

Memory Hierarchy (1): Inside Out the Computer Memory

Hung-Wei Tseng

Disney · PIXAR

INSIDE OUT

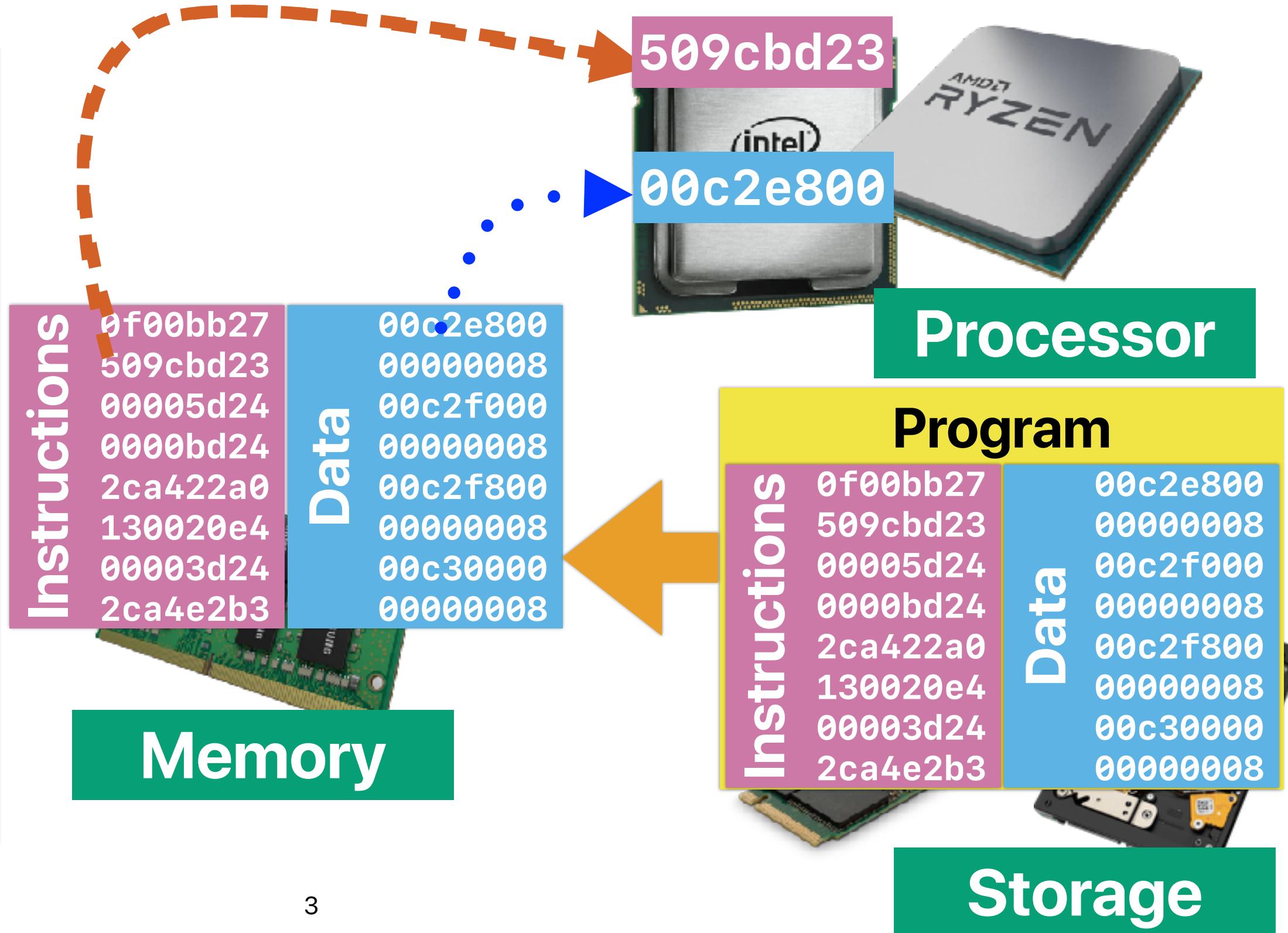
[GET DISNEY+](#)[▶ TRAILER](#)

PG 2015 • 1h 35m • Coming of age, Family, Animation

When 11-year-old Riley moves to a new city, her Emotions team up to help her through the transition. Joy, Fear, Anger, Disgust and Sadness work together, but when Joy and Sadness get lost, they must journey through unfamiliar places to get back home.



Recap: von Neumann Architecture



Recap: Speedup and Amdahl's Law?

- Definition of “Speedup of Y over X” or say Y is n times faster than X:

$$speedup_{Y_over_X} = n = \frac{Execution\ Time_X}{Execution\ Time_Y}$$

- Amdahl's Law — $Speedup_{enhanced}(f, s) = \frac{1}{(1-f) + \frac{f}{s}}$

- Corollary 1 — each optimization has an upper bound

$$Speedup_{max}(f, \infty) = \frac{1}{(1-f)}$$

- Corollary 2 — make the common case (the most time consuming case) fast!

$$Speedup_{max}(f_1, \infty) = \frac{1}{(1-f_1)}$$

$$Speedup_{max}(f_2, \infty) = \frac{1}{(1-f_2)}$$

$$Speedup_{max}(f_3, \infty) = \frac{1}{(1-f_3)}$$

$$Speedup_{max}(f_4, \infty) = \frac{1}{(1-f_4)}$$

- Corollary 3: Optimization has a moving target

- Corollary 4: Exploiting more parallelism from a program is the key to performance gain in modern architectures

$$Speedup_{parallel}(f_{parallelizable}, \infty) = \frac{1}{(1-f_{parallelizable})}$$

