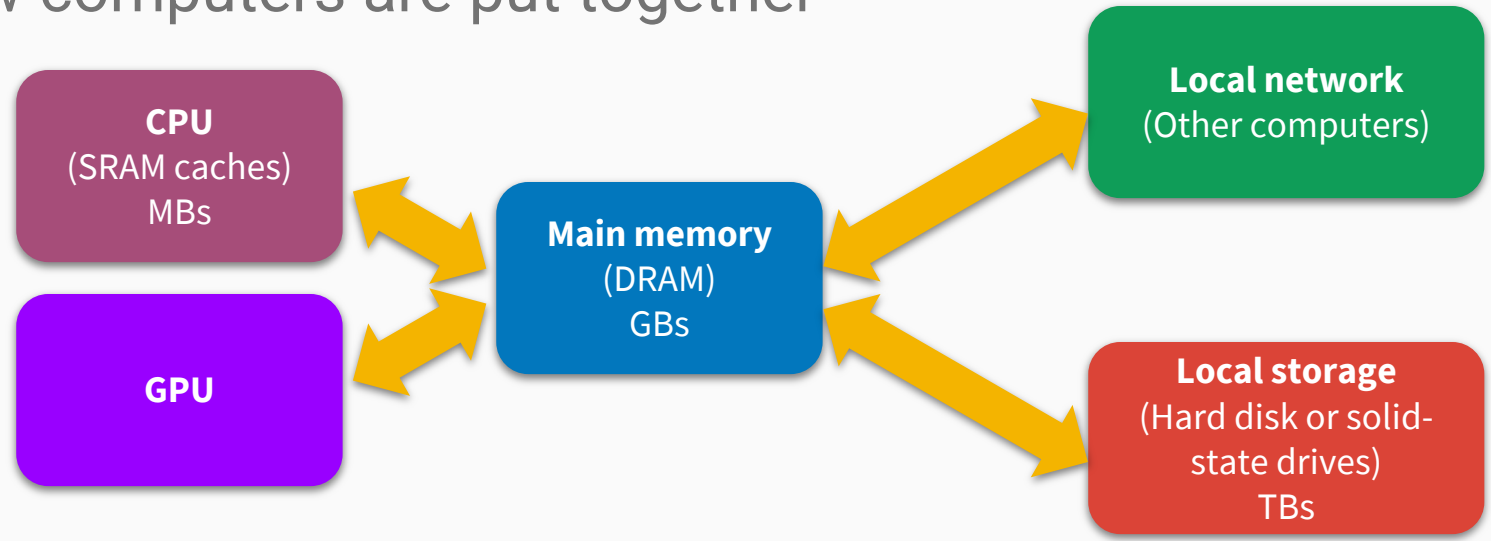
Relatedly: some of you undoubtedly have more experience than others.

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

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 = $11,292.77 in 01/2024

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

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.
- SQ/R GP controller for SA1000 interface.
- SM/R GP controller for storage module drives.
- ST/S, SQ/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.
7560 Trade Street
San Diego, CA 92121
Tel. (714) 271-8730
Telex: 182786

Circle 405 on inquiry card.

$3398 = $11,292.77 in 01/2024

x 2 million

WD Ultrastar HC560 WUH SE SATA HDD 0F38785

4.5 ★ 12

$409.15

FREE delivery **Jan 25 - 29**

Add to Cart

More Buying Choices
$359.99 (25 used & new offers)

