EE 274
Data Compression: Theory and Applications

Lecture 1
Introduction and Logistics

9/27/23

Course Staff

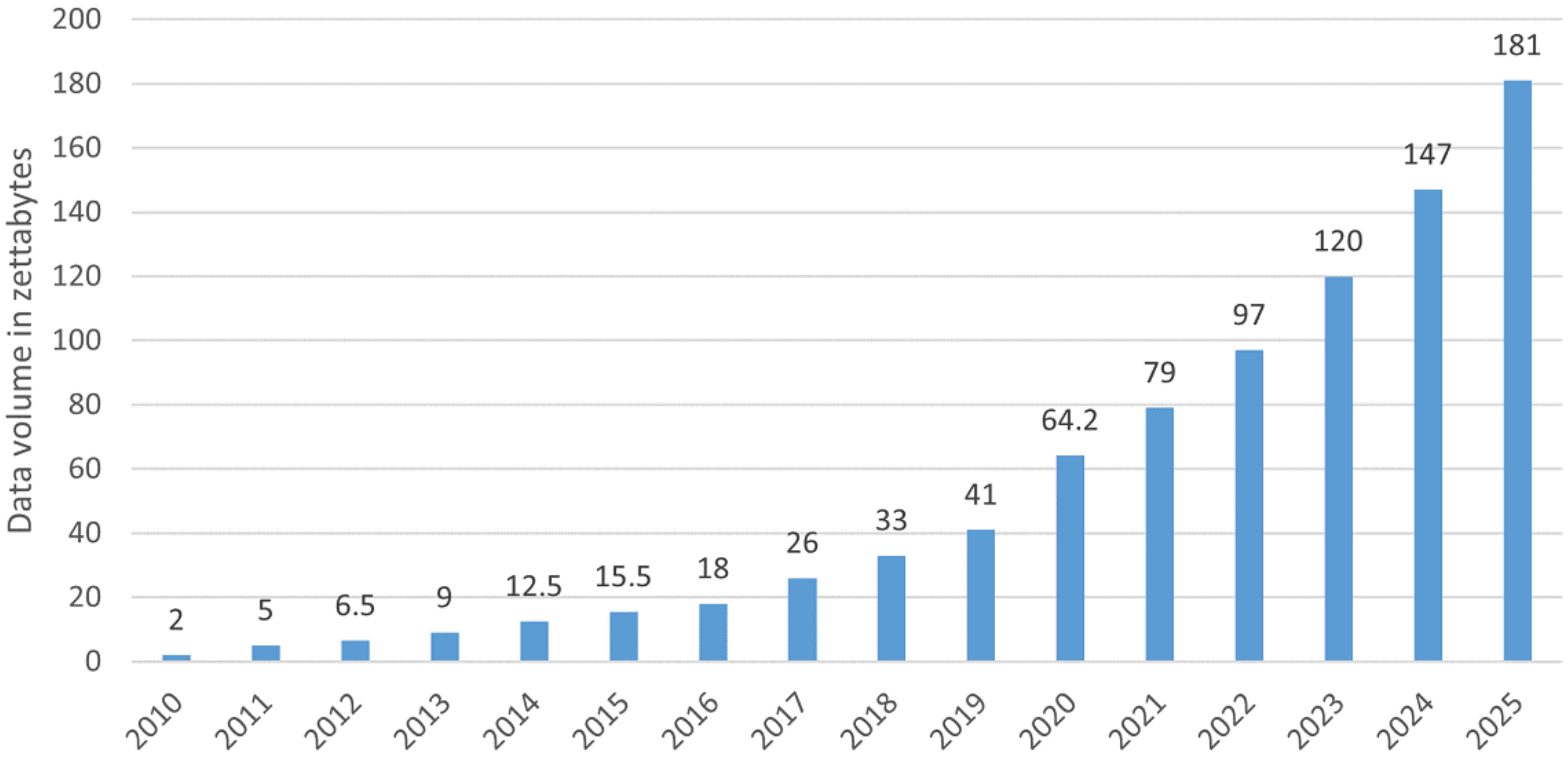
Tsachy Weissman
Professor, EE

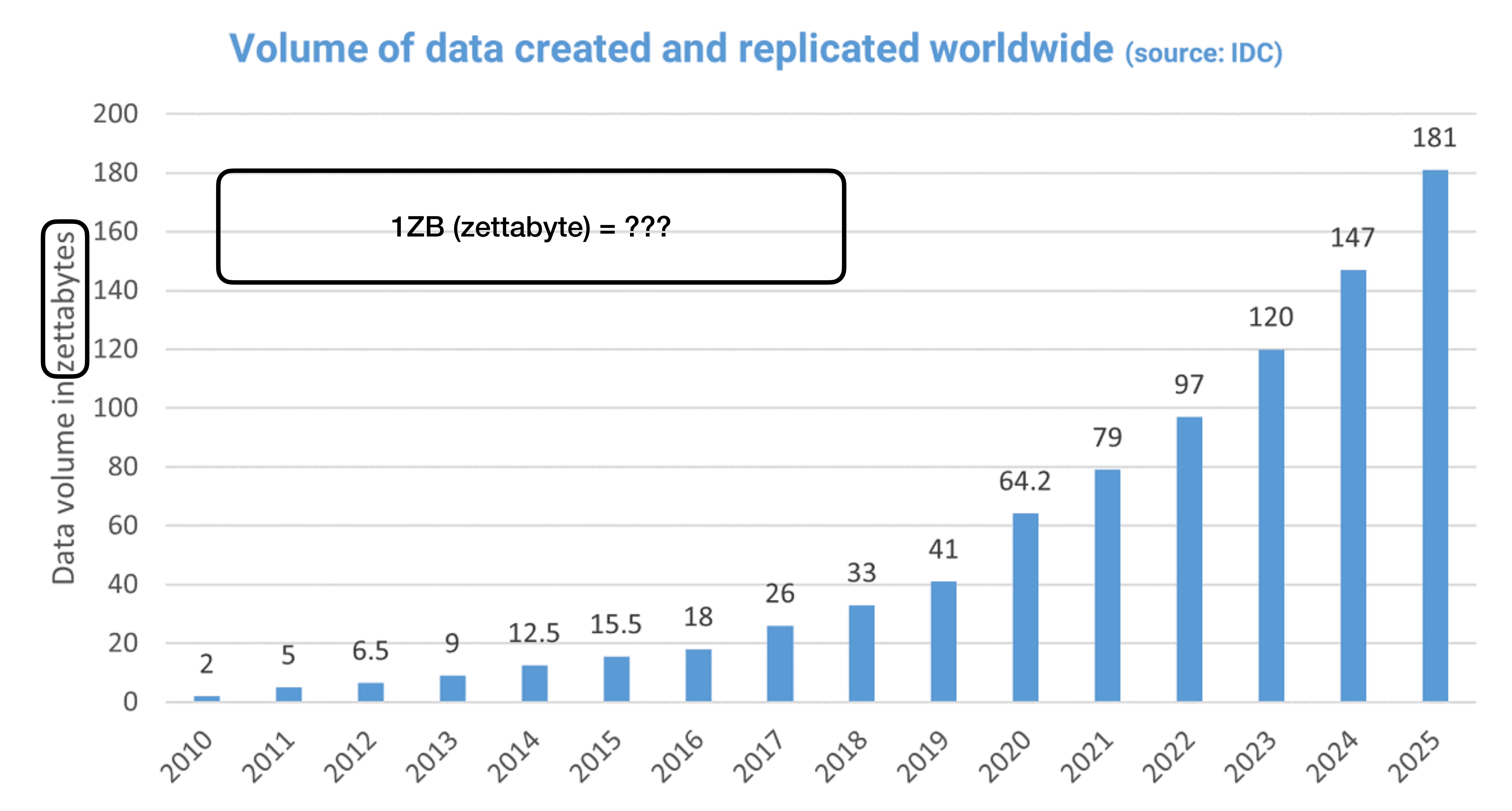
Shubham Chandak
Sr. Applied Scientist at S3, AWS

Pulkit Tandon
Research Scientist at Granica

TA: Noah Huffman
PhD Candidate, Physics

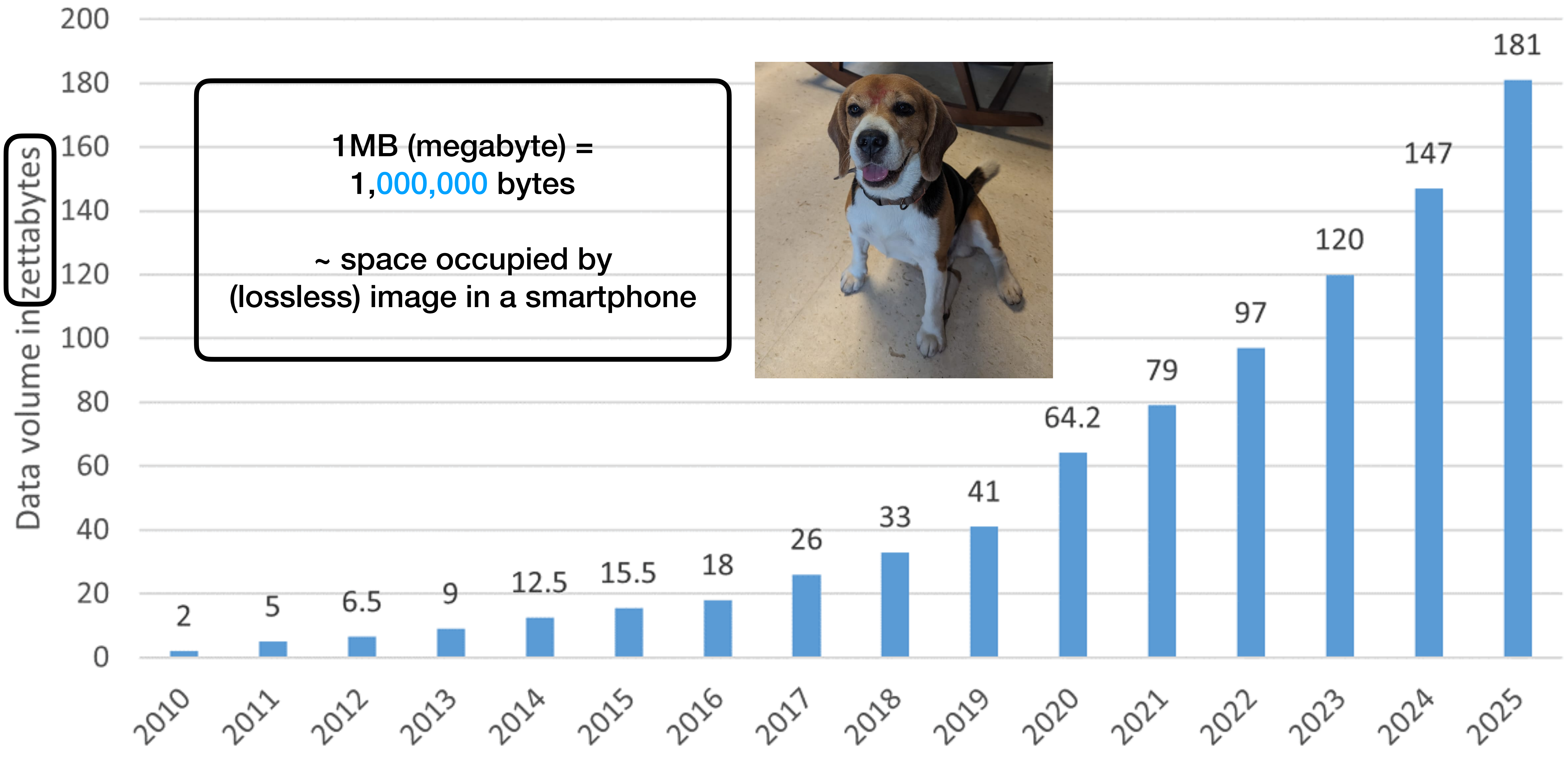
Volume of data created and replicated worldwide (source: IDC)





