Storage Systems (*what is up with Storage Systems and why should you care!*)

Animesh Trivedi
https://animeshtrivedi.github.io/

Briefly: Who am I?

https://animeshtrivedi.github.io/

Assistant Professor (tenure-track)

Research: *I like to build stuff*

- Storage, networking, distributed, operating systems
- Design, implement, and deploy code and measure

Teaching:

- Advanced Network Programming (XB_0048), **Storage Systems (XM_0092)**, Systems Seminar (X_405022)
- https://animeshtrivedi.github.io/teaching/ (*course material available openly*)

Outside computers:

- Cycling, running, reading non-fiction - *talk to me about existential dread ;)*

The goal of this talk is to …

- Tell you all the things that I find exciting - *luckily that is also my job :)*
  - Without pulling hairs and splitting electrons
- Get you excited about storage systems
  - Even if you are not interested "***in***" storage systems you should know *what it can offer to you for your data-driven workloads!*
  - If you ***are*** interested, just chat up with me after this talk
- Convince you it is not just about writing code
  - CS, AI, and Mathematics - we need help from all of you!
- Lastly, right now it is very fun and exciting time to do advanced studies
  - The field of computing is undergoing a fundamental shift - ***you can shape it***

When I say **Storage Systems...**

Storage System Course, https://animeshtrivedi.github.io/course-stosys/
https://studiegids.vu.nl/en/2021-2022/courses/XM_0092#/

Data, Data, and Data Everywhere

ING ABN·AMRO

Financial

Warehouse

Social

Health

Mobility

TESLA

WAYMO

Science

ASTRON

CERN

A minute on the Internet

200 Zettabytes

(by 2025)

200 x 1,000,000,000,000,000,000,000

thousand
million
billion
trillion
quadrillion
quintillion
sextillion

https://new.in-24.com/News/120299.html
https://cybersecurityventures.com/the-world-will-store-200-zettabytes-of-data-by-2025/

If that Zettabytes does not resonate

Assume:

- 1 grain of rice is 1 byte of data

If that Zettabytes does not resonate

Assume:

- 1 grain of rice is 1 byte of data
- **Kilobyte**: a cup of rice

If that Zettabytes does not resonate

Assume:

- 1 grain of rice is 1 byte of data
- **Kilobyte**: a cup of rice
- **Megabyte**: 8 bags of rice

If that Zettabytes does not resonate

Assume:

- 1 grain of rice is 1 byte of data
- **Kilobyte**: a cup of rice
- **Megabyte**: 8 bags of rice
- **Gigabyte**: 3 semi-truck

If that Zettabytes does not resonate

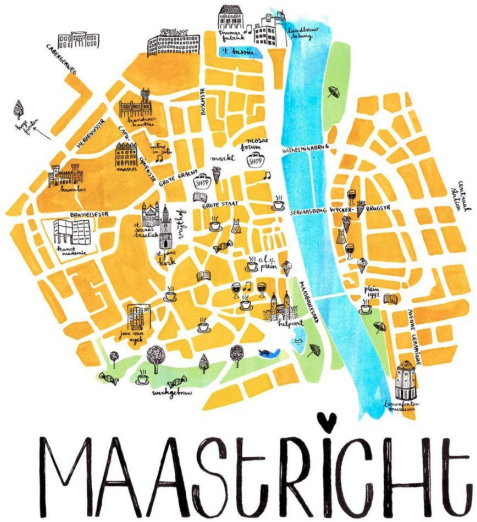
Assume:

- 1 grain of rice is 1 byte of data
- **Kilobyte**: a cup of rice
- **Megabyte**: 8 bags of rice
- **Gigabyte**: 3 semi-truck
- **Terabyte**: 2 container ships

If that Zettabytes does not resonate

Assume:

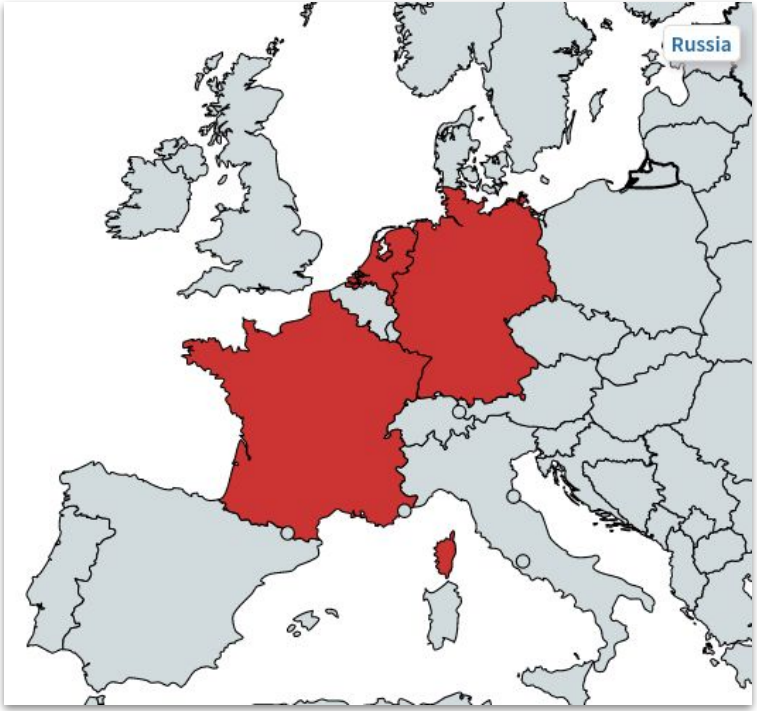
- 1 grain of rice is 1 byte of data
- **Kilobyte**: a cup of rice
- **Megabyte**: 8 bags of rice
- **Gigabyte**: 3 semi-truck
- **Terabyte**: 2 container ships
- **Petabyte**: Covers Maastricht



If that Zettabytes does not resonate

Assume:

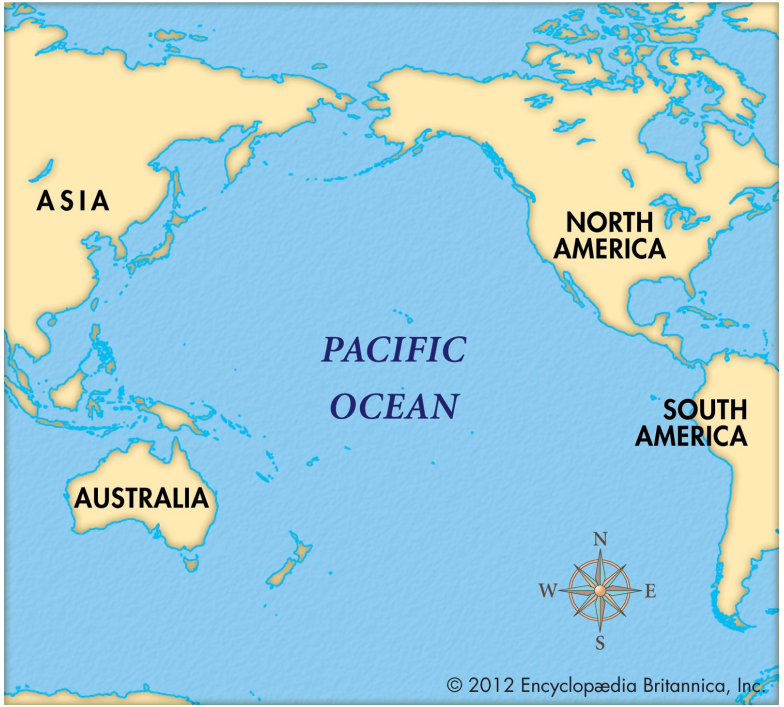
- 1 grain of rice is 1 byte of data
- **Kilobyte**: a cup of rice
- **Megabyte**: 8 bags of rice
- **Gigabyte**: 3 semi-truck
- **Terabyte**: 2 container ships
- **Petabyte**: Covers Maastricht
- **Exabyte**: Covers NL + DE + FR



If that Zettabytes does not resonate

Assume:

- 1 grain of rice is 1 byte of data
- **Kilobyte**: a cup of rice
- **Megabyte**: 8 bags of rice
- **Gigabyte**: 3 semi-truck
- **Terabyte**: 2 container ships
- **Petabyte**: Covers Maastricht
- **Exabyte**: Covers NL + DE + FR
- **Zettabyte**: Fills up the Pacific Ocean



If that Zettabytes does not resonate

Assume:

- 1 grain of rice is 1 byte of data
- **Kilobyte**: a cup of rice
- **Megabyte**: 8 bags of rice
- **Gigabyte**: 3 semi-truck
- **Terabyte**: 2 container ships
- **Petabyte**: Covers Maastricht
- **Exabyte**: Covers NL + DE + FR
- **Zettabyte**: Fills up the Pacific Ocean
- **Yottabytes**: An Earth size rice ball

We are here

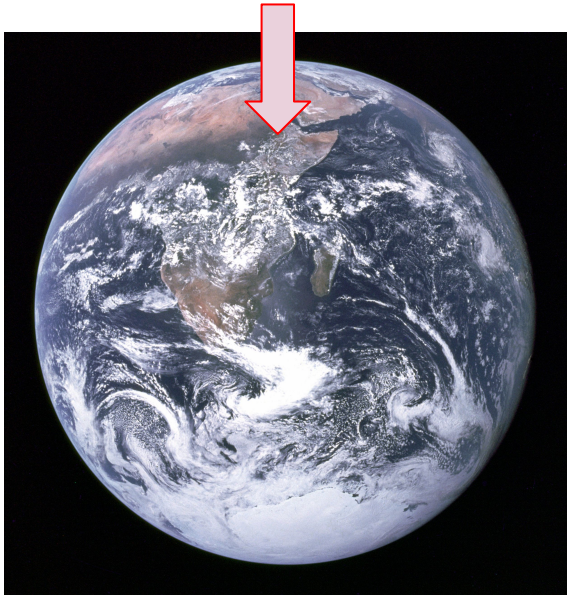
If that Zettabytes does not resonate

Assume:

- 1 grain of rice is 1 byte of data
- **Kilobyte**: a cup of rice
- **Megabyte**: 8 bags of rice
- **Gigabyte**: 3 semi-truck
- **Terabyte**: 2 container ships
- **Petabyte**: Covers Maastricht
- **Exabyte**: Covers NL + DE + FR
- **Zettabyte**: Fills up the Pacific Ocean
- **Yottabytes**: An Earth size rice ball

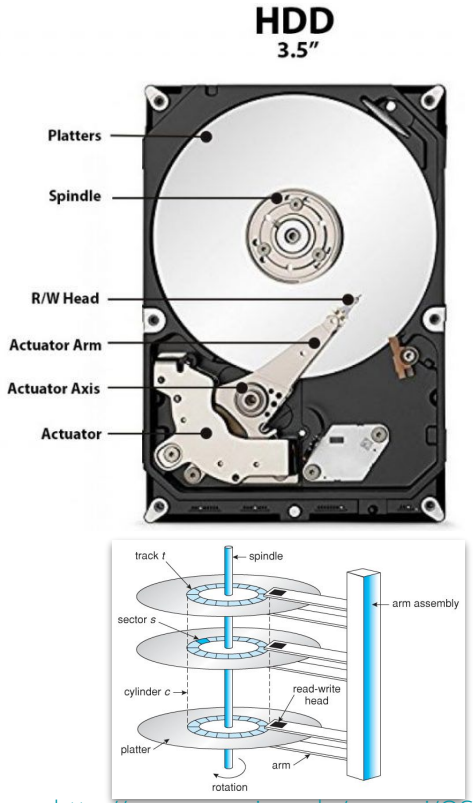
We are here

"Find me an image of a pink cat"



https://www.pinterest.com/LauralikesChai/pink-cat/

How do we store and process this data?



Invented by IBM in **1956**

The primary technology to store data

	Cheetah 15K.5
Capacity	300 GB
RPM	15,000
Average Seek	4 ms
Max Transfer	125 MB/s
Platters	4
Cache	16 MB
Connects via	SCSI

Lots of work and research has been done on it to ...

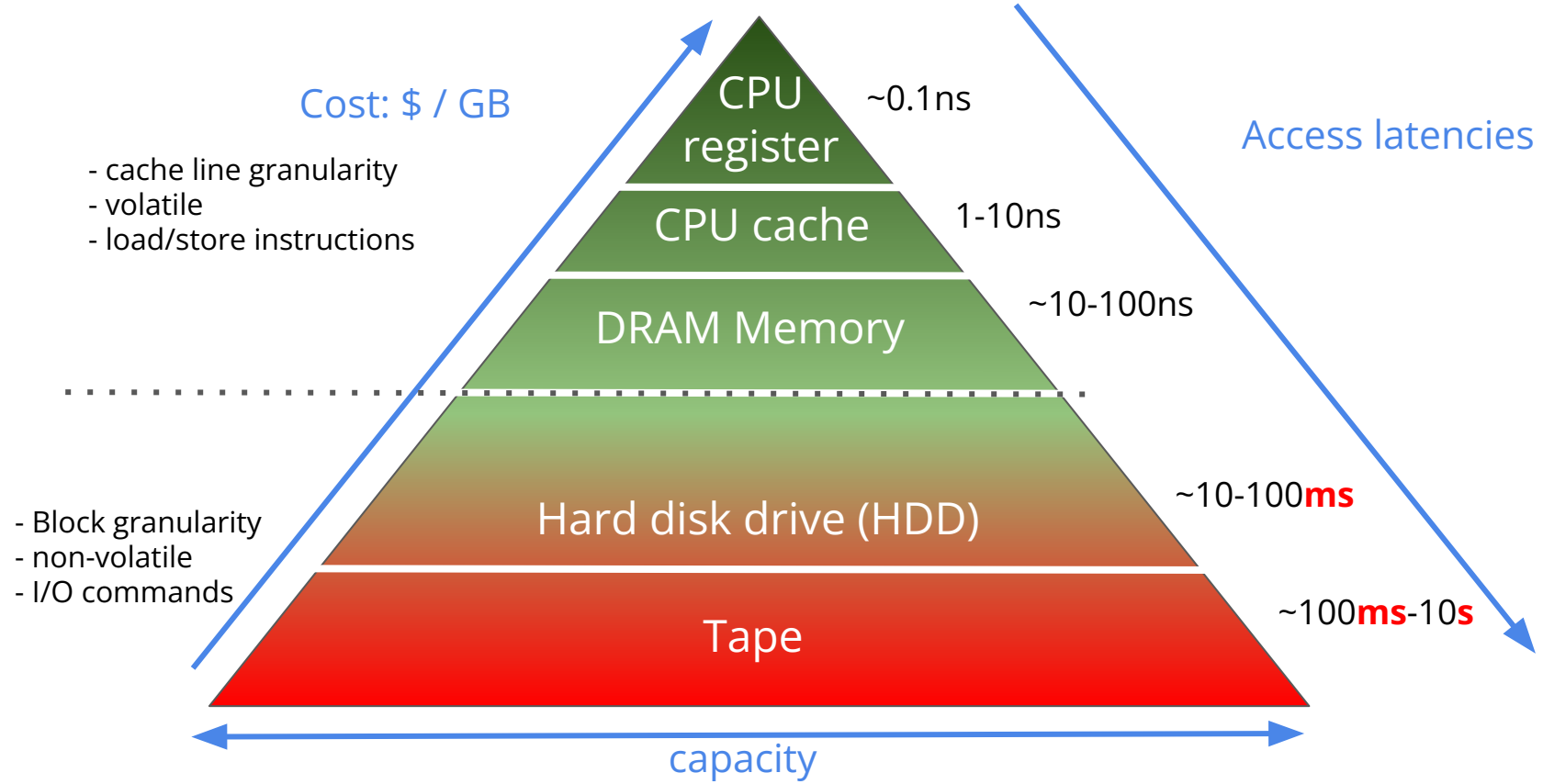
- **Store data** in a single machine
- Store data **reliably** in a single machine
- Store data reliably with **high-performance** in a single machine
- Store data on distributed machines
- Store data **reliably** on distributed machines
- Store data reliably with **high-performance** on distributed machines

http://pages.cs.wisc.edu/~remzi/OSTEP/file-disks.pdf
https://www.partitionwizard.com/help/what-is-chs.html

HDD Design Mantras - Since the 1960s

1. I/O read/write operations happens in a sector granularity (512 bytes)
2. Read and write are symmetric - they both have the same performance
3. Random performance (both read/write) are worse than the sequential performance
4. Small I/O performance is bad - cannot amortize the seek and rotational time
5. [Not so well known] Outer vs inner track performance
    a. Outer tracks rotate with a faster linear speed than Inner tracks
    b. Hence, faster bandwidth (30+%) and lower latencies
    c. See section 4.3 http://cseweb.ucsd.edu/~gmporter/papers/tritonsort-nsdi11.pdf
6. (Theoretically) Infinite durability - magnetic field can be held indefinitely
    a. Disregarding heating, damages, demagnetization, etc.