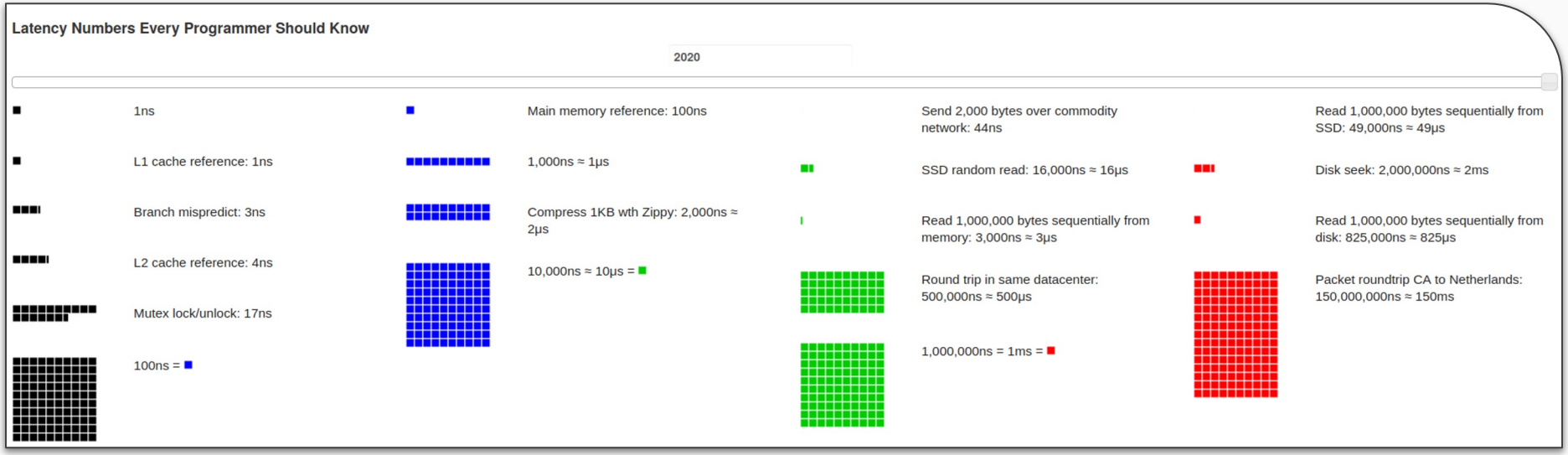
x27
= 540TB

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

westerndigital.com (2024/01)