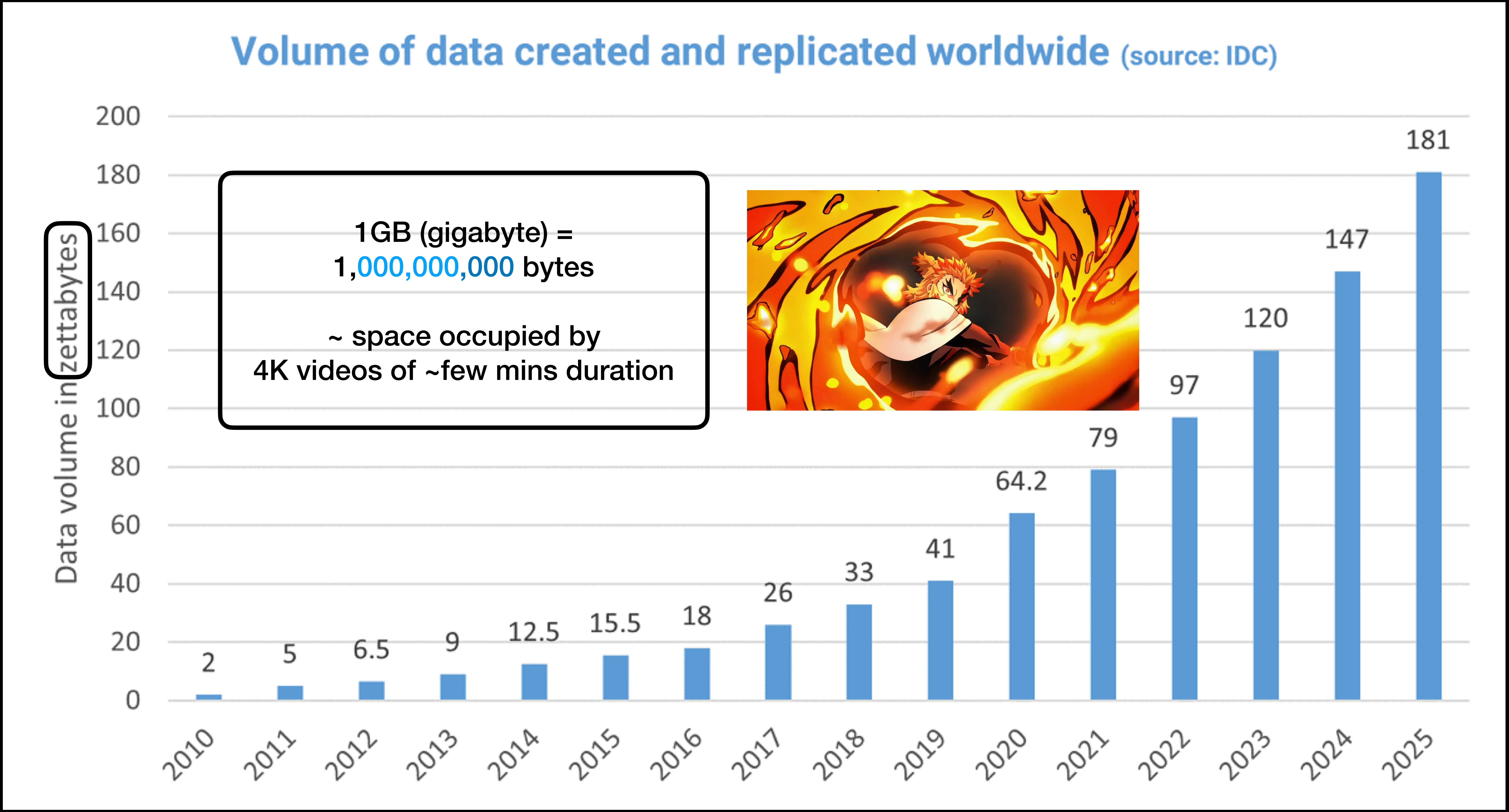
1ZB (zettabyte) = ???

Volume of data created and replicated worldwide (source: IDC)



1MB (megabyte) =
1,000,000 bytes

~ space occupied by
(lossless) image in a smartphone



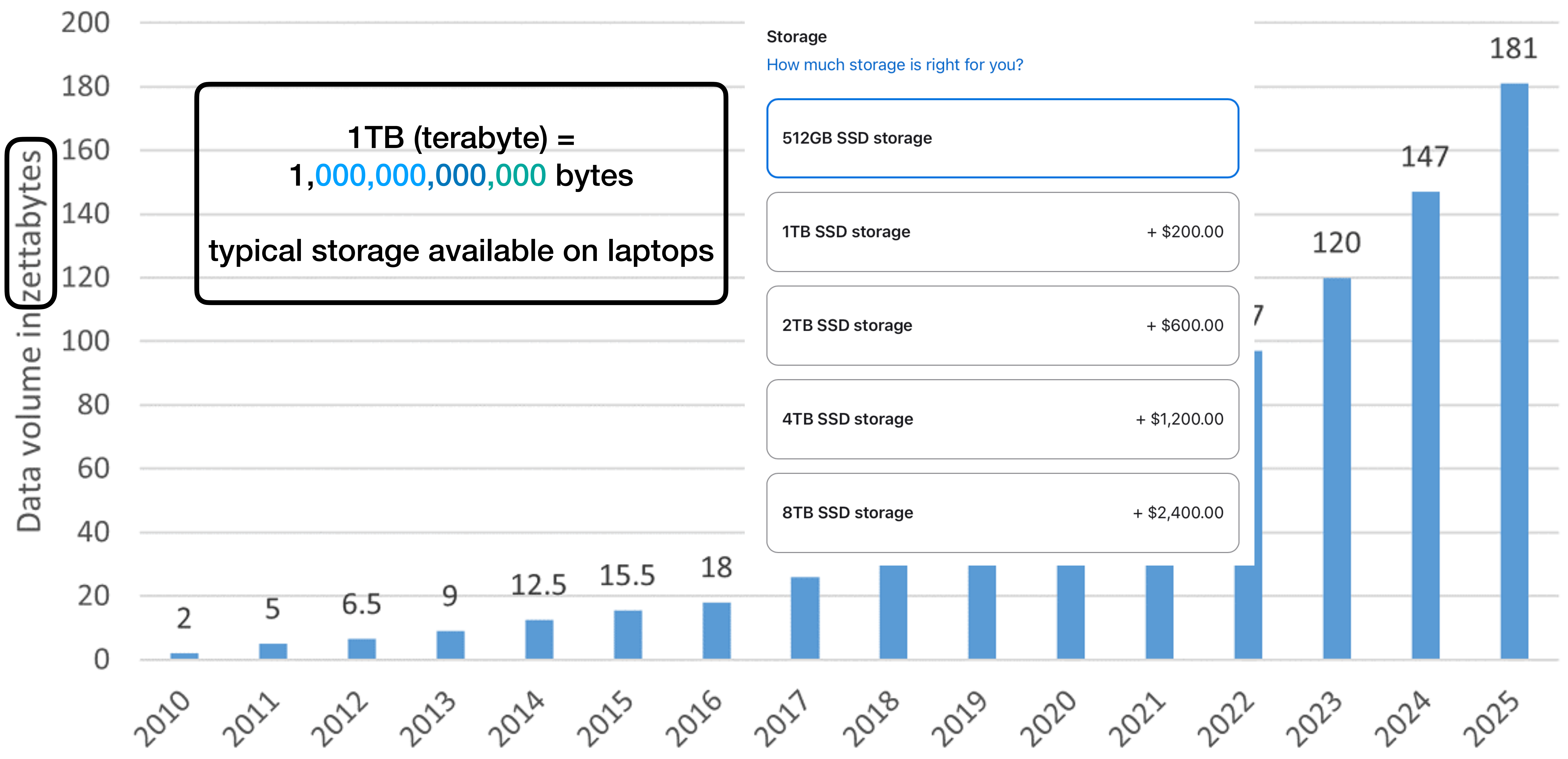
Volume of data created and replicated worldwide (source: IDC)

1GB (gigabyte) =
1,000,000,000 bytes

~ space occupied by
4K videos of ~few mins duration

Source: https://www.youtube.com/watch?v=VYUPMFu4fR8

Volume of data created and replicated worldwide (source: IDC)



