

# Large-scale machine-learning

ENSAI 2019-2020 Arthur Katossky & Rémi Pépin CC-BY-SA

# Motivation

Machine-learning algorithms do not scale well.

- exact solutions do not scale // ex: matrix inversion is of  $O(n^3)$  complexity
- approximate solutions do not scale either

Google's pre-computation of BERT (a neural network for natural language processing) thanks to gradient-descent  
— a typically non-exact solution — took 4 days on a 64-TPU cluster.



Source: Arxiv — Image: Wikimedia

# Motivation

Machine-learning algorithms do not scale well.

- exact solutions do not scale // ex: matrix inversion is of  $O(n^3)$  complexity
- approximate solutions do not scale either

One would thus consider how these computation procedures may be accelerated.

- Using algorithmic tricks?
- Using more memory?
- Parallelizing on several computers?
- Relaxing our precision requirements? ...

These are many many solutions... and each of this problems is far from trivial!

Computing a median can prove challenging to speed up!

# Goal

At the end of this course, you will:

- understand the potential bottlenecks of data-intensive processes
- choose between concurrent solutions for dealing with such bottlenecks
- understand the map-reduce principle and have practical experience with Spark on HDFS
- have an overview of the cloud computing ecosystem and practical experience with one provider (AWS)
- get a glimpse over the *physical, ethical, economical, financial, environmental, statistical, political...* challenges, limitations and dangers of the suggested solutions

# Outline & schedule

## Courses:

- **Course 0:** Introduction to cloud computing (1h30, January 20)
- **Course 1:** Data-intensive computing in a nutshell (3h, January 13)
- **Course 2:** Computing parallelization principles (3h, January 17)
- **Course 3:** Parallelized computing on distributed data (2 × 1h30, January 28, February 6)

## Tutorials:

- **Tutorial 0:** Hands on AWS (1h30)
- **Tutorial 1:** How to optimize a statistical algorithm? (3h, January, 20)
- **Tutorial 2:** How to compute a statistical summary over distributed data? (3h, February, 6)
- **Tutorial 3:** How to accelerate a neural network? (3h, February, 7)

# Evaluation

- Multiple-choice questionnaire at the beginning of every tutorial
- 1 short tutorial report (tutorial 2)
- Desk examination (most probably multiple-choice questionnaire)

# Material

Course and tutorial material will be available on Moodle.

# From "Big Data" to "Large Scale"

# What is "Big Data" ?



# What is "Big Data" ?

The term started to be used in the 1990's and is a ill-defined notion.

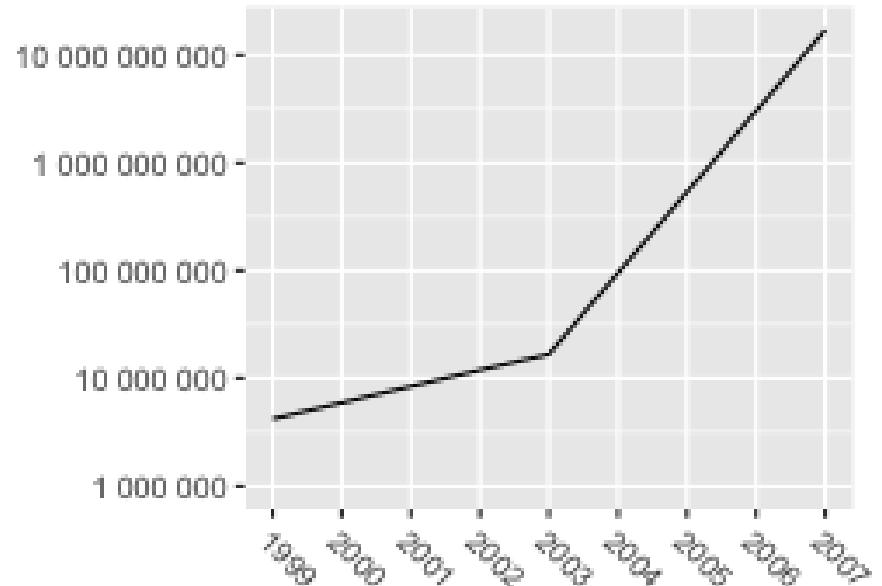
**Big Data** broadly refers to data that **cannot be managed by commonly-used software**.

