Future Computing Architectures

Onur Mutlu
onur.mutlu@inf.ethz.ch
https://people.inf.ethz.ch/omutlu

May 15, 2017

ETH Zurich Inaugural Lecture

SAFARI

ETH zürich

Why Do We Do Computing?

Answer

To Solve Problems

Answer Reworded

To Gain Insight

Hamming, "Numerical Methods for Scientists and Engineers," 1962.

Answer Extended

To Enable

a Better Life & Future

How Does a Computer Solve Problems?

Answer

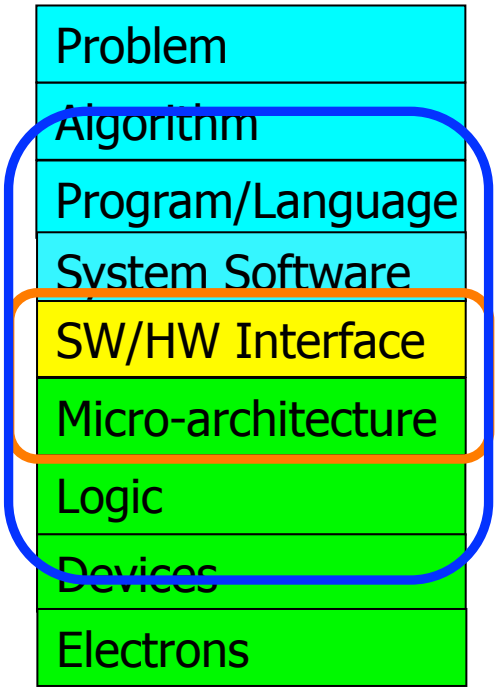
Orchestrating Electrons

In today's dominant technologies

How Do Problems Get Solved by Electrons?

The Transformation Hierarchy

Computer Architecture (expanded view)



Computer Architecture (narrow view)

Computer Architecture

- is the science and art of designing computing platforms (hardware, interface, system SW, and programming model)

- to achieve a set of design goals
  - E.g., highest performance on earth on workloads X, Y, Z
  - E.g., longest battery life at a form factor that fits in your pocket with cost < $$$ CHF
  - E.g., best average performance across all known workloads at the best performance/cost ratio
  - …

  - Designing a supercomputer is different from designing a smartphone → But, many fundamental principles are similar

Different Platforms, Different Goals

Source: http://www.sia-online.org (semiconductor industry association)

Different Platforms, Different Goals

Source: https://iq.intel.com/5-awesome-uses-for-drone-technology/

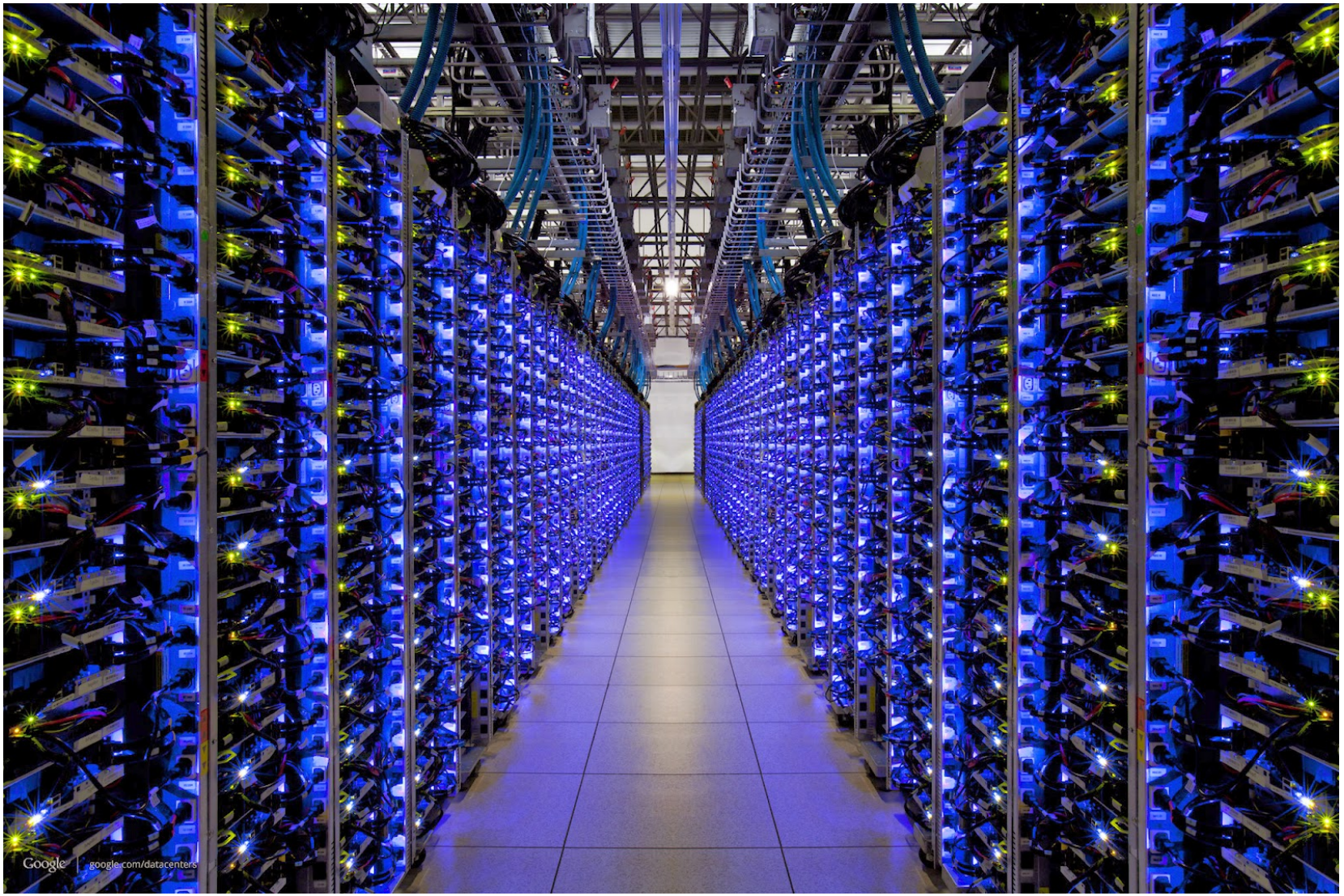
Different Platforms, Different Goals

Source: https://taxistartup.com/wp-content/uploads/2015/03/UK-Self-Driving-Cars.jpg

Different Platforms, Different Goals

Source: http://sm.pcmag.com/pcmag_uk/photo/g/google-self-driving-car-the-guts/google-self-driving-car-the-guts_dwx8.jpg

Different Platforms, Different Goals



Source: http://datacentervoice.com/wp-content/uploads/2015/10/data-center.jpg

Different Platforms, Different Goals

Source: https://fossbytes.com/wp-content/uploads/2015/06/Supercomputer-TIANHE2-china.jpg

Different Platforms, Different Goals

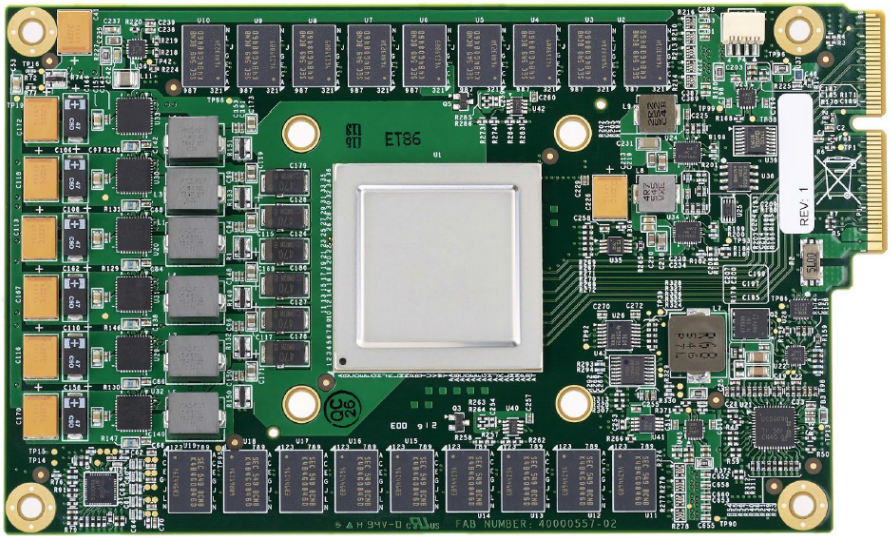


Figure 3. TPU Printed Circuit Board. It can be inserted in the slot for an SATA disk in a server, but the card uses PCIe Gen3 x16.

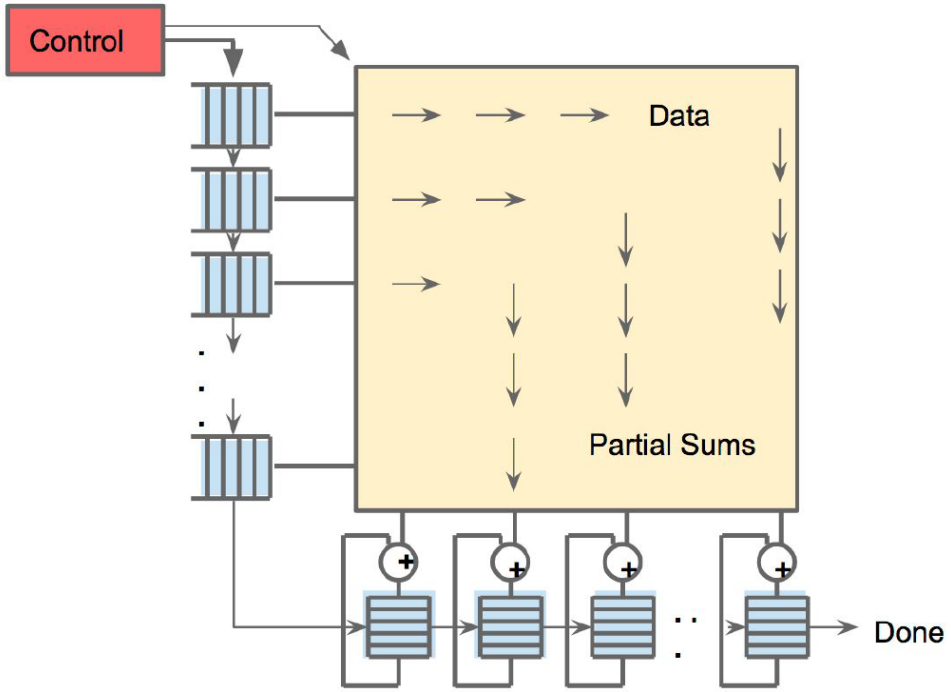


Figure 4. Systolic data flow of the Matrix Multiply Unit. Software has the illusion that each 256B input is read at once, and they instantly update one location of each of 256 accumulator RAMs.

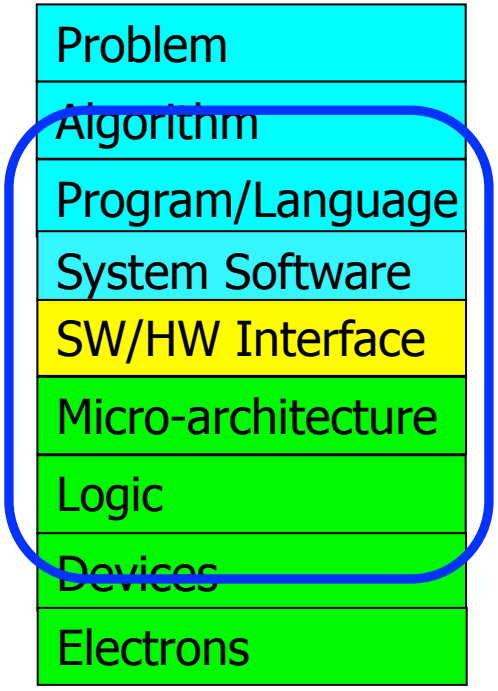
Jouppi et al., "In-Datacenter Performance Analysis of a Tensor Processing Unit", ISCA 2017.

Axiom

To achieve the highest energy efficiency and performance:

we must take the expanded view

of computer architecture



Co-design across the hierarchy: Algorithms to devices

Specialize as much as possible within the design goals

What Kind of a Future Do We Want?

Source: http://www.technologystudent.com/struct1/tacom1.png

Source: AP

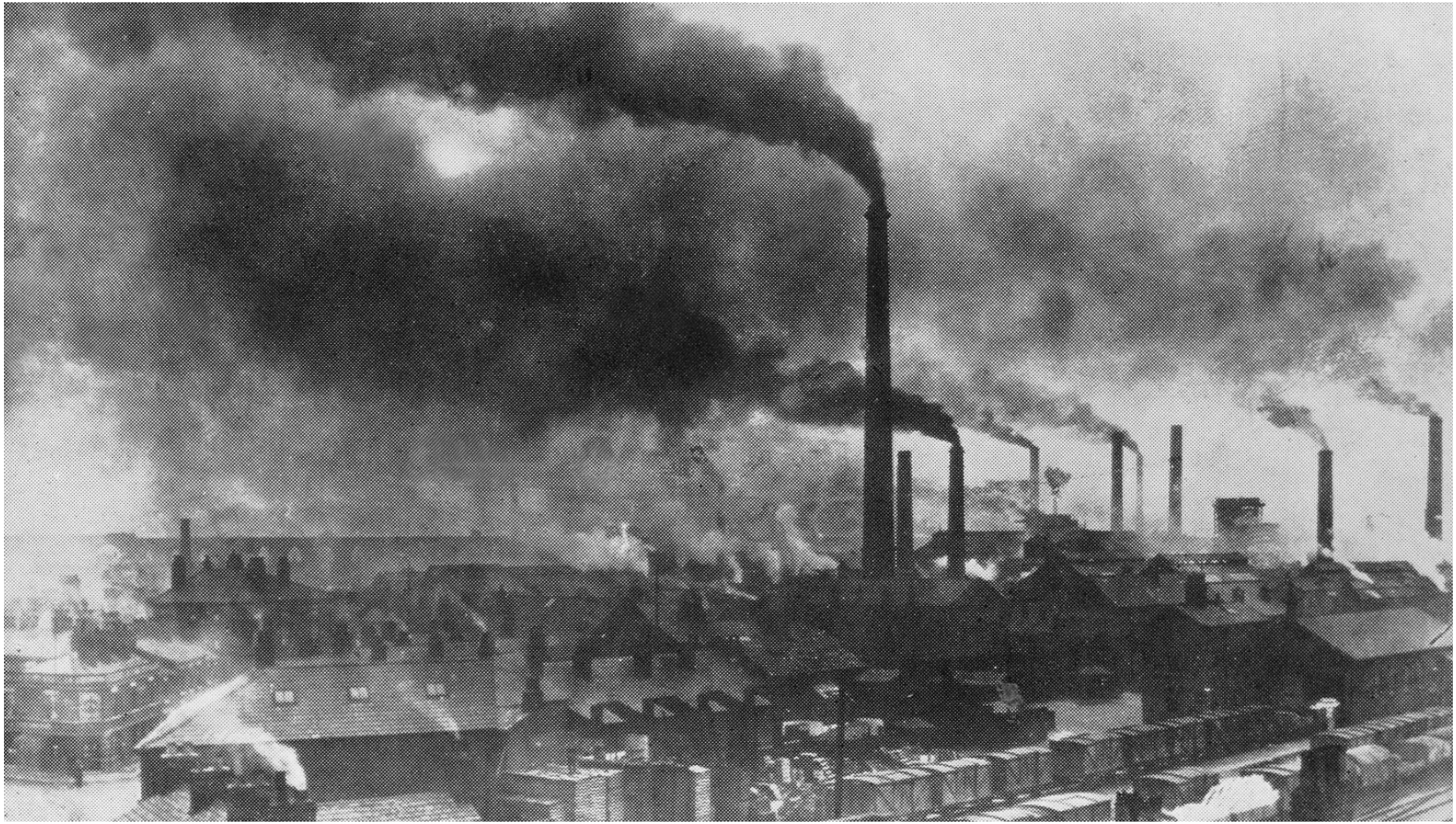
Source: AP Source: http://www.seattlepi.com/science/article/A-Tacoma-Narrows-Galloping-Gertie-bridge-6617030.php

Source: https://s-media-cache-ak0.pinimg.com/originals/48/09/54/4809543a9c7700246a0cf8acdae27abf.jpg

Challenge and Opportunity for Future

Reliable, Secure, Safe

Source: V. Milutinovic



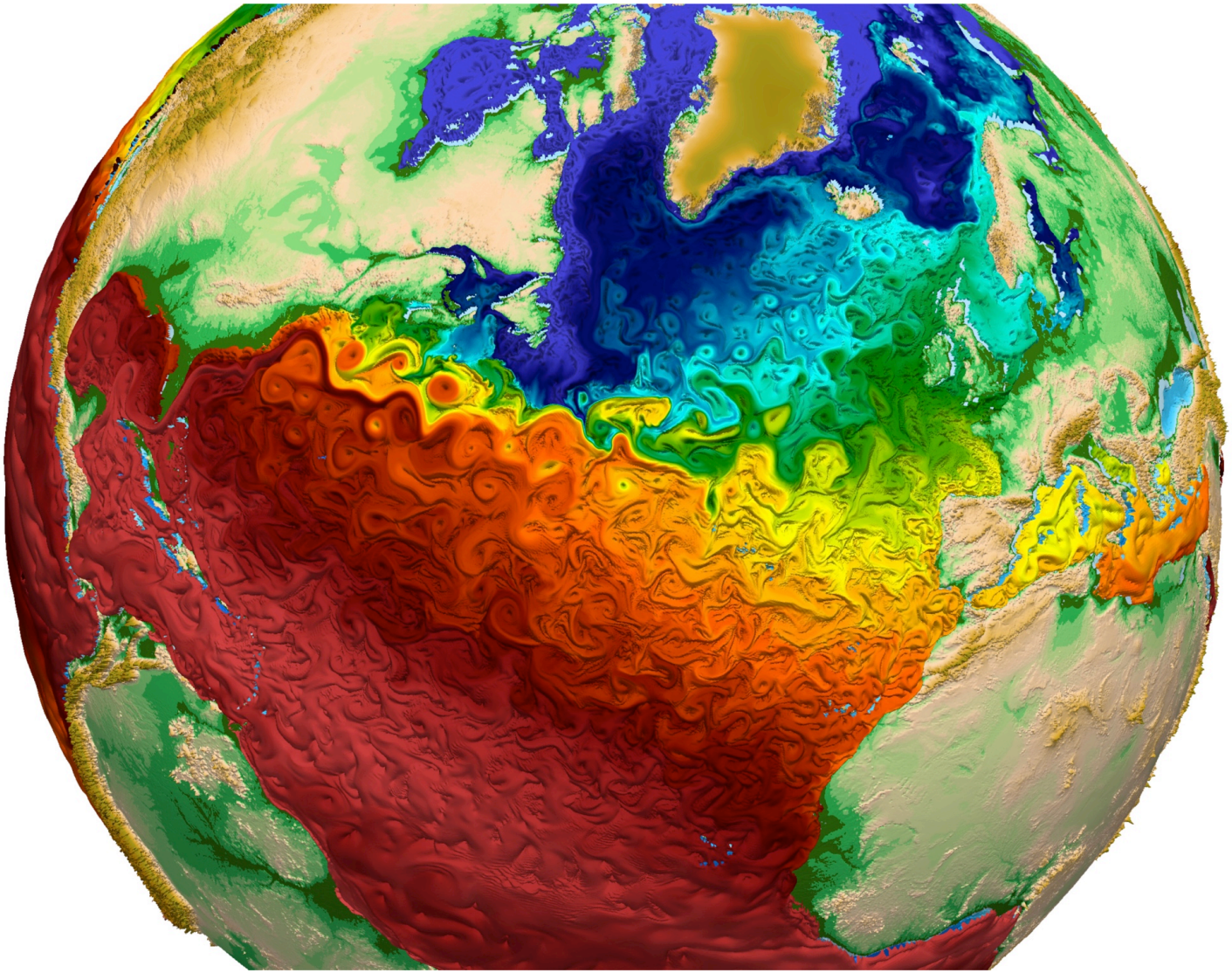
Source: V. Milutinovic

Challenge and Opportunity for Future

Sustainable

and

Energy Efficient



Source: https://farm9.staticflickr.com/8571/16376102935_8628150df8_o.jpg

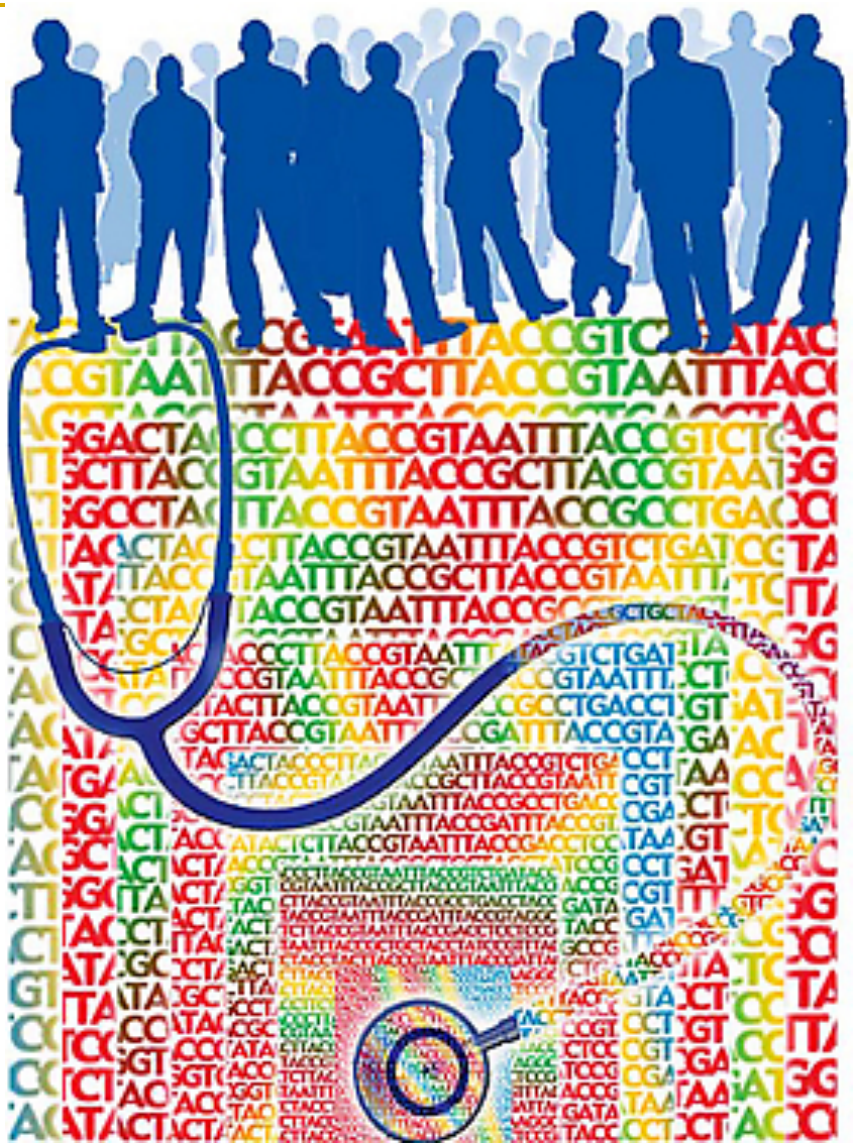
Source: https://blogs-images.forbes.com/jimgorzelany/files/2015/10/China-G4-backup-this-oct-reuters.jpg