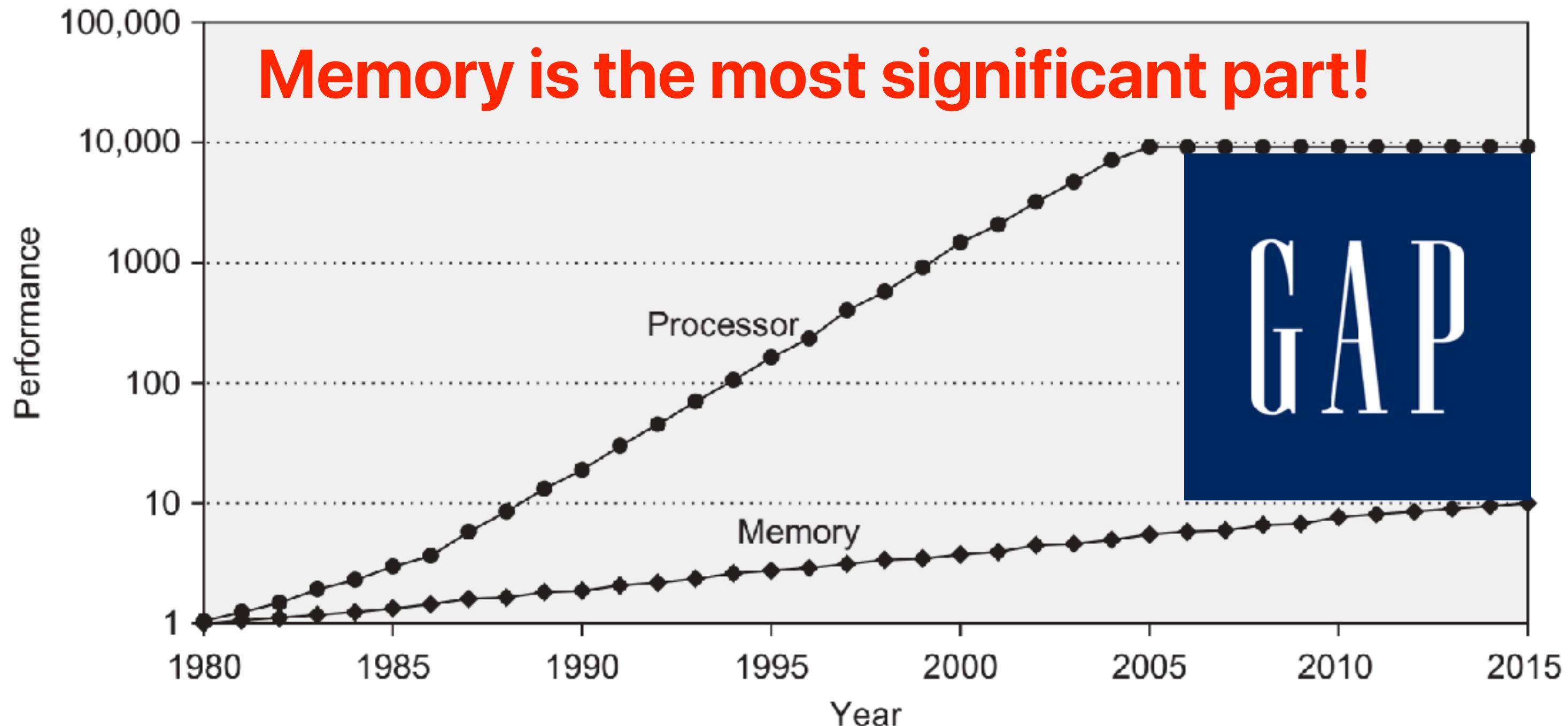
- Corollary 5: Single-core performance still matters

$$Speedup_{parallel}(f_{parallelizable}, \infty) = \frac{1}{(1-f_{parallelizable})}$$

- Corollary 6: Don't hurt the non-common case too much

$$Speedup_{enhanced}(f, s, r) = \frac{1}{(1-f) + perf(r) + \frac{f}{s}}$$

Recap: Performance gap between Processor/Memory



Outline

- Choosing the right metric (cont.)
- Why memory hierarchy
- The “predictable” code behavior
- Designing a cache that captures the predictability

How reflective is FLOPS?

- Given the FLOPS number measured, how many of the followings are true?
 - The FLOPS number remains the same on each architecture if we change the data size
 - The FLOPS number remains the same on each architecture if we change the data type to double
 - The FLOPS number remains the same on each architecture if we change the algorithm implementation
 - The FLOPS number reflects the performance ratio of different architectures when executing floating point applications
- A. 0
B. 1
C. 2
D. 3
E. 4

How reflective is FLOPS?

- Given the FLOPS number measured, how many of the followings are true?
 - The FLOPS number ~~remains the same~~ on each architecture if we change the data size
 - The FLOPS number ~~remains the same~~ on each architecture if we change the data type to double
 - The FLOPS number ~~remains the same~~ on each architecture if we change the algorithm implementation
 - The FLOPS number ~~reflects~~ the performance ratio of different architectures when executing floating point applications
- A. 0
- B. 1
- C. 2
- D. 3
- E. 4

Is TFLOPS (Tera FLoating-point Operations Per Second) a good metric?

$$\begin{aligned}TFLOPS &= \frac{\# \text{ of floating point instructions} \times 10^{-12}}{\text{Execution Time}} \\&= \frac{IC \times \% \text{ of floating point instructions} \times 10^{-12}}{IC \times CPI \times CT} \\&= \frac{\% \text{ of floating point instructions} \times 10^{-12}}{CPI \times CT}\end{aligned}$$

IC is gone!

- Cannot compare different ISA/compiler
 - What if the compiler can generate code with fewer instructions?
 - What if new architecture has more IC but also lower CPI?
- Does not make sense if the application is not floating point intensive

NVIDIA Ships World's Most Advanced AI System — NVIDIA DGX A100 — to Fight COVID-19; Third- Generation DGX Packs Record 5 Petaflops of AI Performance

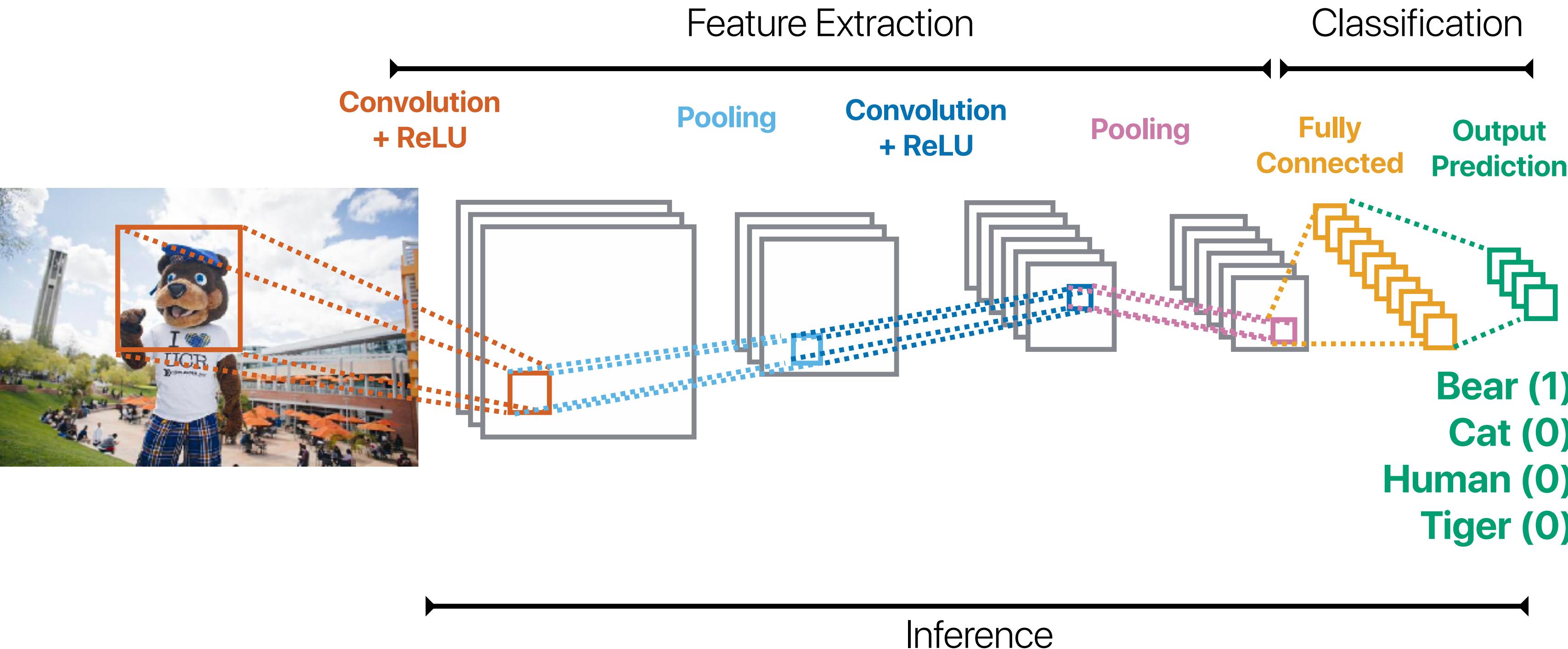
Training, Inference
from One to 56 In
Infrastructure