# What is "Big Data" ?

*Line and row limits of Microsoft's Excel tablesheet*

	Version	Lines	Columns
until 1995	7.0	16 384	256
until 2003	11.0	65 536	256
from 2007	12.0	1 048 576	16 384

*Max. number of items stored in one tablesheet*



# What is "Big Data" ?

Size is **not** the only thing that matters.

In a tablesheet program, what kind of information can't you store properly?

- relationnal data
- images, long texts
- unstructured data (ex: web page)
- rapidly varying data (ex: tweeter feed)



# What is "Big Data" ?

You will often find the reference to the "3 V's of Big Data"

- Volume
- Velocity
- Variety<sup>1</sup>

<sup>1</sup>: not really treated in this course

# How big is "Big Data" ?

# Large-scale computing

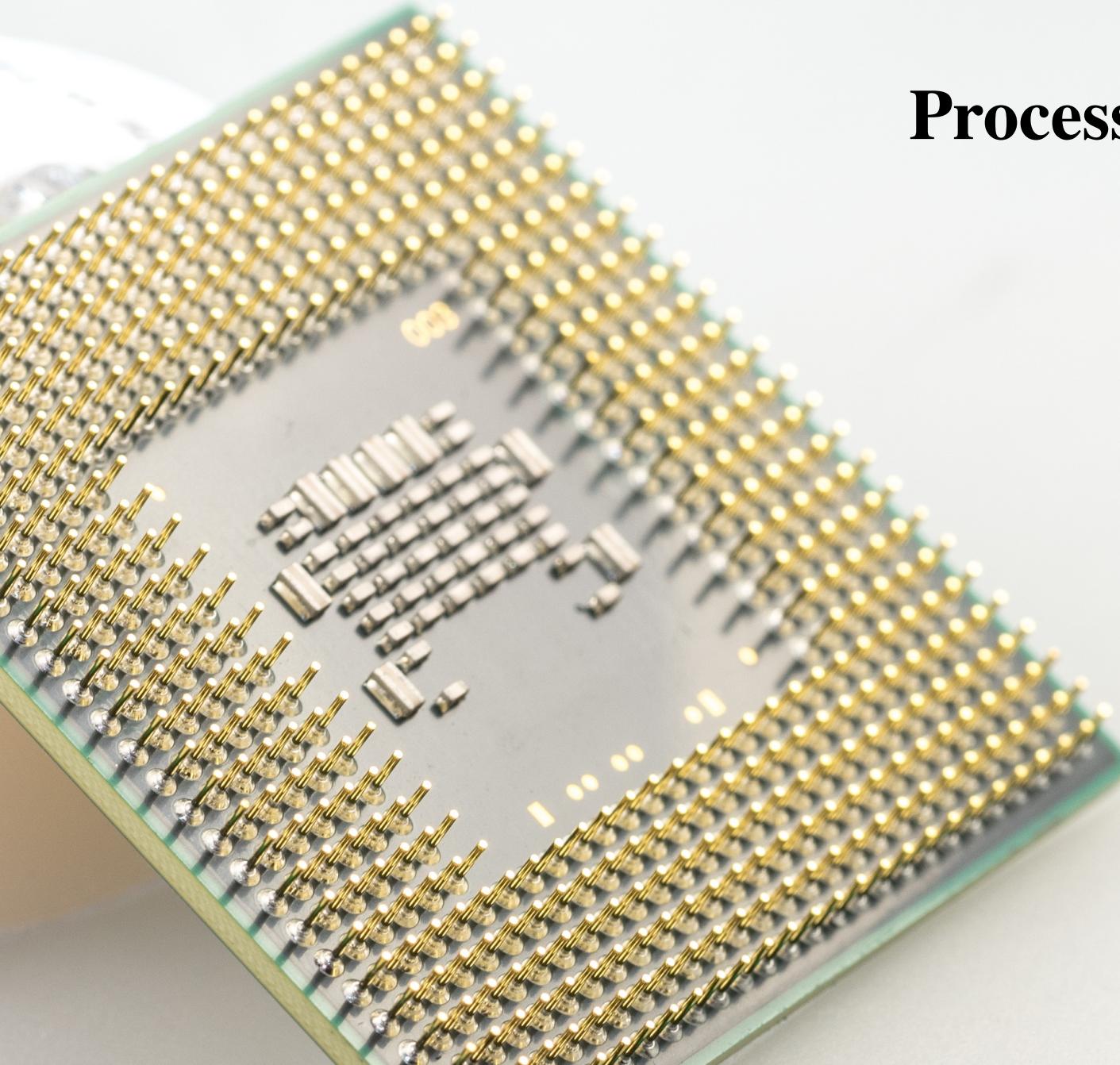
# The limits to storage and computing

# Computer science survival kit

A computer can be abstracted by four key components:

- processing (FR: processeurs)
- memory (FR: mémoire vive)
- storage (FR: stockage)
- wiring / network (FR: connexions / réseau)

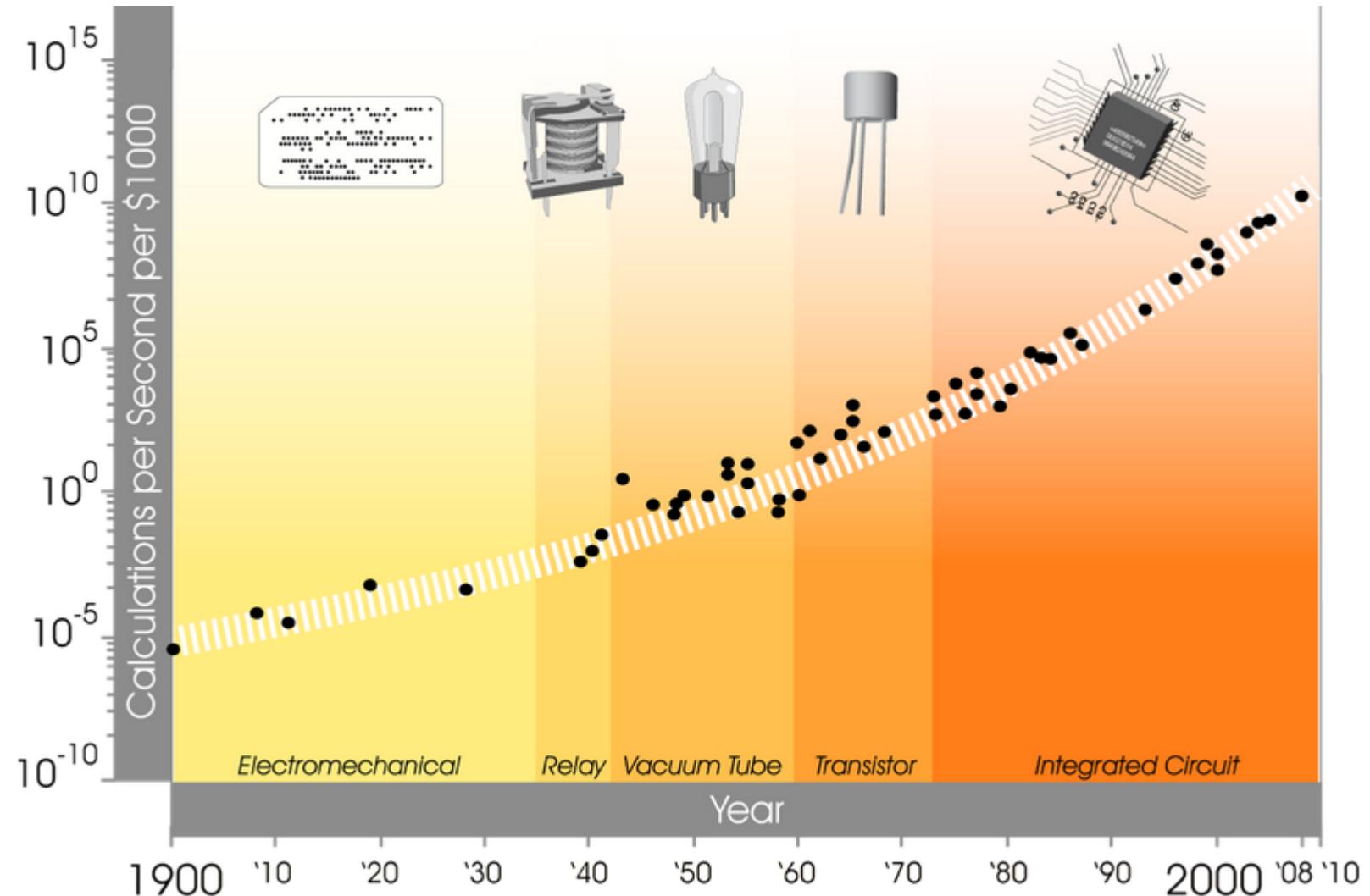
# Processors



# Processors

## What is important for us to know?

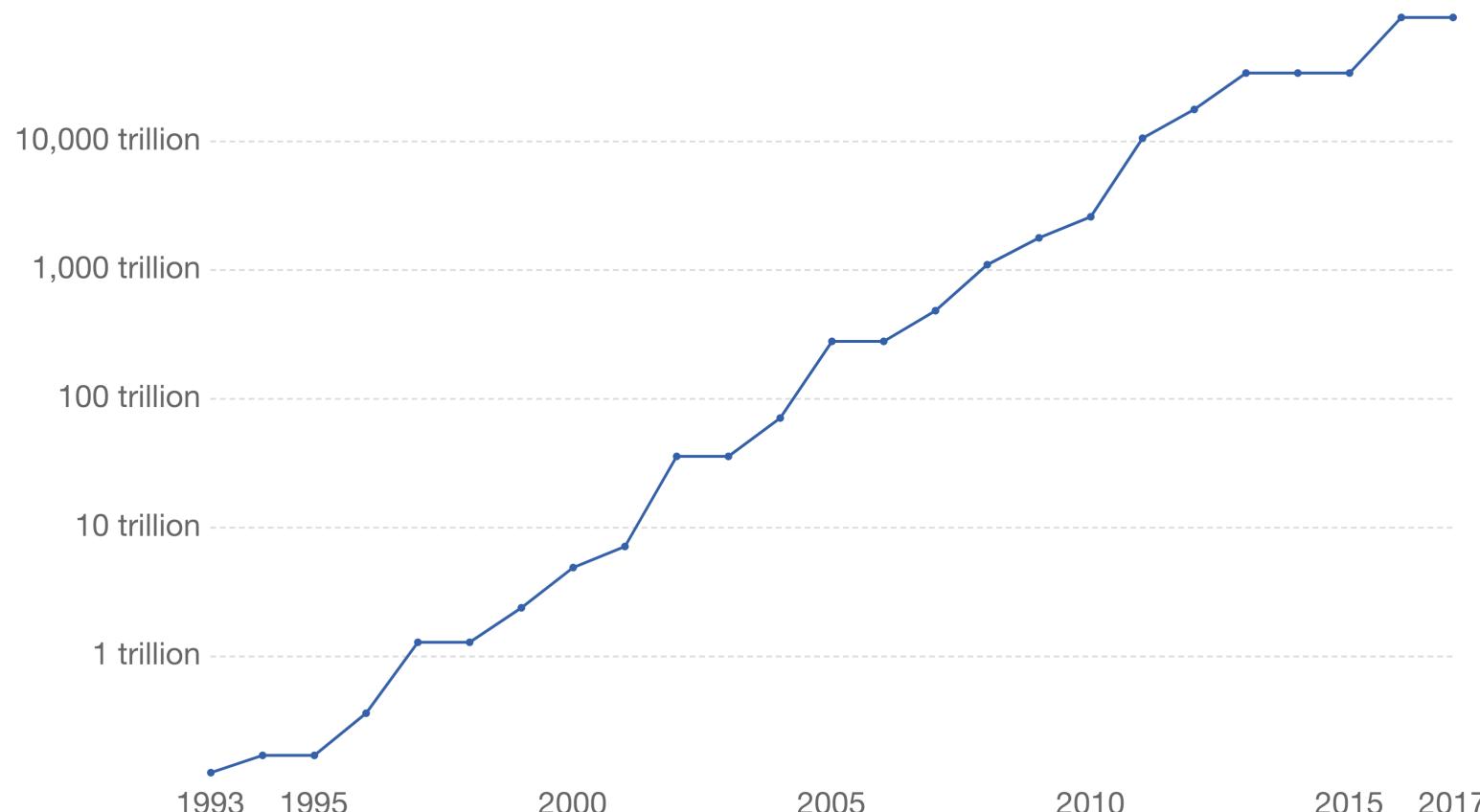
- computation happens *physically*
- programs must be converted to machine-code
- instructions are stored sequentially in a stack / thread
- multiple processors may share a stack, or have each their own
- processors can be specialized



Source: Kurzweil ([link](#)) via Our World in Data ([link](#))

# Supercomputer Power (FLOPS)

The growth of supercomputer power, measured as the number of floating-point operations carried out per second (FLOPS) by the largest supercomputer in any given year. (FLOPS) is a measure of calculations per second for floating-point operations. Floating-point operations are needed for very large or very small real numbers, or computations that require a large dynamic range. It is therefore a more accurate measured than simply instructions per second.