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

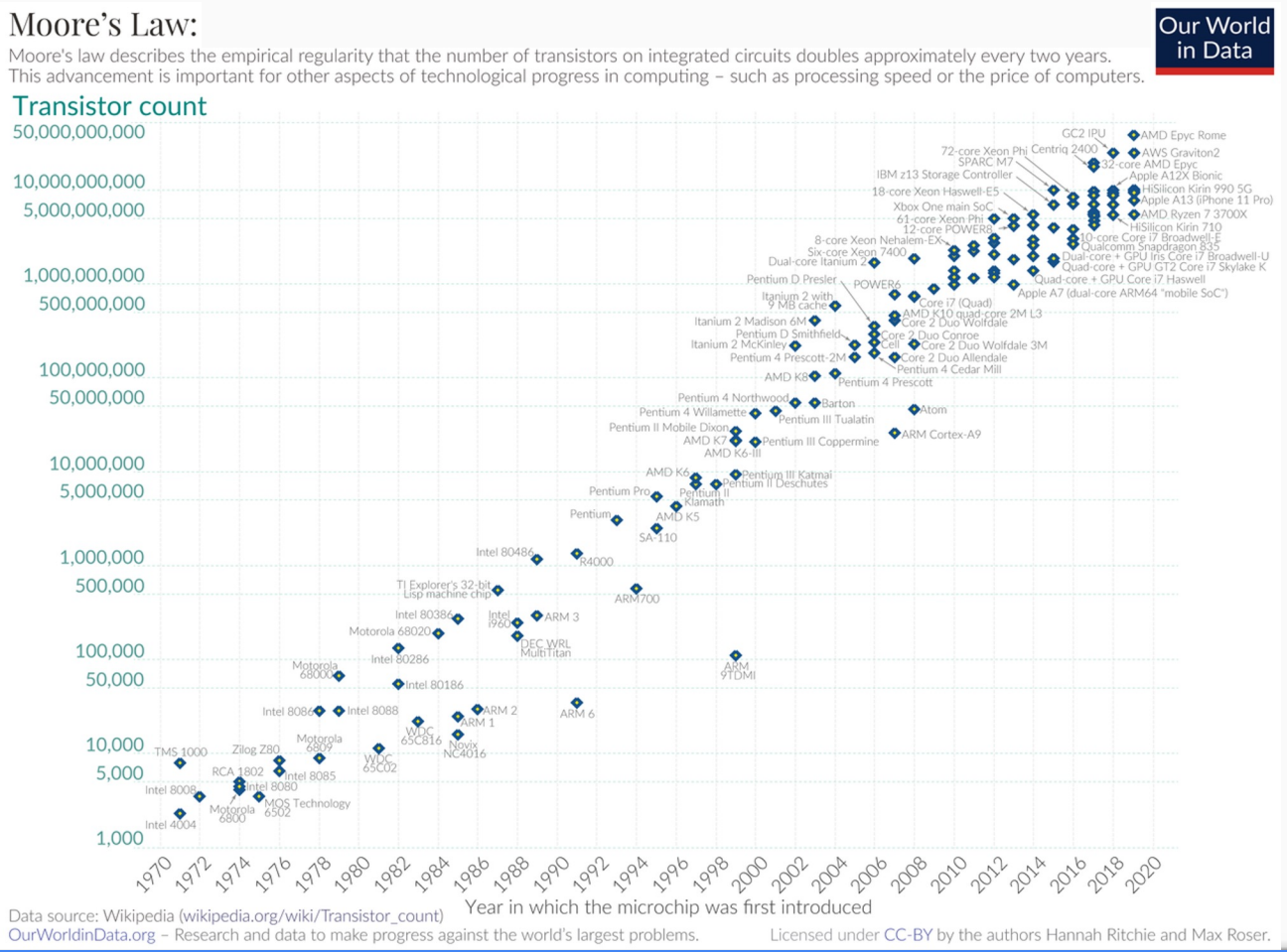


L1 cache read	1 ns
L2 cache read	4 ns
Main memory read	100 ns
SSD random read	16,000 ns
HDD random read	2,000,000 ns