The triangle of storage hierarchy



Latencies in perspective

Numbers every programmer should know

While the other components improved significantly since the 1960s …

Storage has been the Achilles' heel

Incredible amount of work done to hide these latencies

https://gist.github.com/hellerbarde/2843375

Minute:

L1 cache reference	0.5 s	One heart beat (0.5 s)
Branch mispredict	5 s	Yawn
L2 cache reference	7 s	Long yawn
Mutex lock/unlock	25 s	Making a coffee

Hour:

Main memory reference	100 s	Brushing your teeth
Compress 1K bytes with Zippy	50 min	One episode of a TV show (including ad breaks)

Day:

Send 2K bytes over 1 Gbps network	5.5 hr	From lunch to end of work day

Week

SSD random read	1.7 days	A normal weekend
Read 1 MB sequentially from memory	2.9 days	A long weekend
Round trip within same datacenter	5.8 days	A medium vacation
Read 1 MB sequentially from SSD	11.6 days	Waiting for almost 2 weeks for a delivery

Year

Disk seek	16.5 weeks	A semester in university
Read 1 MB sequentially from disk	7.8 months	Almost producing a new human being
The above 2 together	1 year	

Decade

Send packet CA->Netherlands->CA	4.8 years	Average time it takes to complete a bachelor's degree

Enter: USB Drive *(what is inside a USB drive)*

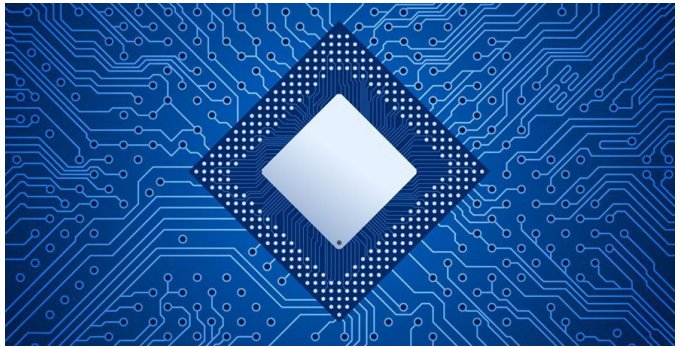
https://www.quill.com/nxt-technologies-32gb-usb-30-flash-drive-nx27996/cbs/55362528.html

Non-Volatile Memory (NVM) Technology

Umbrella term for many related technologies (Solid-State Storage, Storage-Class Memory - *we are rubbish with names :P*)

Overall the idea is to use physical properties of media store bits and data

- Magnetic state
- **Electrical charge** ←
- Crystalline / amorphous state
- Resistance
- Optical



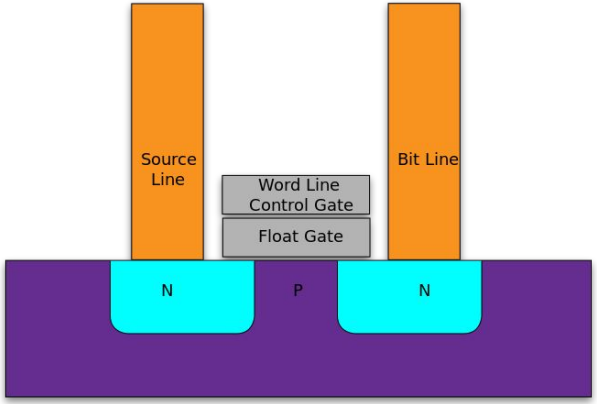
No moving parts !

https://www.rambus.com/blogs/minding-the-memory-gap-2/

Flash Memory

Invented by Toshiba

- Since mid-1980s commercially available
- The basic idea of a floating gate:
  1960s with the MOSFET technology



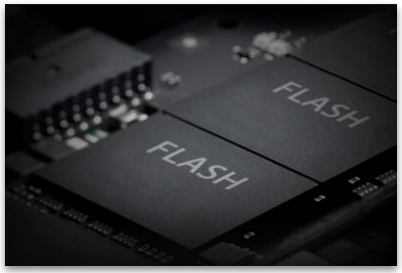
Has been around for more than 30 years

First became popular in with embedded devices

In the mainstream computing from the mid-2000s

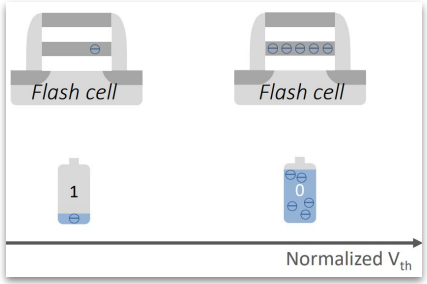
One of the most popular technologies to store data today

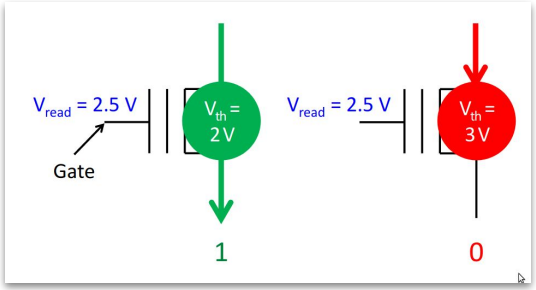
A very fundamental shift how to store and manage data

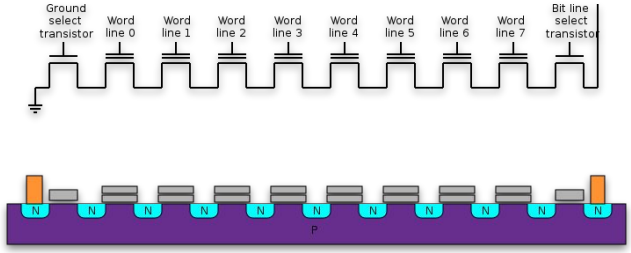


https://www.eweek.com/storage/why-flash-storage-will-remain-dominant-for-several-years
https://en.wikipedia.org/wiki/File:Flash_cell_structure.svg

How does it work?





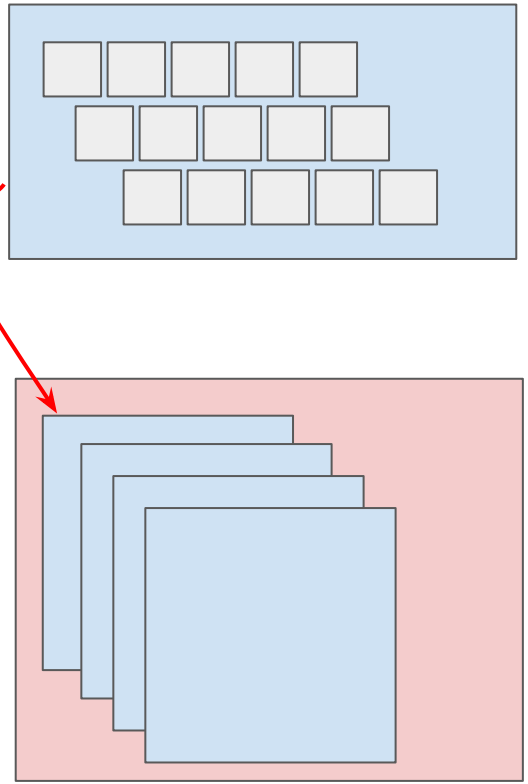


Trapping electrons in the floating gates to either let the current through or not

- You cannot overwrite (you cannot trap more electrons in trapped electrons)
- A FG must be re-programmed / erase (P/E) continuously, leading to an erosion of the the isolated oxide layer - **finite P/E life cycle**
- Multiple reads over the time can also erode the charge : **read disturbance**

For more about the physics of flash cells: https://safari.ethz.ch/architecture/fall2018/lib/exe/fetch.php?media=onur-comparch-fall2018-lecture14b-flashmemory-afterlecture.pdf
https://www.youtube.com/watch?v=XbKOmOPjLtc