Source: http://spectrum.ieee.org/image/MjYzMzAyMg.jpeg

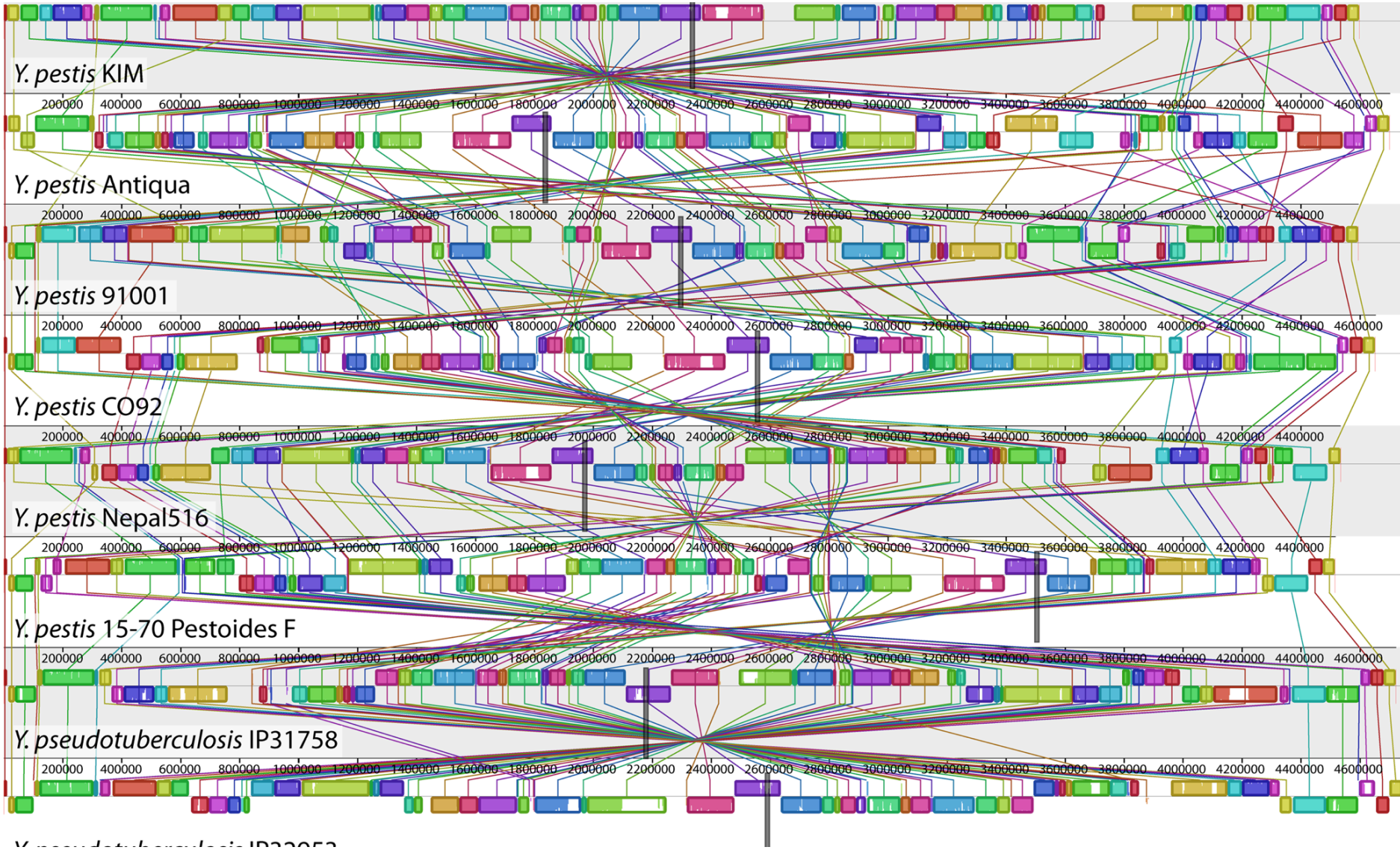
Challenge and Opportunity for Future

High Performance

(to solve
the **toughest** & **all** problems)



Source: Jane Ades, NHGRI



Challenge and Opportunity for Future

Personalized and Private

(in every aspect of life:
health, medicine,
spaces, devices, robotics, …)

This Lecture is About …

- Questioning what limits us in designing the best computing architectures for the future
- Providing directions for fundamentally better designs
- Advocating principled approaches

Increasingly Demanding Applications

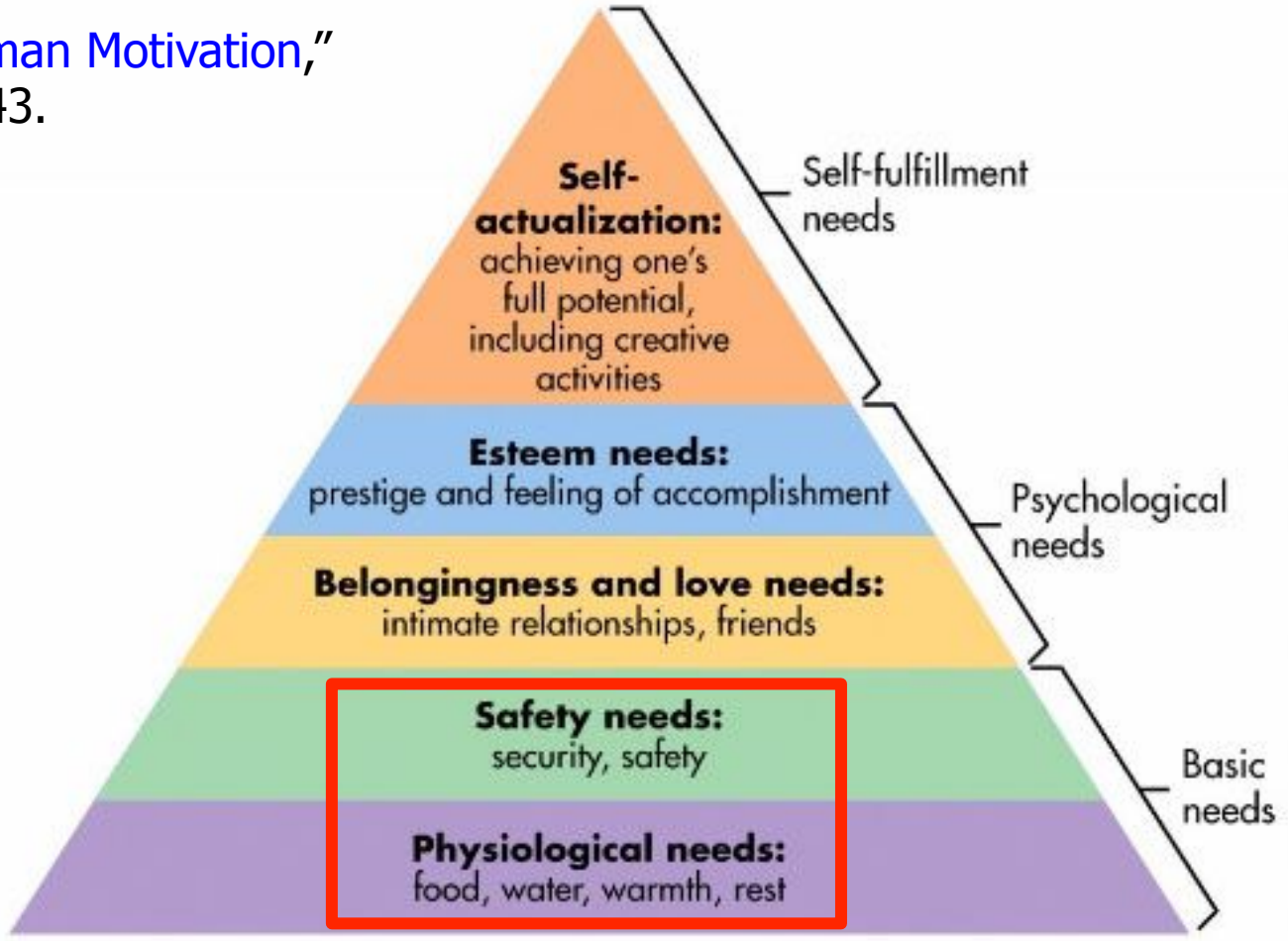
Dream

and, they will come

As applications push boundaries, computing platforms will become increasingly strained.

Maslow's (Human) Hierarchy of Needs

Maslow, "A Theory of Human Motivation," Psychological Review, 1943.



- We need to start with reliability and security…

Source: https://www.simplypsychology.org/maslow.html

Three Key Issues in Future Platforms

- Fundamentally Secure/Reliable/Safe Architectures
- Fundamentally Energy-Efficient Architectures
  - Memory-centric (Data-centric) Architectures
- Architectures for Genomics, Medicine, Health

Source: http://www.technologystudent.com/struct1/tacom1.png

Source: AP

Security is about preventing unforeseen consequences

Source: https://s-media-cache-ak0.pinimg.com/originals/48/09/54/4809543a9c7700246a0cf8acdae27abf.jpg

The Problem

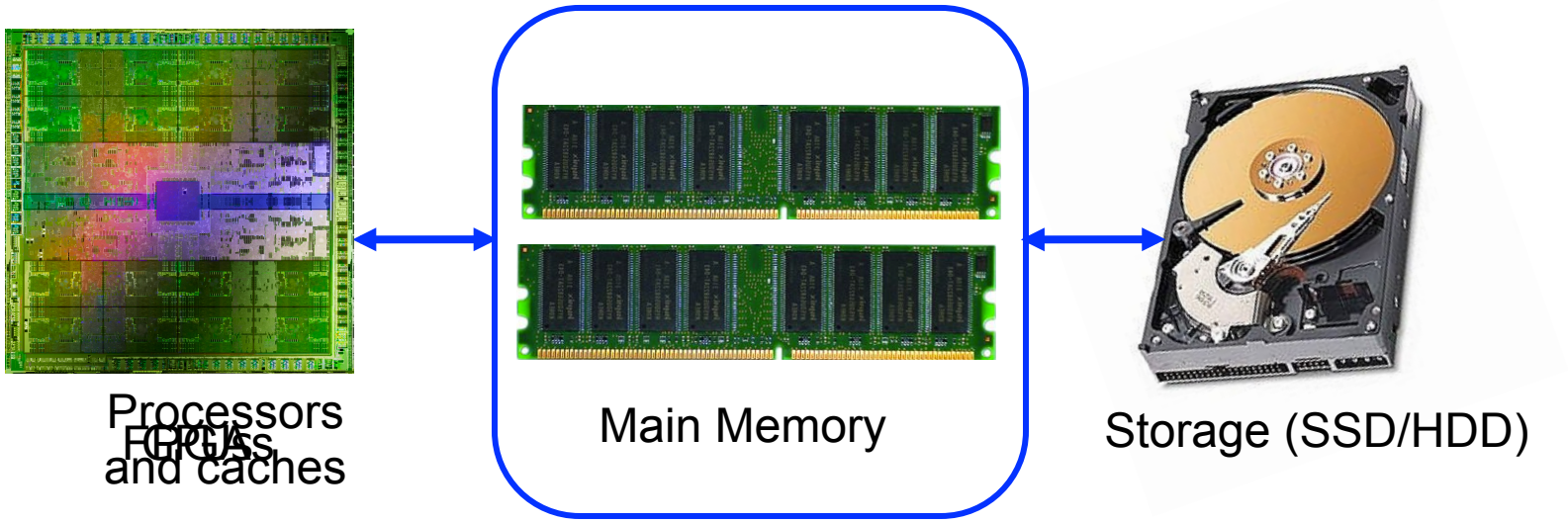
We do not seem to have

design principles for

(guaranteeing)

reliability and security

Focus is on Data Storage Systems (Memory)

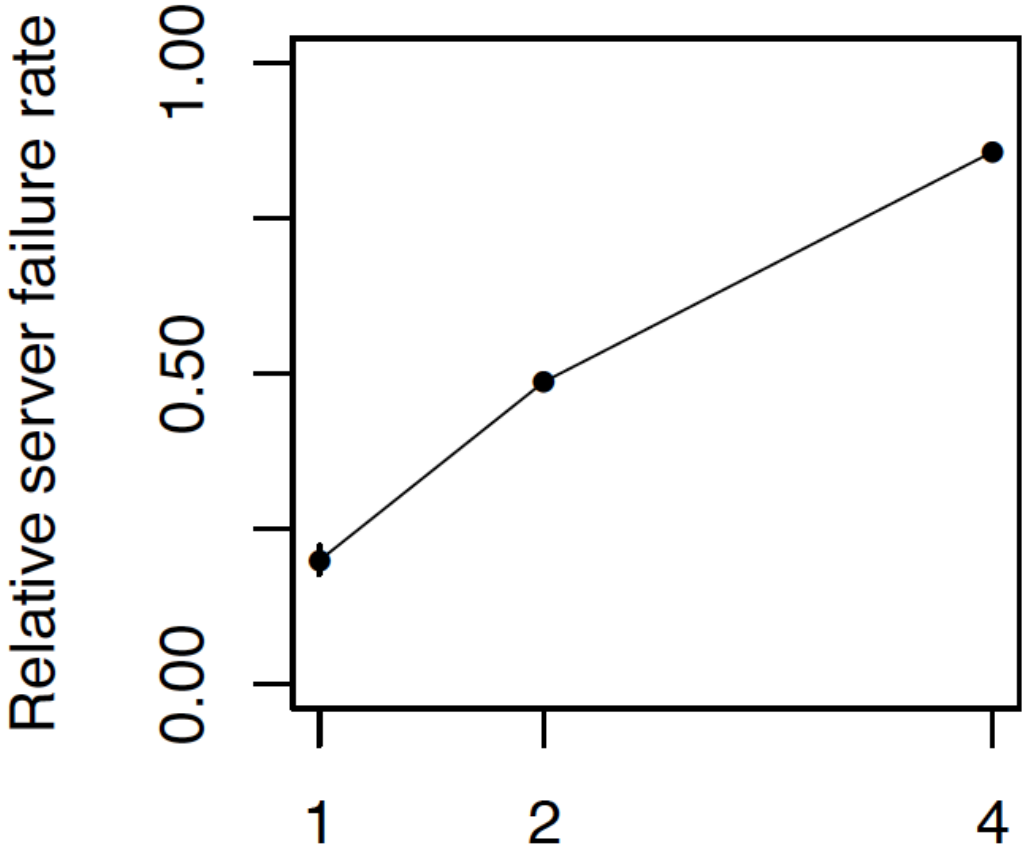


- Memory is a critical component of all computing systems: server, mobile, embedded, desktop, sensor, …

- Main memory system must scale (in *size*, *technology*, *efficiency*, *cost*, and *management algorithms*) to maintain performance growth and technology scaling benefits

As Memory Scales, It Becomes Unreliable

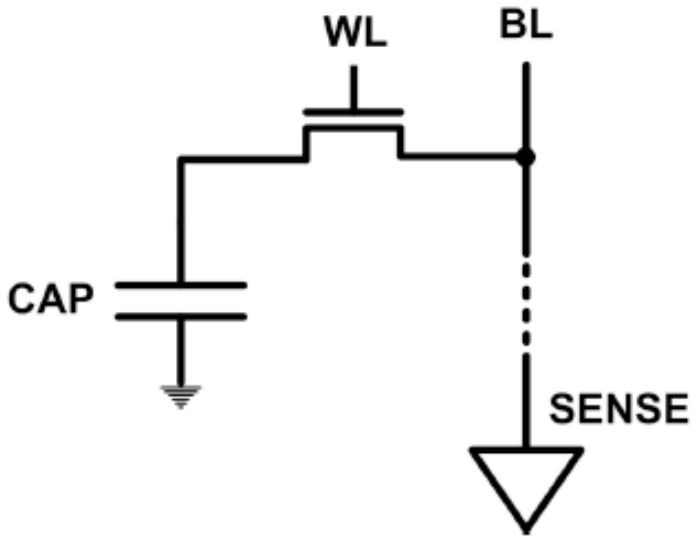
- Data from all of Facebook's servers worldwide
- Meza+, "Revisiting Memory Errors in Large-Scale Production Data Centers," DSN'15.



Intuition: quadratic increase in capacity

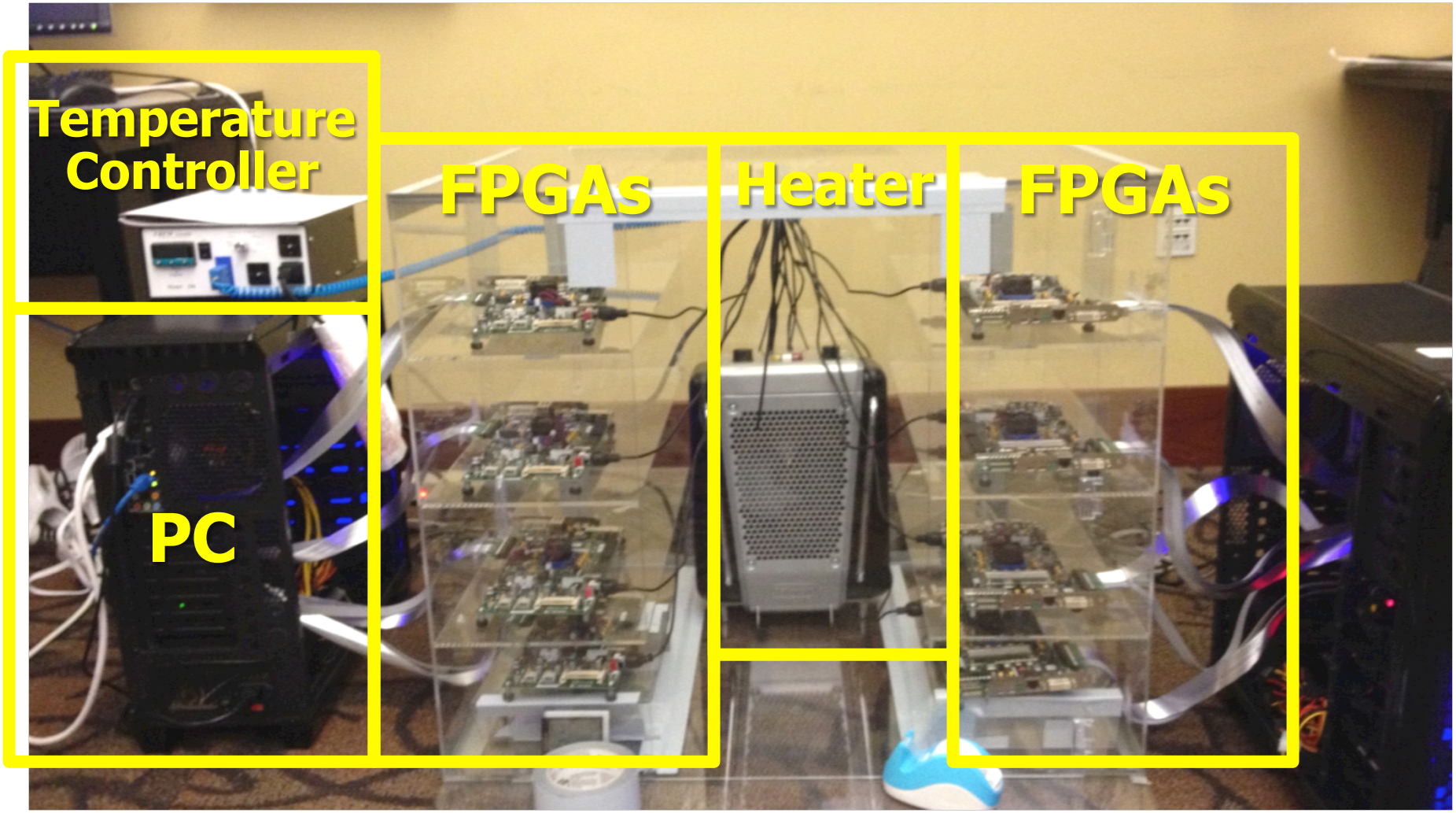
The DRAM Scaling Problem

- DRAM stores charge in a capacitor (charge-based memory)
  - Capacitor must be large enough for reliable sensing
  - Access transistor should be large enough for long data retention time



- As DRAM cell becomes **smaller**, it becomes **more vulnerable**

Infrastructure to Understand Such Issues



Kim+, "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors," ISCA 2014.