May 14, 2020

Technical Specifications			
	H100 SXM	H100 PCIe	H100 NVL ¹
FP64	34 teraFLOPS	26 teraFLOPS	68 teraFLOPS
FP64 Tensor Core	67 teraFLOPS	51 teraFLOPS	134 teraFLOPS
FP32	67 teraFLOPS	51 teraFLOPS	134 teraFLOPS
TF32 Tensor Core	989 teraFLOPS ²	756 teraFLOPS ²	1,979 teraFLOPS ²
BFLOAT16 Tensor Core	1,979 teraFLOPS ²	1,513 teraFLOPS ²	3,958 teraFLOPS ²
FP16 Tensor Core	1,979 teraFLOPS ²	1,513 teraFLOPS ²	3,958 teraFLOPS ²
FP8 Tensor Core	3,958 teraFLOPS ²	3,026 teraFLOPS ²	7,916 teraFLOPS ²
INT8 Tensor Core	3,958 TOPS ²	3,026 TOPS ²	7,916 TOPS ²
GPU memory	80GB	80GB	188GB
GPU memory bandwidth	3.35TB/s	2TB/s	7.8TB/s ³
Decoders	7 NVDEC 7 JPEG	7 NVDEC 7 JPEG	14 NVDEC 14 JPEG
Max thermal design power (TDP)	Up to 700W (configurable)	300-350W (configurable)	2x 350-400W (configurable)

> 5 PetaFLOPS
Only @ 8-bit
floating point

The Machine Learning Inference Pipeline



gemini.google.com

Gemini v.s. ChatGPT Round #1

Gemini was just updated. See update

Hello, Hung-Wei

How can I help you today?

Suggest a Python library to solve a problem

Brainstorm presentation ideas about a topic

Help me comp these college r

1 answer(inference)/6 secs

67 words/6 secs

Enter a prompt here

Gemini may display inaccurate info, including about people, so double-check its responses. [Your privacy & Gemini Apps](#)

chat.openai.com

ChatGPT Round #1



我今天可以如何協助你？

1 answer(inference)/6 secs

93 words/6 secs

傳訊息給 ChatGPT....

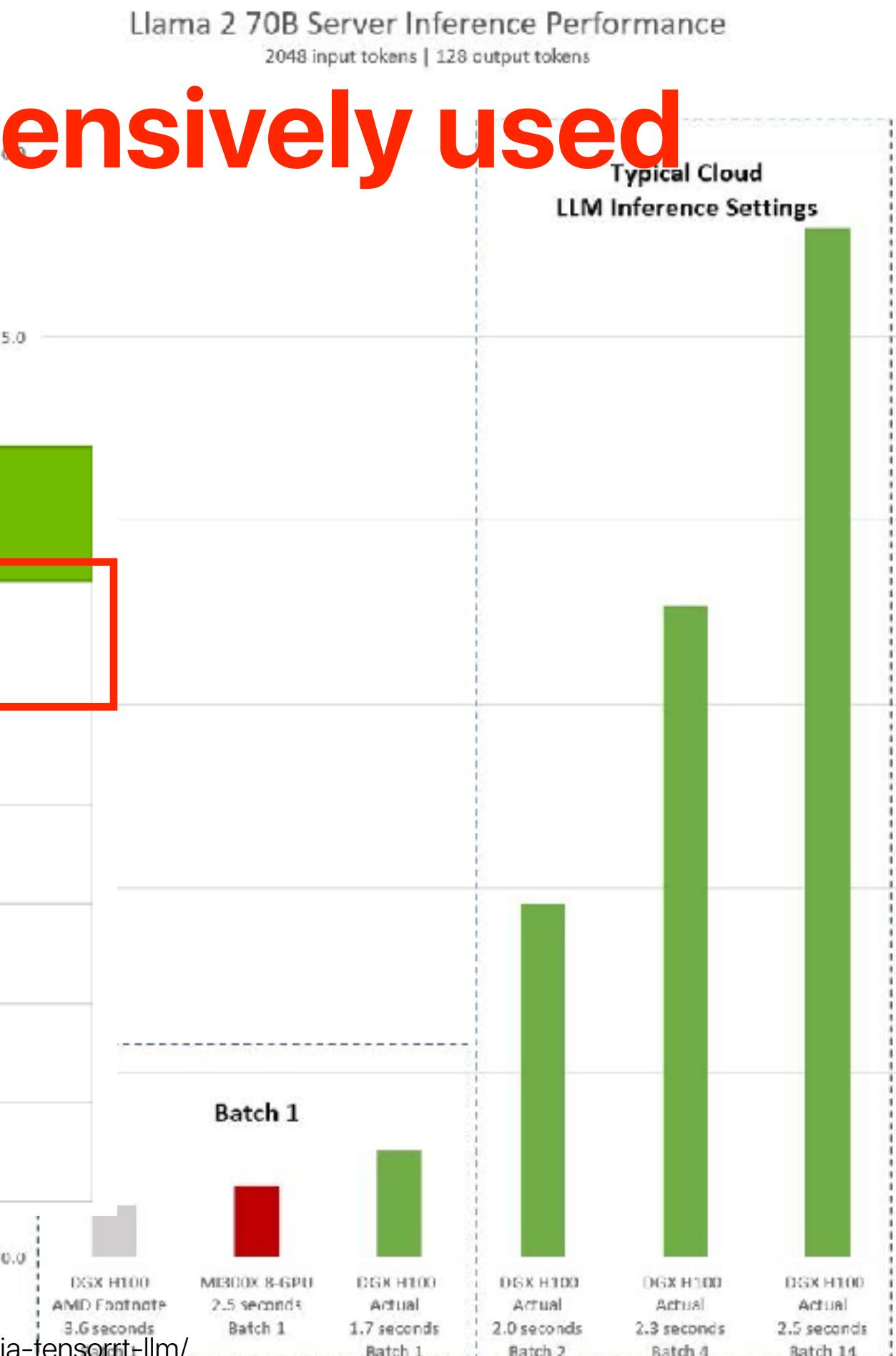
ChatGPT 可能會發生錯誤。建議你查接重要資訊。

Inference per second is intensively used

$$\frac{\text{Inferences}}{\text{Second}} = \frac{\text{Inferences}}{\text{Operation}} \times \frac{\text{Operations}}{\text{Second}}$$