Making flash pages and blocks from flash cells



A bunch of flash cells are packed in a **page** : typically 4kB

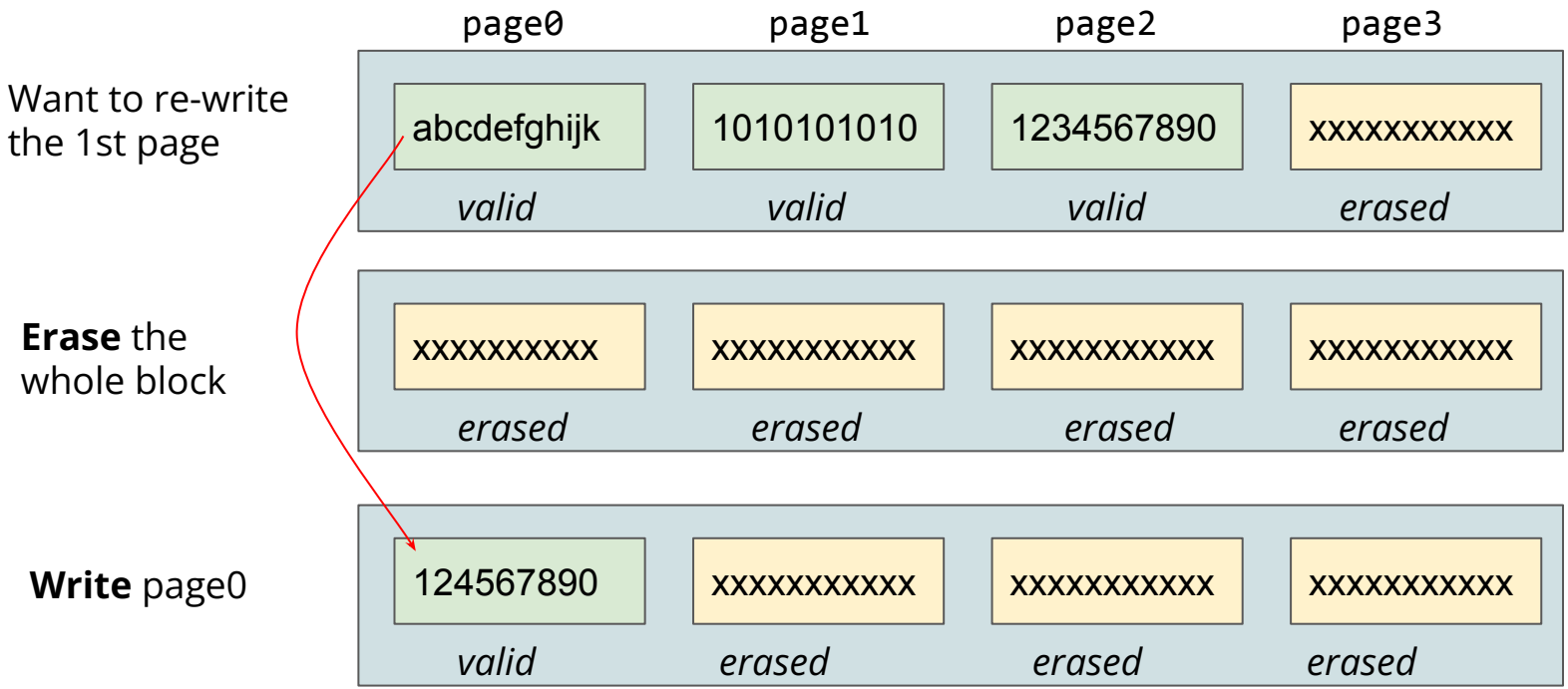
- Typically, a unit of I/O - the page (similar to a sector size in HDD)

A bunch of flash pages packed together as a **block**: typically 64-128-256 pages (hence, a few Megabytes)

- Typically, a unit of erase (bulk erasure)
- Recall: you need to erase before you can write on a flash cell

Lots of parallelism available inside the device - we should use it :)

Write - Erase Operation in Detail



the content of page1 and page2 are lost, hence, if you want to save them then you have to copy out the whole block

Typical Values of Operations

	Read (*usec*)	**Write (*usec*)**	**Erase (*usec*)**	**P/E cycles**
SLC	25	200-300	1,500-2,000	~100,000
MLC	50	~600-900	~3,000	~10,000
TLC	75	~900-1,350	~4,500	~5,000

Performance might depend on:

- The technical generation of the device
- What cell technology it is using - SLC, MLC or TLC
- *How much data you have written to it (?, yes you read it right)*
- How old the file system is
- and a few more parameters...

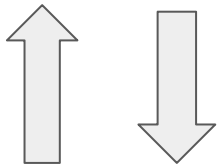
https://www.enterprisestorageforum.com/storage-hardware/slc-vs-mlc-vs-tlc-nand-flash.html
http://pages.cs.wisc.edu/~remzi/OSTEP/file-ssd.pdf

So far: Flash vs HDD (at the cell level)

1. *No moving parts* inside a flash storage
2. Flash read latencies are much lower than HDDs **(10s of msecs vs 10s of usecs)**
3. *Asymmetric* read and write performances, 10 (r) vs 100 (w) useconds
    a. HDD have the same
4. *No overwriting (in-place update)* the same page again (HDD can)
5. *[new operation]* **Erase operation**, must erase a whole block before writing
6. *[new activity]* If there are valid data pages, must copy data before erase
7. *Finite number of times* one can erase and program flash cells
    a. Nonetheless, like with any electron based storage, they leak and data will be eventually lost
    b. HDDs have (theoretically) infinite durability
8. Read performance is uniform - no difference with random or sequential ***at the cell level***
9. Large amount of parallelism *(100s of requests vs 10s of requests)*
10. Also - no mechanical parts, hence, better **energy efficiency and packaging density**

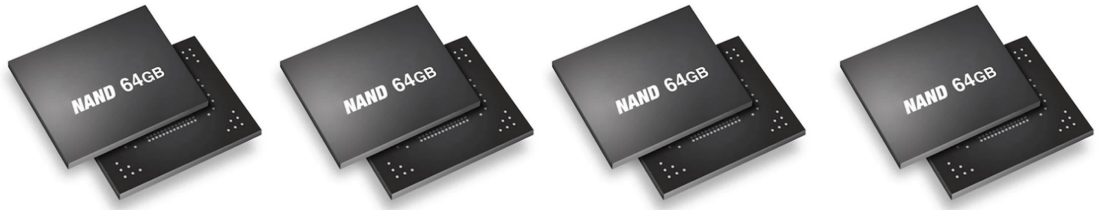
Flash Translation Layer - **The Secret Sauce**

Reads writes from the software (OS, application, fs)



FTL

Address Translation	Garbage Collection	Performance, Parallelism
Wear leveling, Over-provisioning	Free blocks mng.	Dead blocks and Error Corrections

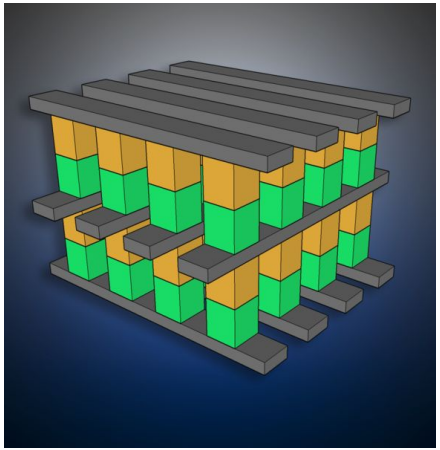


FTL is the secret sauce

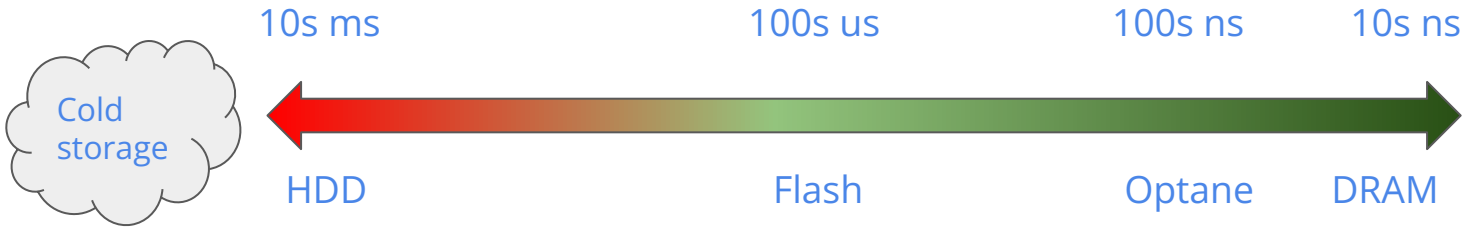
*There is **programmability** and **intelligence** in flash devices*

https://www.sourcetoday.com/news/article/21865991/nand-flash-memory-market-decelerates
https://veritusgroup.com/there-is-no-secret-sauce-build-a-relationship-see-growth/