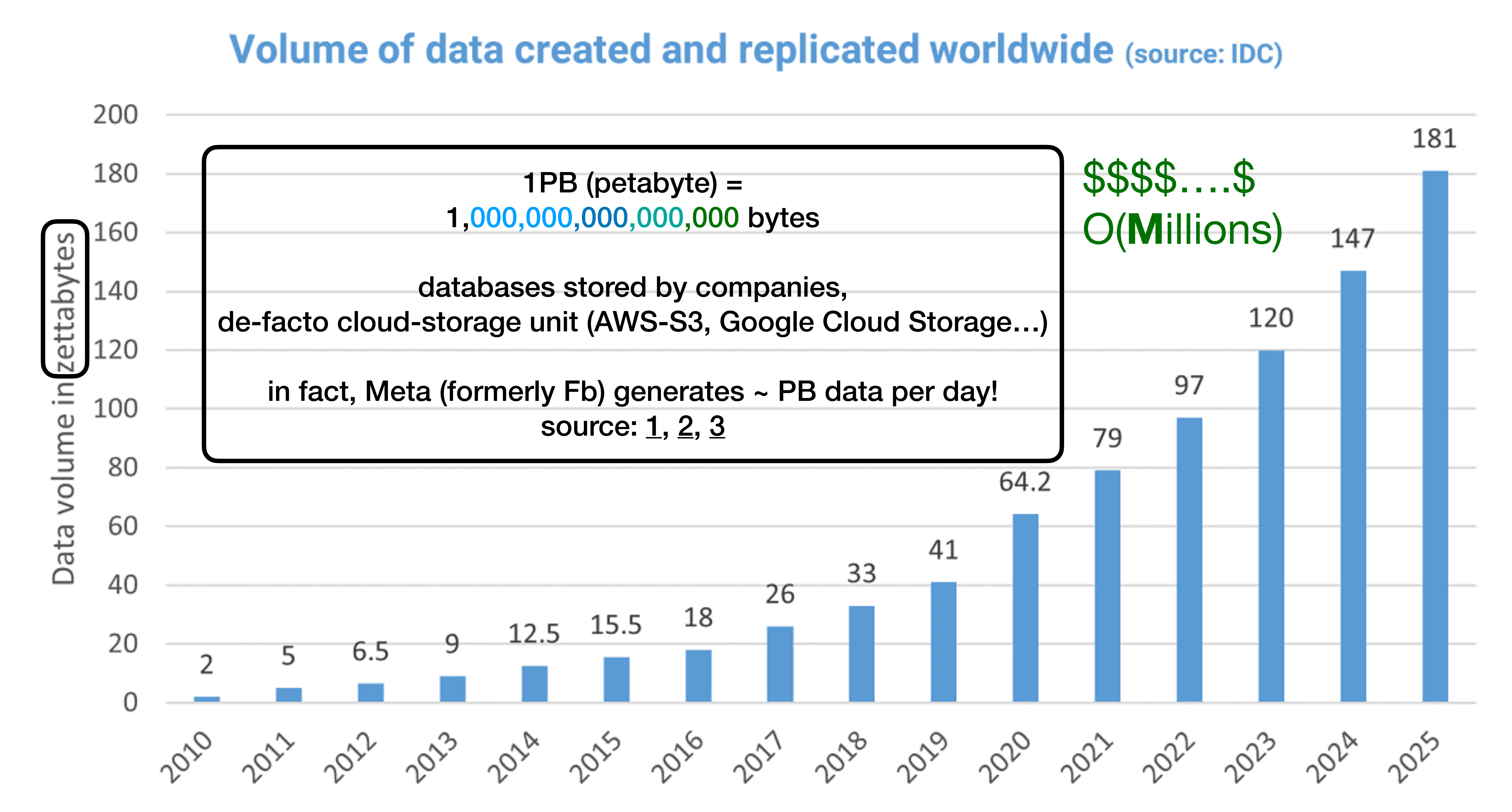
1TB (terabyte) =
1,000,000,000,000 bytes

typical storage available on laptops

Storage

How much storage is right for you?

512GB SSD storage	
1TB SSD storage	+ $200.00
2TB SSD storage	+ $600.00
4TB SSD storage	+ $1,200.00
8TB SSD storage	+ $2,400.00

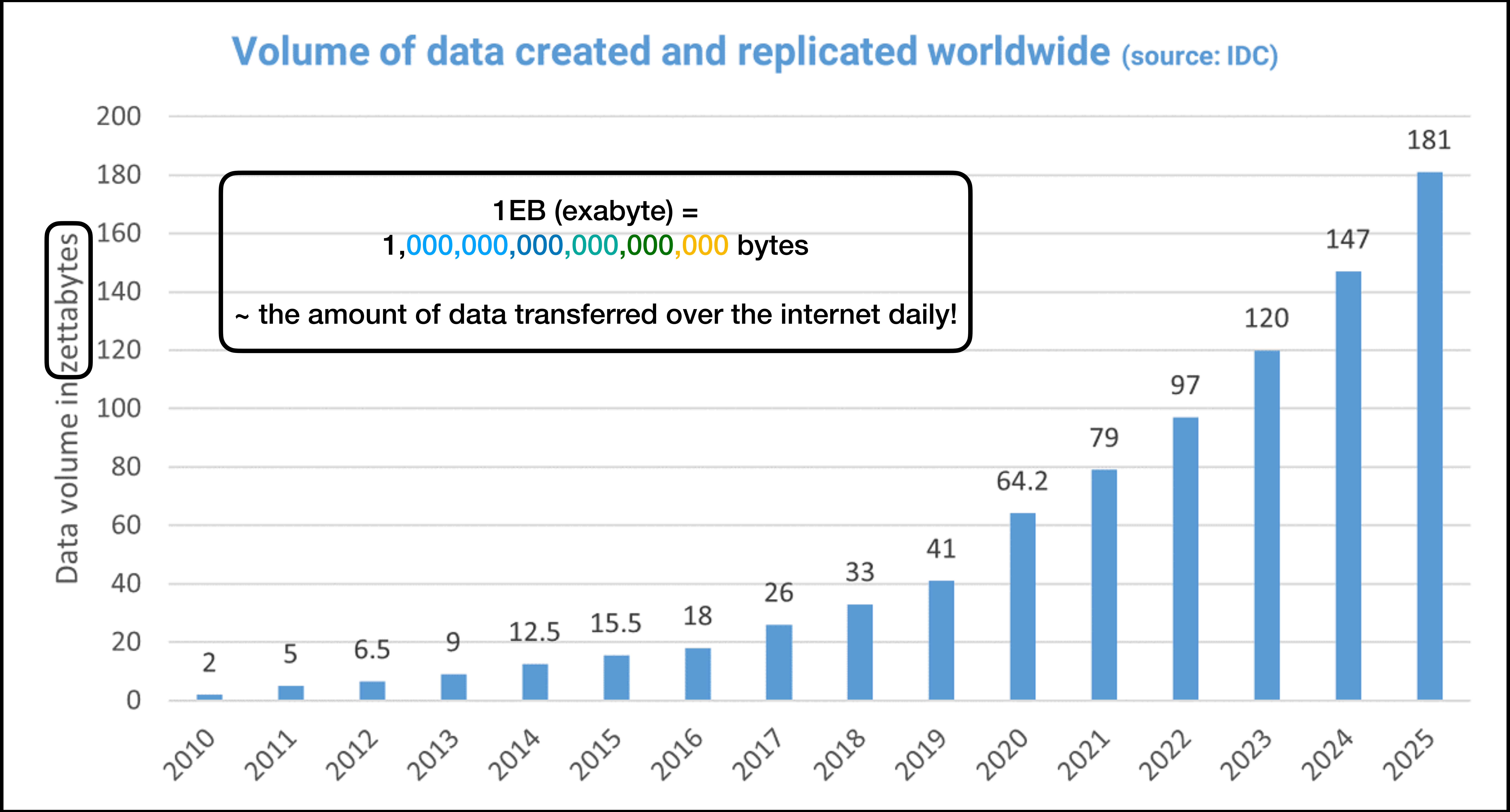


1PB (petabyte) =
1,000,000,000,000,000 bytes

databases stored by companies,
de-facto cloud-storage unit (AWS-S3, Google Cloud Storage…)

in fact, Meta (formerly Fb) generates ~ PB data per day!
source: 1, 2, 3

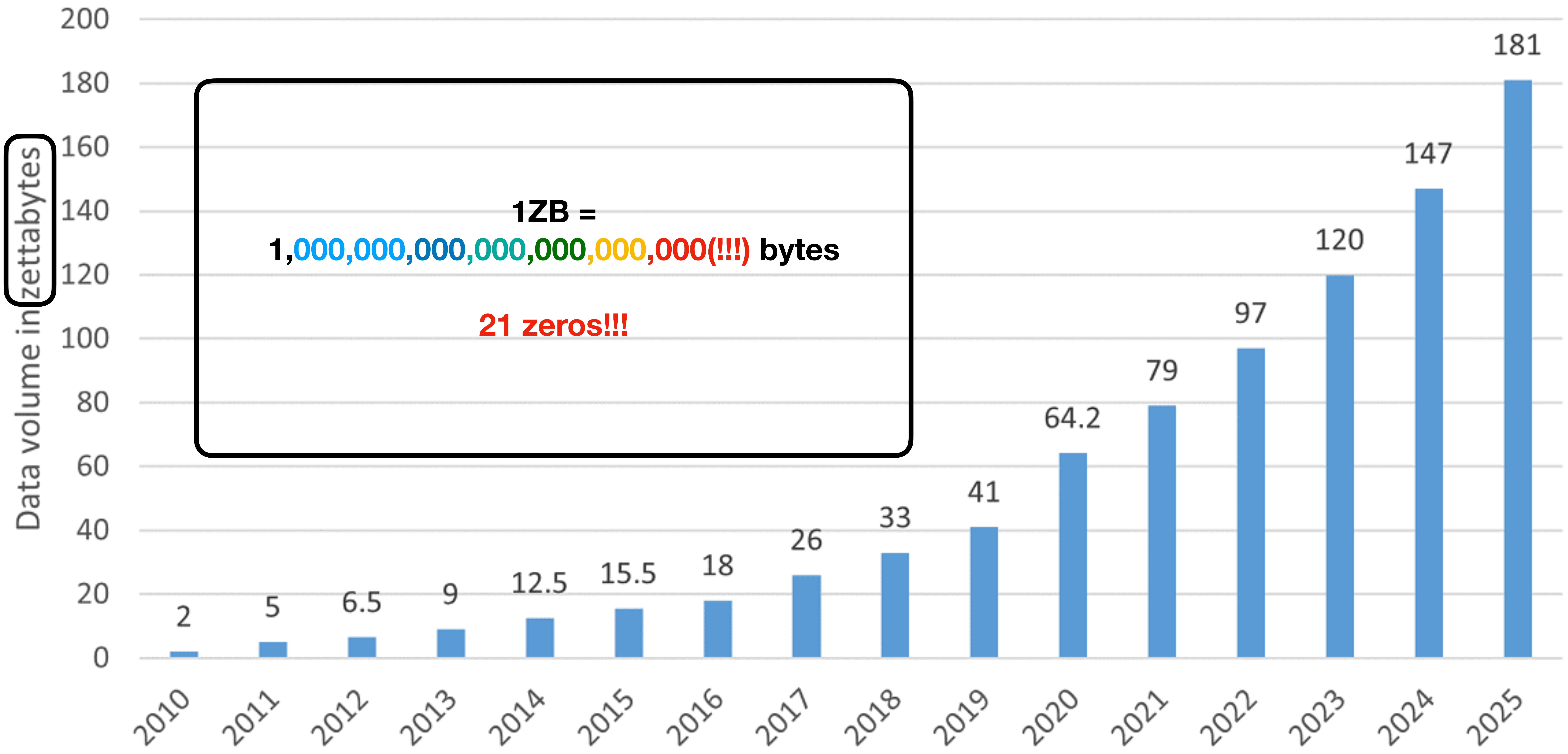
$$$$….$
O(**M**illions)



1EB (exabyte) =
1,000,000,000,000,000,000 bytes

~ the amount of data transferred over the internet daily!