	K80 2012	TPU 2015	P40 2016
Inferences/Sec <10ms latency	1/13X	1X	2X
Training TOPS	6 FP32	NA	12 FP32
Inference TOPS	6 FP32	90 INT8	48 INT8
On-chip Memory	16 MB	24 MB	11 MB
Power	300W	75W	250W
Bandwidth	320 GB/S	34 GB/S	350 GB/S

<https://blogs.nvidia.com/blog/2017/04/10/ai-drives-rise-accelerated-computing-datacenter/>



<https://developer.nvidia.com/blog/achieving-top-inference-performance-with-the-nvidia-h100-tensor-core-gpu-and-nvidia-tensorrt-llm/>

What's (not) considered by Inference per second

$$\frac{\text{Inferences}}{\text{Second}} = \frac{\text{Inferences}}{\text{Operation}} \times \frac{\text{Operations}}{\text{Second}}$$

$$= \frac{\text{Inferences}}{\text{Operation}} \times \left[\frac{\text{operations}}{\text{cycle}} \times \frac{\text{cycles}}{\text{second}} \times \#_{\text{of_PEs}} \times \text{Utilization_of_PEs} \right]$$

	Hardware	Model	Input Data
Operations per inference		v	
Operations per cycle	v		
Cycles per second	v		
Number of PEs	v		
Utilization of PEs	v	v	
Effectual operations out of (total) operations		v	v
Effectual operations plus unexploited ineffectual operations per cycle	v		

gemini.google.com

Gemini v.s. ChatGPT Round #2

Gemini was just updated. See update

Hello, Hung-Wei

How can I help you today?

- Suggest the best parks to visit in a city with descriptions
- Come up with a product name for a new app
- Brainstorm presentation ideas about a topic

Partially correct

1 answer/8 secs

Enter a prompt here

Gemini may display inaccurate info, including about people, so double-check its responses. [Your privacy & Gemini Apps](#)

chat.openai.com

ChatGPT Round #2

HT 你

Can you describe what is computer architecture in one paragraph?

ChatGPT

Computer architecture encompasses the design and organization of computer systems, including the structure of its components and their interrelationships to enable the execution of instructions and the manipulation of data. It defines the principles and methods used to design and implement computer systems, covering aspects such as instruction set architecture (ISA), processor organization, memory hierarchy, input/output systems, and interconnection networks. Computer architecture aims to optimize performance, energy efficiency, cost-effectiveness, and other relevant metrics, considering factors like scalability, reliability, and compatibility, thereby providing the foundational framework upon which computer systems are built and operated.

Completely wrong

目前為止，此對話是否對你帶來幫助？

1 answer/18 secs

Can you describe some research projects from ESCAL at UC Riverside?

ChatGPT 可能會發生錯誤。建議你查核重要資訊。

What's wrong with inferences per second?

- There is no standard on how they inference — but these affect!
 - What model?
 - What dataset?
 - Quality?

“Fair” Comparisons

Andrew Davison. Twelve Ways to Fool the Masses When Giving Performance Results on Parallel Computers.
In Humour the Computer, MITP, 1995
Norman P. Jouppi et. al.. In-Datacenter Performance Analysis of a Tensor Processing Unit
<https://doi.org/10.1145/3079856.3080246>



Extreme Multitasking Performance

- Dual 4K external monitors
- 1080p device display
- 7 applications

What's missing in this video clip?

- The ISA of the “competitor”
- Clock rate, CPU architecture, cache size, how many cores
- How big the RAM?
- How fast the disk?

12 ways to Fool the Masses When Giving Performance Results on Parallel Computers

- Quote only 32-bit performance results, not 64-bit results.
- Present performance figures for an inner kernel, and then represent these figures as the performance of the entire application.
- Quietly employ assembly code and other low-level language constructs.
- Scale up the problem size with the number of processors, but omit any mention of this fact.
- Quote performance results projected to a full system.
- Compare your results against scalar, unoptimized code on Crays.
- When direct run time comparisons are required, compare with an old code on an obsolete system.
- If MFLOPS rates must be quoted, base the operation count on the parallel implementation, not on the best sequential implementation.
- Quote performance in terms of processor utilization, parallel speedups or MFLOPS per dollar.
- Mutilate the algorithm used in the parallel implementation to match the architecture.
- Measure parallel run times on a dedicated system, but measure conventional run times in a busy environment.
- If all else fails, show pretty pictures and animated videos, and don't talk about performance.

Fair comparison in computer architectures

- Metrics: you must consider the fact that performance is composed of IC, CPI, and CT. — any metric that misses one of them is misleading
- Only one variation in each comparison
 - Only change the processor, but not ISA (related to IC) and others
 - Only change the algorithm, but not others
 - The same dataset, must be the same outcome

Why we need benchmark suites

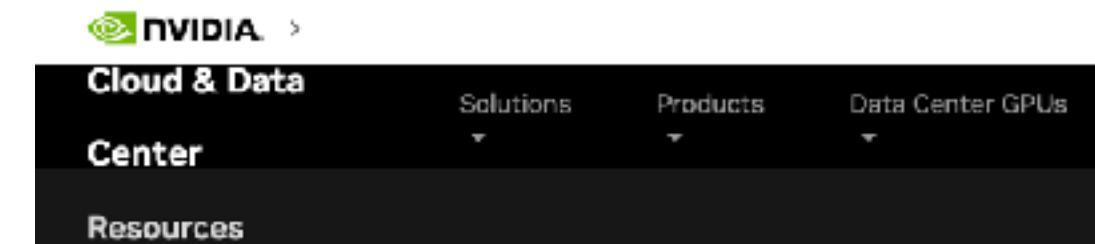
- Allowing people evaluate systems with exactly the same program and the same inputs and validate results from different machines
- Popular benchmark suites
 - SPEC — CPU benchmark
 - MLPerf — ML systems



The screenshot shows the homepage of the SPEC Standard Performance Evaluation Corporation. The header features the SPEC logo and the text "Standard Performance Evaluation Corporation". Below the header is a navigation bar with links for Home, Benchmarks, Tools, Results, Contact, Blog, Join Us, Search, and Help. A sidebar on the left contains links for Published Results, Results Search, Fair Use Policy, and Information, which includes sections for CPURE17, Documentation, Overview, System Requirements, Run & Reporting Rules, Using SPEC-CPURE17, Definitions, Technical Report, and FAQ. The main content area is titled "SPEC CPU® 2017" and describes the benchmark package as containing next-generation, industry-standardized, CPU-intensive suites for measuring and comparing compute intensive performance, assessing a system's processor, memory subsystem and compiler. It mentions a price of \$1000 for new customers, \$250 for qualified non-profit organizations, and \$50 for accredited academic institutions.

- *Pitfall: For NN hardware, Inferences Per Second (IPS) is an inaccurate summary performance metric.*
- Our results show that IPS is a poor overall performance summary for NN hardware, as it's simply the inverse of the complexity of the typical inference in the application (e.g., the number, size, and type of NN layers). For example, the TPU runs the 4-layer MLP1 at 360,000 IPS but the 89-layer CNN1 at only 4,700 IPS, so TPU IPS vary by 75X! Thus, using IPS as the single-speed summary is *even more misleading* for NN accelerators than MIPS or FLOPS are for regular processors [23], so IPS should be even more disparaged. To compare NN machines better, we need a benchmark suite written at a high-level to port it to the wide variety of NN architectures. Fathom is a promising new attempt at such a benchmark suite [3].