Optane: A new class of NVM technology



- Jointly developed by Intel and Micron
- Uses a new type of material
  - Not publicly disclosed - but thought to be a resistive bulk material
- Much faster than flash, and can be packaged as byte-addressable memory
  - Latencies in 100s of nanoseconds, Bandwidths in 10s of GBytes/sec



Optane: Memory and storage form factors

Today available in storage as well as memory form factor

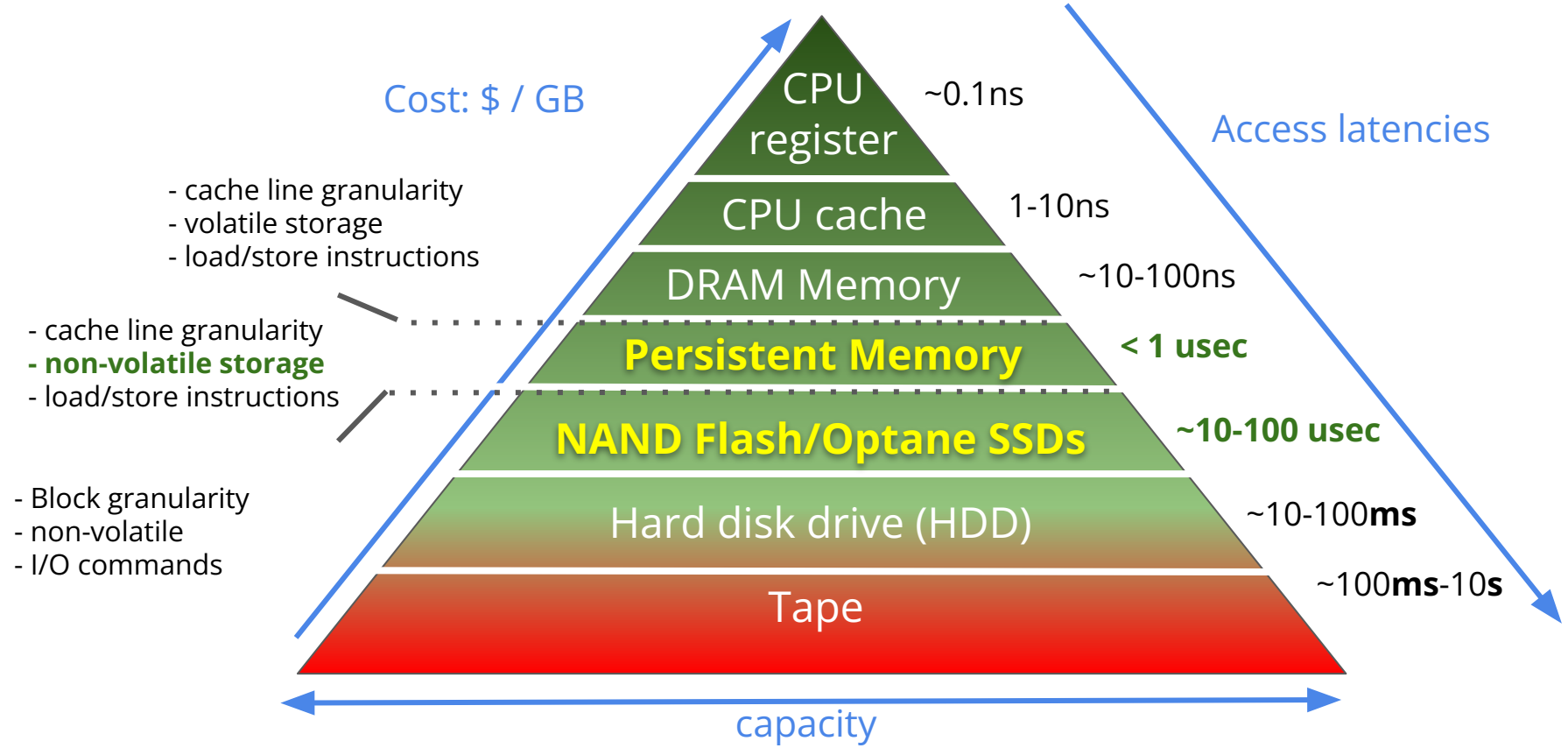
- Optane DIMMs (memory/cache) : 128GB, 256GB and 512GB DIMMs
  - Do direct load/store from the CPU to persistent DRAM
  - Denser and more energy efficient than DRAM

- Optane SSDs (as storage `/dev/optane`)
  - Make a file system, and do file I/O, mmap, read, and write

What does rebooting mean?

https://venturebeat.com/2019/11/21/optane-101-memory-or-storage-yes/

The (new) triangle of storage hierarchy



ACM Turing Award 2017

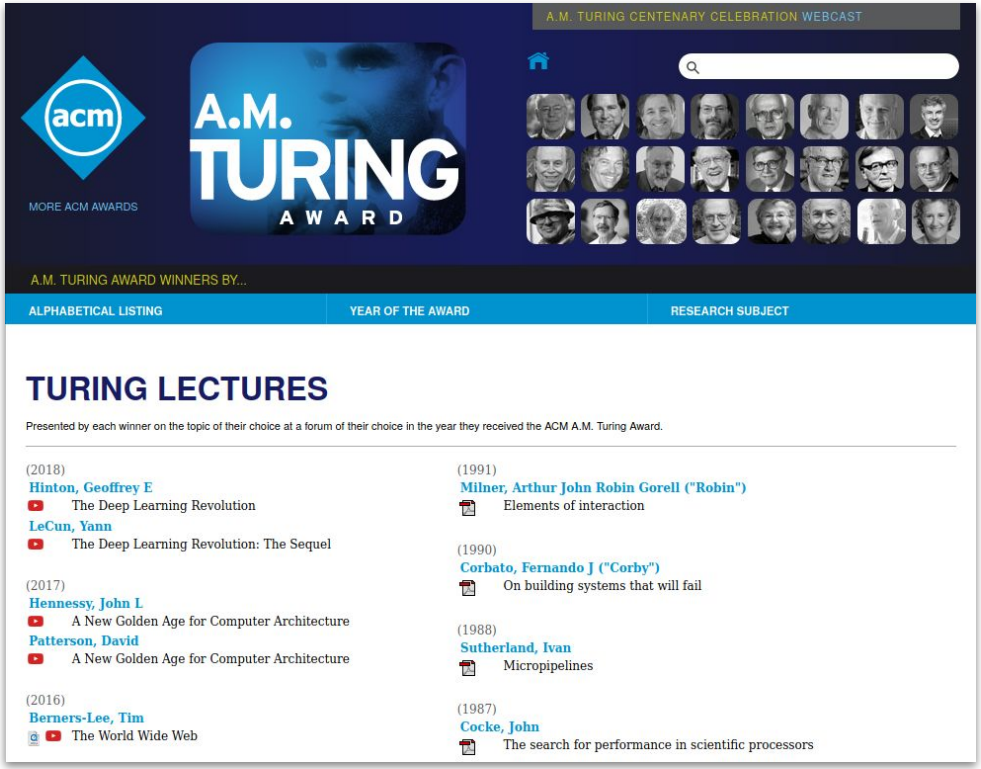
John Hennessy and David Patterson Deliver Turing Lecture at ISCA 2018