- 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

3) Communication

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



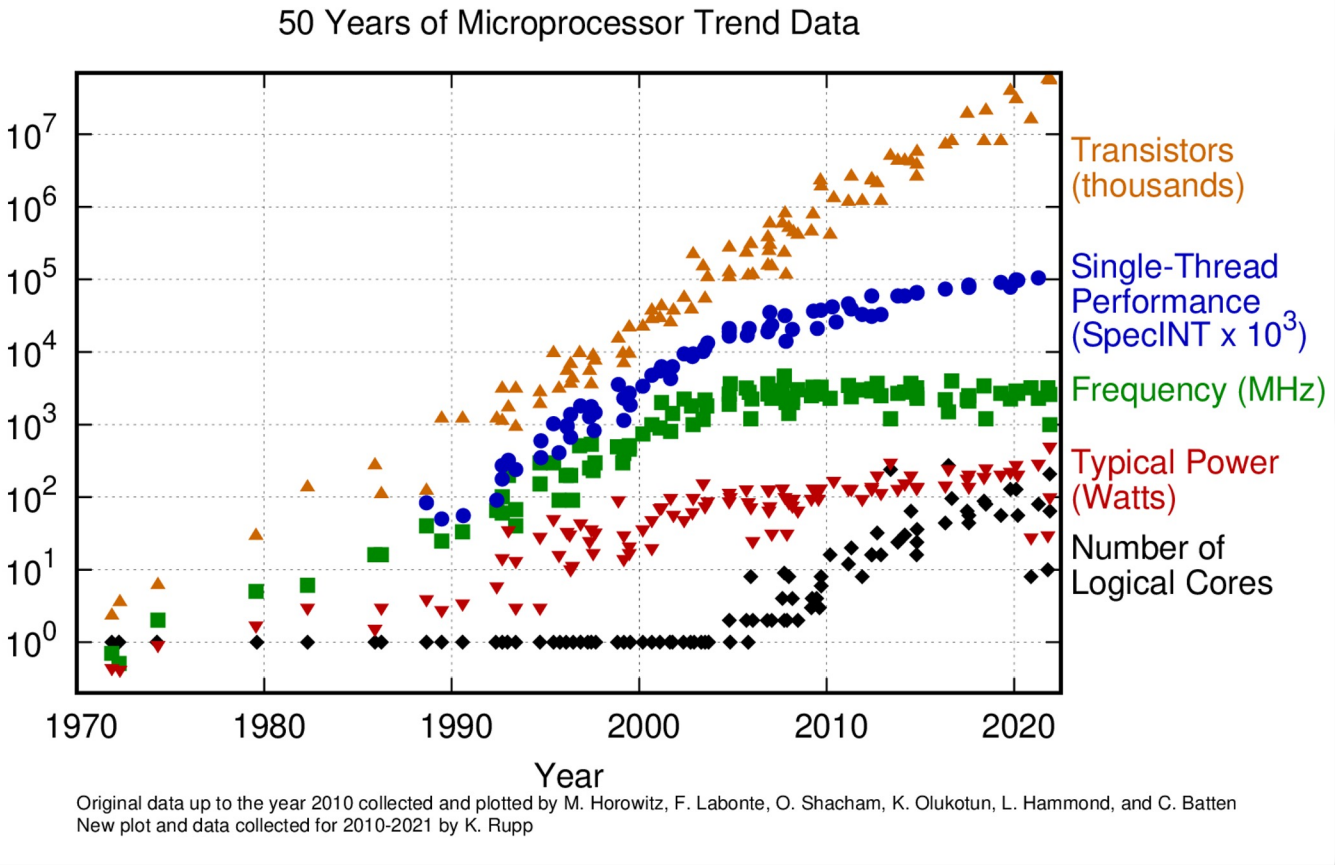
4) Computation

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?

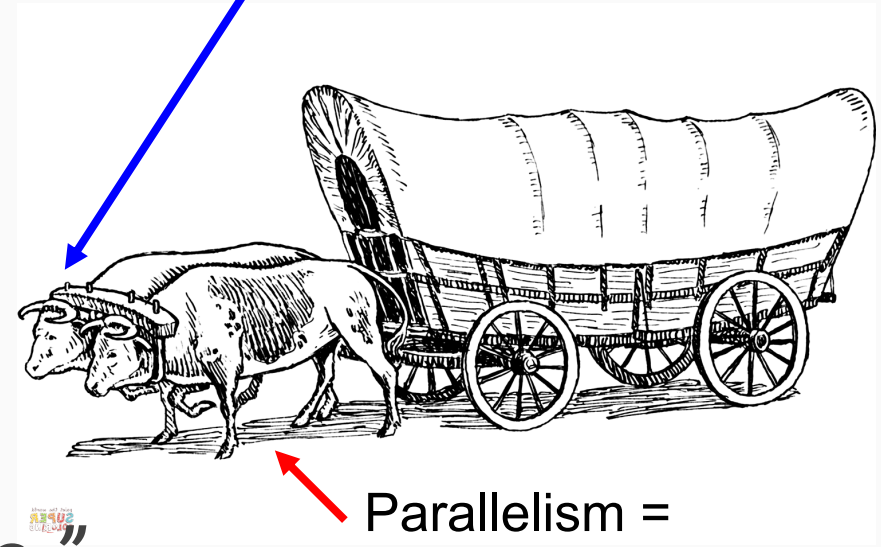
- Storage capacity (per $) continues to increase
- Data velocity is increasing
- But CPU speed is not keeping up

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**.”*



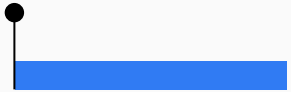
Rear Admiral Grace Hopper (1906-1992)

Parallelism / load distribution can also help with speed

A brief lineage of how we store, manage, access data

File systems

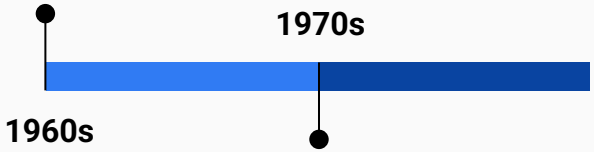
Custom software for
each application / query



1960s

File systems

Custom software for
each application / query



Relational model (Codd, 1970)

Tables are good!

Structured Query Language (SQL)

File systems

Custom software for each application / query

RDBMS takes off

Databases for commodity computers

SQL “standardizes”

Relational model (Codd, 1970)

Tables are good!

Structured Query Language (SQL)

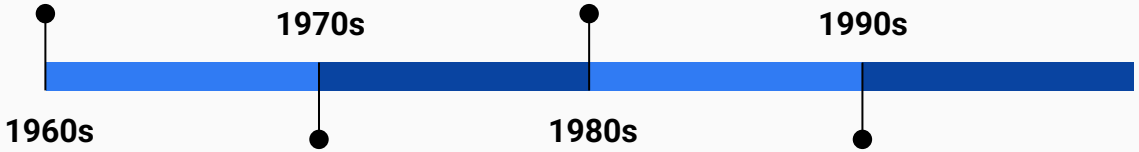
File systems

Custom software for each application / query

RDBMS takes off

Databases for commodity computers

SQL “standardizes”



Relational model (Codd, 1970)

Tables are good!