A Curious Discovery [Kim et al., ISCA 2014]

One can

predictably induce errors

in most DRAM memory chips

DRAM RowHammer

A simple hardware failure mechanism
can create a widespread
system security vulnerability

WIRED

Forget Software—Now Hackers Are Exploiting Physics

BUSINESS | CULTURE | DESIGN | GEAR | SCIENCE

SHARE

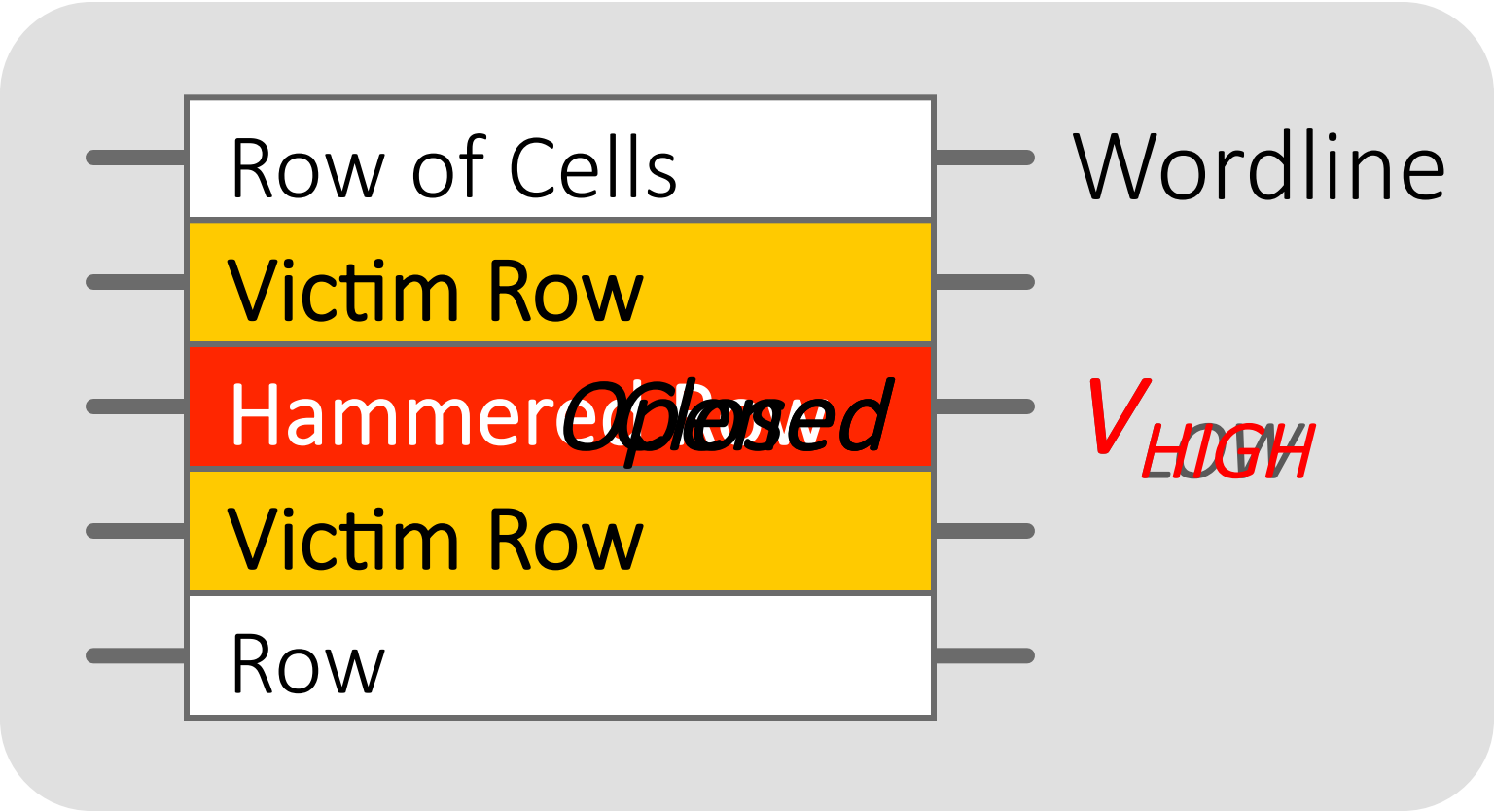
SHARE
18276

TWEET

ANDY GREENBERG SECURITY 08.31.16 7:00 AM

FORGET SOFTWARE—NOW HACKERS ARE EXPLOITING PHYSICS

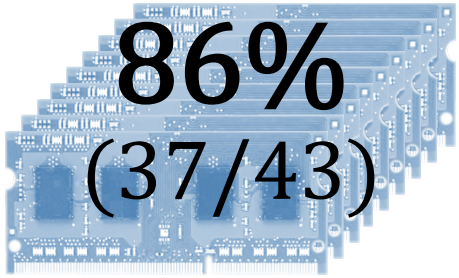
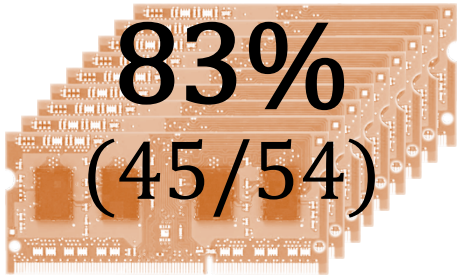
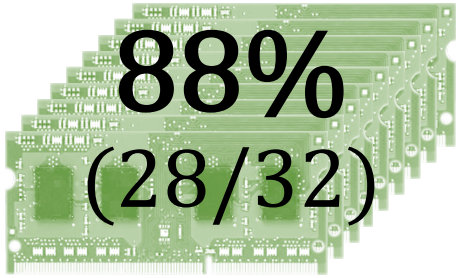
Modern DRAM is Prone to Disturbance Errors



Repeatedly reading a row enough times (before memory gets refreshed) induces disturbance errors in **adjacent rows** in most real DRAM chips you can buy today

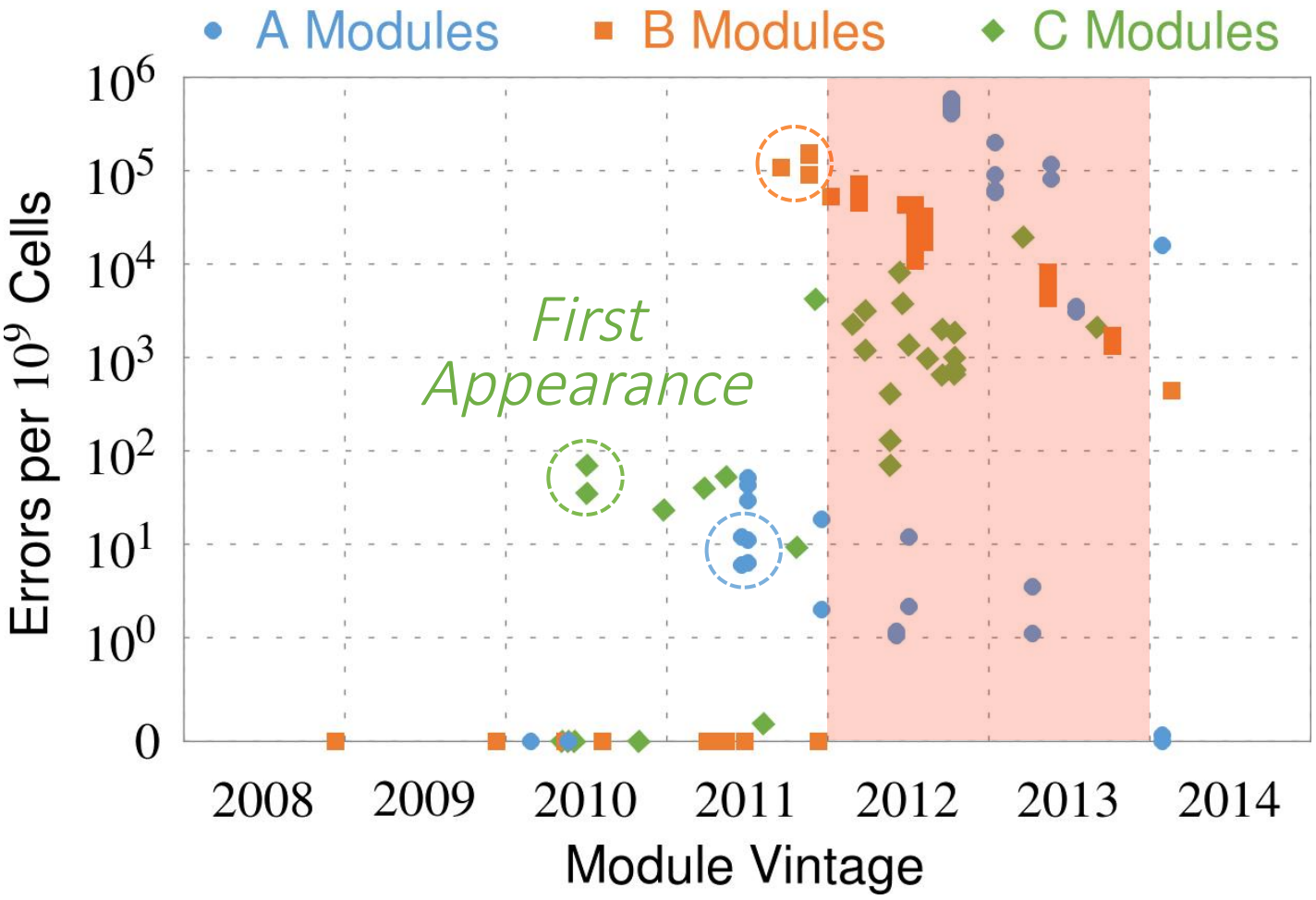
Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors, (Kim et al., ISCA 2014)

Most DRAM Modules Are Vulnerable

A company	B company	C company
		
86% (37/43)	**83%** (45/54)	**88%** (28/32)
Up to **1.0×10^7** errors	Up to **2.7×10^6** errors	Up to **3.3×10^5** errors

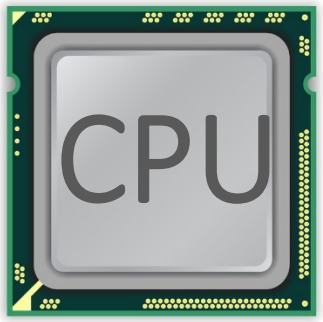
Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors, (Kim et al., ISCA 2014)

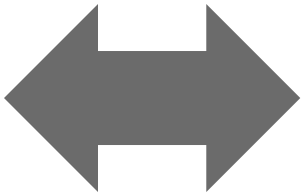
Recent DRAM Is More Vulnerable

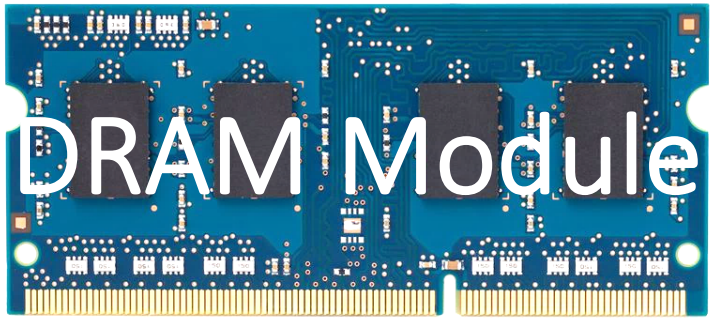


All modules from 2012–2013 are vulnerable

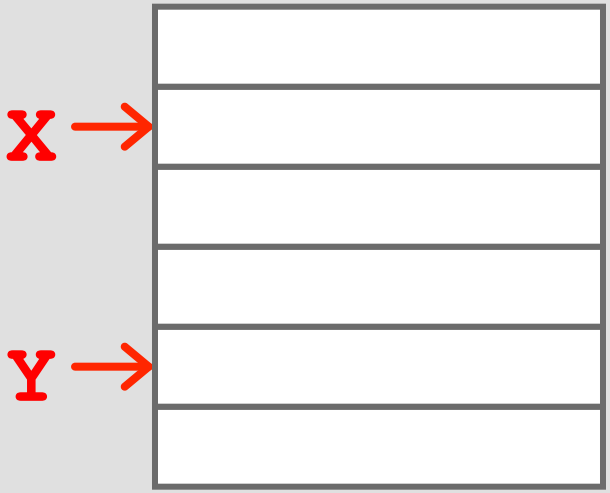
A Simple Program Can Induce Many Errors





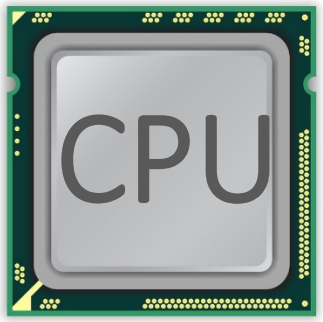


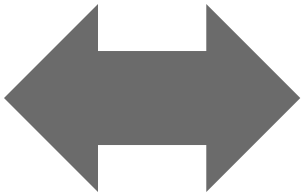

```
loop:
   mov (X), %eax
   mov (Y), %ebx
   clflush (X)
   clflush (Y)
   mfence
   jmp loop
```

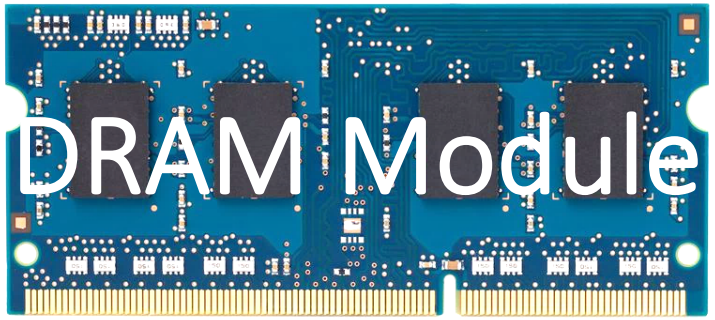


Download from: https://github.com/CMU-SAFARI/rowhammer

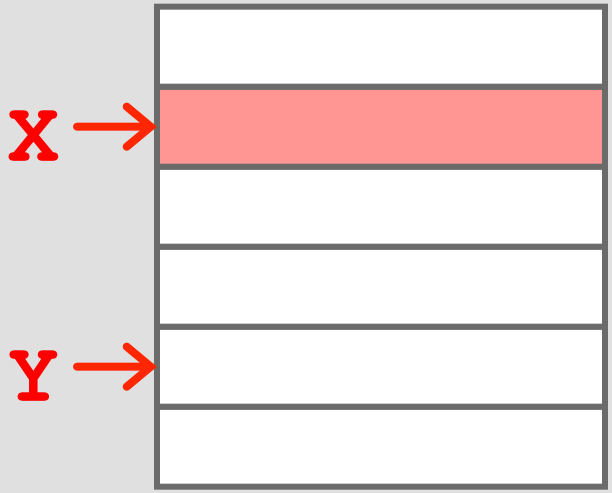
A Simple Program Can Induce Many Errors





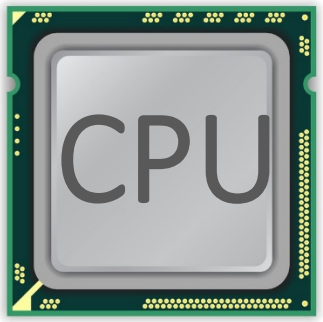


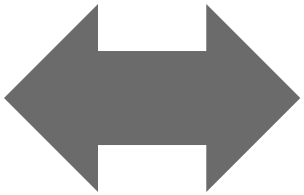

```
loop:
   mov (X), %eax
   mov (Y), %ebx
   clflush (X)
   clflush (Y)
   mfence
   jmp loop
```

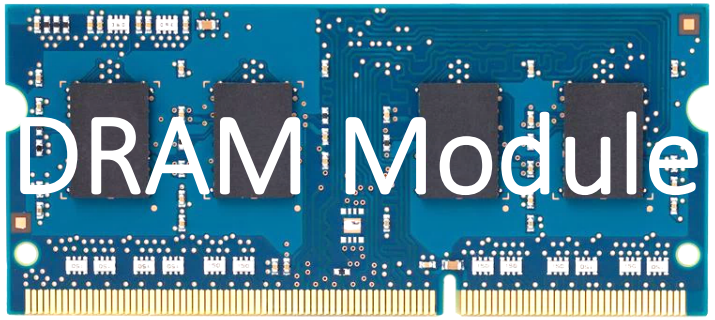


Download from: https://github.com/CMU-SAFARI/rowhammer

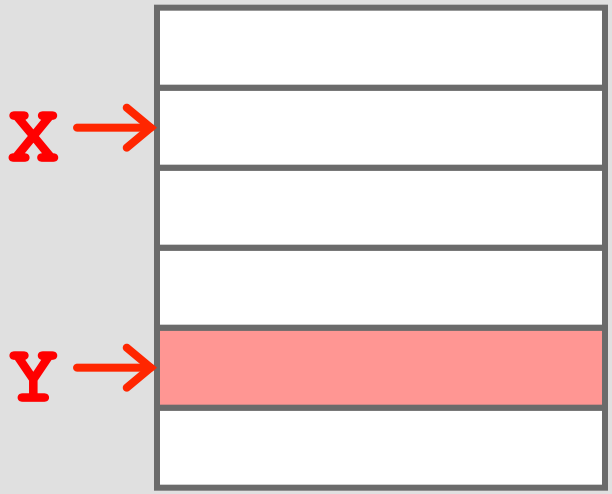
A Simple Program Can Induce Many Errors





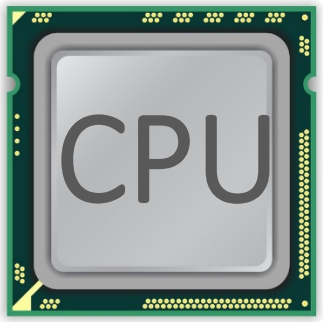


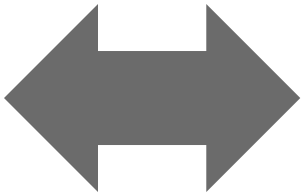

```
loop:
   mov (X), %eax
   mov (Y), %ebx
   clflush (X)
   clflush (Y)
   mfence
   jmp loop
```

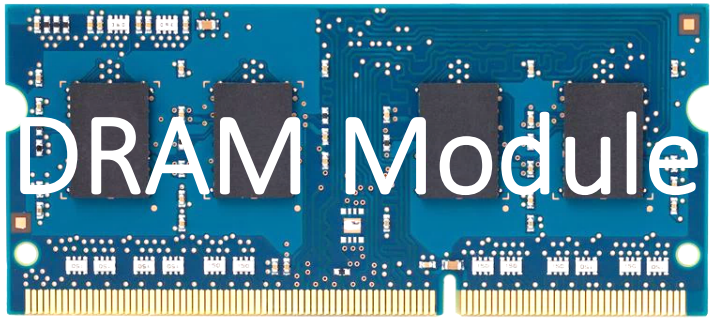


Download from: https://github.com/CMU-SAFARI/rowhammer

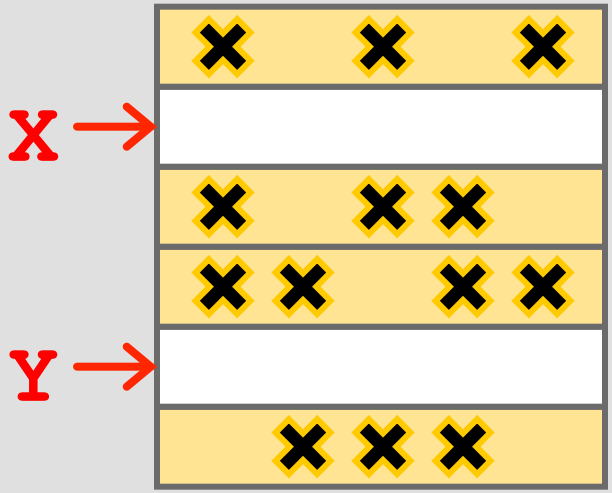
A Simple Program Can Induce Many Errors








```
loop:
   mov (X), %eax
   mov (Y), %ebx
   clflush (X)
   clflush (Y)
   mfence
   jmp loop
```



Download from: https://github.com/CMU-SAFARI/rowhammer

Observed Errors in Real Systems

CPU Architecture	Errors	Access-Rate
Intel Haswell (2013)	22.9K	12.3M/sec
Intel Ivy Bridge (2012)	20.7K	11.7M/sec
Intel Sandy Bridge (2011)	16.1K	11.6M/sec
AMD Piledriver (2012)	59	6.1M/sec

A real reliability & security issue

Kim+, "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors," ISCA 2014.

One Can Take Over an Otherwise-Secure System

Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors

Abstract. Memory isolation is a key property of a reliable and secure computing system — an access to one memory address should not have unintended side effects on data stored in other addresses. However, as DRAM process technology

Project Zero

Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors (Kim et al., ISCA 2014)

News and updates from the Project Zero team at Google

Exploiting the DRAM rowhammer bug to gain kernel privileges (Seaborn+, 2015)

Monday, March 9, 2015

Exploiting the DRAM rowhammer bug to gain kernel privileges

RowHammer Security Attack Example

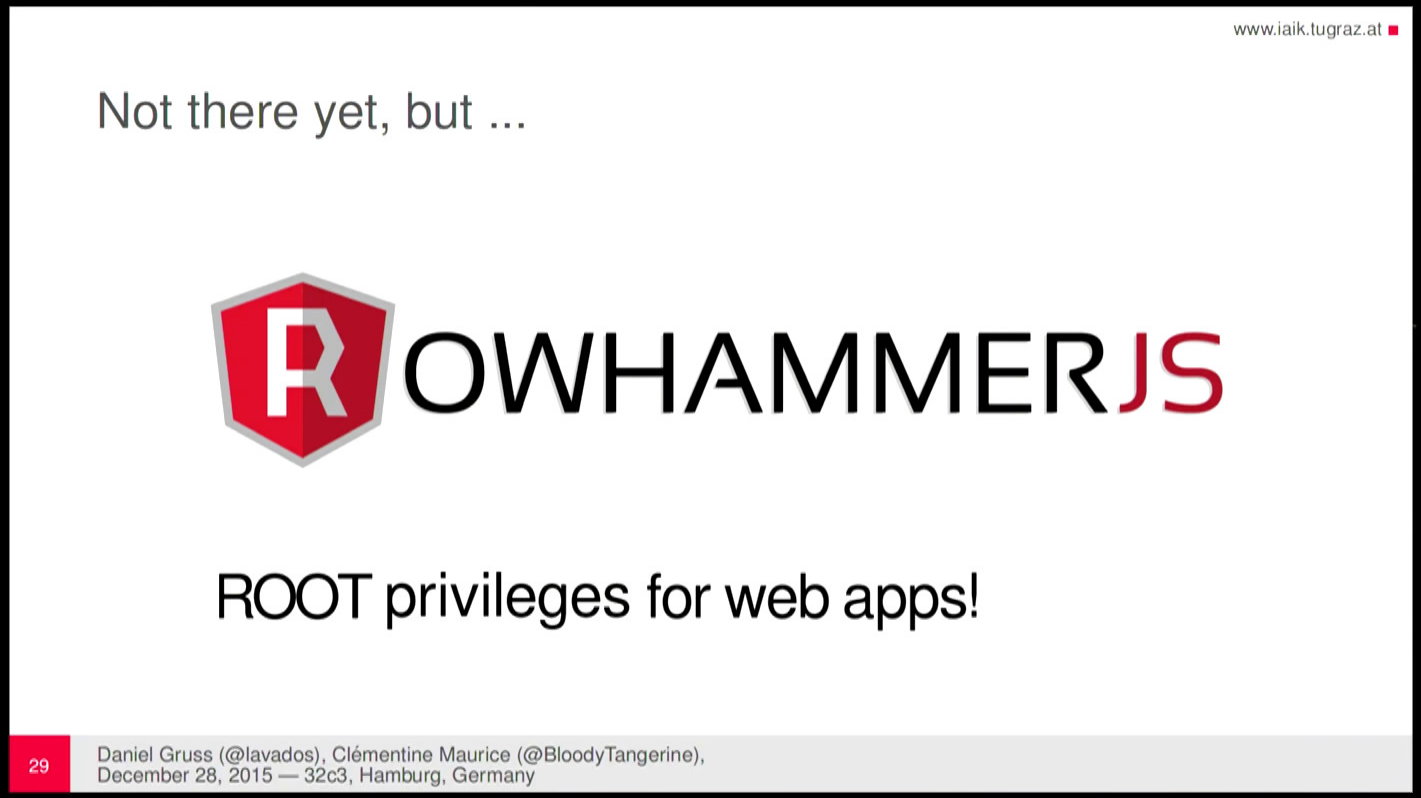
- "Rowhammer" is a problem with some recent DRAM devices in which repeatedly accessing a row of memory can cause bit flips in adjacent rows (Kim et al., ISCA 2014).
  - Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors (Kim et al., ISCA 2014)
- We tested a selection of laptops and found that a subset of them exhibited the problem.
- We built two working privilege escalation exploits that use this effect.
  - Exploiting the DRAM rowhammer bug to gain kernel privileges (Seaborn+, 2015)
- One exploit uses rowhammer-induced bit flips to **gain kernel privileges** on x86-64 Linux when run as an unprivileged userland process.
- When run on a machine vulnerable to the rowhammer problem, the process was able to induce bit flips in page table entries (PTEs).
- It was able to use this to gain write access to its own page table, and hence gain read-write access to all of physical memory.