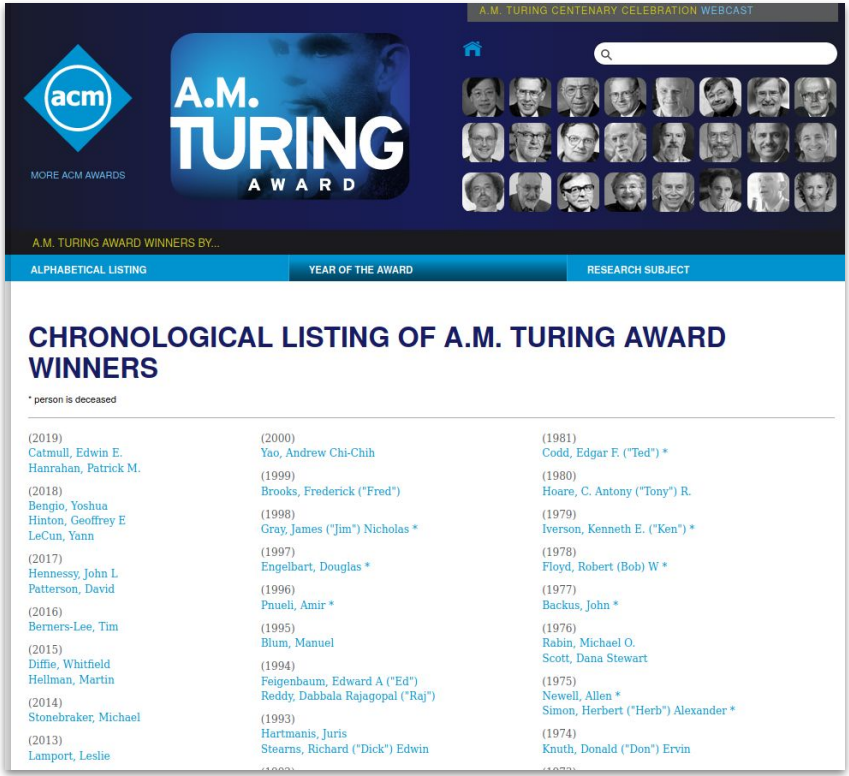
2017 ACM A.M. Turing Award recipients John Hennessy and David Patterson delivered the Turing Lecture on June 4 at ISCA 2018 in Los Angeles. The lecture took place from 5 to 6 p.m. PDT and was open to the public. A video of the lecture can be viewed below.

Titled "A New Golden Age for Computer Architecture: Domain-Specific Hardware/Software Co-Design, Enhanced Security, Open Instruction Sets, and Agile Chip Development," the talk covers recent developments and future directions in computer architecture.

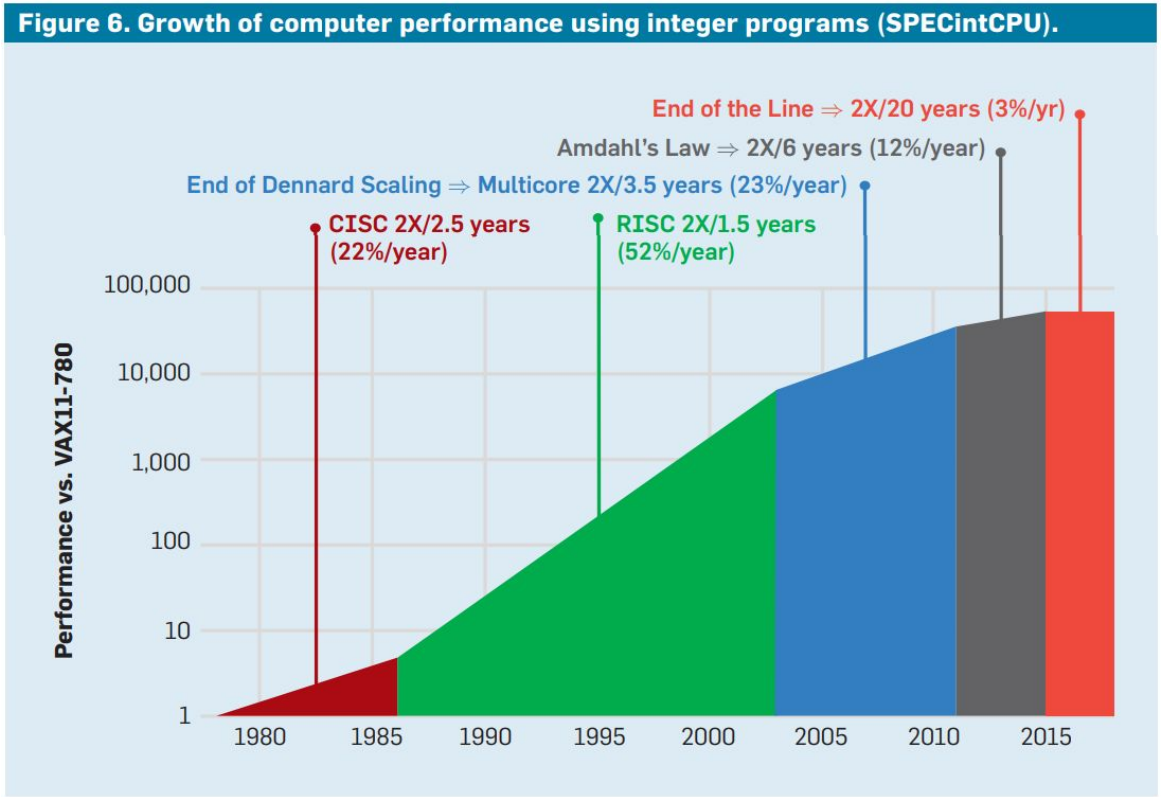
Hennessy and Patterson were recognized with the Turing Award for "pioneering a systematic, quantitative approach to the design and evaluation of computer architectures with enduring impact on the microprocessor industry."

Watch and read them all ;) (really you should)



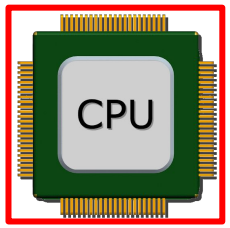


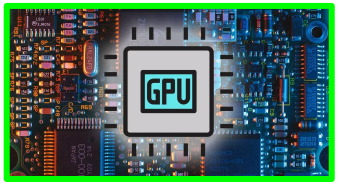
CPU Performance Stalling

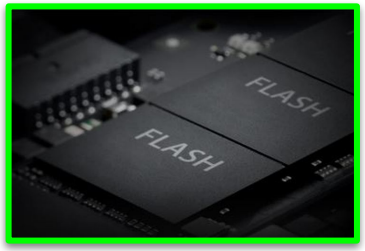


- Stalled **frequency** scaling
- Stalled **core** scaling
- Limited TDP
- CMOS scaling (7- nm)
- Memory wall

The Bigger Context







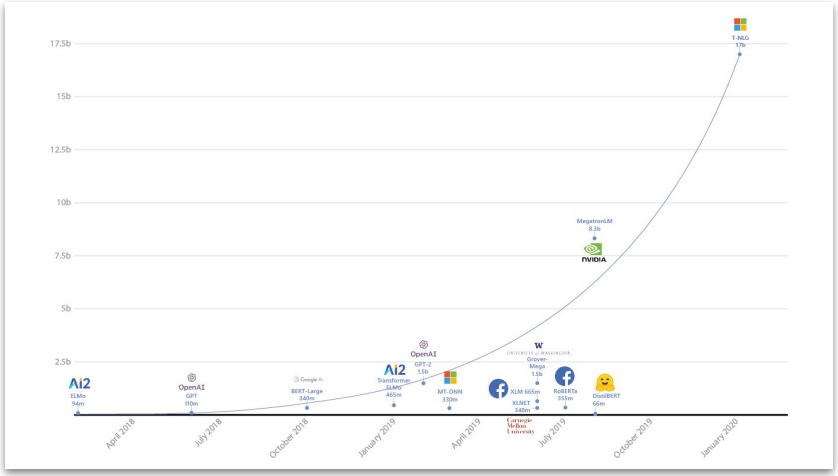
- **The compute accelerators are mainstream : GPUs, FPGA, DSAs**
  - 100x performance improvements
- **Network is improving** - we have 200 Gigabits/s links now, and Terabit/s in works
  - Expected 10-100x performance improvements
- **NVM storage is improving** - 10+ Gigabytes/s with single digit microsecond latencies
  - Expected 10-100x performance improvements
- **The CPU performance is stalled**

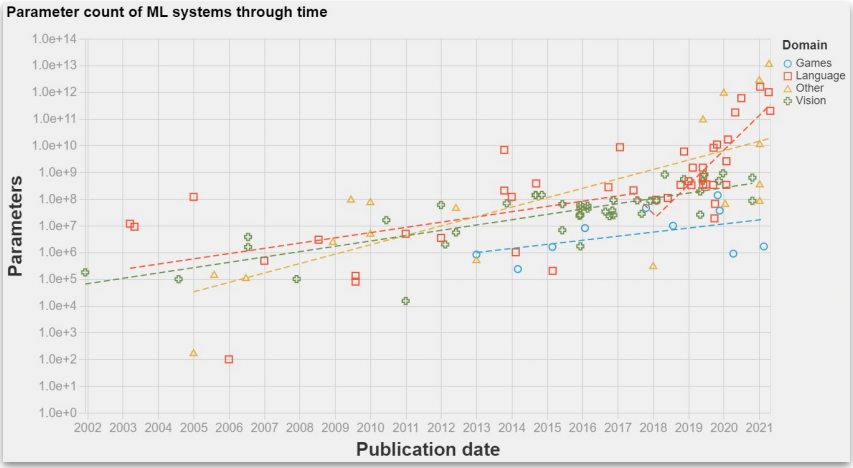
Example: *The work that the CPU was doing for HDD (in ~msecs)*

(a) Now must be done in ~usec! ⇒ ***Efficiency (new implementations and data structures)***

(b) Cannot be done at all ⇒ ***Specialization : for computation, for network, and for storage!***

1/3: Implications for AI/ML





How much DRAM? The training for a 40.8 MBytes of a 1001-layer ResNet model can take up to 67.2 GBytes (1000x). How much memory do you need for 1 Trillion parameter model?
I am sure many times you run out of memory even for current models on the GPU?