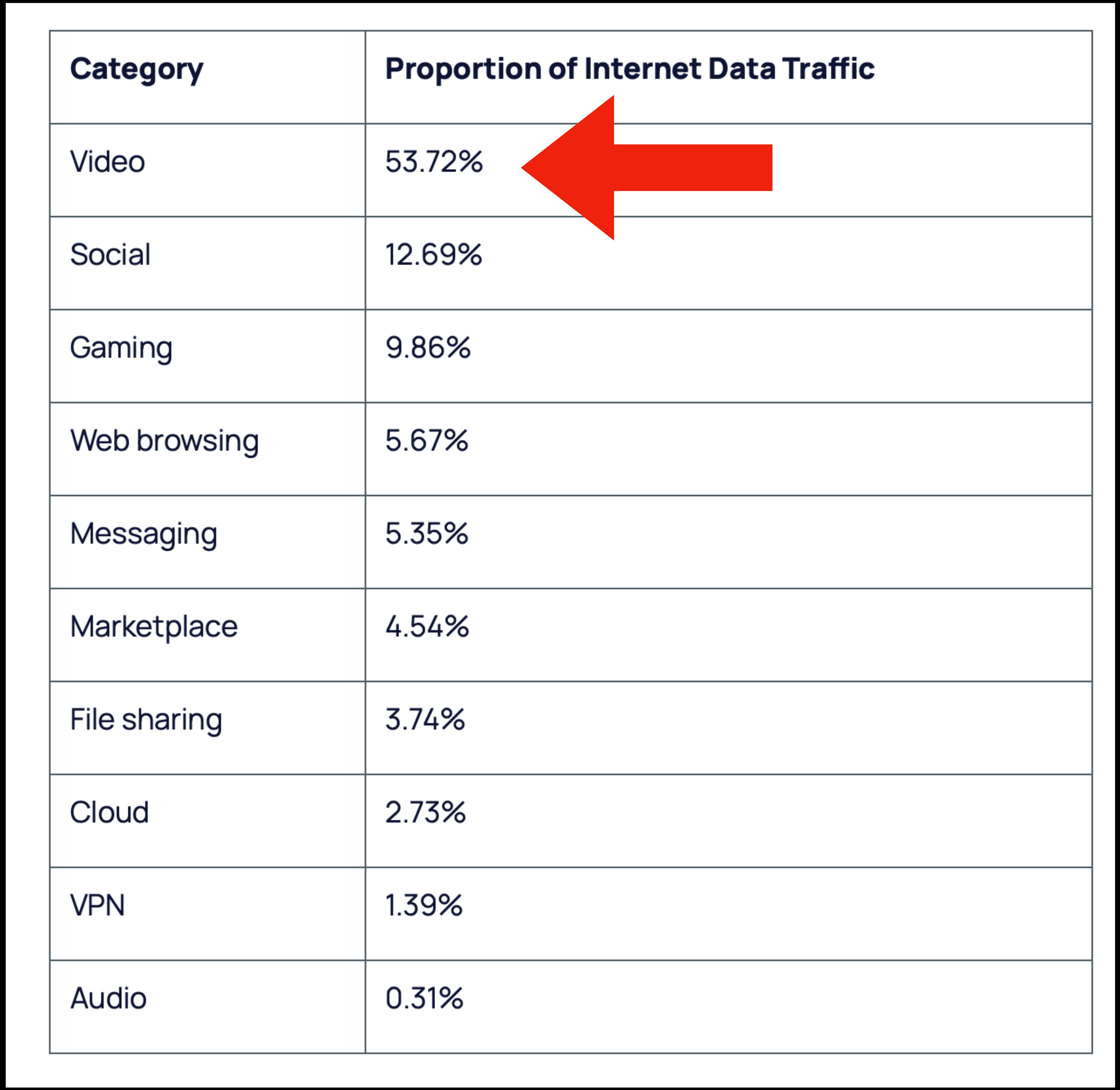
Volume of data created and replicated worldwide (source: IDC)



1ZB =
1,000,000,000,000,000,000,000(!!!) bytes

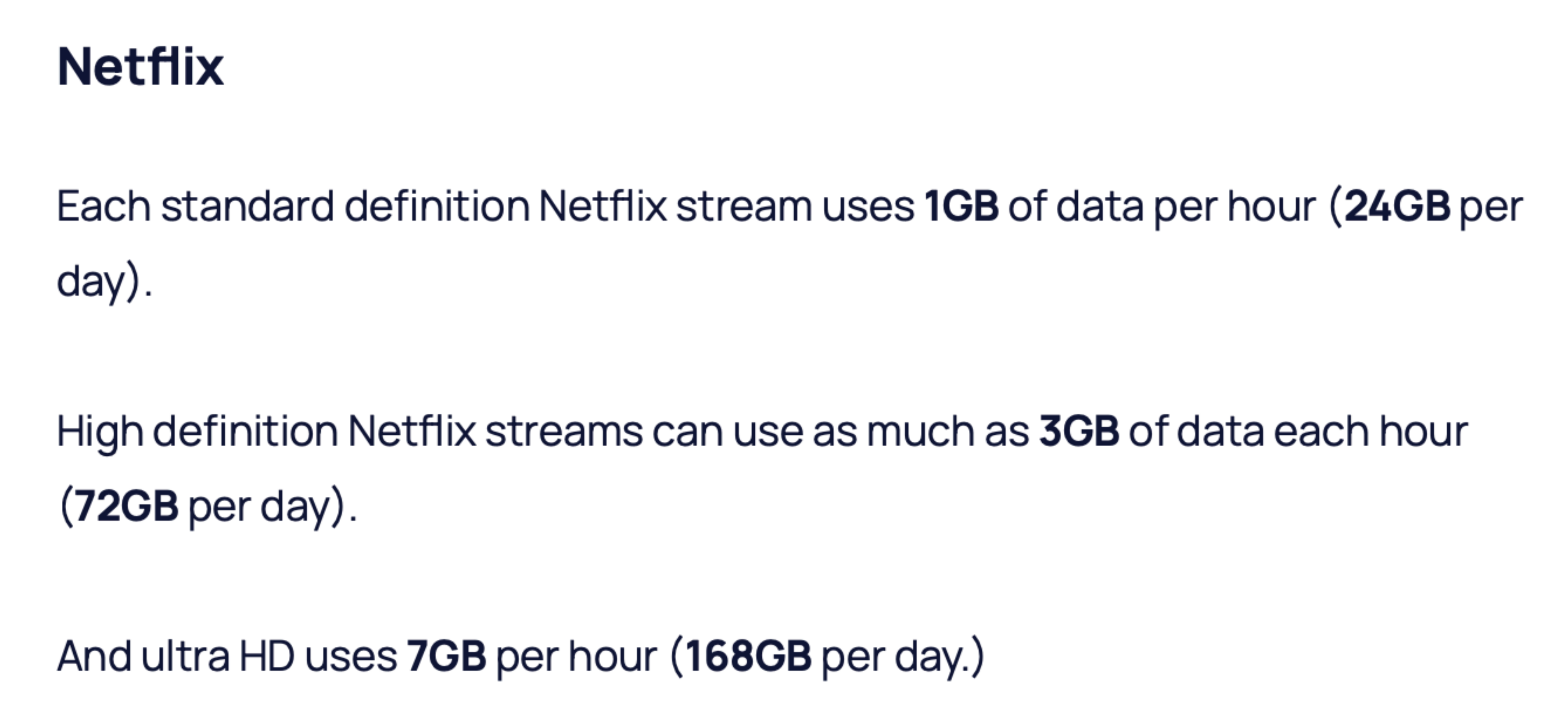
21 zeros!!!

It's not just about data *"at rest"* but also data *"in motion"*



Category	Proportion of Internet Data Traffic
Video	53.72%
Social	12.69%
Gaming	9.86%
Web browsing	5.67%
Messaging	5.35%
Marketplace	4.54%
File sharing	3.74%
Cloud	2.73%
VPN	1.39%
Audio	0.31%

Netflix will reduce streaming quality in Europe for 30 days



Netflix

Each standard definition Netflix stream uses **1GB** of data per hour (**24GB** per day).

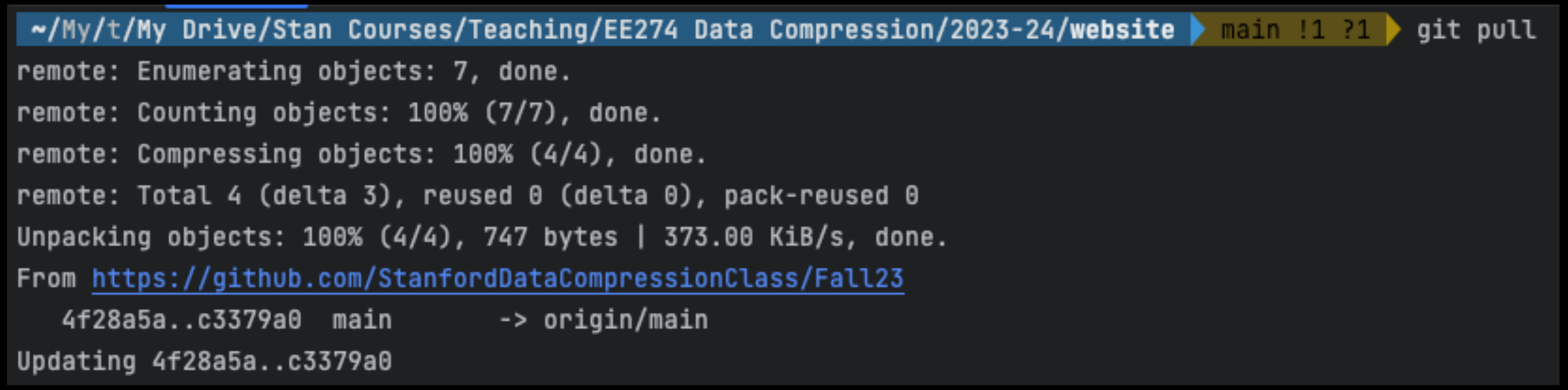
High definition Netflix streams can use as much as **3GB** of data each hour (**72GB** per day).

And ultra HD uses **7GB** per hour (**168GB** per day.)