Security Implications

It's like breaking into an apartment by repeatedly slamming a neighbor's door until the vibrations open the door you were after

More Security Implications

"We can gain unrestricted access to systems of website visitors."



Rowhammer.js: A Remote Software-Induced Fault Attack in JavaScript (DIMVA'16)

Source: https://lab.dsst.io/32c3-slides/7197.html

More Security Implications

"Can gain control of a smart phone deterministically"

Drammer: Deterministic Rowhammer Attacks on Mobile Platforms, CCS'16

Source: https://fossbytes.com/drammer-rowhammer-attack-android-root-devices/

More Security Implications?

Apple's Patch for RowHammer

- https://support.apple.com/en-gb/HT204934

Available for: OS X Mountain Lion v10.8.5, OS X Mavericks v10.9.5

Impact: A malicious application may induce memory corruption to escalate privileges

Description: A disturbance error, also known as Rowhammer, exists with some DDR3 RAM that could have led to memory corruption. This issue was mitigated by increasing memory refresh rates.

CVE-ID

CVE-2015-3693 : Mark Seaborn and Thomas Dullien of Google, working from original research by Yoongu Kim et al (2014)

HP, Lenovo, and other vendors released similar patches

Better Solution Directions: Principled Designs

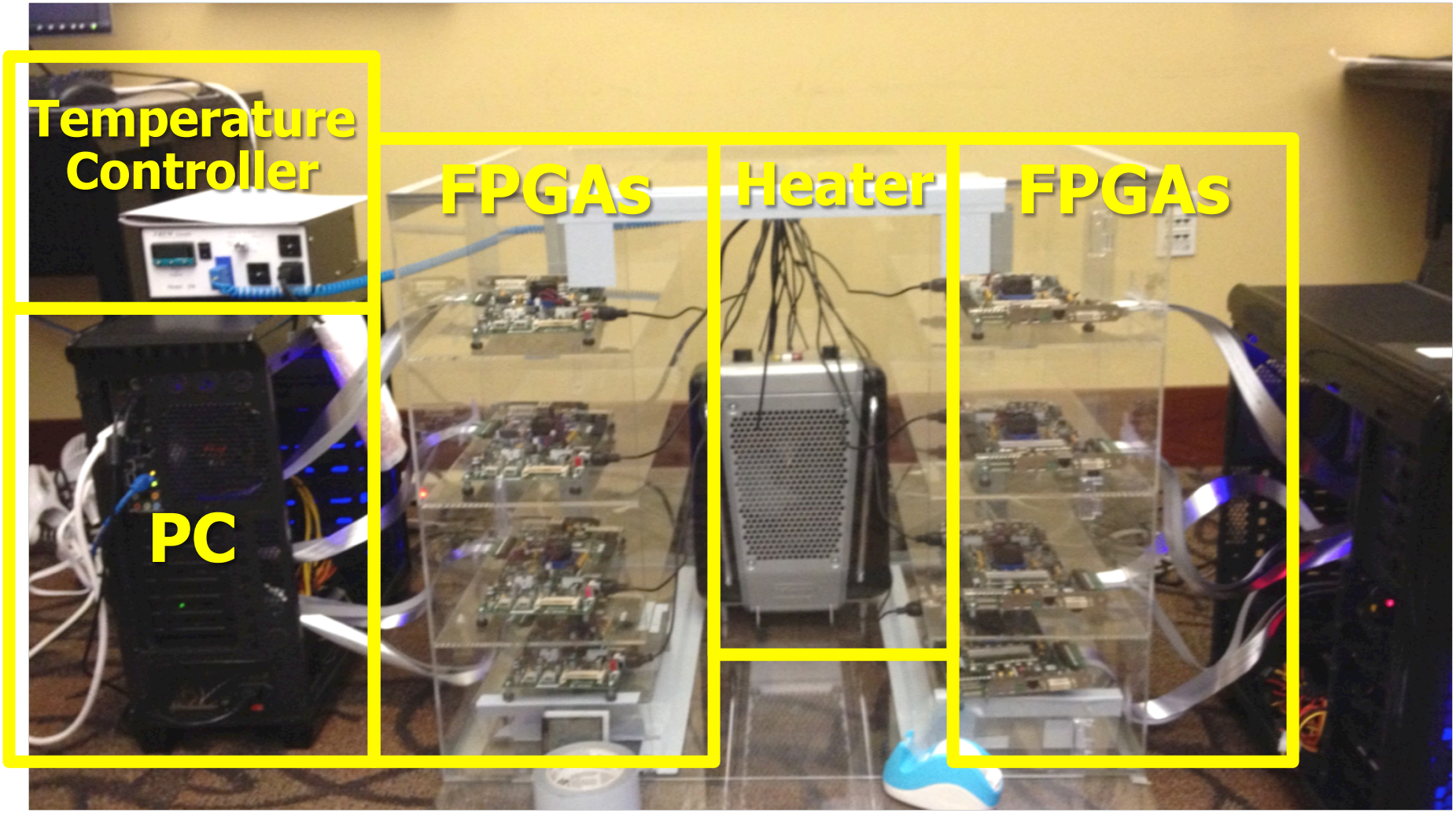
Design fundamentally secure computing architectures

Predict and prevent such safety issues

How Do We Keep Memory Secure?

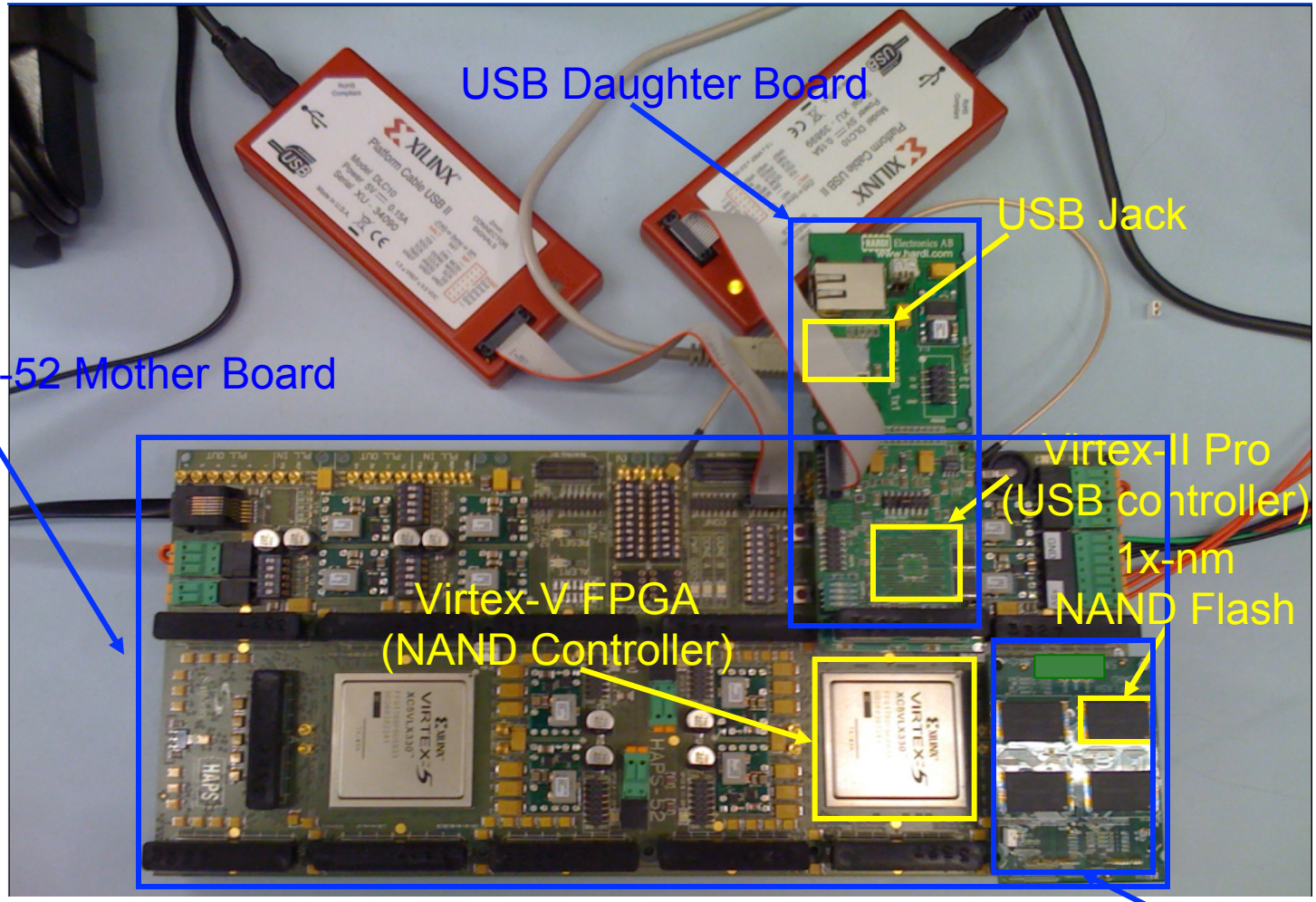
- **Understand**: Methodologies for failure modeling and discovery
  - Modeling and prediction based on real (device) data
- **Architect**: Principled co-architecting of system and memory
  - Good partitioning of duties across the stack
- **Design & Test**: Principled design, automation, testing
  - High coverage and good interaction with system reliability methods

Understand and Model with Experiments (DRAM)



Kim+, "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors," ISCA 2014.

Understand and Model with Experiments (Flash)



[DATE 2012, ICCD 2012, DATE 2013, ITJ 2013, ICCD 2013, SIGMETRICS 2014, HPCA 2015, DSN 2015, MSST 2015, JSAC 2016, HPCA 2017, DFRWS 2017]

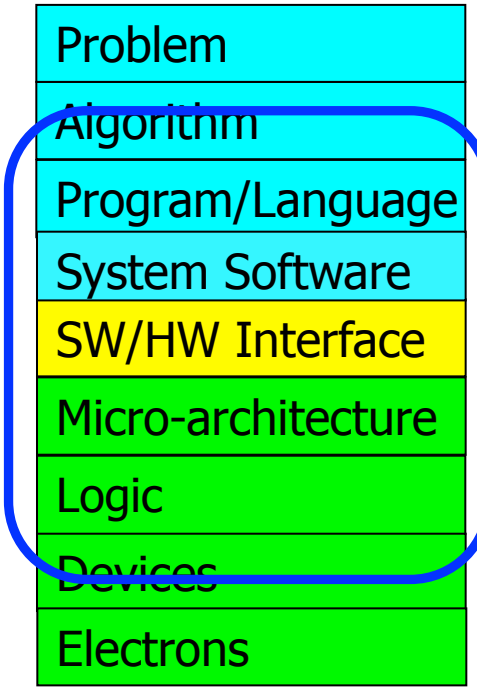
There are Two Other Other Solutions

- **New Technologies:** Replace or (more likely) augment DRAM with a different technology
  - Non-volatile memories

- **Embracing Un-reliability:**

  Design memories with different reliability and store data intelligently across them

- …



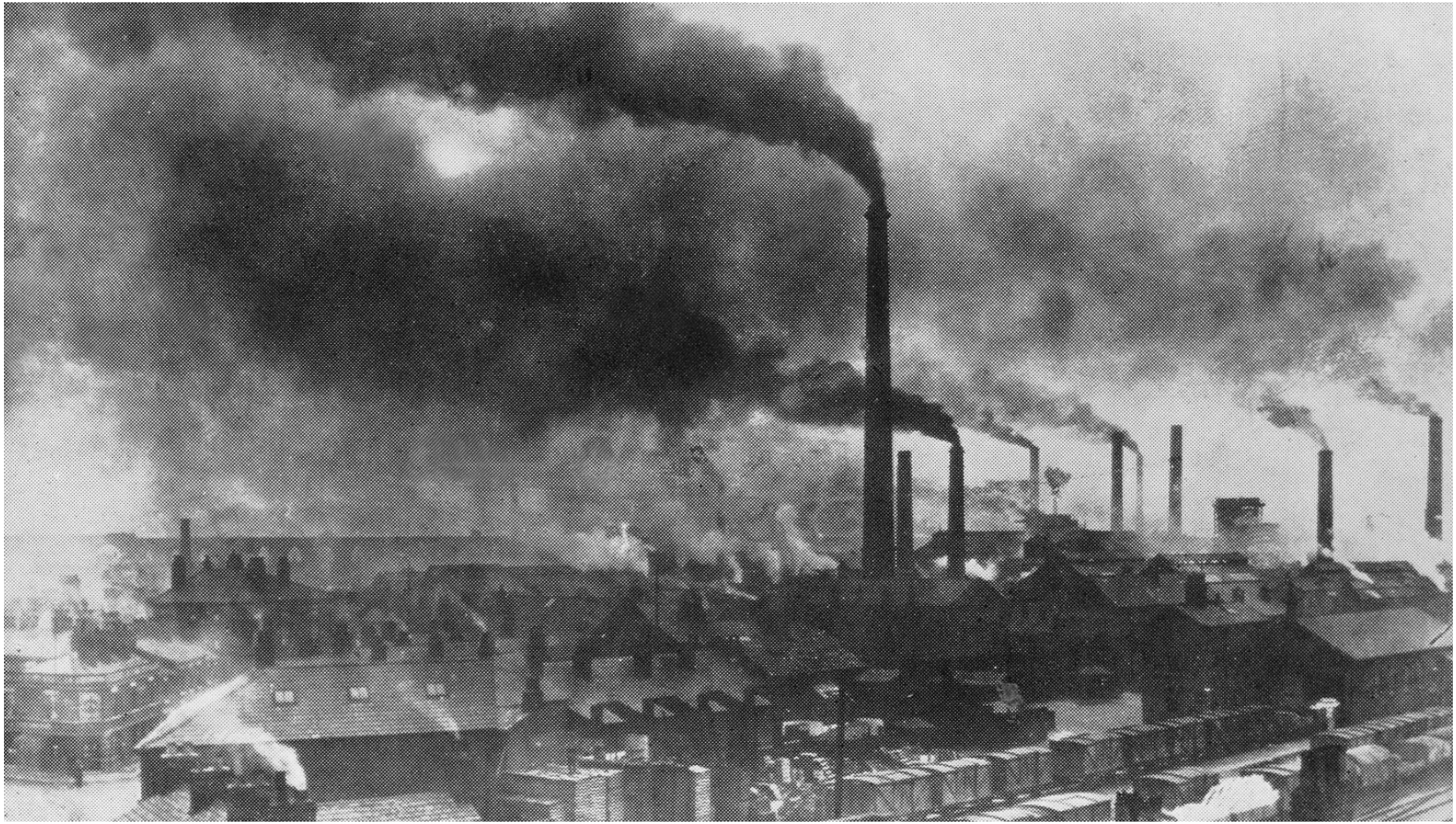
Fundamental solutions to security require co-design across the hierarchy

Challenge and Opportunity for Future

Fundamentally
Secure, Reliable, Safe
Computing Architectures

Three Key Issues in Future Platforms

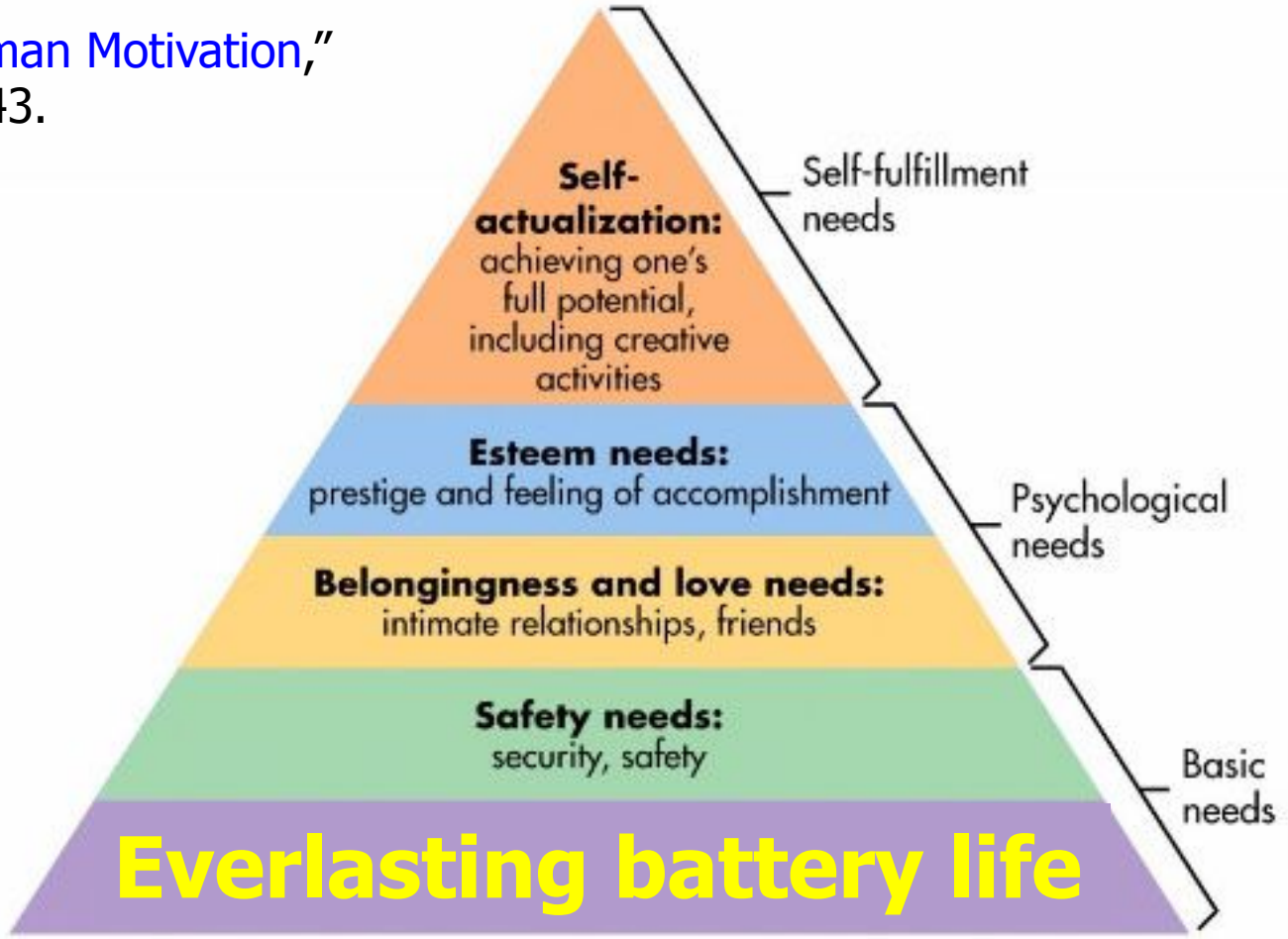
- Fundamentally Secure/Reliable/Safe Architectures
- Fundamentally Energy-Efficient Architectures
  - Memory-centric (Data-centric) Architectures
- Architectures for Genomics, Medicine, Health



Source: V. Milutinovic

Maslow's (Human) Hierarchy of Needs, Revisited

Maslow, "A Theory of Human Motivation," Psychological Review, 1943.