In-Datacenter Performance Analysis of a Tensor Processing Unit
<https://doi.org/10.1145/3079856.3080246>



The screenshot shows the NVIDIA Cloud & Data Center website. The top navigation bar includes the NVIDIA logo, a search bar, and links for Cloud & Data Center, Solutions, Products, and Data Center GPUs. The "Cloud & Data Center" link is highlighted. Below the navigation is a section titled "MLPerf Benchmarks" with the subtext: "The NVIDIA AI platform showcases leading performance and versatility in MLPerf Training, Inference, and HPC for the most demanding, real-world AI workloads." To the right of the text is a small image of a server rack.

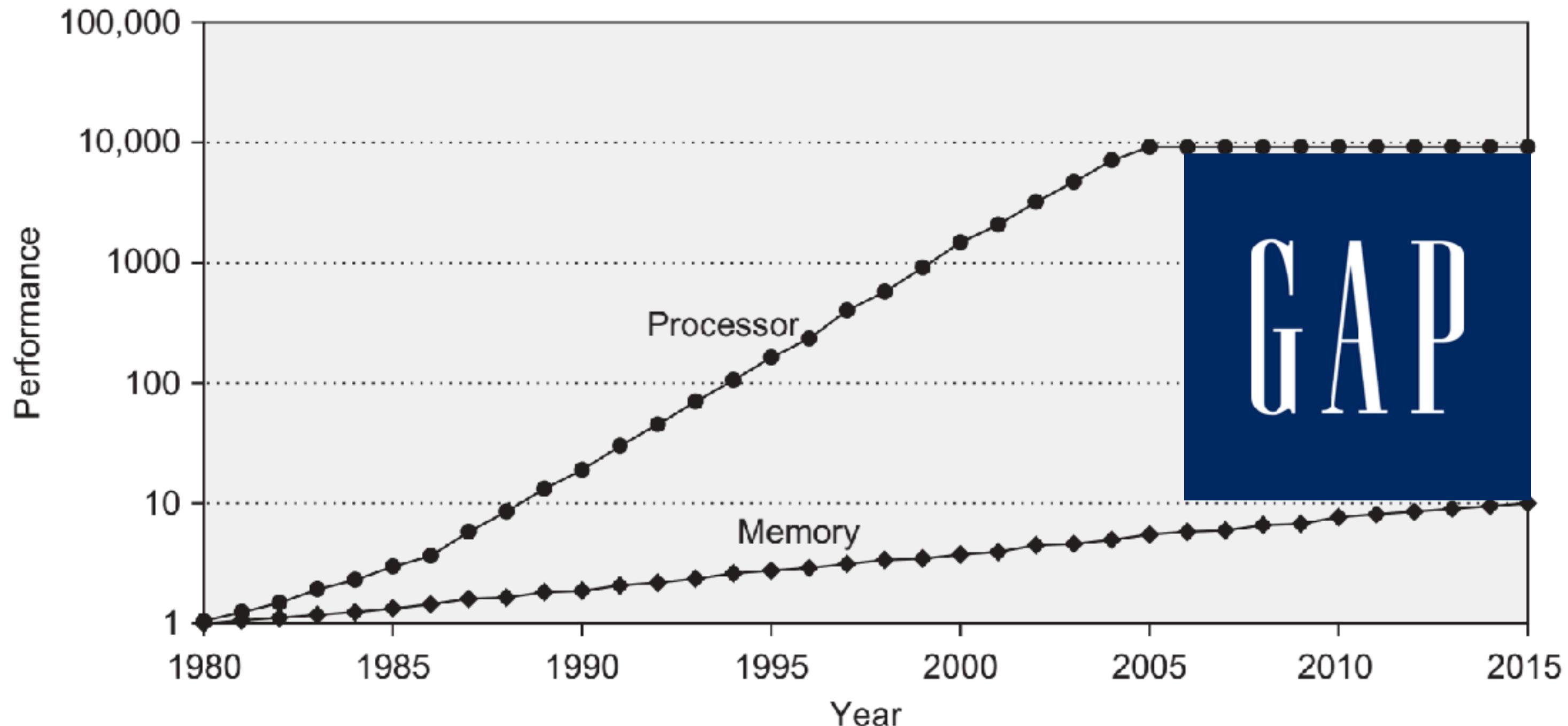
Takeaways: Do the right thing

- Choose the right performance metric
 - A good metric should consider all factors of performance equation
 - IC, CPI, and CT
 - Performance is not determined by just a single factor

Modern DRAM performance

SDRAM					DDR				
Data Rate MT/s	Bandwidth GB/s	CAS (clk)	Latency (ns)	Year	Data Rate MT/s	Bandwidth GB/s	CAS (clk)	Latency (ns)	Year
100	0.80	3	24.00	1992	400	3.20	5	25.00	1998
133	1.07	3	22.50		667	5.33	5	15.00	
					800	6.40	6	15.00	
DDR 2					DDR 3				
400	3.20	5	25.00	2003	800	6.40	6	15.00	2007
667	5.33	5	15.00		1066	8.53	8	15.00	
800	6.40	6	15.00		1333	10.67	9	13.50	
					1600	12.80	11	13.75	
					1866	14.93	13	13.93	
					2133	17.07	14	13.13	
DDR 4					DDR 5				
1600	12.80	11	13.75	2014	3200	25.60	22	13.75	2020
1866	14.93	13	13.92		3600	28.80	26	14.44	
2133	17.07	15	14.06		4000	32.00	28	14.00	
2400	19.20	17	14.17		4400	35.20	32	14.55	
2666	21.33	19	14.25		4800	38.40	34	14.17	
2933	23.46	21	14.32		5200	41.60	38	14.62	
3200	25.20	22	13.75		5600	44.80	40	14.29	
					6000	48.00	42	14.00	
					6400	51.20	46	14.38	

Recap: Performance gap between Processor/Memory





The impact of “slow” memory

- Assume that we have a processor running @ 4 GHz and a program with 20% of load/store instructions. If the instruction has no memory access, the CPI is just 1. Now, consider we have DDR5. The program is well-optimized so precharge is never necessary — the memory access latency is 13.75 ns. What's the average CPI (pick the closest one)?
 - 9
 - 12
 - 15
 - 56
 - 67

The impact of “slow” memory

- Assume that we have a processor running @ 4 GHz and a program with 20% of load/store instructions. If the instruction has no memory access, the CPI is just 1. Now, consider we have DDR5. The program is well-optimized so precharge is never necessary — the memory access latency is 13.75 ns. What's the average CPI (pick the closest one)?