Parameter counts in Machine Learning, https://www.lesswrong.com/posts/GzoWcYibWYwJva8aL/parameter-counts-in-machine-learning
Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, the World's Largest and Most Powerful Generative Language Model, https://www.microsoft.com/en-us/research/blog/using-deepspeed-and-megatron-to-train-megatron-turing-nlg-530b-the-worlds-largest-and-most-powerful-generative-language-model/
Turing-NLG: A 17-billion-parameter language model by Microsoft, https://www.microsoft.com/en-us/research/blog/turing-nlg-a-17-billion-parameter-language-model-by-microsoft/

WILL KNIGHT BUSINESS 01.21.2020 07:00 AM

AI Can Do Great Things—if It Doesn't Burn the Planet

The computing power required for AI landmarks, such as recognizing images and defeating humans at Go, increased 300,000-fold from 2012 to 2018.

LAST MONTH, RESEARCHERS at OpenAI in San Francisco revealed an algorithm capable of learning, through trial and error, how to manipulate the pieces of a Rubik's Cube using a robotic hand. It was a remarkable research feat, but it required more than 1,000 desktop computers plus a dozen machines running specialized graphics chips crunching intensive calculations for several months.

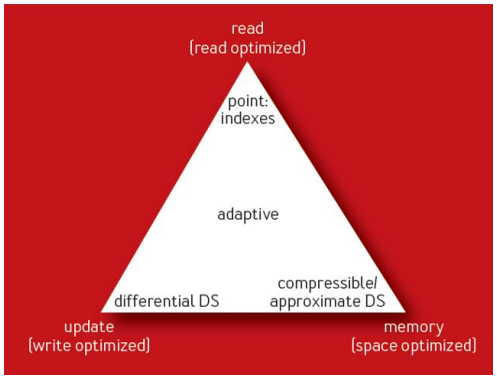
The effort may have consumed about 2.8 gigawatt-hours of electricity, estimates Evan Sparks, CEO of Determined AI, a startup that provides software to help companies manage AI projects. That's roughly equal to the output of three nuclear power plants for an hour. A

*This is not to say that AI/ML is bad, but to point out that we have an opportunity to **smartly design storage and memory systems** to deliver efficiency*

- *Scale*
- *Cost*
- *Energy and Sustainability*

2/3: Implications for Data Structures

FIGURE 9: **THE RUM CONJECTURE**



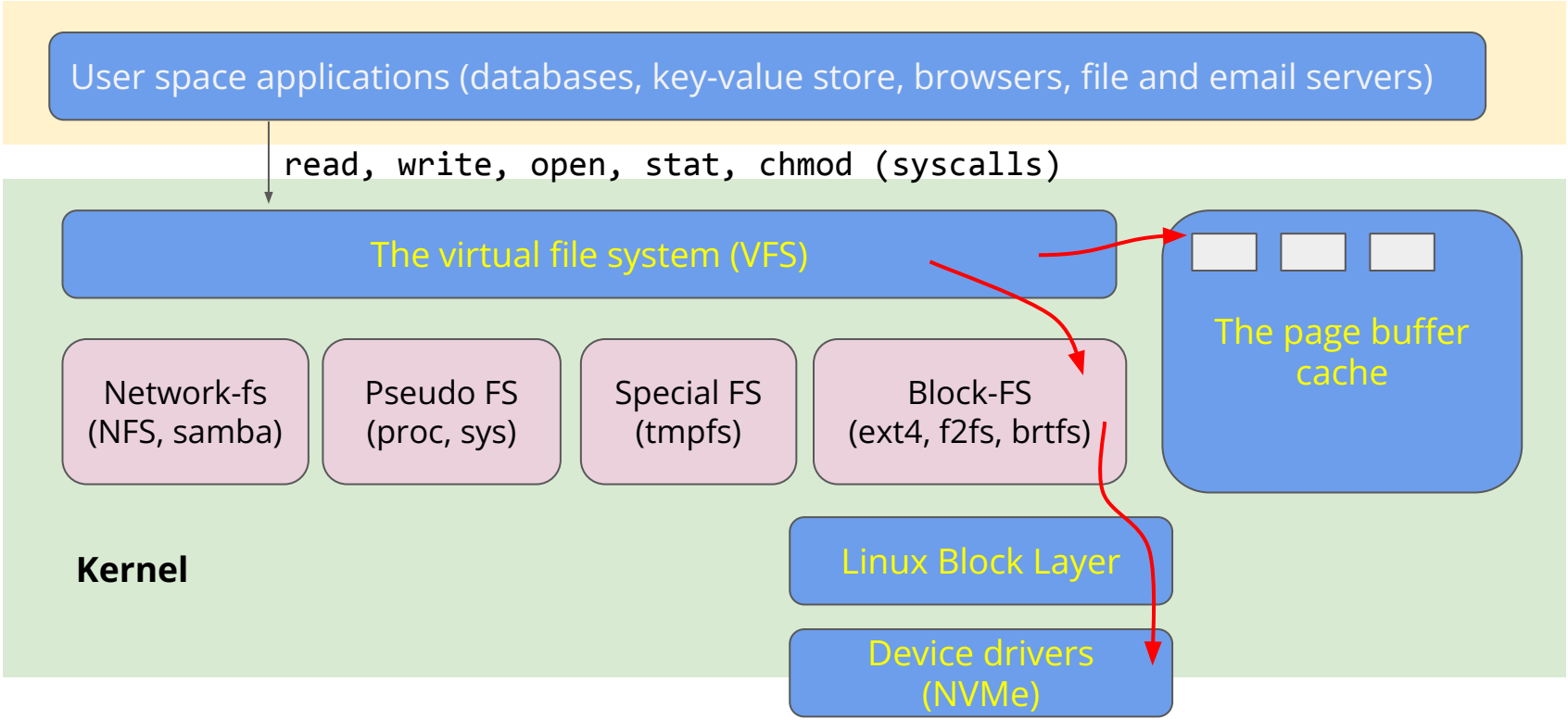
The RUM Conjecture: Need for efficient data-structure designs

- Read-heavy, write-heavy, mixed, range scans, concurrency, batch operations
- Modeling, statistics, and analysis