- We need to start with reliability and security…

Challenge and Opportunity for Future

Sustainable

and

Energy Efficient

The Problem

Data access is the major performance and energy bottleneck

Our current

design principles

cause great energy waste

The Problem

Processing of data

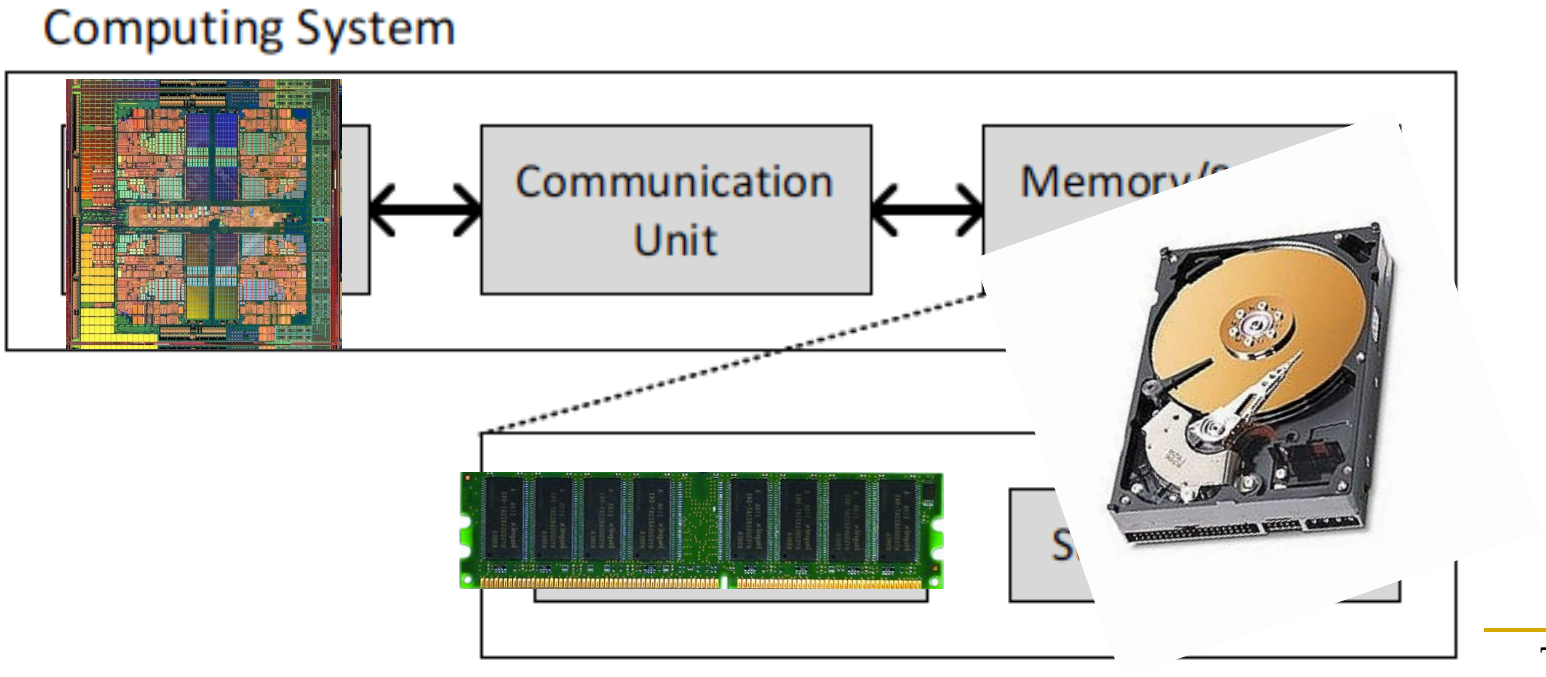
is performed

far away from the data

A Computing System

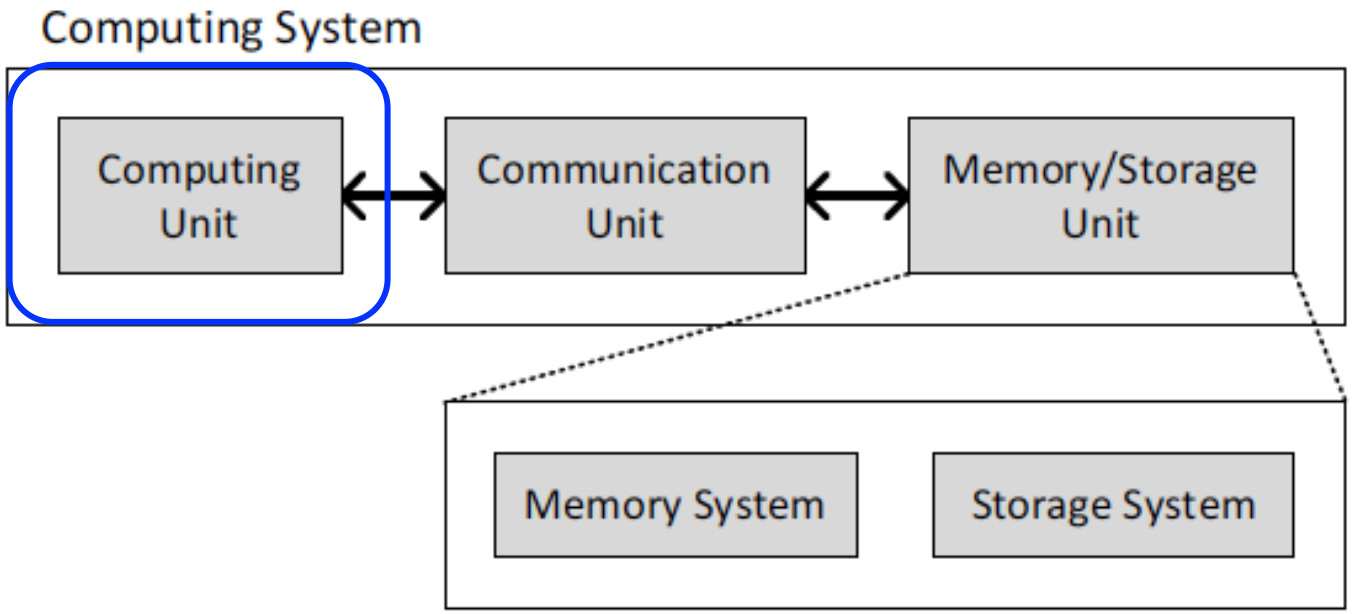
- Three key components
- Computation
- Communication
- Storage/memory

Burks, Goldstein, von Neumann, "Preliminary discussion of the logical design of an electronic computing instrument," 1946.



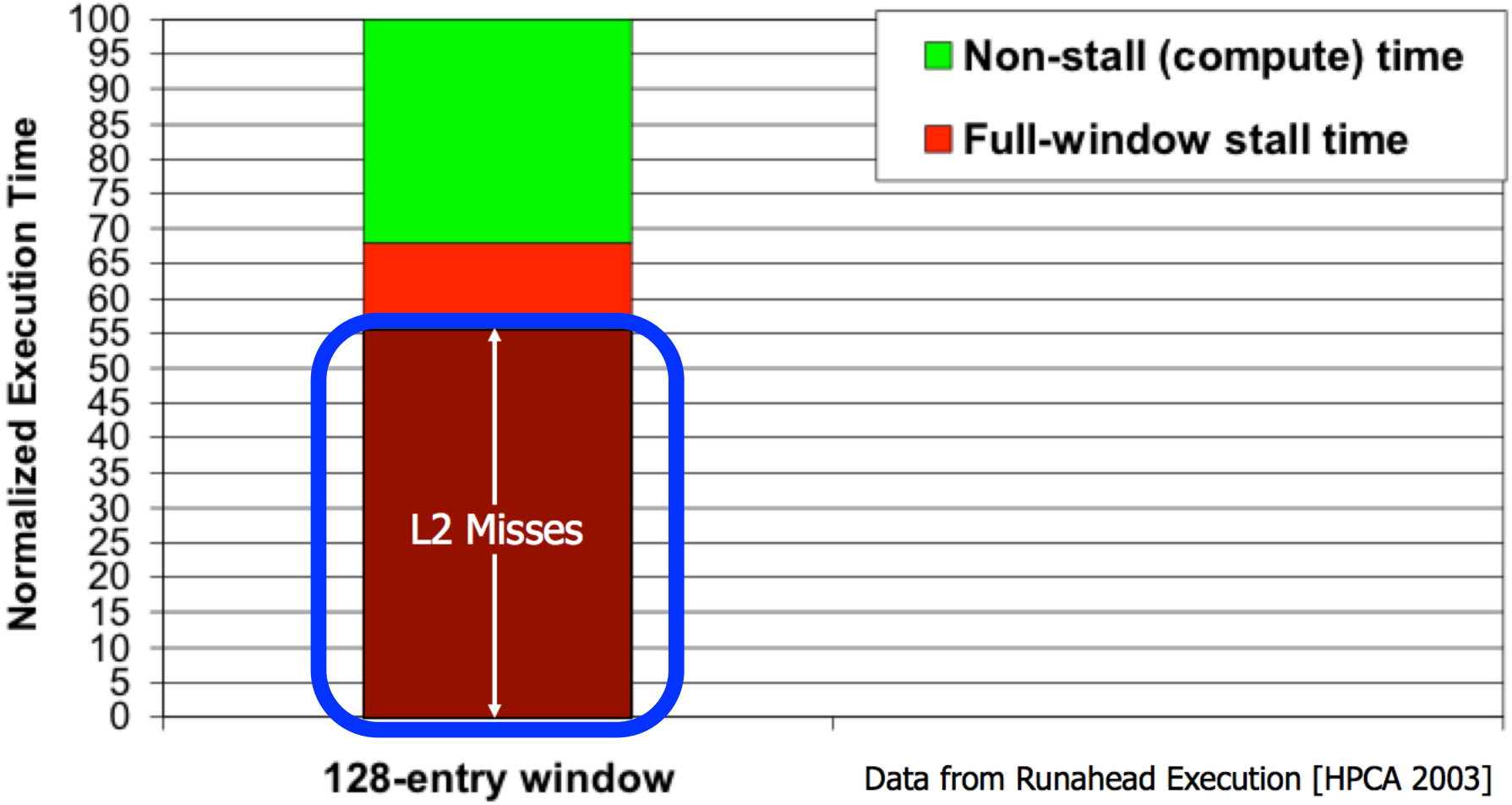
Today's Computing Systems

- Are overwhelmingly processor centric
- All data processed in the processor → at great system cost
- Processor is heavily optimized and is considered the master
- Data storage units are dumb slaves and are largely unoptimized (except for some that are on the processor die)



Yet …

- "**It's the Memory, Stupid!**" (Richard Sites, MPR, 1996)



Perils of Processor-Centric Design

- Grossly-imbalanced systems
  - Processing done only in **one place**
  - Everything else just stores and moves data: **data moves a lot**

  → Energy inefficient

  → Low performance

  → Complex

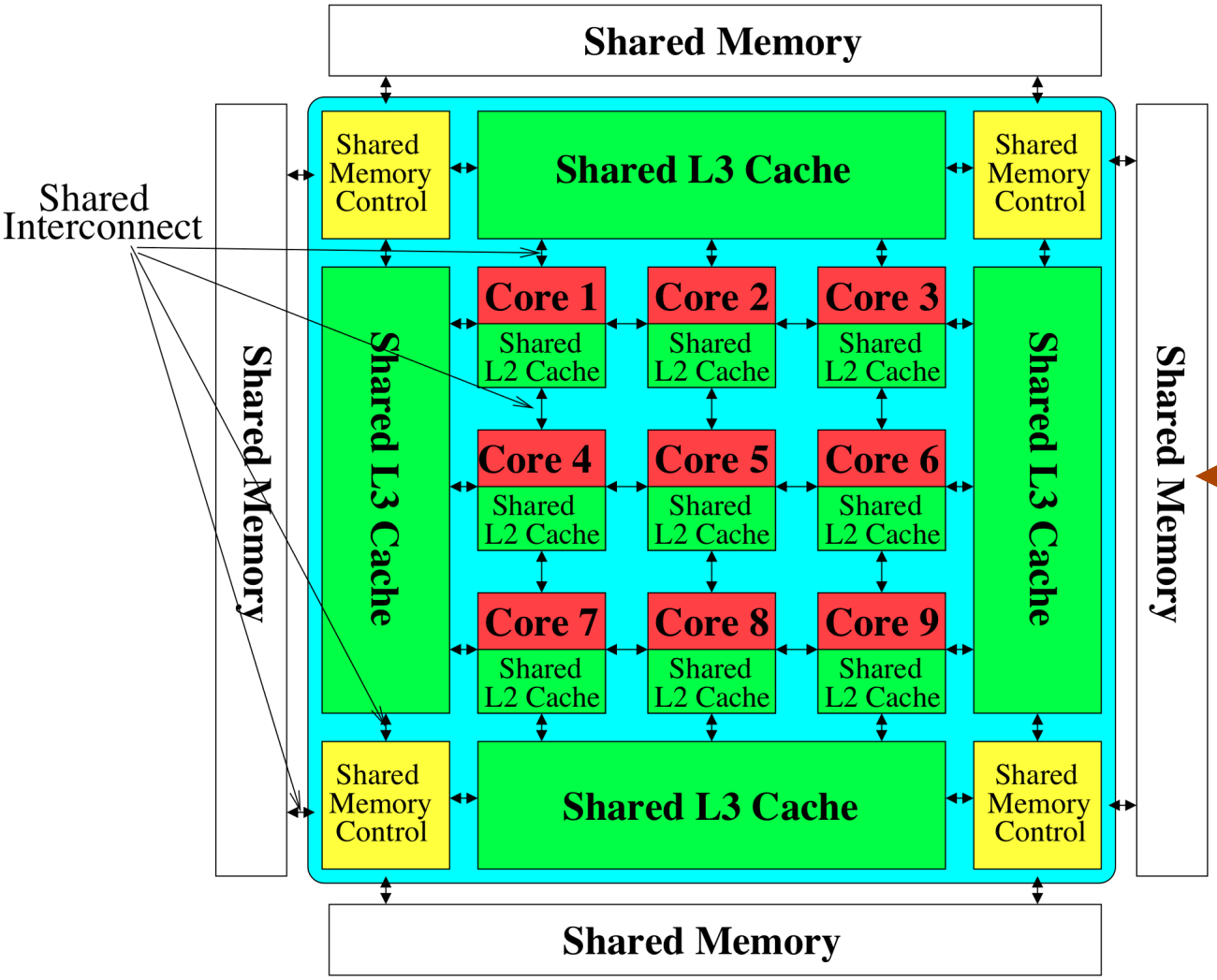
- Overly complex and bloated processor (and accelerators)
  - To tolerate data access from memory
  - Complex hierarchies and mechanisms

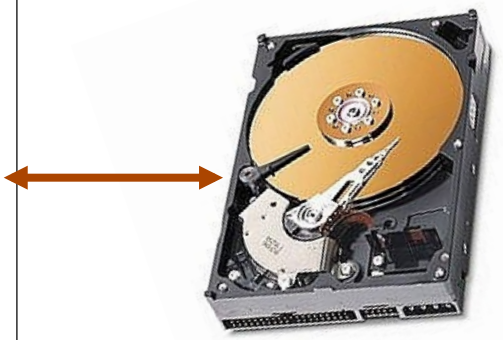
  → Energy inefficient

  → Low performance

  → Complex

Perils of Processor-Centric Design





Three Key Systems Trends

1. Data access is a major bottleneck

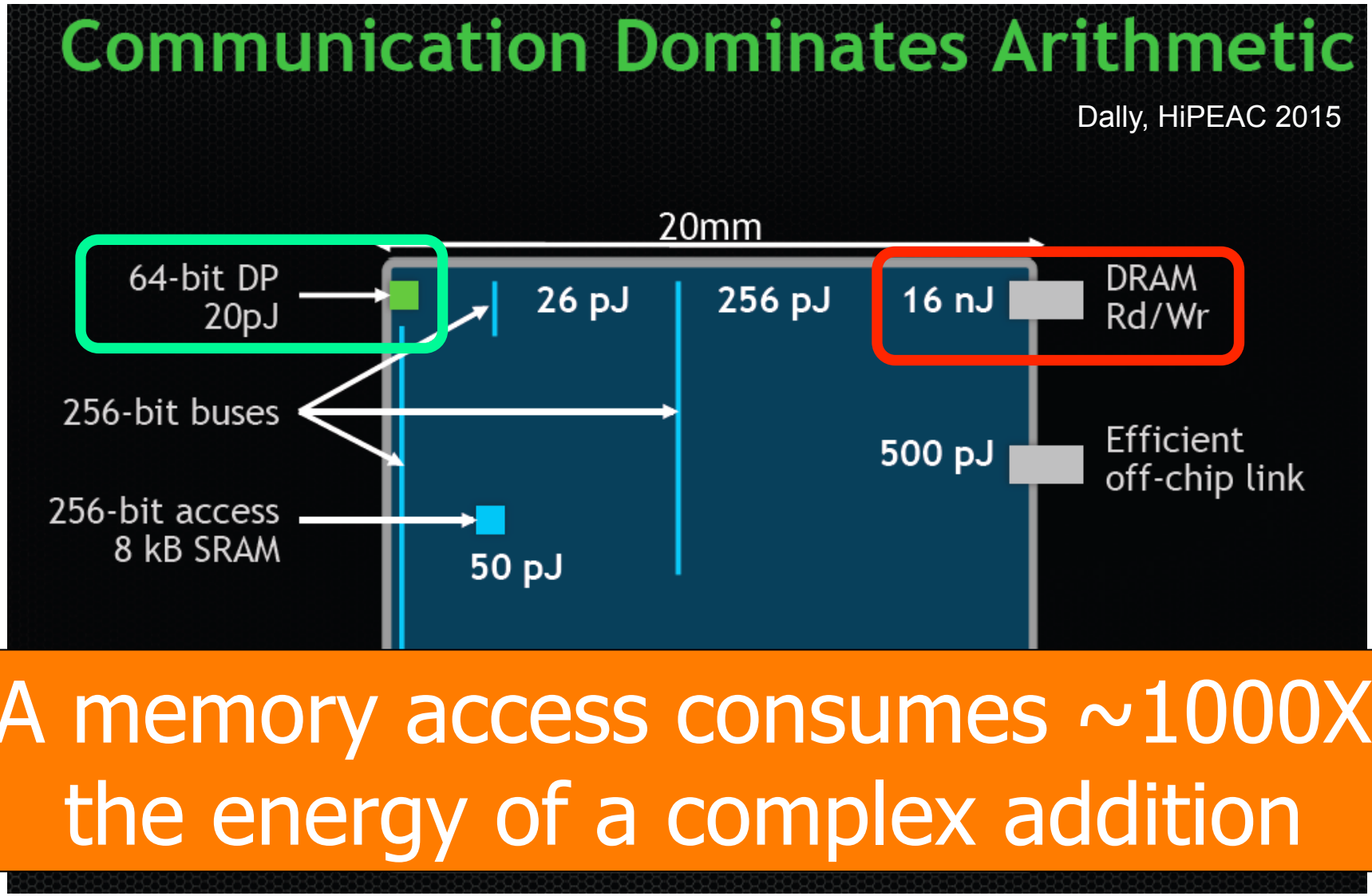
- Applications are increasingly data hungry

2. Energy consumption is a key limiter

3. Data movement energy dominates compute

- Especially true for off-chip to on-chip movement

Data Movement vs. Computation Energy

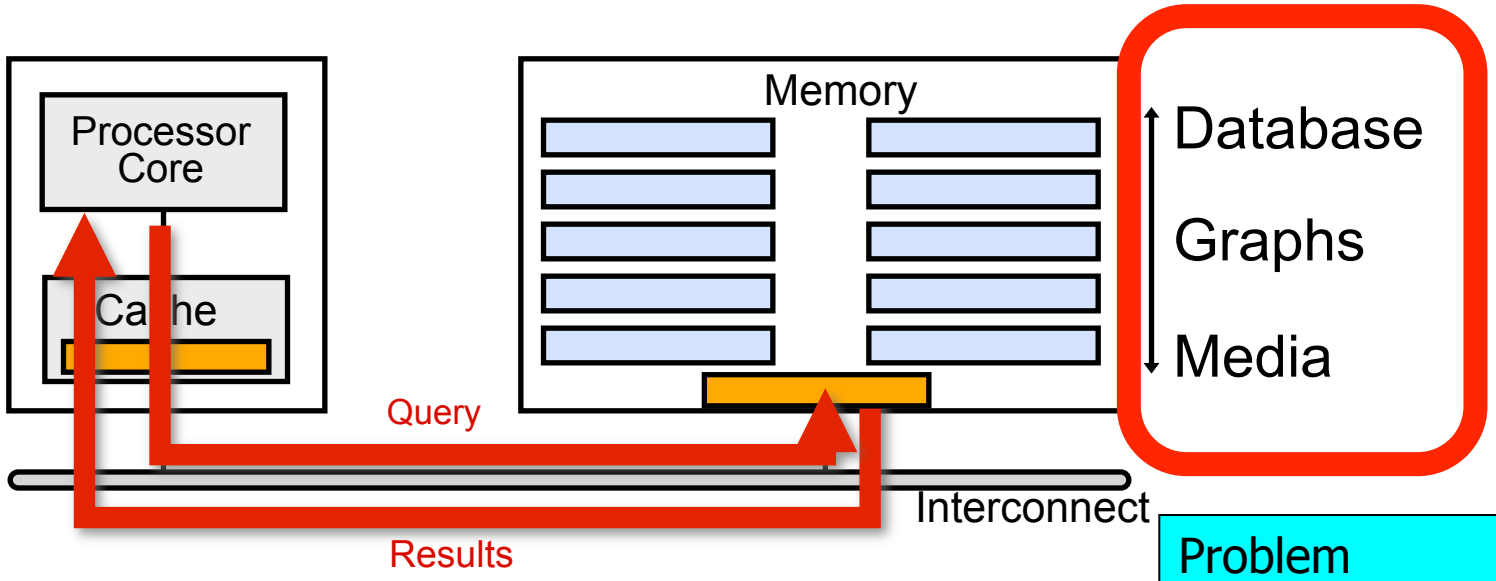


A memory access consumes ~1000X the energy of a complex addition

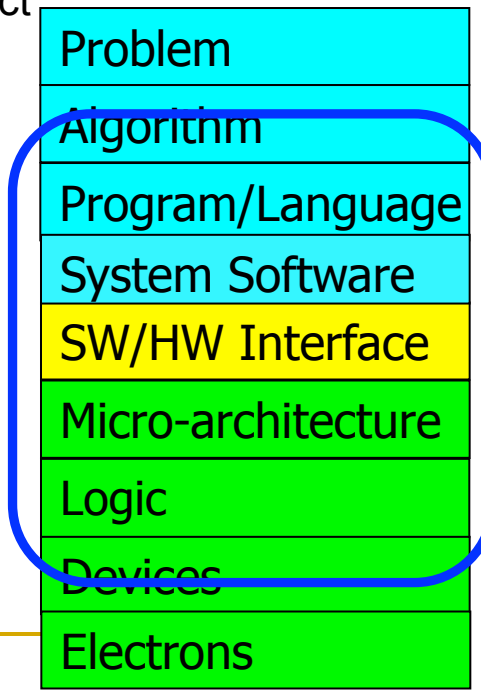
We Need A Paradigm Shift To …

- Enable computation with minimal data movement
- Compute where it makes sense (where data resides)
- Make computing architectures more data-centric

Goal: In-Memory Computation Engine

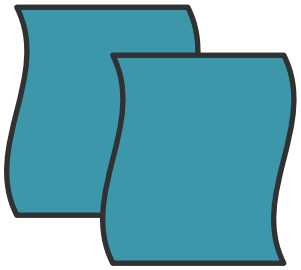


- Many questions … How do we design the:
  - compute-capable memory?
  - processor chip?
  - software interface?
  - system software and languages?
  - algorithms?