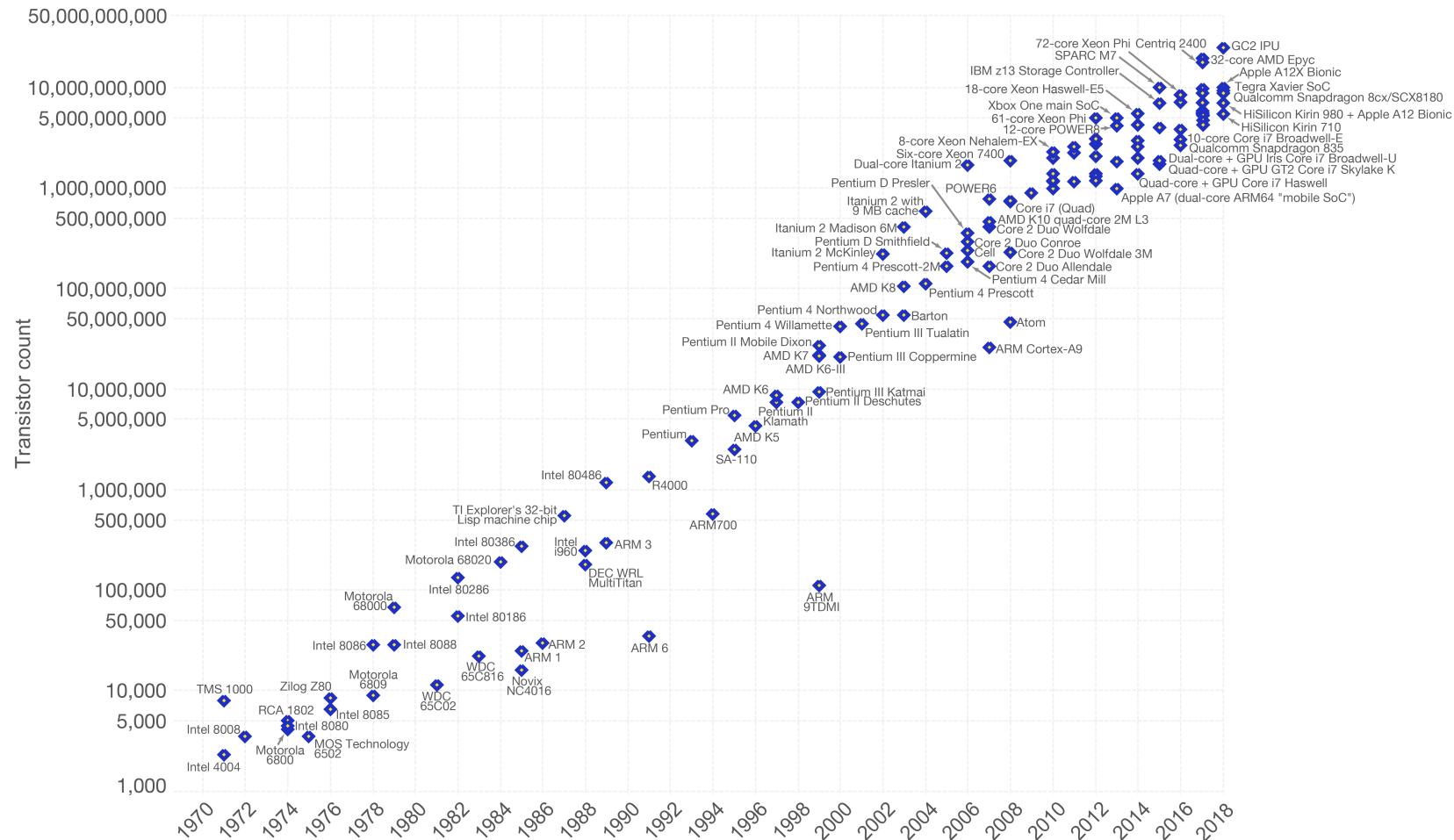


Source: TOP500 Supercomputer Database

CC BY

# Moore's Law – The number of transistors on integrated circuit chips (1971-2018)

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are linked to Moore's law.