Structured Query Language (SQL)

The web happens

Distributed / parallel computing takes off

File systems

Custom software for each application / query

RDBMS takes off

Databases for commodity computers

SQL “standardizes”

MapReduce

Simpler abstractions for distributed computation and storage

Tables are bad!

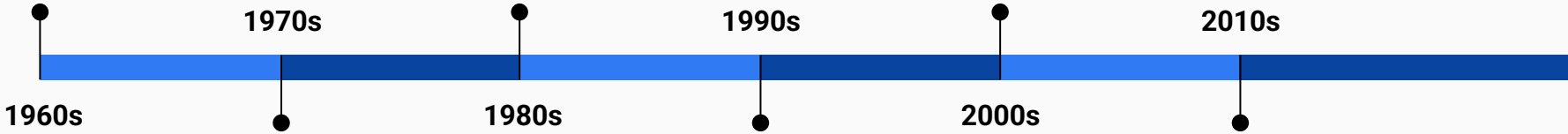
Relational model (Codd, 1970)

Tables are good!

Structured Query Language (SQL)

The web happens

Distributed / parallel computing takes off



File systems

Custom software for each application / query

Relational model (Codd, 1970)

Tables are good!

Structured Query Language (SQL)

RDBMS takes off

Databases for commodity computers

SQL “standardizes”

The web happens

Distributed / parallel computing takes off

MapReduce

Simpler abstractions for distributed computation and storage

Tables are bad!

THE CLOUD ERA

Flexible cluster management

High-level languages

Focus on interactivity

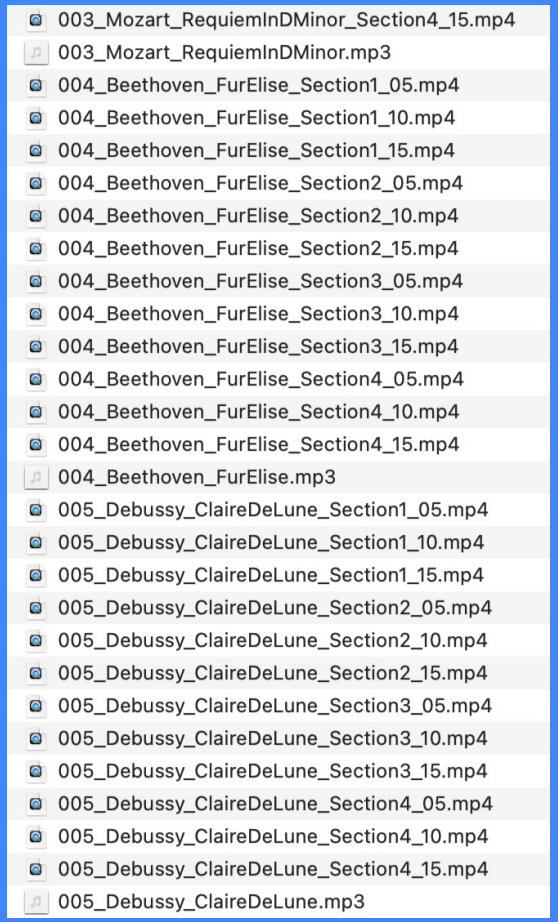
Tables are good if you call them ✨data frames✨

Going back to the beginning:

File systems!

- To understand modern tools, it helps to understand previous generations.
- What problems did people face in designing:
  - Relational databases?
  - Map-reduce?
  - Spark?
  - Dask?