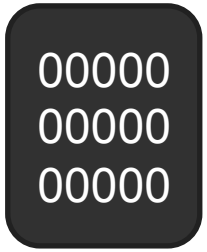


Starting Simple: Data Copy and Initialization

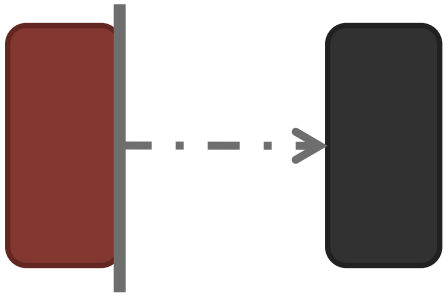
memmove & memcpy: 5% cycles in Google's datacenter [Kanev+ ISCA'15]



Forking

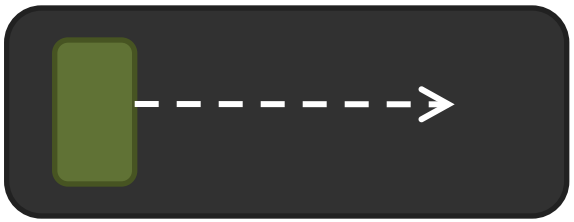


Zero initialization (e.g., security)



Checkpointing

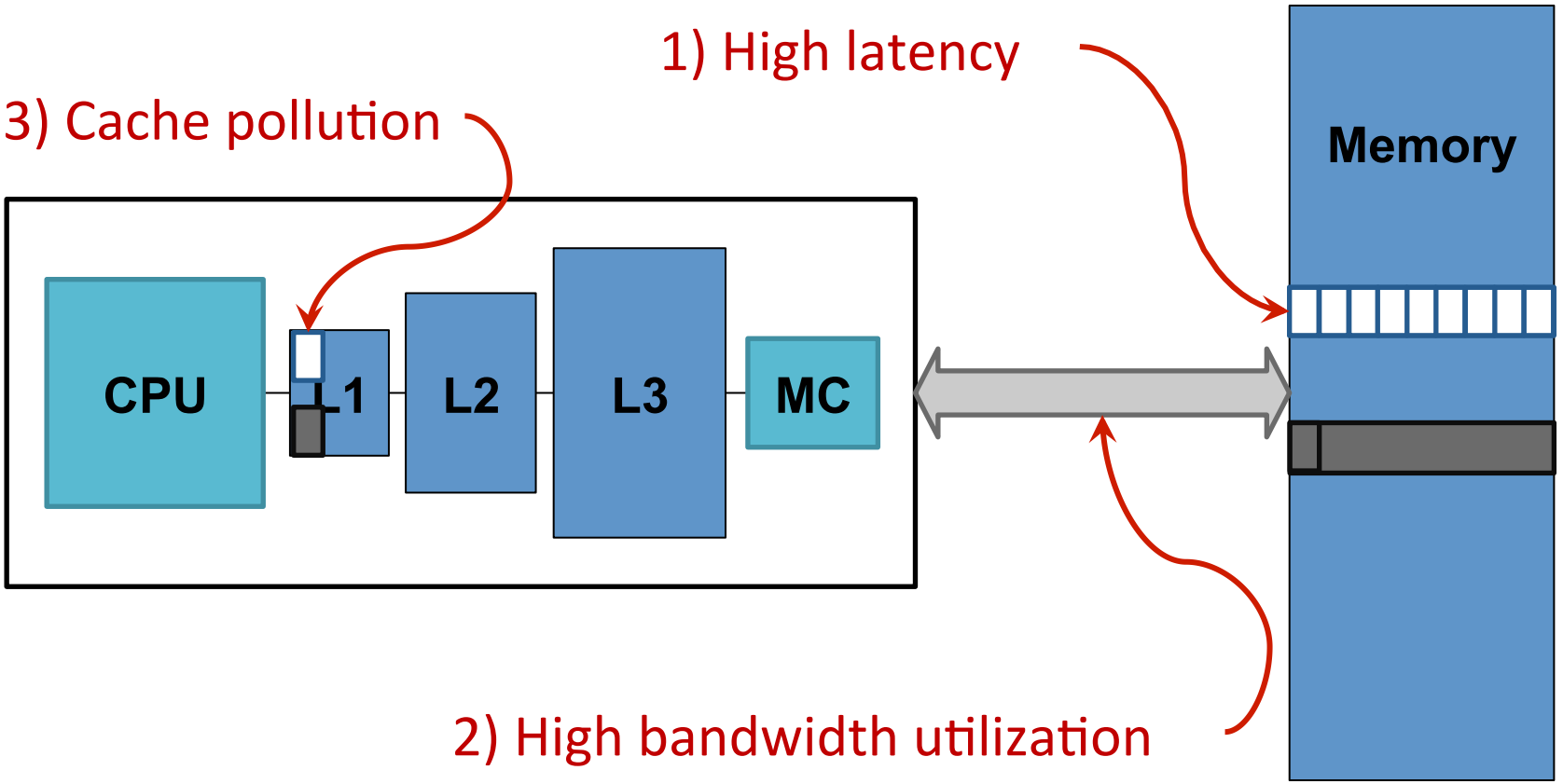
VM Cloning
Deduplication



Page Migration

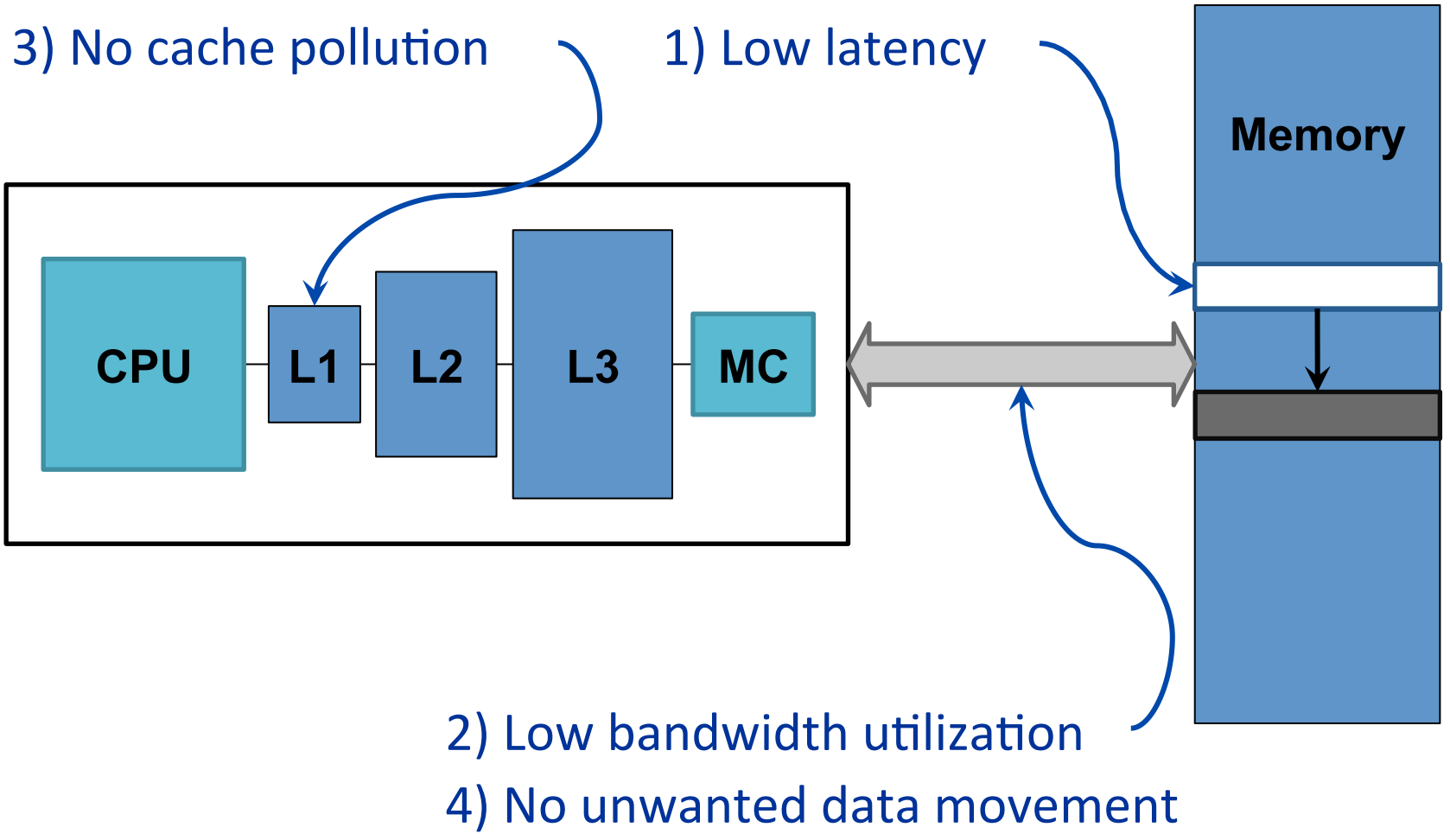
• • •
Many more

Today's Systems: Bulk Data Copy

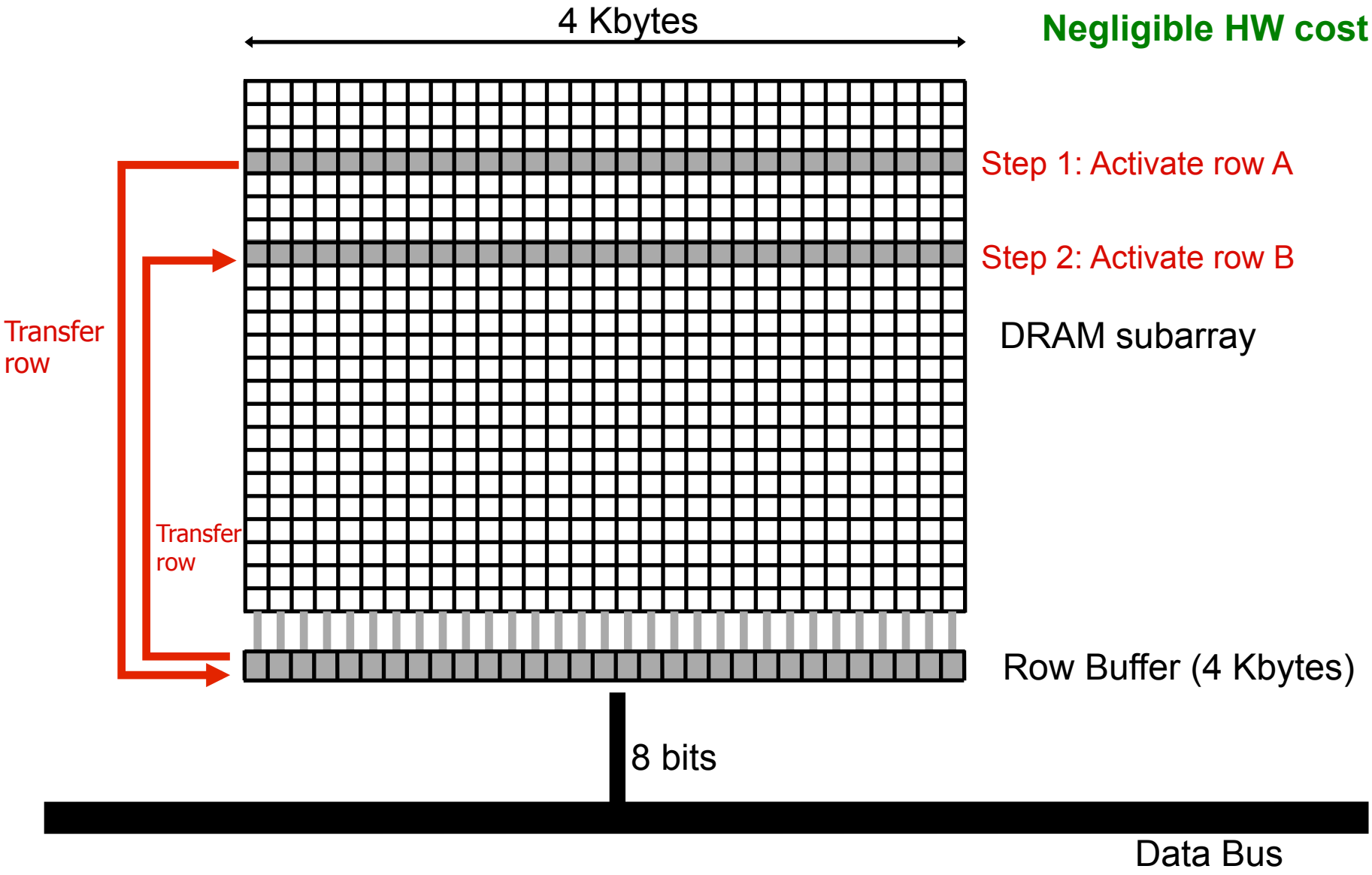


1046ns, 3.6uJ (for 4KB page copy via DMA)

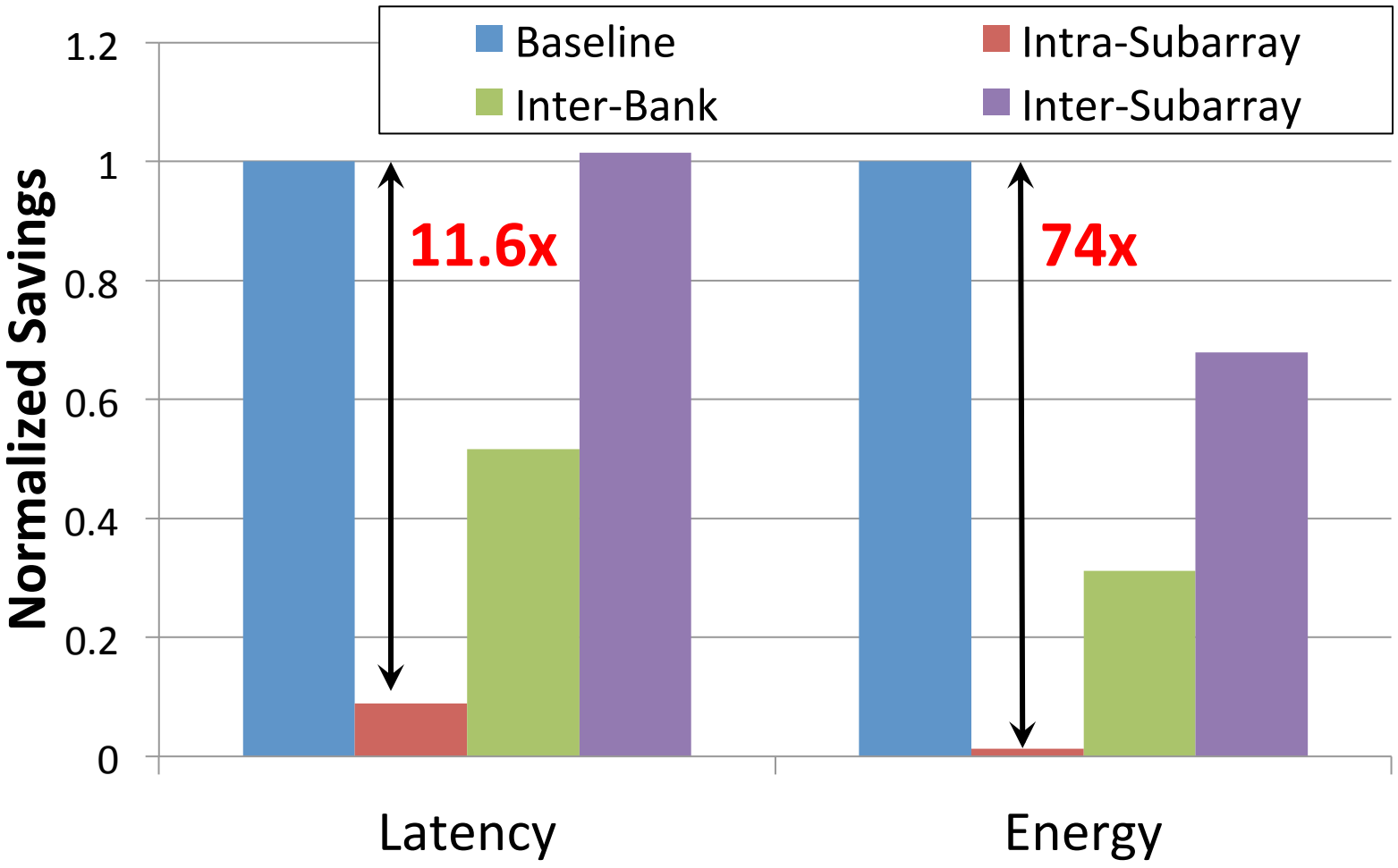
Future Systems: In-Memory Copy



RowClone: In-DRAM Row Copy



RowClone: Latency and Energy Savings



Seshadri et al., "RowClone: Fast and Efficient In-DRAM Copy and Initialization of Bulk Data," MICRO 2013.

(Truly) In-Memory Computation

- Similarly, we can support in-DRAM AND, OR, NOT
- At low cost
- Using analog behavior of memory
- 30-60X performance and energy improvement

- New memory technologies enable even more opportunities
  - Memristors, resistive RAM, phase change memory
  - Can operate on data with minimal movement

Another Example: In-Memory Graph Processing

- Large graphs are everywhere (circa 2015)

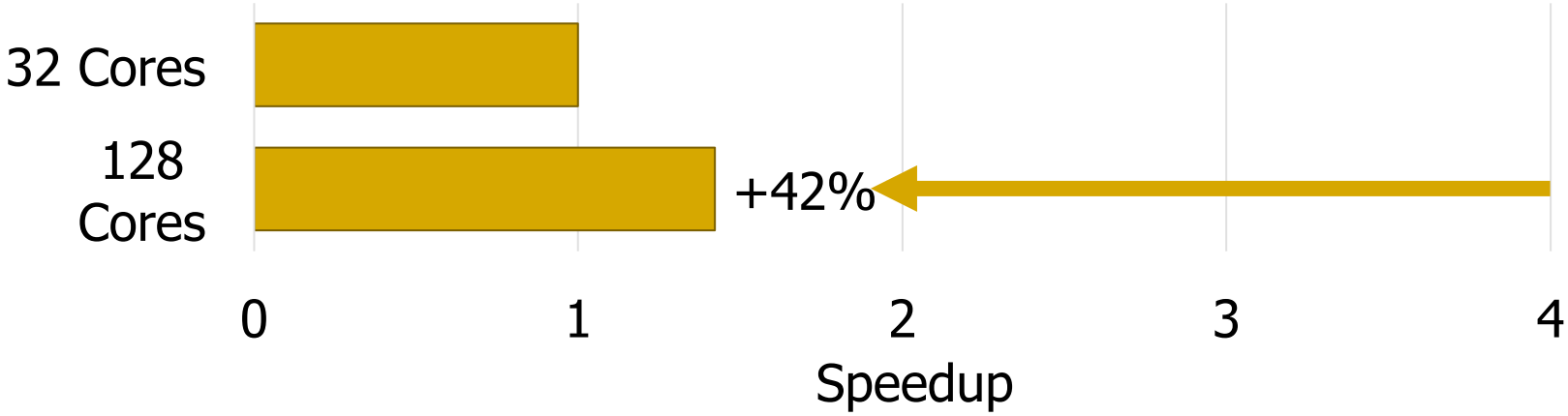
36 Million
Wikipedia Pages

1.4 Billion
Facebook Users

300 Million
Twitter Users

30 Billion
Instagram Photos

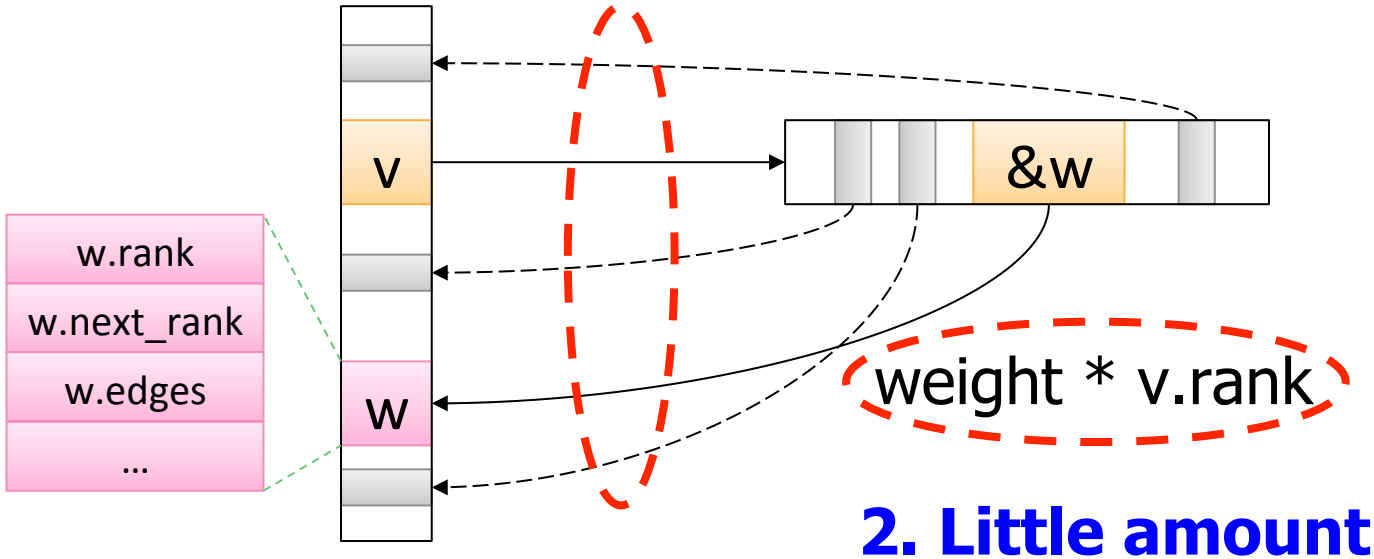
- Scalable large-scale graph processing is challenging



Key Bottlenecks in Graph Processing

```
for (v: graph.vertices) {
    for (w: v.successors) {
        w.next_rank += weight * v.rank;
    }
}
```