A. 9

B. 12

C. 15

D. 56

E. 67

$$CPU \text{ cycle time} = \frac{1}{4 \times 10^9} = 0.25 \text{ ns}$$

$$Each \text{ DRAM access} = \frac{13.75}{0.25} = 55 \text{ cycles}$$

$$CPI_{average} = 1 + 100\% \times 55 + 20\% \times 55 = 67 \text{ cycles}$$

Don't forget, instructions are also from “memory”

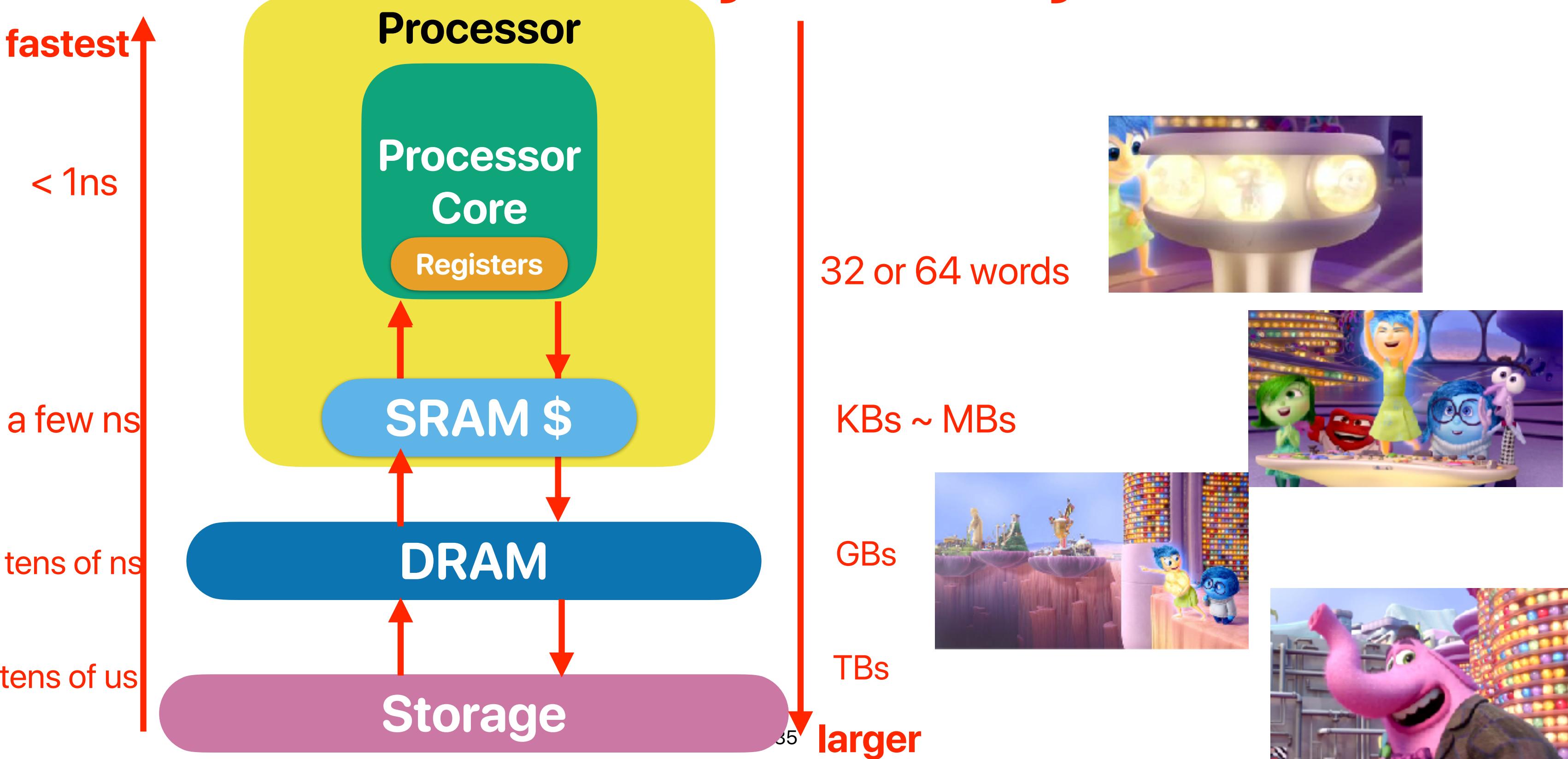
$$\frac{66}{67} = 98.5 \% \text{ of time, we're dealing with memory accesses!}$$

Alternatives?

Memory technology	Typical access time	\$ per GiB in 2012
SRAM semiconductor memory	0.5–2.5 ns	\$500–\$1000
DRAM semiconductor memory	50–70 ns	\$10–\$20
Flash semiconductor memory	5,000–50,000 ns	\$0.75–\$1.00
Magnetic disk	5,000,000–20,000,000 ns	\$0.05–\$0.10

Fast, but expensive \$\$\$

Memory Hierarchy





How can “memory hierarchy” help in performance?

- Assume that we have a processor running @ 4 GHz and a program with 20% of load/store instructions. If the instruction has no memory access, the CPI is just 1. Now, in addition to we DDR5, whose latency 13.75 ns, we also got an SRAM cache with latency of just at 0.5 ns and can capture 90% of the desired data/instructions. what's the average CPI (pick the closest one)?

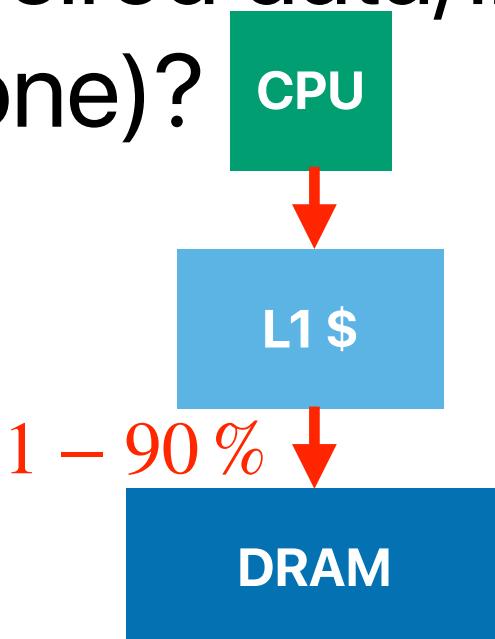
- A. 6
- B. 8
- C. 10
- D. 12
- E. 67



How can “memory hierarchy” help in performance?

- Assume that we have a processor running @ 4 GHz and a program with 20% of load/store instructions. If the instruction has no memory access, the CPI is just 1. Now, in addition to we DDR5, whose latency 13.75 ns, we also got an SRAM cache with latency of just at 0.5 ns and can capture 90% of the desired data/instructions. what's the average CPI (pick the closest one)?

- A. 6
- B. 8
- C. 10
- D. 12
- E. 67



$$CPU \text{ cycle time} = \frac{1}{4 \times 10^9} = 0.25\text{ns}$$

$$\text{Each \$ access} = \frac{0.5}{0.25} = 2 \text{ cycles}$$

$$\text{Each DRAM access} = \frac{13.75}{0.25} = 55 \text{ cycles}$$

$$CPI_{average} = 1 + 100\% \times [2 + (1 - 90\%) \times 55] + 20\% \times [2 + (1 - 90\%) \times 55] = 10 \text{ cycles}$$

L1? L2? L3?