Data source: Wikipedia ([https://en.wikipedia.org/wiki/Transistor\\_count](https://en.wikipedia.org/wiki/Transistor_count))

The data visualization is available at [OurWorldinData.org](http://OurWorldinData.org). There you find more visualizations and research on this topic.

Licensed under CC-BY-SA by the author Max Roser.

# Processors

**Processing is a limiting factor.**

# Memory



# Memory

What is important for us to know ?

- moving data around takes time
- memory is fast (ns)
- memory is volatile
- memory-processor units are **heavily** optimized

# Memory

**Memory is a limiting factor.**

# Storage



# Storage

**What is important for us to know ?**

- storage is failible
- writing and reading is slow (ms)
- data is always cached for computation

# Storage

THE GREAT THING ABOUT  
DIGITAL DATA IS THAT  
IT NEVER DEGRADES.



HARD DRIVES FAIL,  
OF COURSE, BUT THEIR  
BITS CAN BE COPIED  
FOREVER WITHOUT LOSS.



FILM DEGRADES, PRINT  
OPTIONS, BUT A COPY OF A  
CENTURY-OLD DATA FILE IS  
IDENTICAL TO THE ORIGINAL.



IF HUMANITY HAS A  
PERMANENT RECORD,  
WE ARE THE FIRST  
GENERATION IN IT.