Index	Search	Insertion | deletion	Space	Experimental evaluation
SCATTERED LOGGING				
BFTL [149, 151]	$h * c$	$2(\frac{1}{M-1} + \tilde{N}_{\text{split}} + \tilde{N}_{\text{merge/rotate}})$	$n * c + B$	B-tree
WOBF [51]	–	–	–	BFTL, IBSF
SCATTERED LOGGING & NODE MODIFICATION				
FlashB-tree [73]	$(h, h * c)$	$\frac{1}{n_{in}}(\frac{2}{3} * \tilde{N}_{\text{split}})$	$n + B$	BFTL, IBSF
NODE MODIFICATION				
uB+tree [141]	–	–	–	B+tree
BF-tree [7]	$h + \lceil p_{fp} * n_{pl} \rceil_{sr}$	–	$n * n_{pl}$	B+tree, FD-tree, hashing
IN MEMORY BUFFERING				
IBSF [88]	h	$\frac{1}{n_{in}}(\tilde{N}_{\text{split}} + \tilde{N}_{\text{merge/rotate}})$	$n + B$	BFTL
RBFTL [152]	–	–	–	B-tree
LU B+tree [116]	–	–	–	B+tree
TNC [59]	–	–	–	–
AS B-tree [123]	–	–	–	B+tree, BFTL, LA-tree
FLASH BUFFERING				
FD-tree [98, 99]	$\log_k n$	$\lceil \frac{k}{f-k} \log_k n \rceil_{srw}$	$c * n$	B+tree, BFTL, LSM-tree
FD+tree, FD+FC [139]	$\log_\gamma \frac{n}{\kappa\rho\beta}$	$\lceil \frac{\gamma}{\beta-\gamma} \log_\gamma \frac{n}{\kappa\rho\beta} \rceil_{srw}$	$c * n$	FD+XM, FB+DS [139]
BSMVBT [34]	–	–	–	TMVBT [57]
FLASH BUFFERING & NODE MODIFICATION				
AB-tree [64]	$\sum_{l=1}^{h} \frac{M^{l-1}}{N_l} l$	h/s_n	n	B+tree, BFTL, FD-tree
WPCB-tree [61]	h	$\lceil 1 \rceil_{sw} + 3 * n_{sp} + \lceil n_b \rceil_{bm}$ $\lceil 1 \rceil_{sw} + \lceil n_b \rceil_{bm}$	$n + B$	B-tree, the d-IPL, μ*-tree
IN MEMORY BUFFERING & NODE MODIFICATION				
MB-tree [124]	$2 + \lceil \log_M \frac{2*n}{M*n_l} \rceil$	$\lceil 3/n_t \rceil_w + \lceil (n_t + \lceil \log_M \frac{2*n}{M*n_l} \rceil) / n_t \rceil_r$	$n + B$	BFTL, B+tree(ST), B-tree
FB-tree [75]	–	–	–	B+tree
Bw-tree [92, 93]	–	–	–	BerkeleyDB, Skip List
Bloom tree [66]	$h + p_{fp} * d + 2$	–	$n + B$	B+tree, B+tree(ST), FD-tree, MB-tree
IN MEMORY BUFFERING & IN MEMORY BATCH READ BUFFERING & NODE MODIFICATION				
PIOB-tree [125, 126]	$h - 1 + t_L$	$\lceil \sum_{l=\lceil \eta \rceil}^{h-2} \frac{1}{G(l)} \rceil_r + \lceil \frac{1}{G(h-1)} \rceil_r + \lceil \frac{1}{G(h-1)} \rceil_w - \frac{1/M^{\lceil \eta \rceil}}{G(\log_{M'}(\mu - B) - 1)}$	$n + \mu$	BFTL, FD-tree, B+tree

Algorithms Behind Modern Storage Systems, https://queue.acm.org/detail.cfm?id=3220266
Indexing in flash storage devices: a survey on challenges, current approaches, and future trends, https://link.springer.com/article/10.1007/s00778-019-00559-8

3/3: Implications for CS/Systems - **Pick Anything ;)**



What you should remember from this talk

1. **Data sizes are exploding**
   a. Bioinformatics, edge computing, social media, scientific experiments, farming ...

2. **Storage is no longer a bottleneck**
   a. Data accesses in 10s of microseconds with millions operations /sec
   b. Can help to compliment DRAM data accesses (Optane)

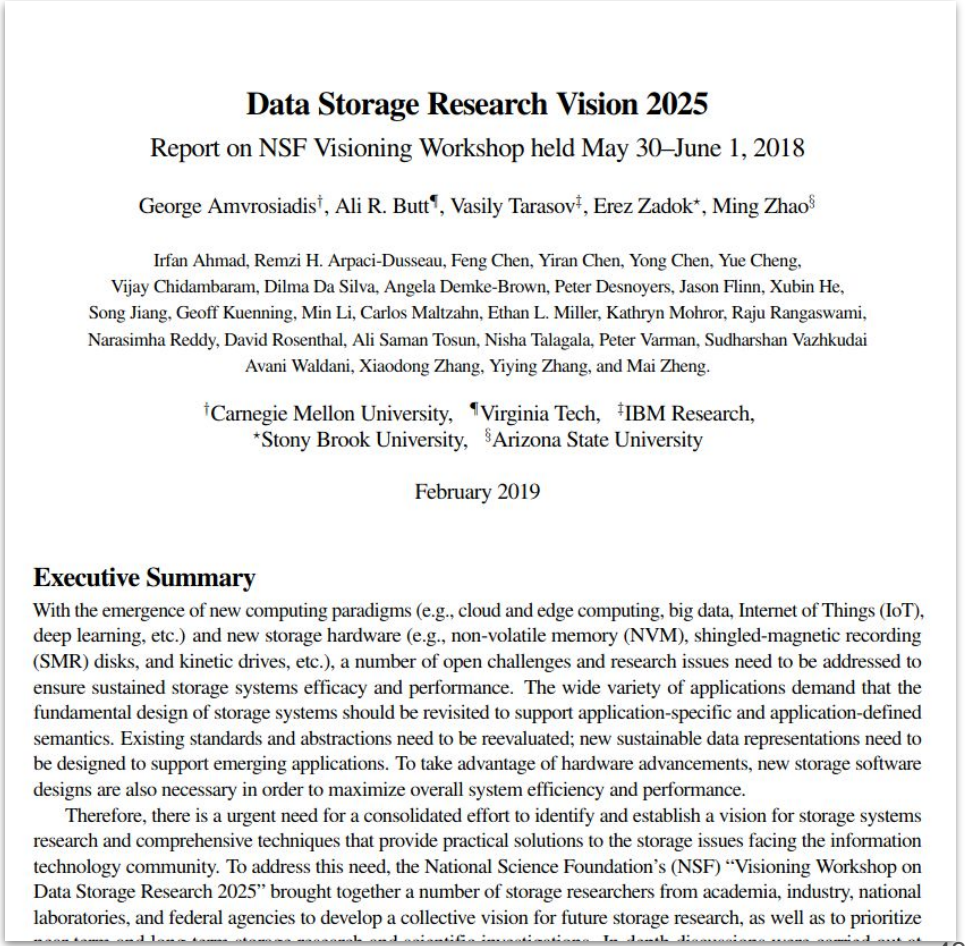
3. The field of computing is changing ***towards specialization***
   a. **Where you store data:** DRAM, Flash, Optane, HDDs, Tape
   b. **How do you store data:** block, objects, files, tensors, tree, link lists, hash tables
   c. **How do you access:** synchronous, interrupts, scheduling, fast/slow paths, programmability
   d. ***You can ask to tailor a storage stack for you own needs*** - performance, and cost

4. **Efficiency of data storage** - cost and computation - is important

To Conclude

Storage Research is fundamentally changing and reshaping what **kind of systems we can build tomorrow**

- Better performance
- New specialized abstractions
- Better cost and energy efficiency
- Superior scalability
- And much more …



Data Storage Research Vision 2025

Report on NSF Visioning Workshop held May 30–June 1, 2018

George Amvrosiadis[†], Ali R. Butt[¶], Vasily Tarasov[‡], Erez Zadok[⋆], Ming Zhao[§]

Irfan Ahmad, Remzi H. Arpaci-Dusseau, Feng Chen, Yiran Chen, Yong Chen, Yue Cheng, Vijay Chidambaram, Dilma Da Silva, Angela Demke-Brown, Peter Desnoyers, Jason Flinn, Xubin He, Song Jiang, Geoff Kuenning, Min Li, Carlos Maltzahn, Ethan L. Miller, Kathryn Mohror, Raju Rangaswami, Narasimha Reddy, David Rosenthal, Ali Saman Tosun, Nisha Talagala, Peter Varman, Sudharshan Vazhkudai Avani Waldani, Xiaodong Zhang, Yiying Zhang, and Mai Zheng.

[†]Carnegie Mellon University, [¶]Virginia Tech, [‡]IBM Research, [⋆]Stony Brook University, [§]Arizona State University

February 2019

Executive Summary

With the emergence of new computing paradigms (e.g., cloud and edge computing, big data, Internet of Things (IoT), deep learning, etc.) and new storage hardware (e.g., non-volatile memory (NVM), shingled-magnetic recording (SMR) disks, and kinetic drives, etc.), a number of open challenges and research issues need to be addressed to ensure sustained storage systems efficacy and performance. The wide variety of applications demand that the fundamental design of storage systems should be revisited to support application-specific and application-defined semantics. Existing standards and abstractions need to be reevaluated; new sustainable data representations need to be designed to support emerging applications. To take advantage of hardware advancements, new storage software designs are also necessary in order to maximize overall system efficiency and performance.

Therefore, there is a urgent need for a consolidated effort to identify and establish a vision for storage systems research and comprehensive techniques that provide practical solutions to the storage issues facing the information technology community. To address this need, the National Science Foundation's (NSF) "Visioning Workshop on Data Storage Research 2025" brought together a number of storage researchers from academia, industry, national laboratories, and federal agencies to develop a collective vision for future storage research, as well as to prioritize

The (new) triangle of storage hierarchy