CPU-Z - ID : vfljg

CPU Mainboard Memory SPD Graphics Bench About

Processor

Name	AMD Ryzen 7 7700X		
Code Name	Raphael	Max TDP	105 W
Package	Socket AM5 (LGA1718)		
Technology	5 nm	Core Voltage	1.288 V

Specification

AMD Ryzen 7 7700X 8-Core Processor			
Family	F	Model	1
Ext. Family	19	Ext. Model	61
Stepping	2	Revision	RPL-B2

Instructions

MMX(+), SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, SSE4A x86-64, AMD-V, AES, AVX, AVX2, AVX512, FMA3, SHA			
--	--	--	--

Clocks (Core #0)

Core Speed	5188.99 MHz
Multiplier	x 52.0 (4 - 55.5)
Bus Speed	99.79 MHz
Rated FSB	

Cache

L1 Data	8 x 32 KB
L1 Inst.	8 x 32 KB
Level 2	8 x 1024 KB
Level 3	32 MBytes

Selection Socket #1 Cores 8 Threads 16

CPU-Z Ver. 2.09.0.x64 Tools Validate Close

CPU-Z - ID : pk15b

CPU Mainboard Memory SPD Graphics Bench About

Processor

Name	Intel Core i7 14700K		
Code Name	Raptor Lake	Max TDP	125 W
Package	Socket 1700 LGA		
Technology	10 nm	Core Voltage	1.412 V

Specification

Intel(R) Core(TM) i7-14700K			
Family	6	Model	7
Ext. Family	6	Ext. Model	B7
Stepping	1	Revision	B0

Instructions

MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, EM64T VT-x, AES, AVX, AVX2, FMA3, SHA			
--	--	--	--

Clocks (Core #0)

Core Speed	5287.07 MHz
Multiplier	x 53.0 (8 - 55)
Bus Speed	99.76 MHz
Rated FSB	

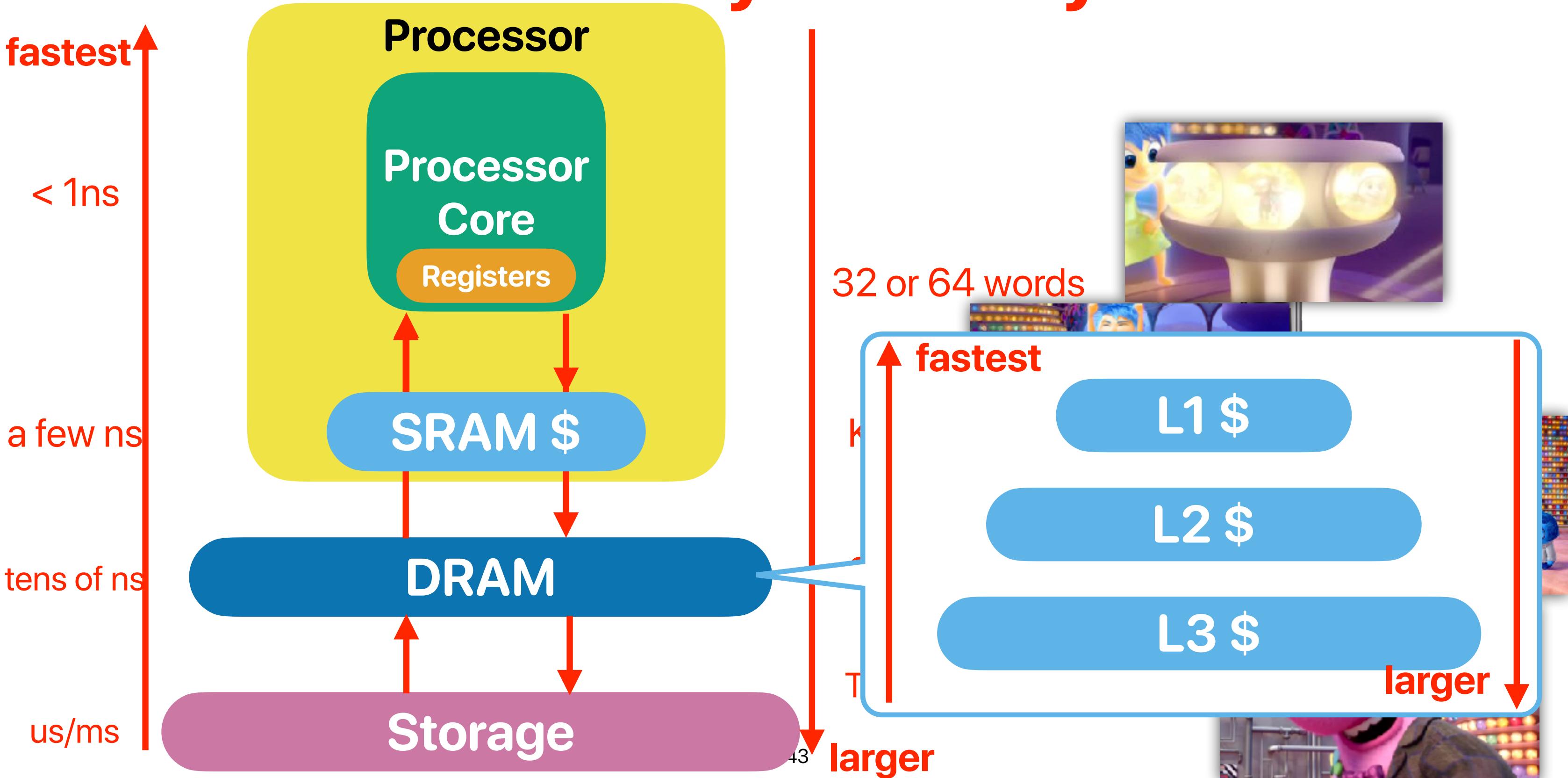
Cache

L1 Data	8 x 48 KB + 12 x 32 KB
L1 Inst.	8 x 32 KB + 12 x 64 KB
Level 2	8 x 2 MB + 3 x 4 MB
Level 3	33 MBytes

Selection Socket #1 Cores 8 + 12 Threads 28

CPU-Z Ver. 2.08.0.x64 Tools Validate Close

Memory Hierarchy





How can a deeper memory hierarchy help in performance?

- Assume that we have a processor running @ 4 GHz and a program with 20% of load/store instructions. If the instruction has no memory access, the CPI is just 1. Now, in addition to we DDR5, whose latency 13.75 ns, we also got a 2-level SRAM caches with
 - it's 1st-level one at latency of 0.5ns and can capture 90% of the desired data/instructions.
 - the 2nd-level at latency of 5 ns and can capture 60% of the desired data/instructions

We also got an SRAM cache with latency of just at 0.5 ns and can capture 90% of the desired data/instructions.

what's the average CPI (pick the closest one)?

- A. 6
- B. 8
- C. 10
- D. 12
- E. 67

How can a deeper memory hierarchy help in performance?