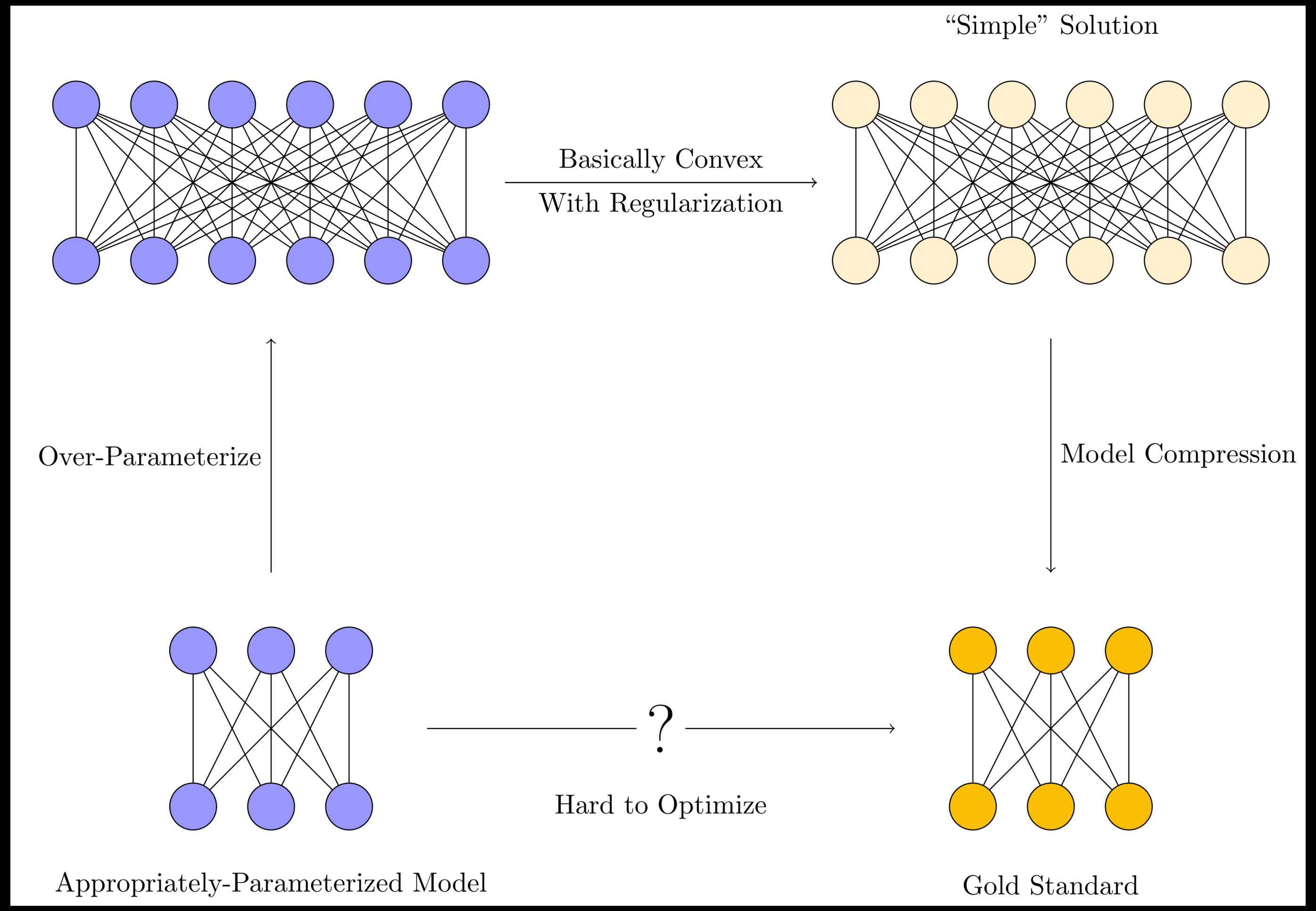
Source: https://explodingtopics.com/blog/data-generated-per-day#category

And it’s not just about one type of data

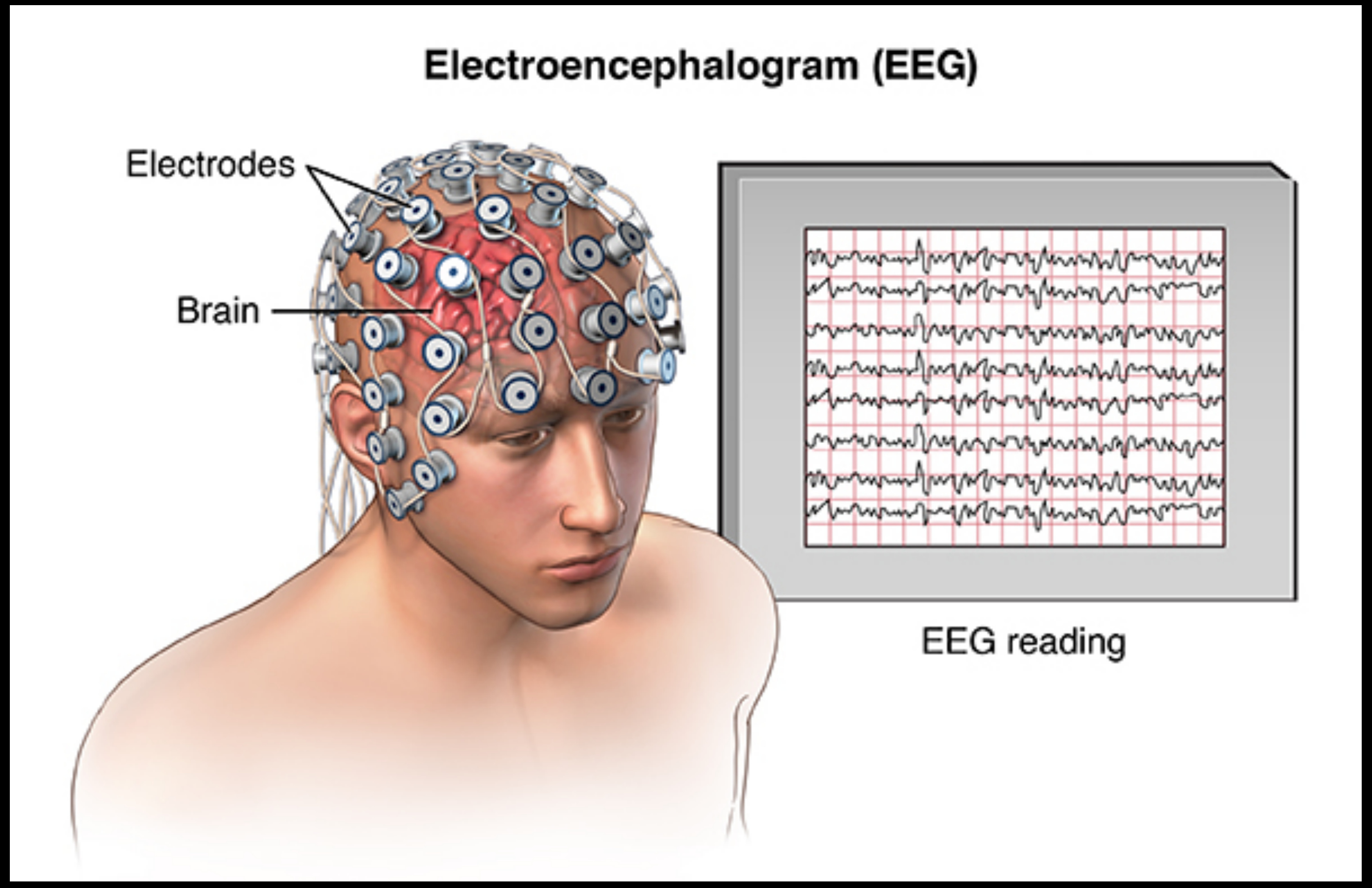
Text, Log files, Code, Genomic Data, Emails, Google Searches, “Tweets”, Images, Videos, EEG data, Sensor data, Neural Recordings, Multiverse, LLMs, ML models, …



```
~/My/t/My Drive/Stan Courses/Teaching/EE274 Data Compression/2023-24/website  main !1 ?1  git pull
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 3), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), 747 bytes | 373.00 KiB/s, done.
From https://github.com/StanfordDataCompressionClass/Fall23
   4f28a5a..c3379a0  main       -> origin/main
Updating 4f28a5a..c3379a0
```



Source: http://mitchgordon.me/machine/learning/2020/01/13/do-we-really-need-model-compression.html



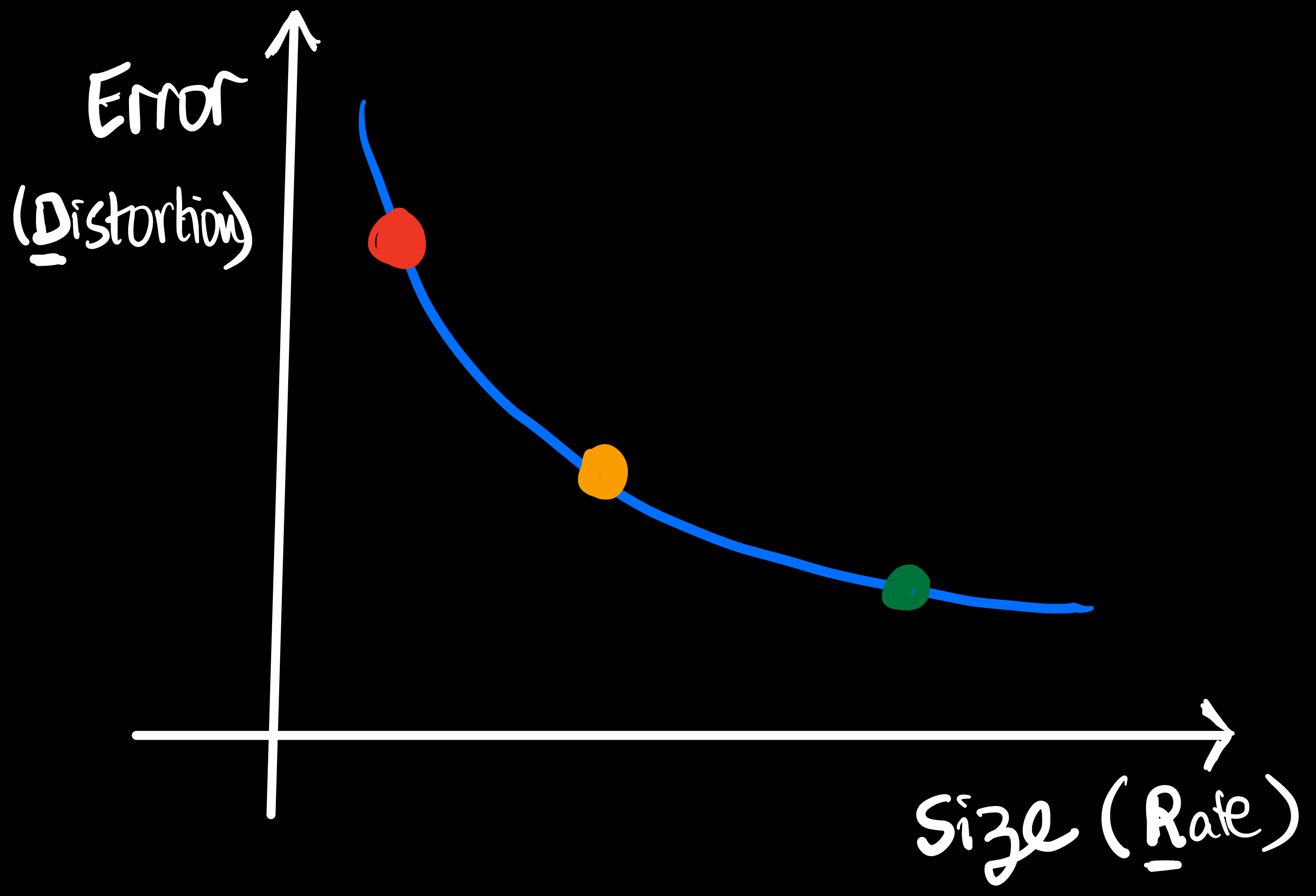
Source: https://www.kaggle.com/code/ruslankl/eeg-data-analysis