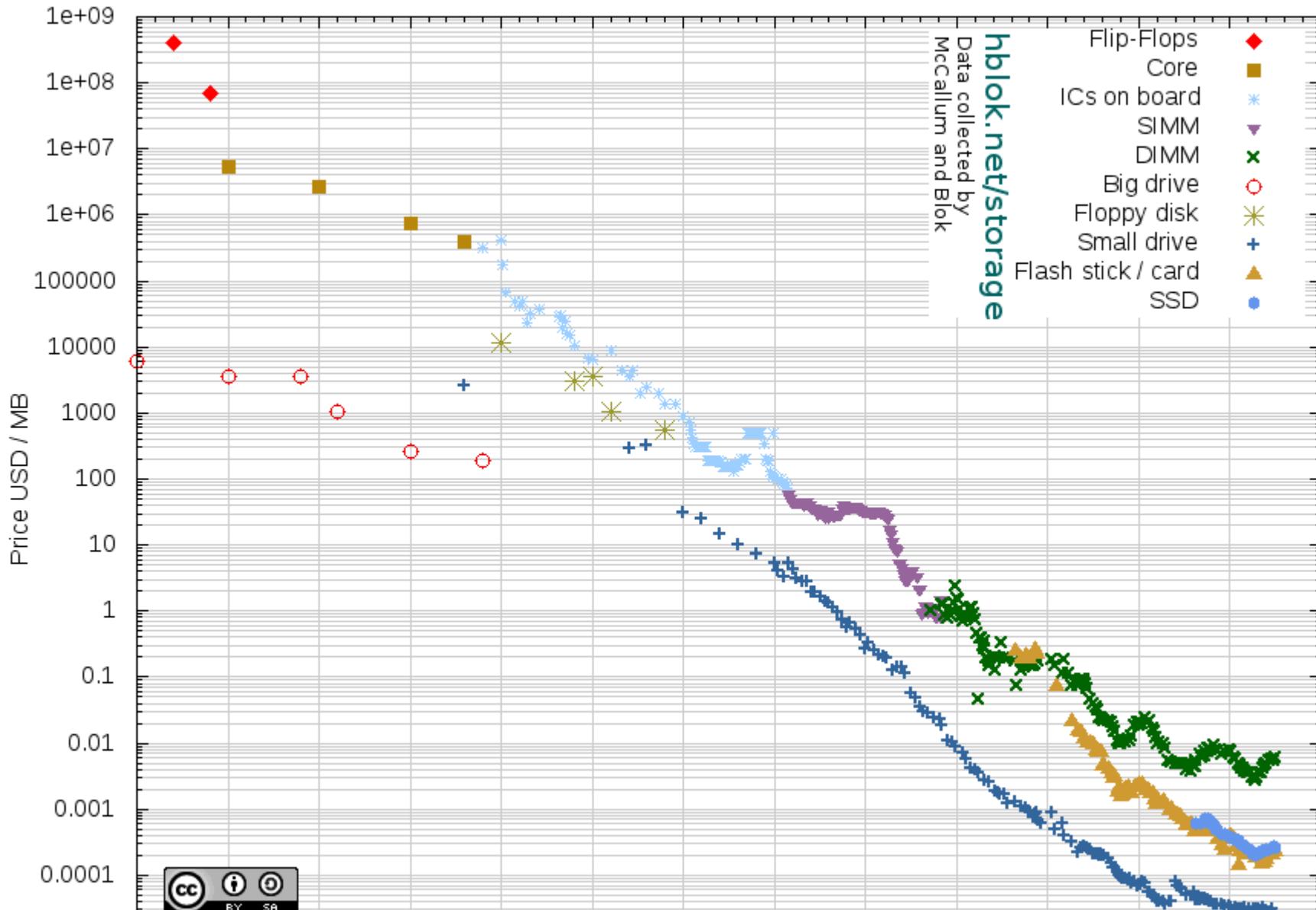
AMAZING

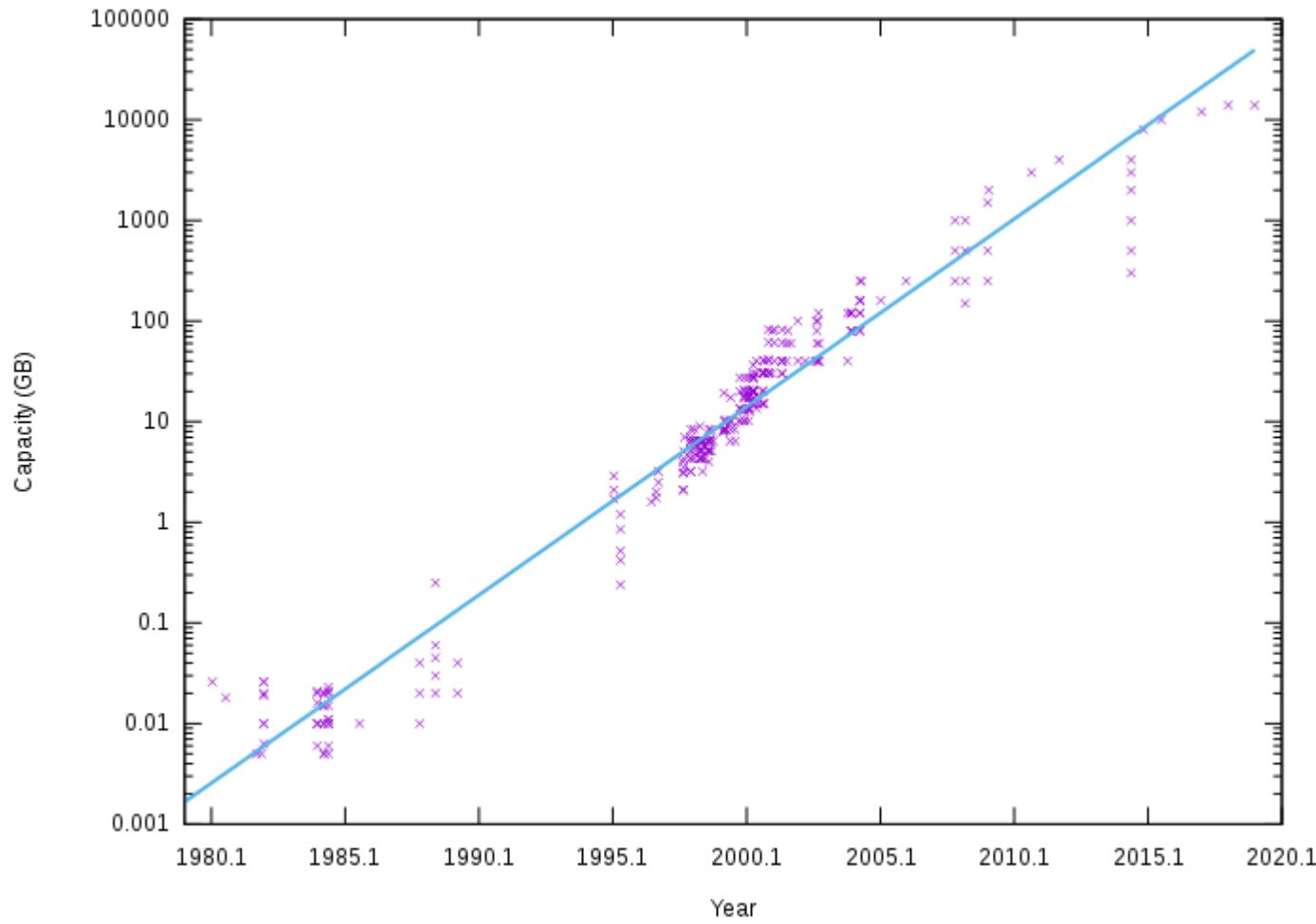
9GAG

# Storage

**Storage is a limiting factor.**

# Historical Cost of Computer Memory and Storage





# Network

# Interpreting vs. compiling

# Challenges with large-scale machine-learning

# What if ... ?

# What if data is too big to fit in RAM ?

1. do less (filter, sample)
2. buy more/bigger memory
3. do things in chunks (stream, pipeline)
4. distribute the data



# What if ... ?

# What if file is too big for file system ?

1. do less (filter, sample)
2. change file system
3. cut file into pieces and process in chunk
4. go on databases, that's what they are made for
5. go on cloud, they take care of the problems for you



# What if file is too big for file system ?

**Q: How much data can I store?**



The total volume of data and number of objects you can store are unlimited. Individual Amazon S3 objects can range in size from 1 byte to 5 terabytes. The largest object that can be uploaded in a single PUT is 5 gigabytes. For objects larger than 100 megabytes, customers should consider using the [Multipart Upload](#) capability.

# What if file is too big for file system ?

## General Availability: Larger Block Blobs in Azure Storage

Posted on December 22, 2016



[Michael Hauss](#), Program Manager, Azure Storage

Azure Blob Storage is a massively scalable object storage solution capable of storing and serving tens to hundreds of petabytes of data per customer across a diverse set of data types including media, documents, log files, scientific data and much more. Many of our customers use Blobs to store very large data sets, and have requested support for larger files. The introduction of larger Block Blobs increases the maximum file size from 195 GB to 4.77 TB. The increased blob size better supports a diverse range of scenarios, from media companies storing and processing 4K and 8K videos to cancer researchers sequencing DNA.