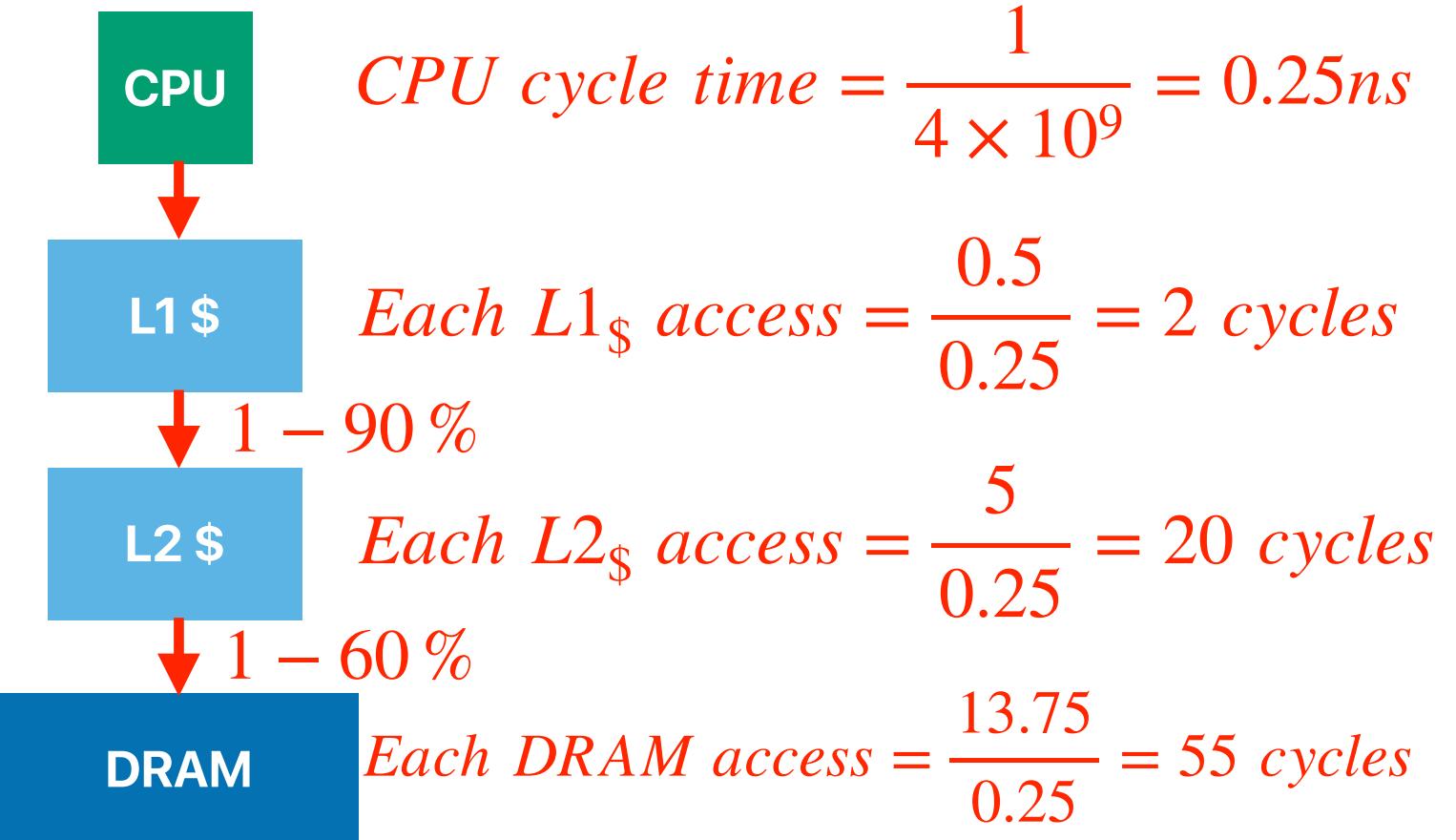
- Assume that we have a processor running @ 4 GHz and a program with 20% of load/store instructions. If the instruction has no memory access, the CPI is just 1. Now, in addition to we DDR5, whose latency 13.75 ns, we also got a 2-level SRAM caches with
 - it's 1st-level one at latency of 0.5ns and can capture 90% of the desired data/instructions.
 - the 2nd-level at latency of 5 ns and can capture 60% of the desired data/instructions

We also got an SRAM cache with latency of just at 0.5 ns and can capture 90% of the desired data/instructions.

what's the average CPI (pick the closest one)?

- A. 6
- B. 8
- C. 10
- D. 12
- E. 67

$$\begin{aligned} CPI_{average} &= 1 + 100\% \times [2 + (1 - 90\%) \times (20 + (1 - 60\%) \times 55) + \\ &\quad 20\% \times [2 + (1 - 90\%) \times (20 + (1 - 60\%) \times 55)]] \\ &= 8.44 \text{ cycles} \end{aligned}$$



L1? L2? L3?

Can we really “predict” upcoming data accurately (e.g., 90%) with such small caches?

The image shows two side-by-side CPU-Z software interfaces. Both are titled "CPU-Z - ID : vfljg" and "CPU-Z - ID : pk15b". Each interface has tabs for CPU, Mainboard, Memory, SPD, Graphics, Bench, and About. The "Processor" section displays the processor name, code name, max TDP, package, technology, specification, ext. family, instructions, and clock information. The "Cache" section lists L1 Data, L1 Inst., Level 2, and Level 3 cache sizes. A large red box highlights the Cache sections of both processors.

Processor

Name	AMD Ryzen 7 7700X	Intel Core i7 14700K
Code Name	Raphael	Raptor Lake
Max TDP	105 W	125 W
Technology	5 nm	18 nm
Core Voltage	1.283 V	1.112 V
Specification	AMD Ryzen 7 7700X 8-Core Processor	Intel(R) Core(TM) i7-14700K
Ext. Family	19	6
Ext. Model	1	67
Revision	RP1.2	B0
Instructions	MMX(+), SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, SSE4A x86-64, AMD-V, AES, AVX, AVX2, AVX512, FMA3, SHA	MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, EM64T VT-x, AES, AVX, AVX2, FMA3, SHA
Clocks (Core #0)		
Core Speed	5188.99 MHz	5287.07 MHz
Multiplier	x 52.0 (4 - 55.5)	x 53.0 (8 - 55)
Bus Speed	99.79 MHz	99.76 MHz
Rated FSB		
Cache		
L1 Data	8 x 32 KB	8 x 48 KB + 12 x 32 KB
L1 Inst.	8 x 32 KB	8 x 32 KB + 12 x 64 KB
Level 2	8 x 1024 KB	8 x 2 MB + 3 x 4 MB
Level 3	32 MBytes	33 MBytes
Selection	Socket #1	Socket #1
Cores	8	8 + 12
Threads	16	28

CPU-Z Ver. 2.09.0.x64 Tools Validate Close

CPU-Z Ver. 2.08.0.x64 Tools Validate Close

Announcement

- Reading quiz #3 due **next Tuesday** before the lecture
- Assignment #2 due **this Thursday**
 - Review the “demo”s of previous lectures if you need inspiration for programming assignment
 - You should run the experiments yourself and calculate results based on that — everyone should have a different answer
 - If you consult your classmates, you need to put their names in the cell where you state your name

Computer Science & Engineering

203

つづく