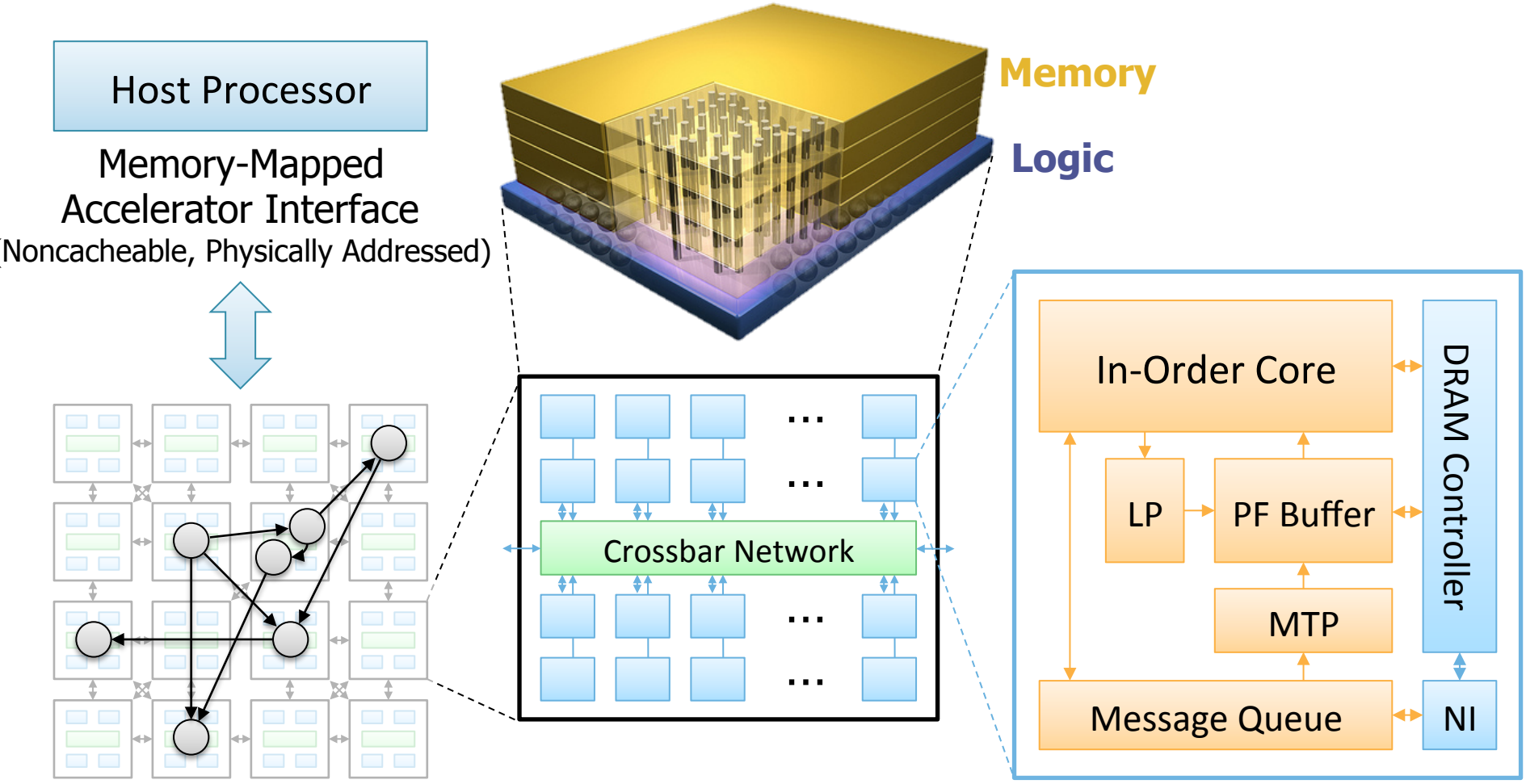
1. Frequent random memory accesses



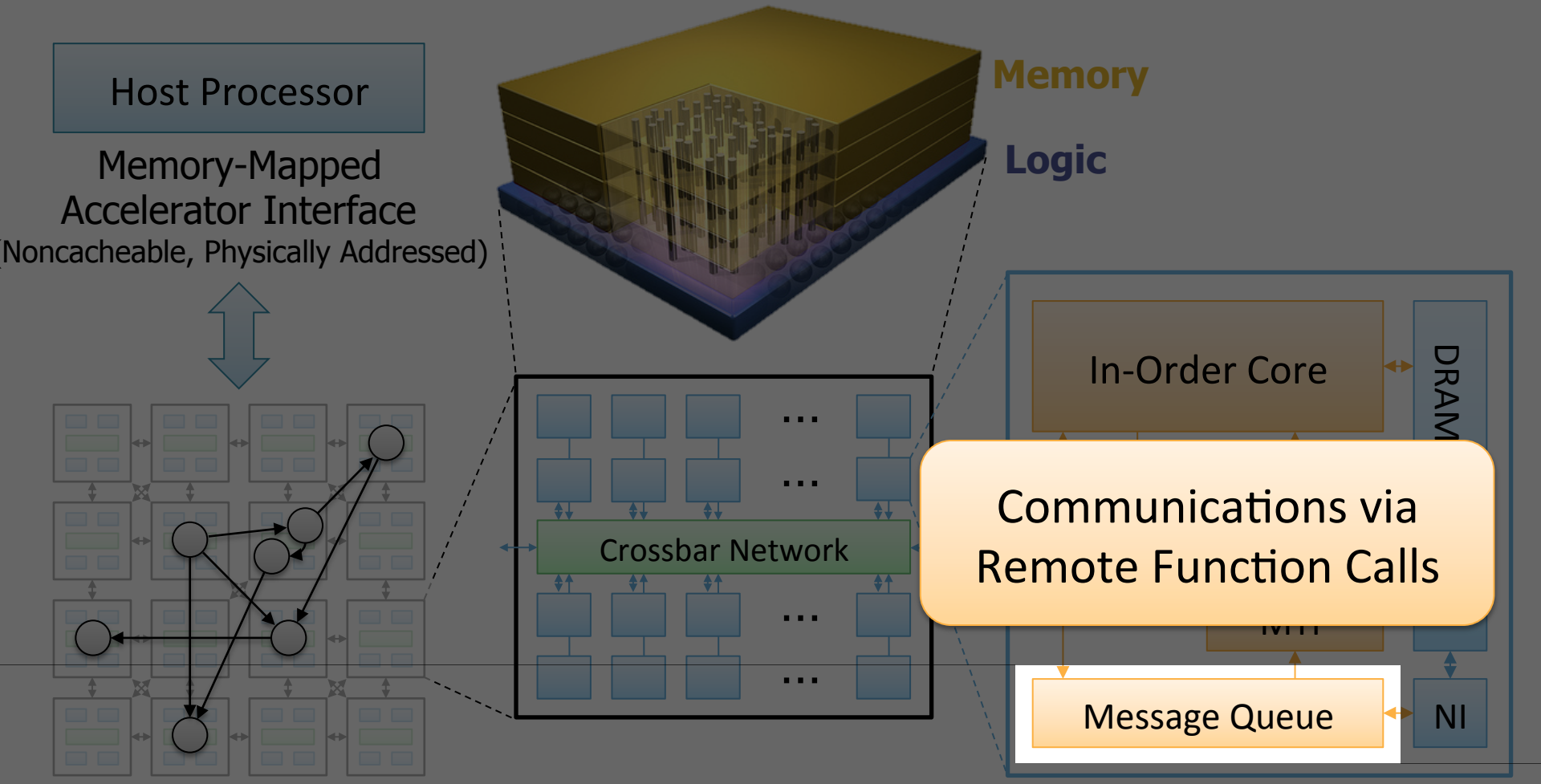
2. Little amount of computation

Tesseract System for Graph Processing

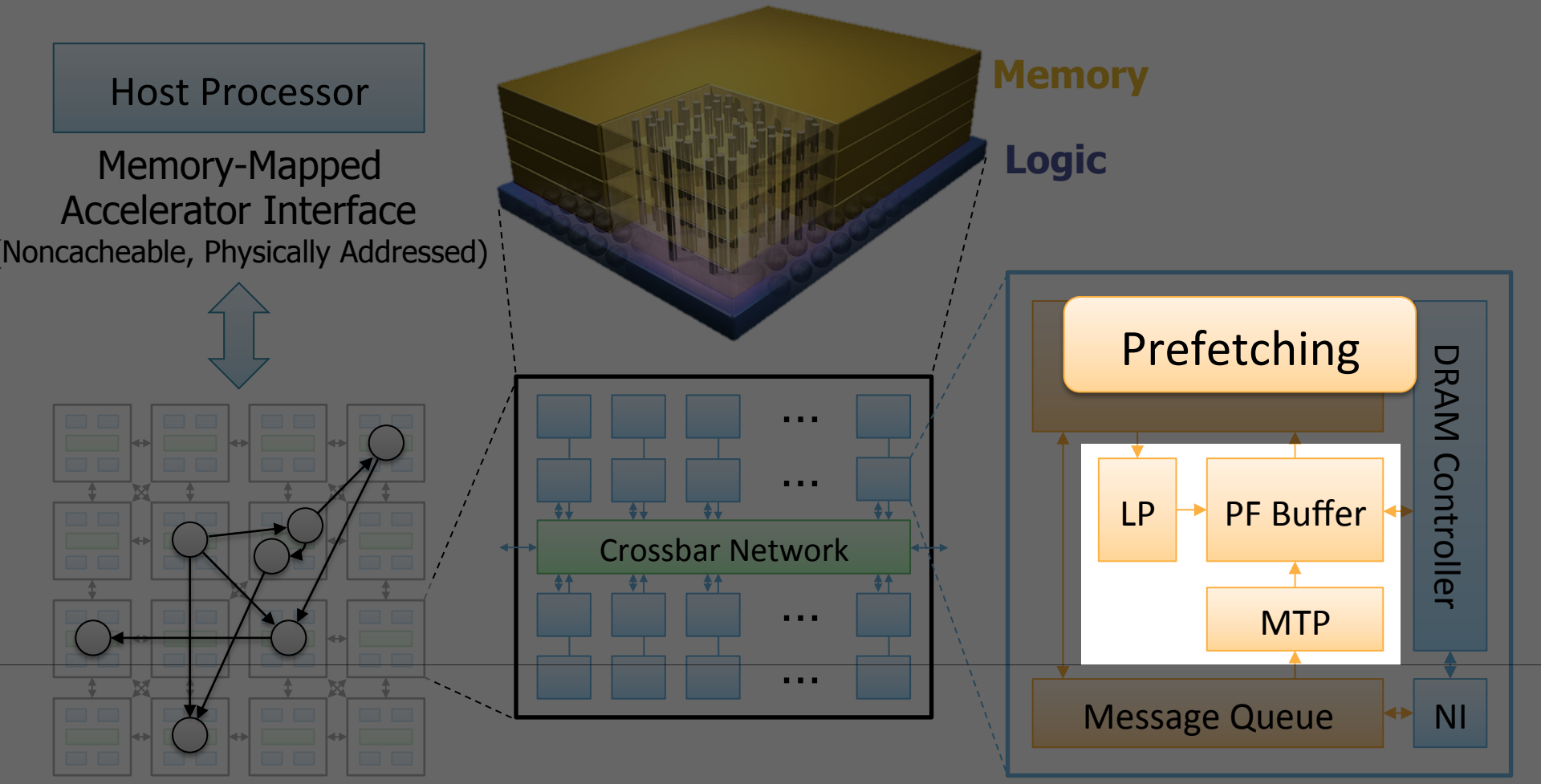
Interconnected set of 3D-stacked memory+logic chips with simple cores