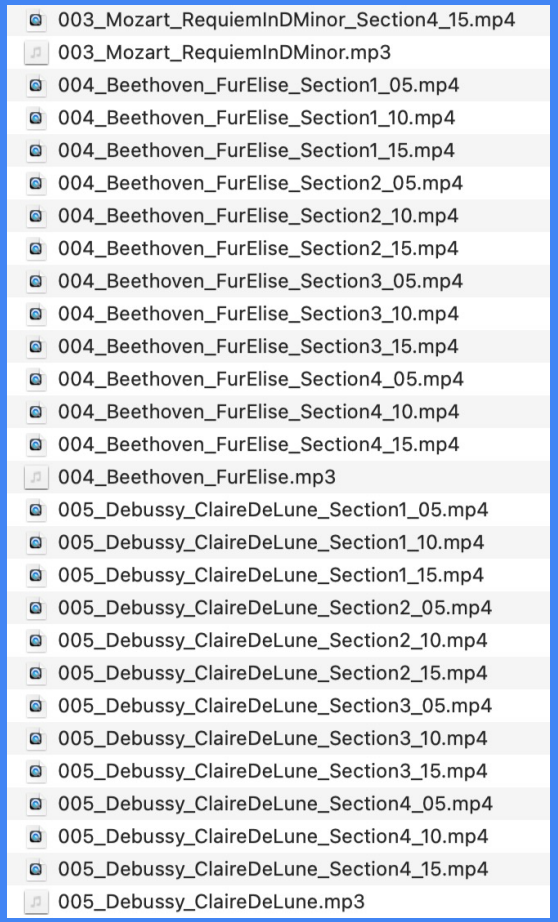
File systems



- Use directories to organize your data
- Structured data can be stored as files

  ⇒ data persists across application runs

File systems



- Use directories to organize your data
- Structured data can be stored as files

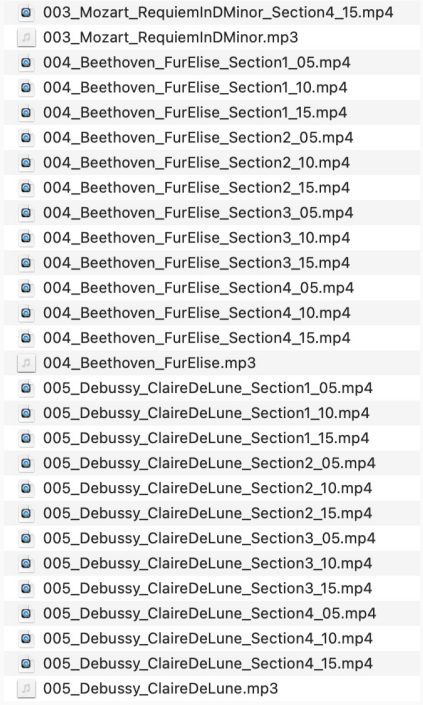
  ⇒ data persists across application runs
- Some great properties:
  - **Easy** to implement
  - **Data** does not vanish
  - **Inherently** organized (tree structure)
  - **Portable** across systems

File systems can be awesome!

So why are we not always using them (exclusively)?

Reasons not to rely (only) on file systems

- Does not expose or exploit the structure of data
  - What if I want to search by file contents?
    Better options than brute force?
- Each query / analysis required writing a new program
  - Little re-usability between similar analyses
  - Or the same analysis with slightly different data structures
- Directory hierarchies may be too restrictive



When are file systems not awesome?

- When your data is structured along multiple axes
- When data have complex interactions
- When your analyses are complex
- **Relational databases to the rescue!**

Databases did *not* replace file systems

- File archives are still the most common way to share large datasets
  - But we usually include some metadata/indexing structure as well
- As we'll see soon, **Hadoop** relies on a **distributed** file system
  - DB **abstractions** can be built on top
  - But this comes with **restrictions** on file contents and structure
- Key differences now:
  - Standardized (restricted) file formats ("CSV", Parquet, HDF5, NPY, etc.)

File-based storage

- Often, data of lives (permanently) **on disk / in the file-system**
- If it's small, we can load it into **main memory:**
  - df ← read_csv('my_data.csv')
  - analyze(df)
- If it's **too big**, we have several options:
  - **Sampling** / approximate computation
  - **Stream processing** (one or few records at a time)
  - **Data structures** / index structures
  - **Parallel computation**
  - **...**
  - Buy more memory

Good solutions often combine two or more of these strategies!

We'll see that often this semester.

Some things to consider

- Is an **exact answer** required?
  - Would an estimate be good enough?
  - Can we bound estimation errors if sampling?
- Do we expect **erroneous** or ill-formatted records?
  - What if errors are correlated with the data?
  - We may have to examine each record anyway
- Will this dataset grow over time?
  - Streaming might be beneficial
- Might we add more features to this analysis later on?
- How difficult is the implementation?

Where we’re going next

- **Database management systems** (DBMS)
  - Provide a standardized interface to store, load, and process data
  - We’ll see many additional benefits next week
- The **relational model** imposes constraints on how data is organized
  - i.e., tables / spreadsheets / dataframes
- Putting these two ideas together = **RDBMS**!

Next time…

Relational model and databases

- Reading for next week

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