Moreover, filled with tradeoffs…

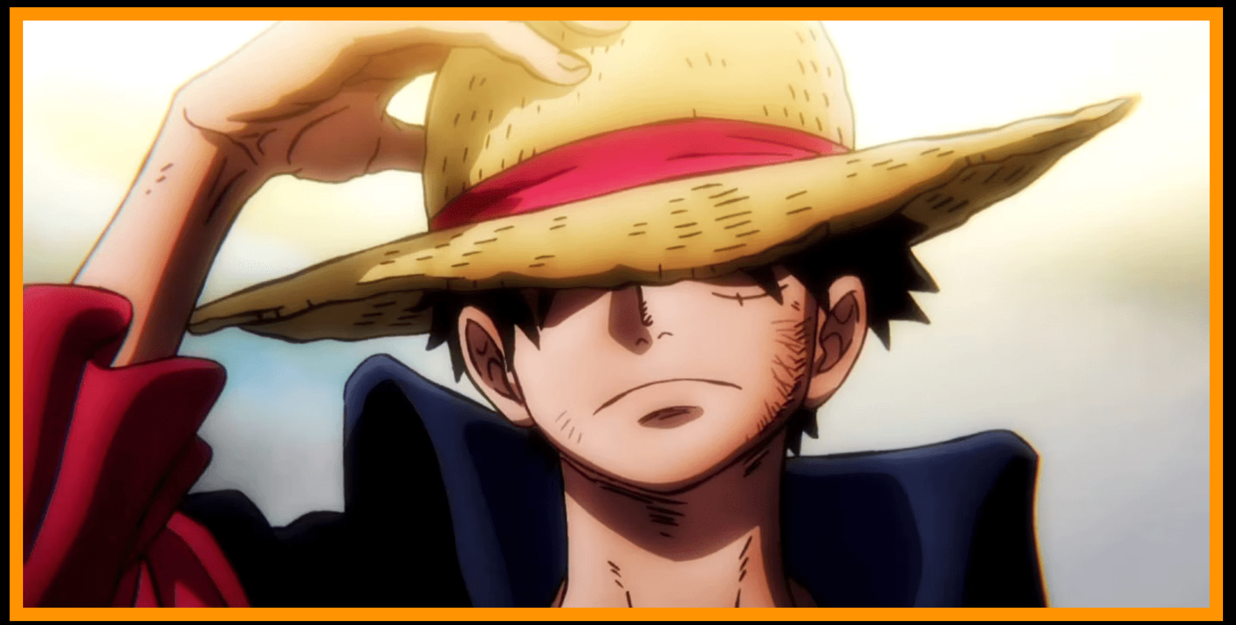
5.3 MB 🟢

2.5 MB 🟠

872 KB 🔴



585 KB



280 KB

13 KB

“Data compression =
succinct representation of information”

Tsachy Weissman

Why Compression?

Storage is costly

Purest form of information processing

- distilling your information to its essence
- communication without the noise

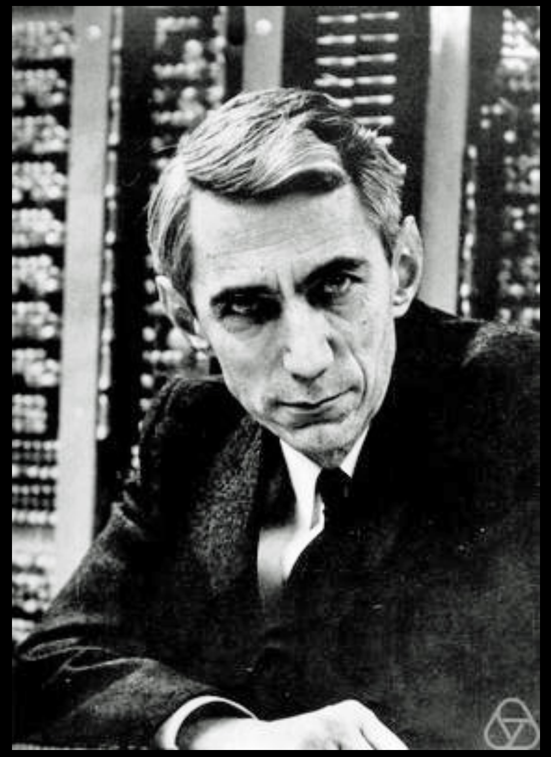
Compressed representations

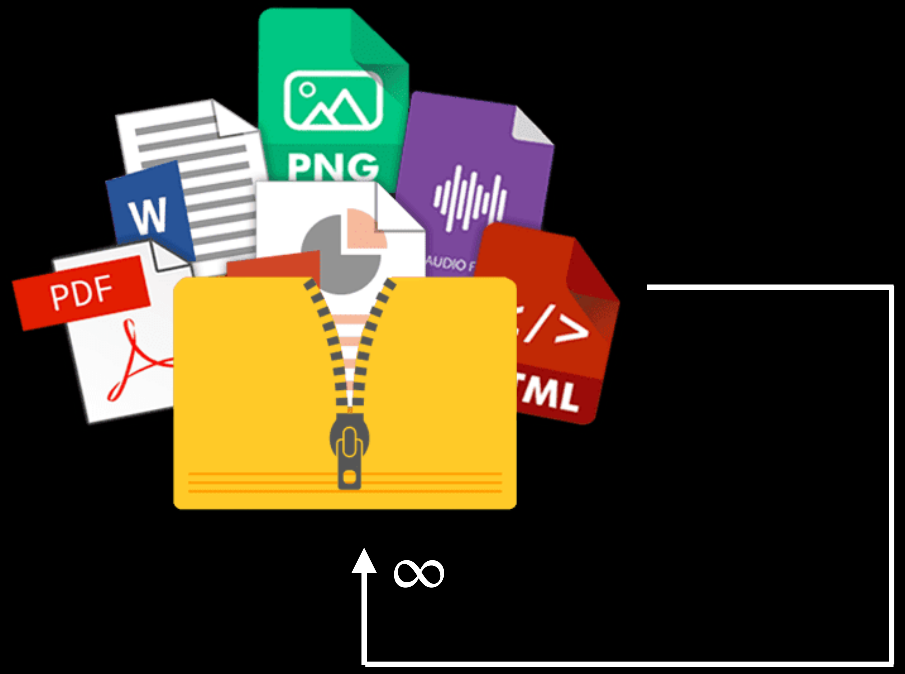
- easy to transmit and search
- can simplify implementations

Connection with data modeling, prediction, ML etc.

Critical building block in scalable and efficient systems

There is something for everyone…





Beautiful Theory

e.g.

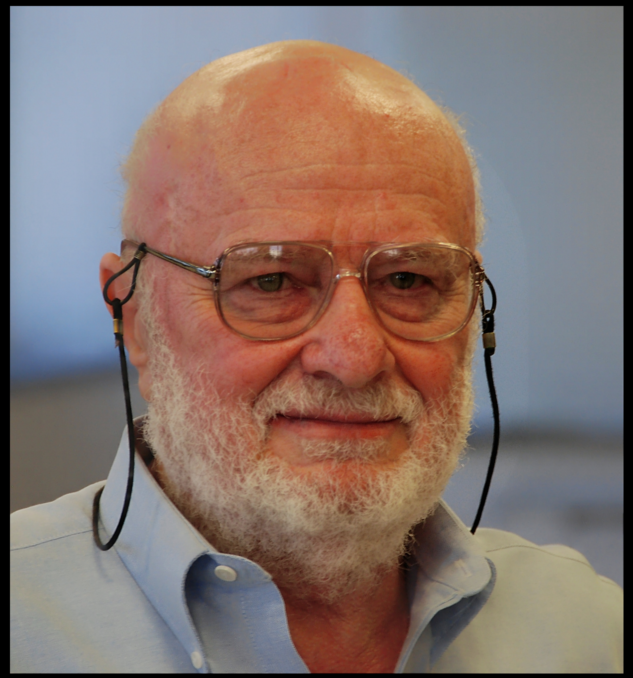
1. why bits?
   digital information age but not coincidentally or arbitrarily
   basics in this course but for more details check out EE 276

2. can I keep applying my compressor recursively and get a smaller file — ***entropy***

3. how much can I compress if I am ok to loose some information — ***rate-distortion***

There is something for everyone…

Elegant Algorithms



e.g.