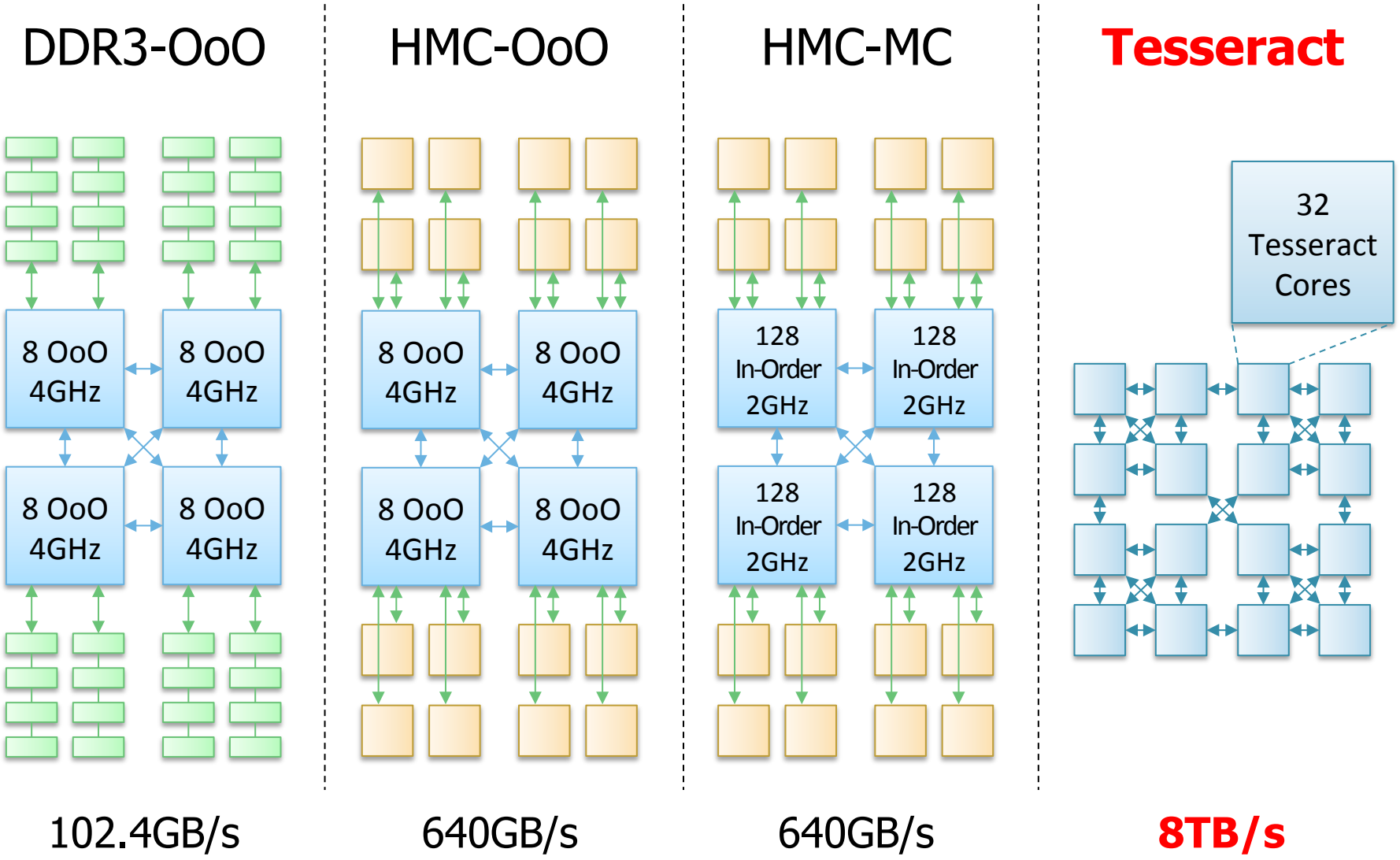
Tesseract System for Graph Processing



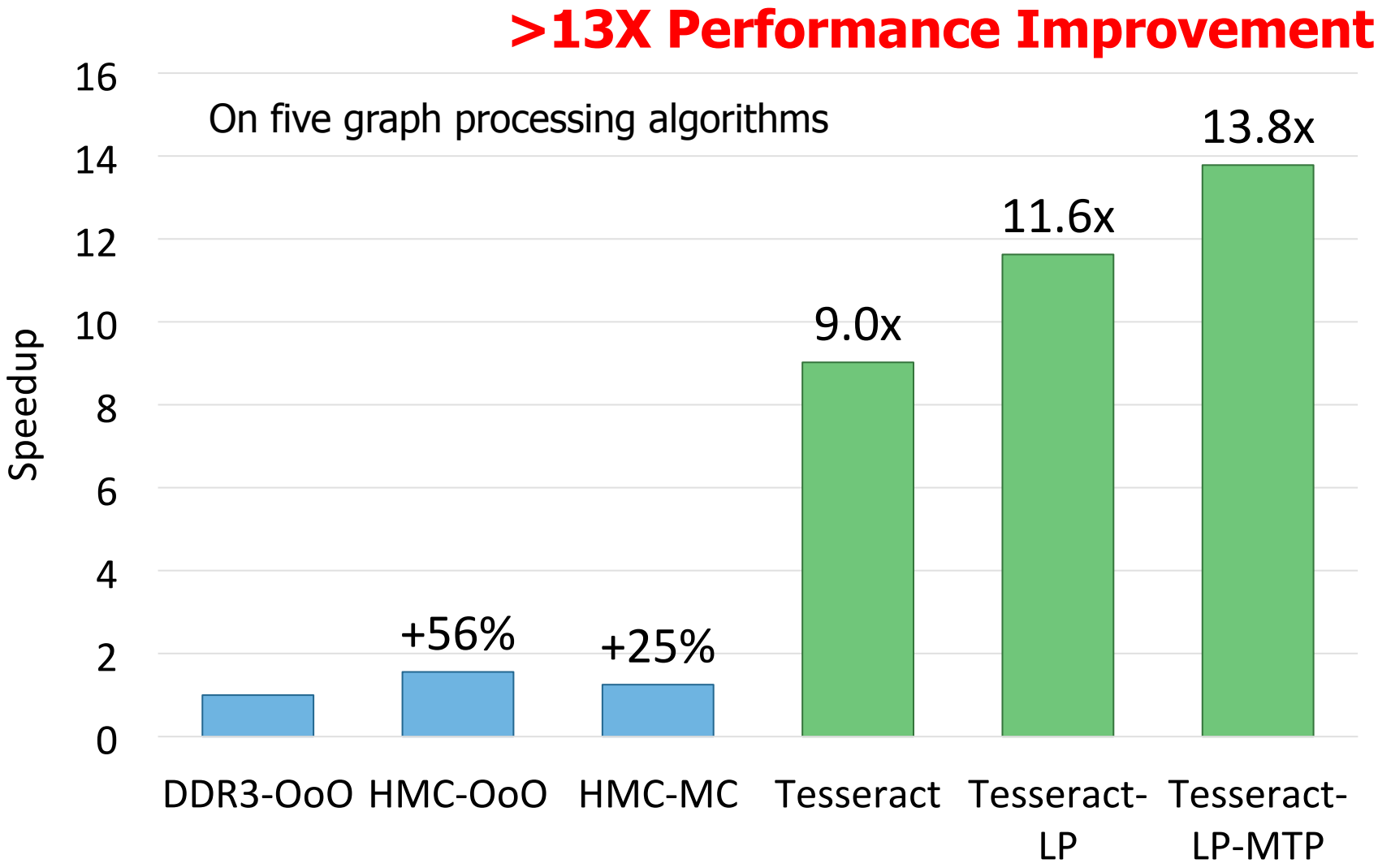
Tesseract System for Graph Processing



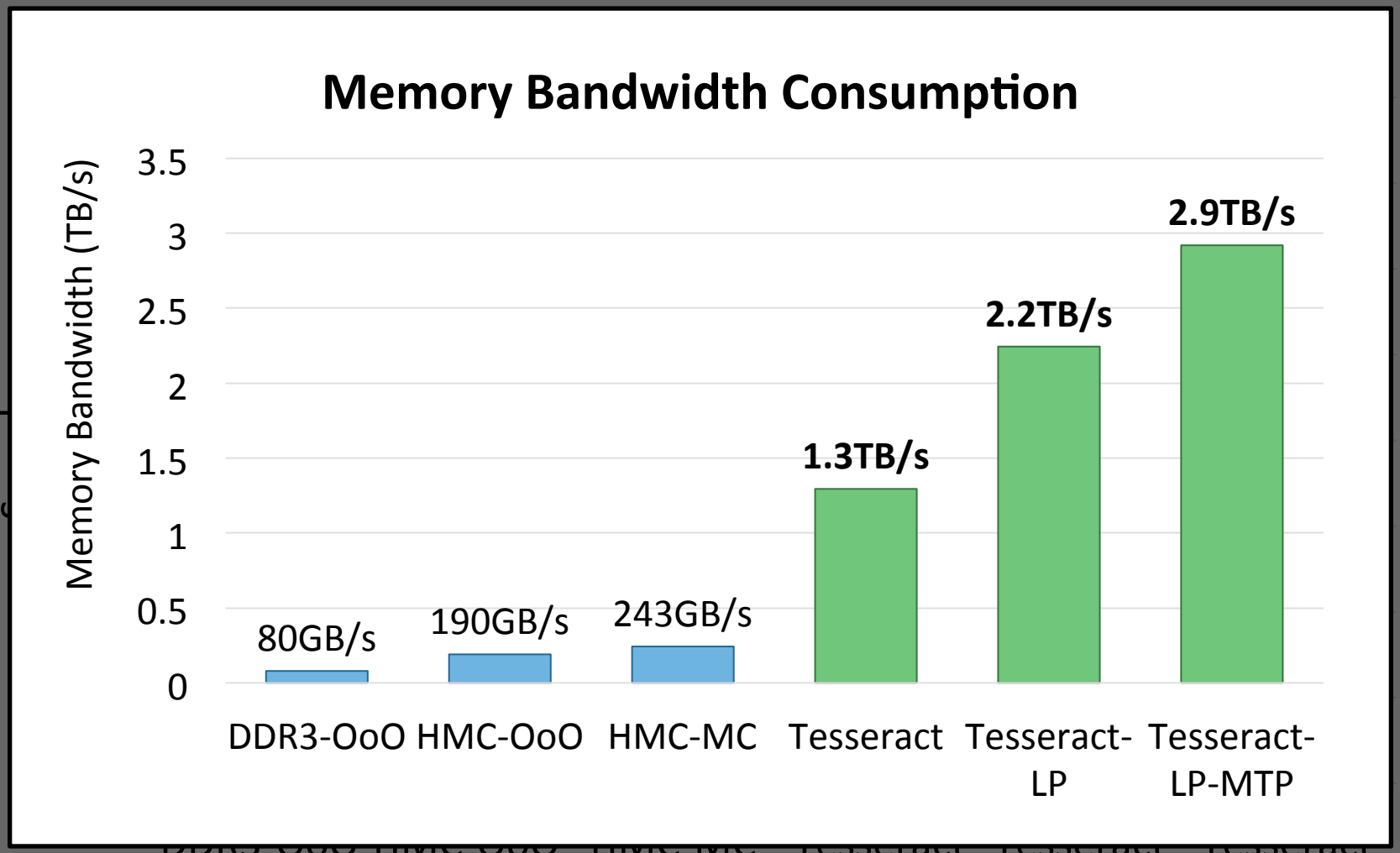
Evaluated Systems



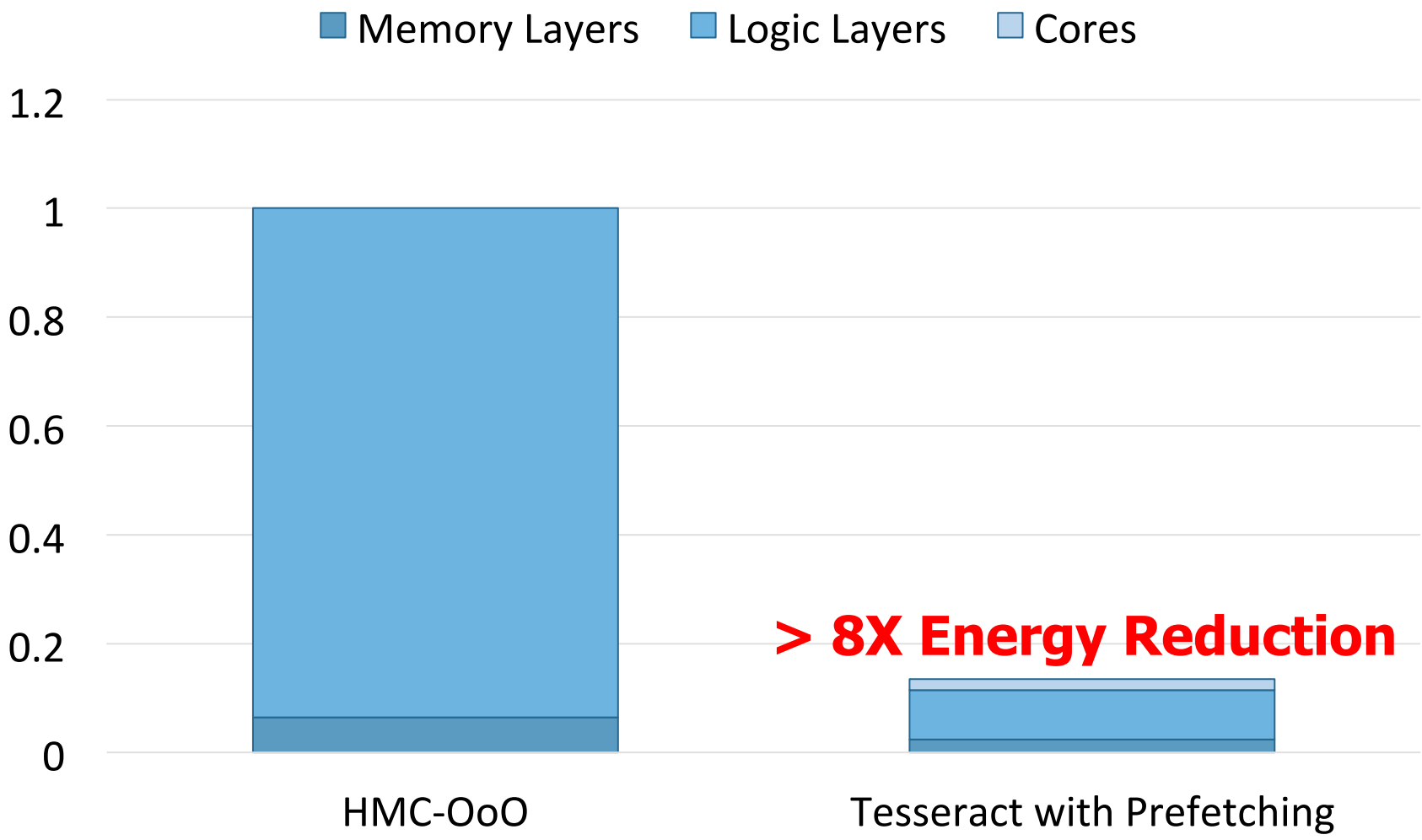
Tesseract Graph Processing Performance



Tesseract Graph Processing Performance



Tesseract Graph Processing Energy



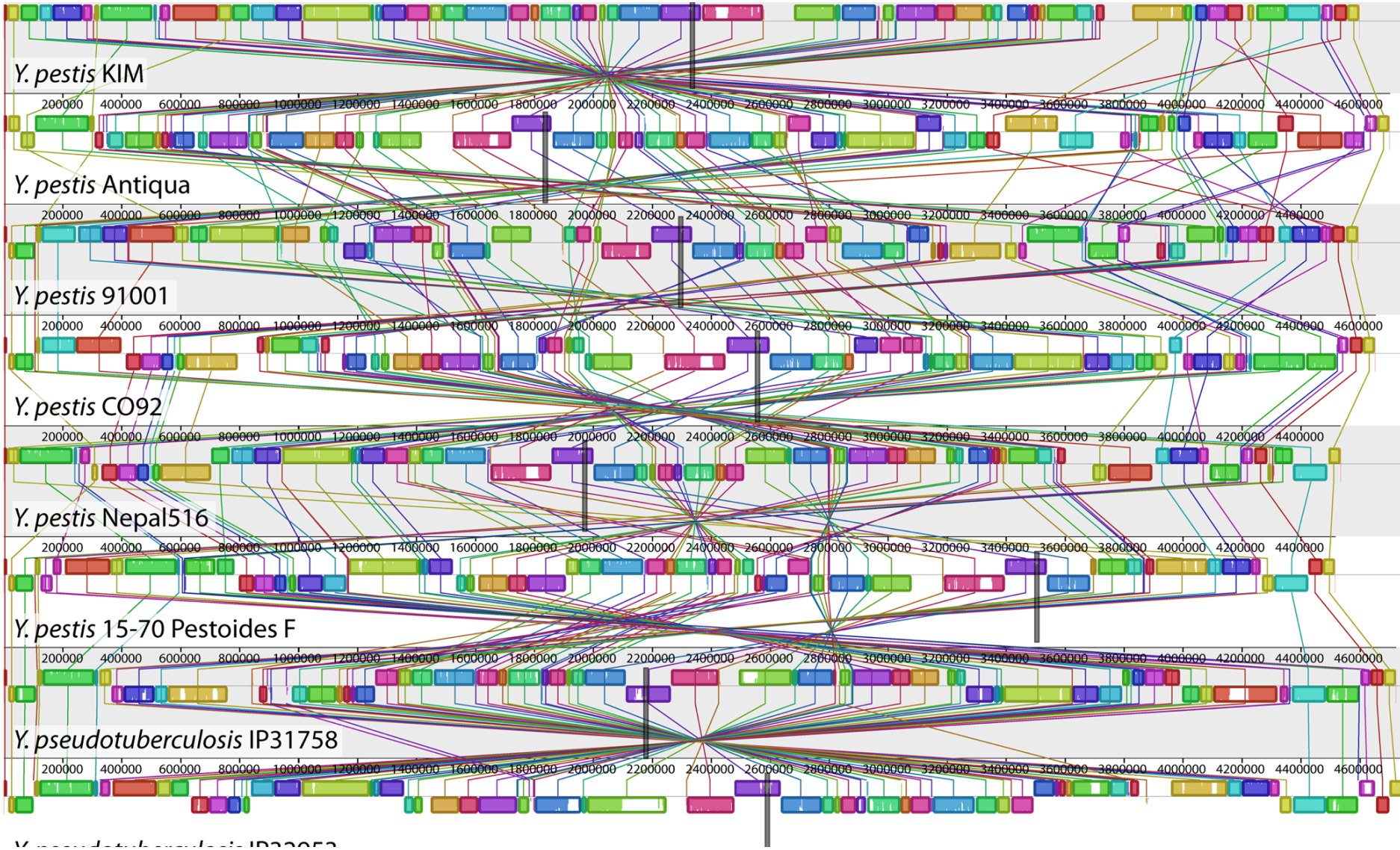
Challenge and Opportunity for Future

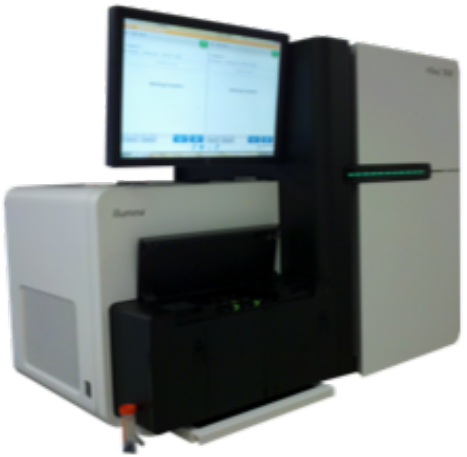
Fundamentally Energy-Efficient (Data-Centric) Computing Architectures

Three Key Issues in Future Platforms

- Fundamentally Secure/Reliable/Safe Architectures
- Fundamentally Energy-Efficient Architectures
  - Memory-centric (Data-centric) Architectures
- Architectures for Genomics, Medicine, Health

Genome Sequence Alignment

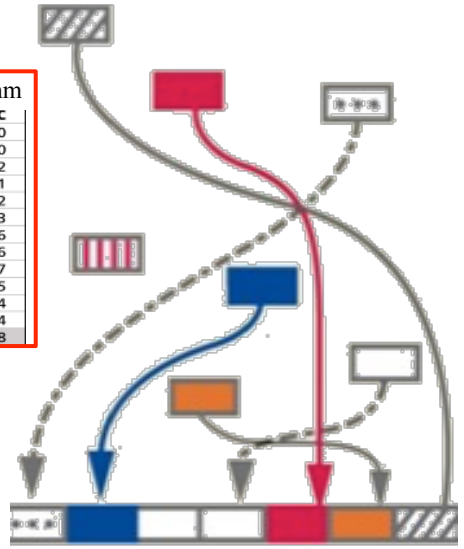




```
read1: ATCGCTTCC
read2: TATCGCTTC
read3: CTTCCATGA
read4: CGCTTCCAT
read5: CCATGACGC
read6: TTCCATGAC
```

e.g. Smith-Waterman alignment algorithm

		0	A	C	G	T	A	T	G	C
	0	0	0	0	0	0	0	0	0	0
A	0	0	2	0	0	0	2	0	0	0
C	0	0	0	4	2	1	0	1	0	2
G	0	0	0	2	6	4	3	2	3	1
A	0	2	1	4	5	6	4	3	2	2
A	0	2	1	3	3	7	5	4	3	3
C	0	2	4	2	2	5	6	4	6	4
C	0	0	0	2	3	1	4	4	5	6
C	0	0	0	2	1	2	3	3	3	7
T	0	0	0	0	1	3	2	5	3	5
T	0	0	0	0	0	3	2	4	4	4
G	0	0	0	0	2	1	2	2	6	4
C	0	0	0	2	0	1	0	1	4	8



1 Sequencing

Genome

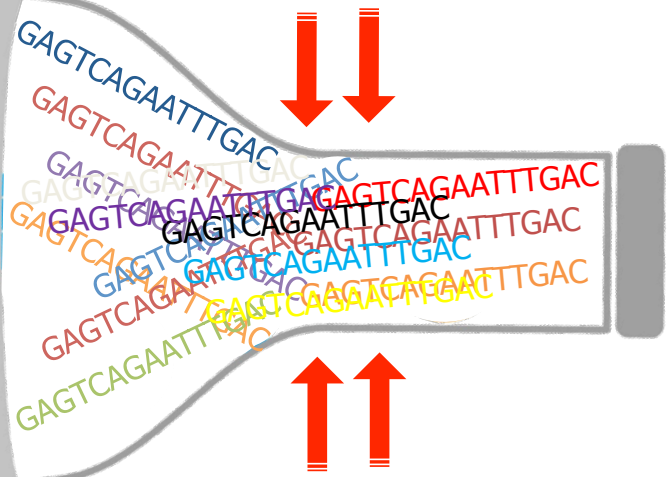
Mapping 2

Bottlenecked in Mapping!

Illumina HiSeq4000

300 M

bases/min



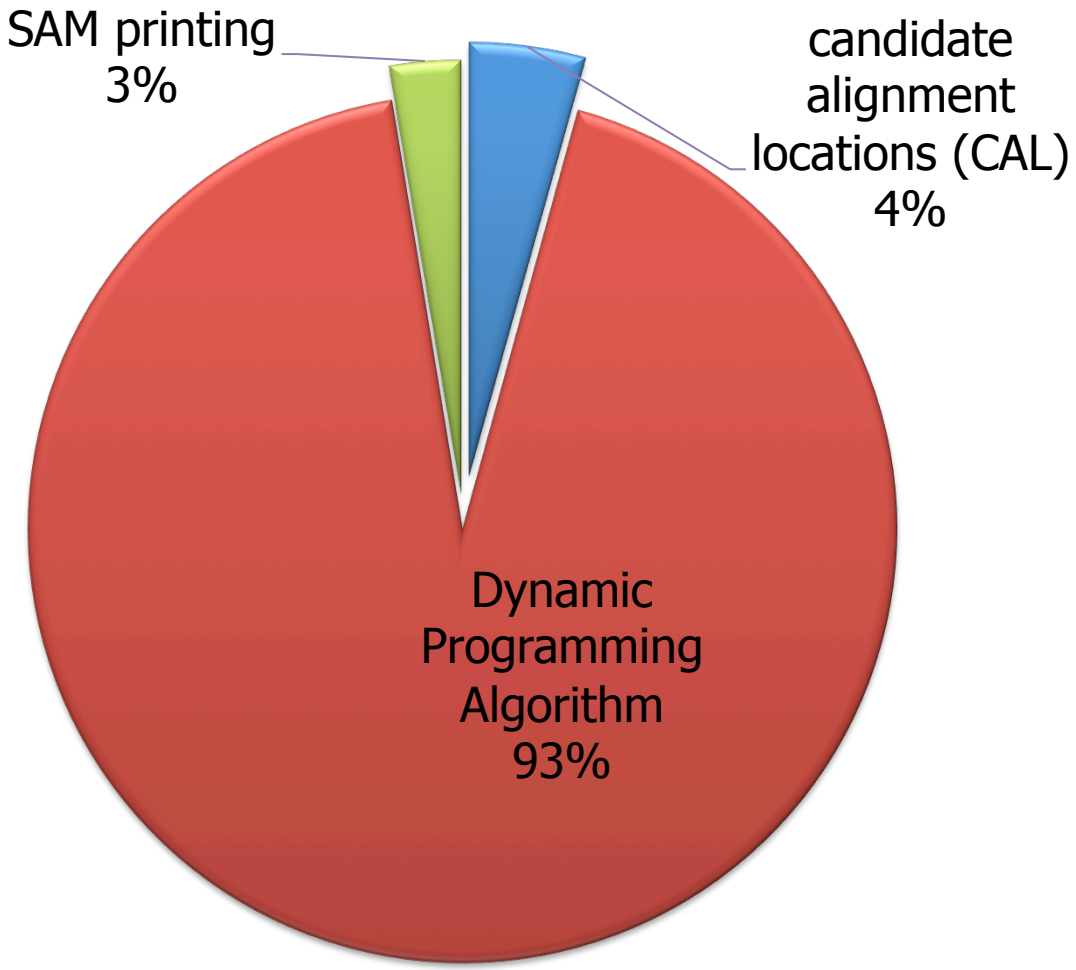
on average

2 M

bases/min

(0.6%)

Total Processing Time Breakdown



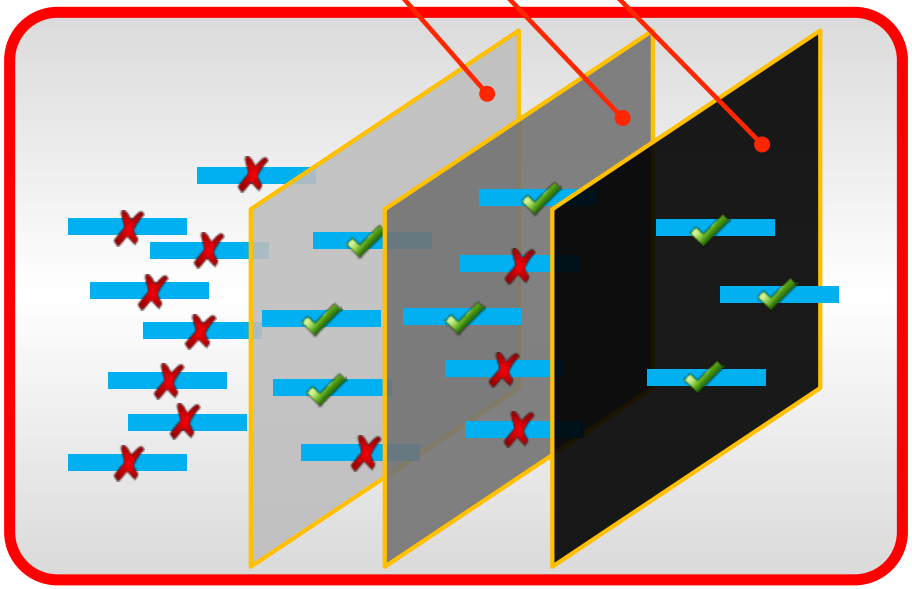
An Example Solution: GateKeeper

Novel Filter Algorithm

= 1st FPGA-based Alignment Filter.

Low Speed & High Accuracy
Medium Speed, Medium Accuracy
High Speed, Low Accuracy

$x10^{12}$ mappings



$x10^{3}$ mappings

Scoring Matrix

	0	A	C	G	T	A	T	G	C
0	0	0	0	0	0	0	0	0	0
A	0	2	0	0	0	2	0	0	0
C	0	0	4	2	1	0	1	0	2
G	0	0	2	6	4	3	2	3	1
A	0	2	1	4	5	6	4	3	2
A	0	2	1	3	3	7	5	4	3
C	0	2	4	2	2	5	6	4	6
C	0	0	2	3	1	4	4	5	6
C	0	0	2	1	2	3	3	3	7
T	0	0	0	1	3	2	5	3	5
T	0	0	0	0	3	2	4	4	4
G	0	0	0	2	1	2	2	6	4
C	0	0	2	0	1	0	1	4	8

Some Key Principles and Results

- Two key principles:
  - Exploit the structure of the genome to minimize computation
  - Morph and exploit the structure of the underlying hardware to maximize performance and efficiency

- Algorithm-architecture co-design for DNA read mapping
  - Improves performance by 20-100X
  - Improves accuracy of alignment in the presence of errors
  - Leads to a much more comprehensive read mapper

Xin et al., "Accelerating Read Mapping with FastHASH," BMC Genomics 2013.

Xin et al., "Shifted Hamming Distance: A Fast and Accurate SIMD-friendly Filter to Accelerate Alignment Verification in Read Mapping," Bioinformatics 2015.

Alser et al., "GateKeeper: A New Hardware Architecture for Accelerating Pre-Alignment in DNA Short Read Mapping," arxiv 2016.

Kim et al., "Genome Read In-Memory (GRIM) Filter," PSB 2017.

Concluding Remarks

A Quote from A Famous Architect

- "architecture [...] based upon principle, and not upon precedent"

Another Example: Precedent-Based Design

Source: http://cookiemagik.deviantart.com/art/Train-station-207266944

Principled Design

Source: By Toni_V, CC BY-SA 2.0, https://commons.wikimedia.org/w/index.php?curid=4087256

Principle Applied to Another Structure

Source: By 準建築人手札網站 Forgemind ArchiMedia - Flickr: IMG_2489.JPG, CC BY 2.0, https://commons.wikimedia.org/w/index.php?curid=37493386, https://en.wikipedia.org/wiki/Santiago_Calatrava

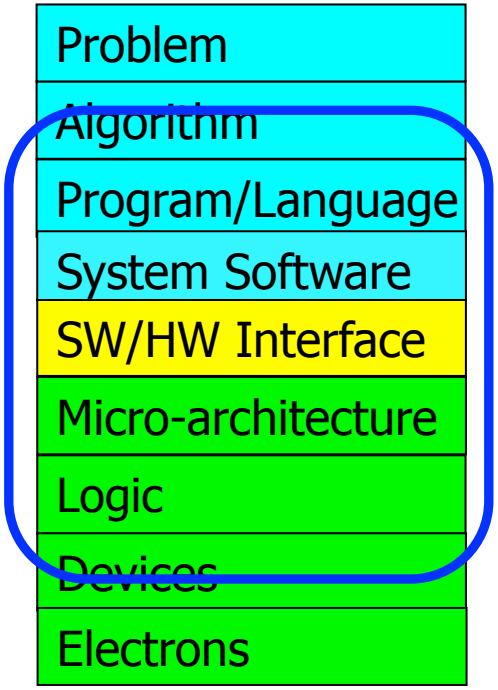
Source: https://www.dezeen.com/2016/08/29/santiago-calatrava-oculus-world-trade-center-transportation-hub-new-york-photographs-hufton-crow/

Concluding Remarks

- It is time to design principled computing architectures to achieve the highest security, performance, and efficiency
- Discover design principles for fundamentally secure and reliable computer architectures
- Design complete systems to be balanced, i.e., data-centric (or memory-centric)
- The expanded view of computer architecture can
  - Lead to **orders-of-magnitude** improvements
  - **Enable new applications & computing platforms**
  - **…**

The Future is Very Bright

- Regardless of challenges
  - in underlying technology and overlying problems/requirements



Future Computing Architectures

Onur Mutlu
onur.mutlu@inf.ethz.ch
https://people.inf.ethz.ch/omutlu

May 15, 2017

ETH Zurich Inaugural Lecture

SAFARI

ETH zürich

Acknowledgments

- My current and past students and postdocs
  - **Mohammed Alser**, Rachata Ausavarungnirun, Abhishek Bhowmick, Amirali Boroumand, Rui Cai, Yu Cai, Kevin Chang, Saugata Ghose, Kevin Hsieh, Tyler Huberty, Ben Jaiyen, Samira Khan, Jeremie Kim, **Yoongu Kim**, Yang Li, Jamie Liu, Lavanya Subramanian, Donghyuk Lee, Yixin Luo, Justin Meza, Gennady Pekhimenko, **Vivek Seshadri**, Lavanya Subramanian, Nandita Vijaykumar, Hongyi Xin, HanBin Yoon, Jishen Zhao, …
- My collaborators
  - **Junwhan Ahn**, Can Alkan, Chita Das, Phil Gibbons, Sriram Govindan, Norm Jouppi, Mahmut Kandemir, Mike Kozuch, Konrad Lai, Ken Mai, Todd Mowry, Yale Patt, Moinuddin Qureshi, Partha Ranganathan, Bikash Sharma, Kushagra Vaid, Chris Wilkerson, …
- ETH Zurich, CMU, Microsoft Research, Intel, Google, VMware, AMD

Funding Acknowledgments

- ETH Zurich
- NSF
- GSRC
- SRC
- CyLab
- AMD, Google, Facebook, HP Labs, Huawei, IBM, Intel, Microsoft, Nvidia, Oracle, Qualcomm, Rambus, Samsung, Seagate, VMware