# What if file is too big for file system ?



Google Cloud

• There is a maximum size limit of 5 TB for individual objects stored in Cloud Storage.

# What if ... ?

# What if data is too big to fit on disk ?

1. store less (sample, filter)
2. buy bigger physical disk
3. buy a storage server
4. rent storage space in the cloud
5. distribute data yourself on serval nodes



# What if ... ?

# What if computation takes ages ?

1. do less (filter, sample), especially in development stages
2. do less (less tests, less formatting)
3. go low-level (compiling ... or building chips!)
4. profile your code (how does it scale ? where are the bottlenecks?)
5. buy bigger or more cores
6. rent cores on the cloud
7. avoid i/o or network operations
8. **parallelize on non-local nodes**



\*

\*\*

# What if ... ?

# What if computation / storage is too expensive ?

1. store or compute less (filter, sample)
2. consider cloud computing
3. be carefull with databases requests
4. go from RAM to disk
5. use smaller but many computing units (ex:  
scientific grids)



# What if ... ?

# What if data i/o is too fast ?

1. for computing: stream / pipeline
2. for storage: fast databases



# Other issues with Large-Scale Machine-Learning

# Statistical issues

# **Environmental issues**

# Political issues

# Today's conclusion

1. MOST DATA IS SMALL BUT COMPUTATION ON IT CAN STILL BE A CHALLENGE
2. DISTIBUTING DOES NOT HELP ; DO IT ONLY IF YOU CAN'T AVOID IT
3. SAMPLING IS OFTEN A VERY GOOD SOLUTION
4. THERE ARE MANY MANY WAYS TO SPEED UP A COMPUTATION
5. THERE ARE ENVIRONMENTAL, ECONOMICAL, ETHICAL AND STATISTICAL CHALLENGES WITH THE ACCUMULATION OF SUCH A QUANT