1. existence of “universal” lossless data compression algorithms — ***LZ***

2. usage of transforms in multimedia compression — ***KLT, FFT, DCT***

3. how are numeral systems related to compression — ***AC, rANS, tANS***

There is something for everyone…

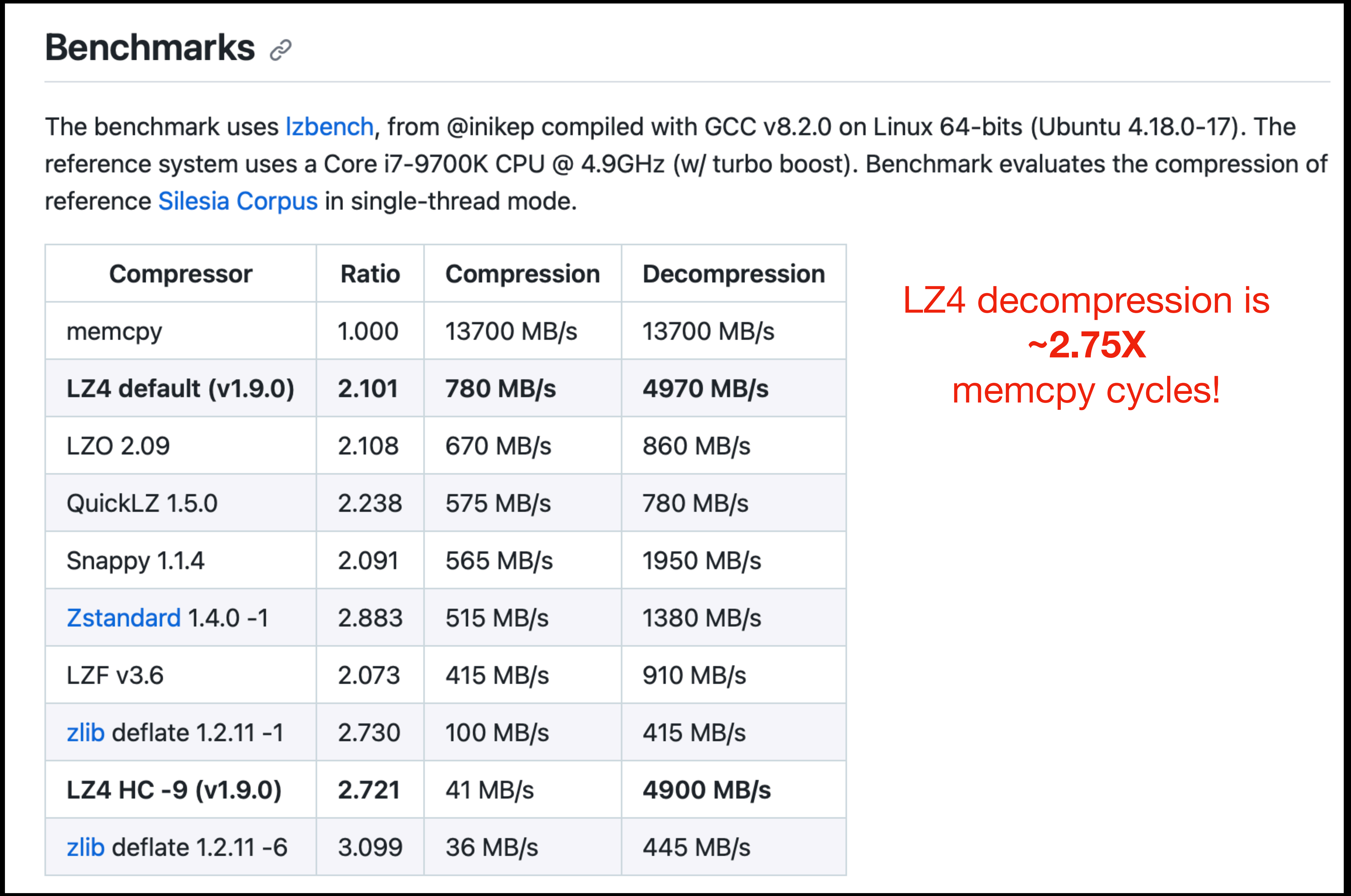
Clever Implementations

e.g.

1. can we “cache” computations

2. limitations on seeks during video streaming —
   existence of I/P/B frames

3. interesting data structures —
   suffix trees in ***BWT transform***



Benchmarks

The benchmark uses lzbench, from @inikep compiled with GCC v8.2.0 on Linux 64-bits (Ubuntu 4.18.0-17). The reference system uses a Core i7-9700K CPU @ 4.9GHz (w/ turbo boost). Benchmark evaluates the compression of reference Silesia Corpus in single-thread mode.

Compressor	Ratio	Compression	Decompression
memcpy	1.000	13700 MB/s	13700 MB/s
LZ4 default (v1.9.0)	**2.101**	**780 MB/s**	**4970 MB/s**
LZO 2.09	2.108	670 MB/s	860 MB/s
QuickLZ 1.5.0	2.238	575 MB/s	780 MB/s
Snappy 1.1.4	2.091	565 MB/s	1950 MB/s
Zstandard 1.4.0 -1	2.883	515 MB/s	1380 MB/s
LZF v3.6	2.073	415 MB/s	910 MB/s
zlib deflate 1.2.11 -1	2.730	100 MB/s	415 MB/s
LZ4 HC -9 (v1.9.0)	**2.721**	41 MB/s	**4900 MB/s**
zlib deflate 1.2.11 -6	3.099	36 MB/s	445 MB/s

LZ4 decompression is
~2.75X
memcpy cycles!

Source: https://github.com/lz4/lz4

There is something for everyone…

Beautiful **Theory**

Elegant **Algorithms**

Clever **Implementations**

and we will try to give you a flavor of all of these in the class.

In particular

First half — **Lossless** Compression

Second half — **Lossy** Compression

and throughout we will show you workings of various tools and code snippets!

Lossless Compression

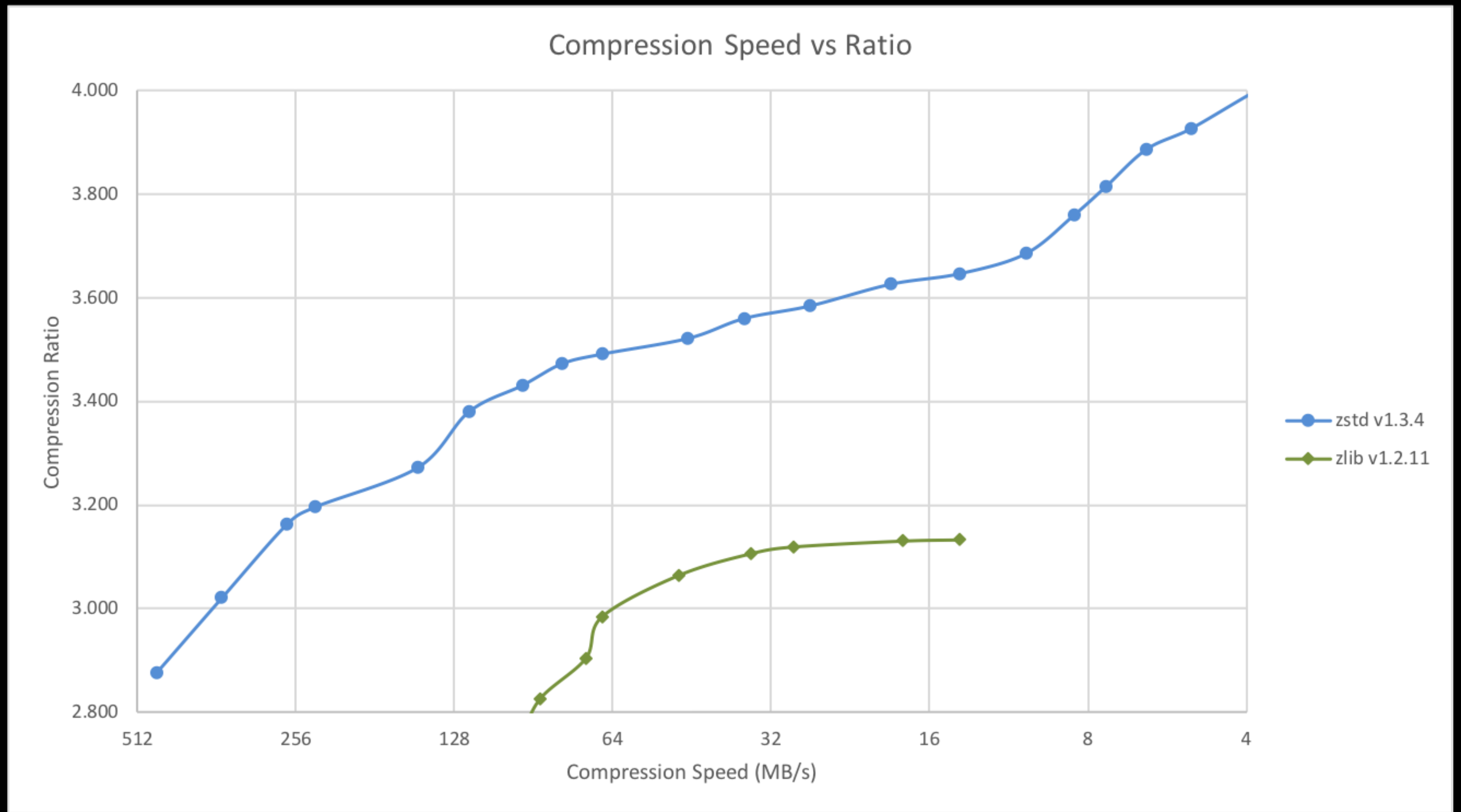
Entropy and its fundamental role in compression

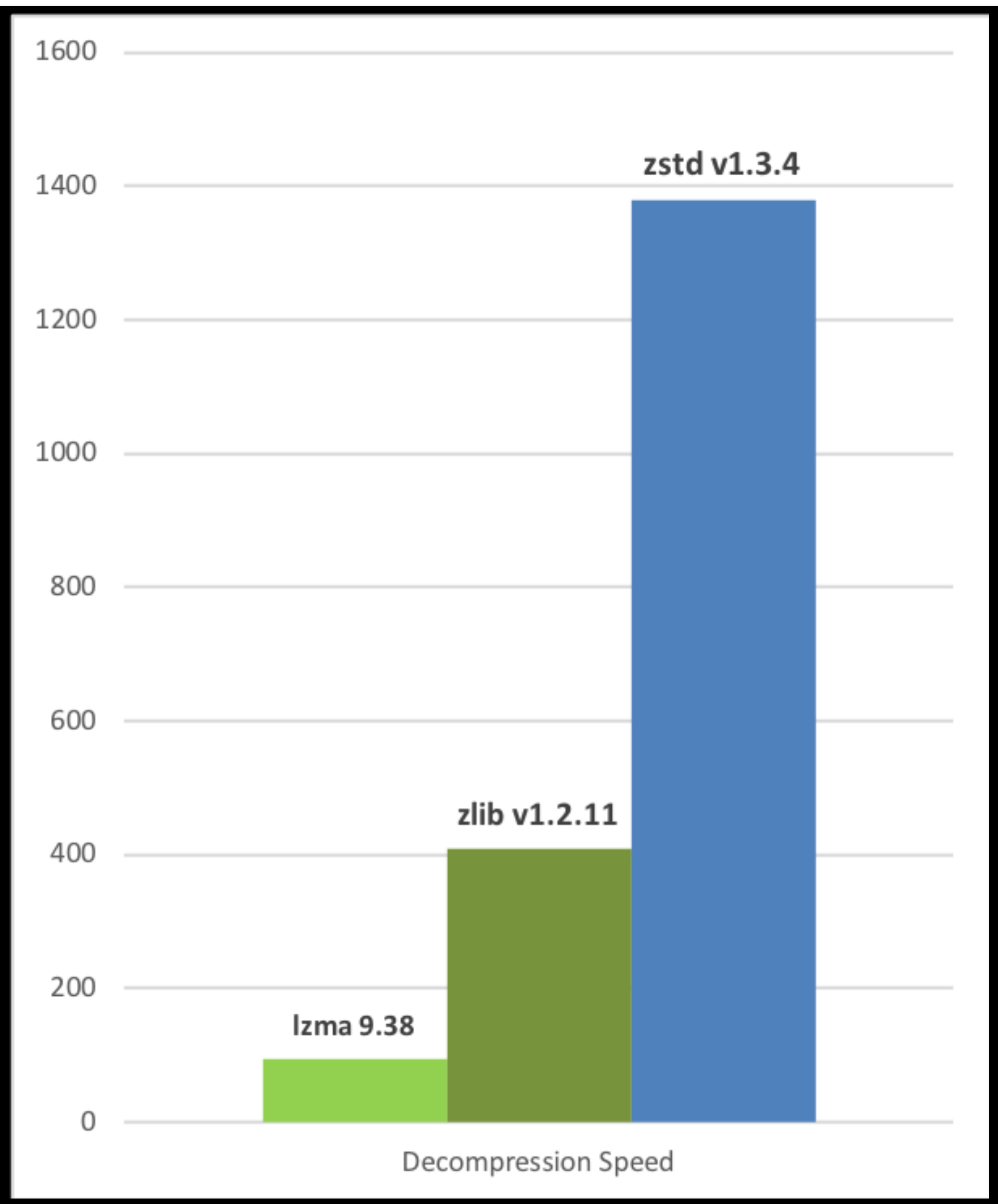
Lossless Compressors

- Huffman
- Arithmetic Coding (**AC**)
- Asymmetric Numeral Systems (**ANS**)
- Lempel-Ziv (**LZ 77/78**) Schemes, **gzip**

Handling correlated sources

- Adaptive Arithmetic Coding
- Universal Compression





Source: http://facebook.github.io/zstd/

Lossy Compression

Rate-Distortion Theory, Mutual Information

Quantization

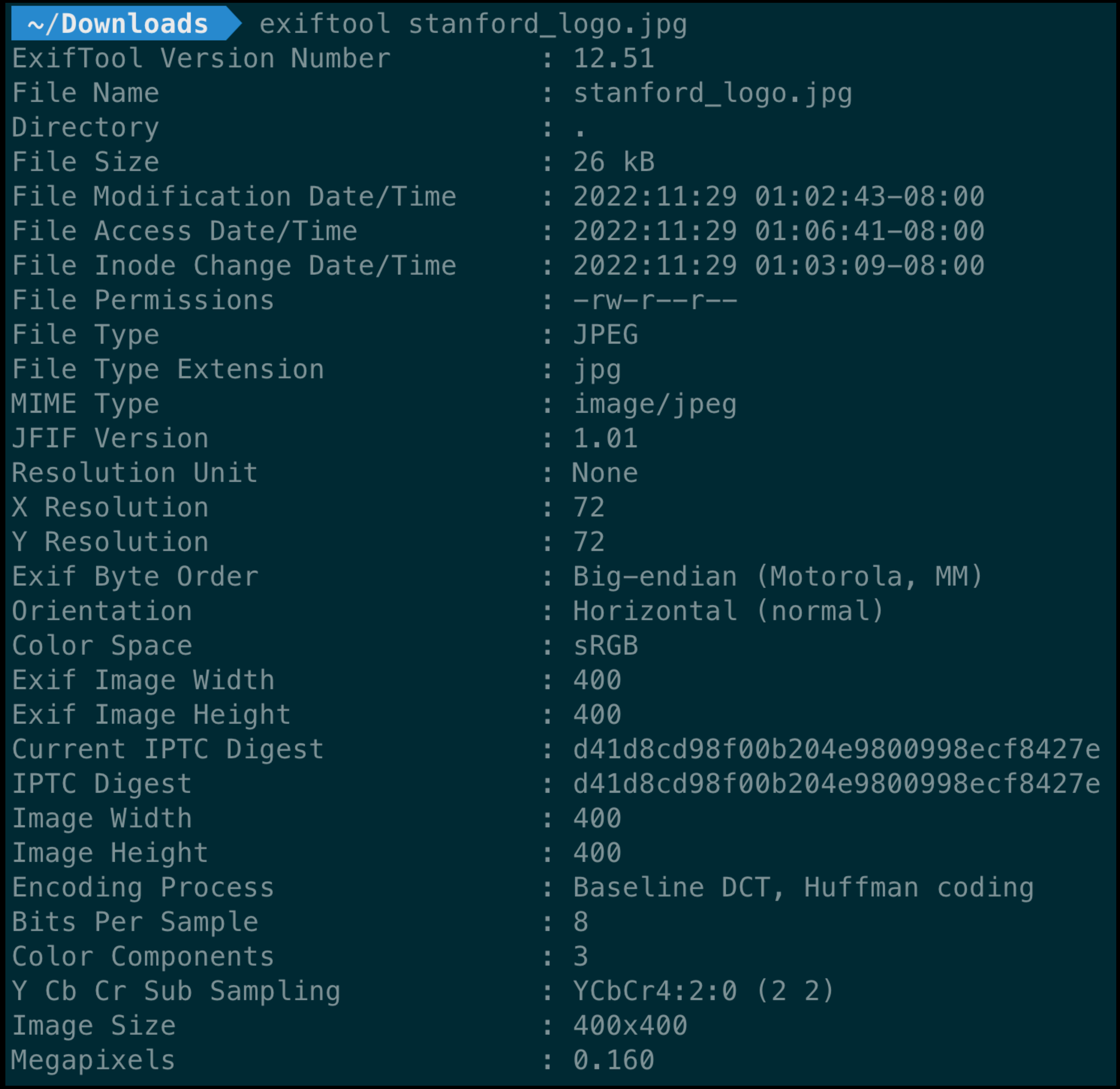
Transform Coding — KLT, FFT, DCT

Deep-dive into Images

- Traditional — **JPEG**, **BPG**
- ML + Compression

Video Compression

Role of human perception



```
 ~/Downloads  exiftool stanford_logo.jpg
ExifTool Version Number         : 12.51
File Name                       : stanford_logo.jpg
Directory                       : .
File Size                       : 26 kB
File Modification Date/Time     : 2022:11:29 01:02:43-08:00
File Access Date/Time           : 2022:11:29 01:06:41-08:00
File Inode Change Date/Time     : 2022:11:29 01:03:09-08:00
File Permissions                : -rw-r--r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Resolution Unit                 : None
X Resolution                    : 72
Y Resolution                    : 72
Exif Byte Order                 : Big-endian (Motorola, MM)
Orientation                     : Horizontal (normal)
Color Space                     : sRGB
Exif Image Width                : 400
Exif Image Height               : 400
Current IPTC Digest             : d41d8cd98f00b204e9800998ecf8427e
IPTC Digest                     : d41d8cd98f00b204e9800998ecf8427e
Image Width                     : 400
Image Height                    : 400
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:2:0 (2 2)
Image Size                      : 400x400
Megapixels                      : 0.160
```

Compression is a rich field: we will only touch a tip-of-the-iceberg

Successive refinement

Distributed compression

Compression and neural nets

Random access and succinct data structures

Lossy compression and denoising

Interplay between compression and inference

Compression for and using perceptual distortion metrics

Genomic data compression

Storage in DNA

Hardware decoders

RDP formulation……….

but check-out

IT Forum

Other classes such as
EE 276, 278, 261, 376 series

Course Elements

Lectures

Office Hours

Short Quizzes

Homeworks

Project

For CR/NC — see the grading slide

Prerequisites

We will not assume information theory background and re-introduce the relevant concepts

Knowledge of probability concepts at the level of EE 178 or equivalent

Programming at the level of CS 106B or equivalent (we will use **Python**)

Signal Processing background can be helpful, but again, not needed

Disclaimer:

First two weeks might be bit slow for some of you with Information Theory background but things will pick up pretty fast as we go-ahead.

Not sure? Reach out to us.

Useful Links

Course Webpage: https://stanforddatacompressionclass.github.io/Fall23/

check regularly — used for releasing lectures, HWs and project details

ED: https://edstem.org/us/courses/47776/discussion/

used for offline discussions and time-sensitive announcements

Gradescope: https://www.gradescope.com/courses/625620

used for quizzes, HW and project submissions

Staff mailing list: ee274-aut2324-staff@lists.stanford.edu

for admin questions, though a private ED post is encouraged for this

IT Forum: https://web.stanford.edu/group/it-forum/talks/

for talks which are complementary to lectures

Stanford Compression Library: https://github.com/kedartatwawadi/stanford_compression_library

used for many of the class examples + HWs
we will release a tutorial on SCL in week 2

Lectures and Office Hours

Bi-weekly, in-person, Mon & Wed 4:30-5:50pm at Shriram 104

Remote Attendees: Lectures will be recorded and released using Panopto Course Video tool in Canvas

> Note for on-campus students:
> not an invitation for students to miss classes
> we will keep it engaging and interactive — with perks to participation

(tentative) Lecture Plan available at the course website

Lecture Notes and Resources

> released via website — do bookmark and revisit as the material is under-preparation and will be changed during the offering

Mix of slides and white-board (used mostly for theoretical part of the course)

Office Hours timings

> available via website, learning from peers via ED participation is encouraged (and rewarded!)

Recording disclaimer

Video cameras located in the back of the room will capture the instructor presentations in this course. For your convenience, you can access these recordings by logging into the course Canvas site. These recordings might be reused in other Stanford courses, viewed by other Stanford students, faculty, or staff, or used for other education and research purposes. Note that while the cameras are positioned with the intention of recording only the instructor, occasionally a part of your image or voice might be incidentally captured. If you have questions, please contact a member of the teaching team.

(short) Quizzes

Available on Gradescope

Released after every lecture, due before next (4:30pm next lecture)

(a few) questions (mostly multiple-choice type) on basic concepts covered in that lecture

Shouldn’t take more than 15 mins ***if*** attentively attending the lecture

Homeworks

4 homeworks

Roughly covers 4 lectures each; released after the third one and due in two weeks

(tentative) HW release and due dates are available at the course website

Both theoretical/conceptual and programming problems

First HW will be released on 4th October (Wednesday) after third lecture

Project

Deeper dive into compression algorithm, techniques or analysis beyond the topics covered in class

Should include a significant implementation component and a writeup (blog/wiki/report)

    e.g. contribute to Stanford Compression Library!

Project Elements (tentative dates on class website)

- Proposal — around week 5
- Milestone — right after Thanksgiving break
- Presentation — during last lecture (make sure you attend this!)
- Final Report — due 15th Dec

More details, such as recommended projects and guidelines, will be released on the website
and you will be updated

Grading

Homeworks: 60% [4 x 15%]

Quizzes: 10% [18 x ~0.5%]

Final Project: 30% [5% + 5% + 5% + 15% for report]

Participation (bonus): 5%
[class participation, ED discussions, GitHub issues on notes, contributions to SCL beyond project, …]

For CR/NC, we require that you get 50% of the total grade.

Late Day Policy

Everyone is given **three late-days by default**, communicate with teaching-staff via ED to avail them
Note:
late-days ***will not apply*** to final project presentation and report! Make sure you have the times blocked in your calendar.

Honor Code

When in doubt follow the detailed Honor Code:
https://communitystandards.stanford.edu/policies-guidance/honor-code

[thumb rules]

You are permitted to seek assistance from peers, online resources, and your favorite LLM, etc.; however, it is imperative that you ***submit your own original work*** and *acknowledge all the external resources* consulted.

If ever in doubt, reach out to us.

We will regretfully, but strictly, follow honor code guidelines in case of any violations.

Questions?!

Have fun!

(and let us know if you are not)