Memory & Caches I
CSE 351 Winter 2024

Instructor:
Justin Hsia

Teaching Assistants:
Adithi Raghavan
Aman Mohammed
Connie Chen
Eyoel Gebre
Jiawei Huang
Malak Zaki
Naama Amiel
Nathan Khuat
Nikolas McNamee
Pedro Amarante
Will Robertson
Relevant Course Information

❖ HW14 due Monday, HW15 due Wednesday

❖ Lab 3 due next Friday (2/16)
  ▪ Make sure to look at HW14 before starting

❖ Midterm starts tomorrow (2/8-10)
  ▪ Only private posts on Ed Discussion
  ▪ Staff cannot help you study during the exam window – only point you to resources and clarify the questions
  ▪ We will post clarifications and corrections about the exam on Ed as we go
Lesson Summary (1/3)

- IEC prefixes are unambiguously powers of 2:

- 2^{XY} “things” = \begin{align*}
    Y = 0 &\rightarrow 1 \\
    Y = 1 &\rightarrow 2 \\
    Y = 2 &\rightarrow 4 \\
    Y = 3 &\rightarrow 8 \\
    Y = 4 &\rightarrow 16 \\
    Y = 5 &\rightarrow 32 \\
    Y = 6 &\rightarrow 64 \\
    Y = 7 &\rightarrow 128 \\
    Y = 8 &\rightarrow 256 \\
    Y = 9 &\rightarrow 512 \\
\end{align*} + \begin{align*}
    X = 0 &\rightarrow \\
    X = 1 &\rightarrow \text{Kibi-} \\
    X = 2 &\rightarrow \text{Mebi-} \\
    X = 3 &\rightarrow \text{Gibi-} \\
    X = 4 &\rightarrow \text{Tebi-} \\
    X = 5 &\rightarrow \text{Pebi-} \\
    X = 6 &\rightarrow \text{Ebi-} \\
    X = 7 &\rightarrow \text{Zebi-} \\
    X = 8 &\rightarrow \text{Yobi-} \\
\end{align*} + “things”
Lesson Summary (2/3)

❖ Memory Hierarchy

- Successively higher levels contain “most used” data from lower levels
- Caches are intermediate storage levels used to optimize data transfers between any system elements with different characteristics
- Exploits *temporal and spatial locality*:

![Diagram of memory hierarchy]

- Smaller, faster, costlier per byte
- Larger, slower, cheaper per byte
Lesson Summary (3/3)

❖ Cache Performance

▪ Ideal case: found in cache (*cache hit*), return requested data immediately

▪ Bad case: not found in cache (*cache miss*), search in next level
  • Bring entire *cache block* containing requested data into this cache once found

▪ **Average Memory Access Time** (AMAT) = HT + MR × MP
  • Hurt by Miss Rate and Miss Penalty
Lesson Q&A

❖ Learning Objectives:
  ▪ Describe the memory hierarchy and explain the relationship between cost, size, and access speed of its layers.
  ▪ Analyze how changes [to cache parameters and policies] affect performance metrics such as AMAT

❖ What lingering questions do you have from the lesson?
  ▪ Chat with your neighbors about the lesson for a few minutes to come up with questions
Caches I – Practice
Polling Questions (1/2)

- Convert the following to or from IEC:
  - $512$ Ki-books $= 2^{19}$ books
  - $2^{27}$ caches $= 128$ Mi-caches

- Compute the average memory access time (AMAT) for the following system properties:
  - Hit time of $1$ ns
  - Miss rate of $1\%$
  - Miss penalty of $100$ ns

$$\text{AMAT} = \text{HT} + \text{MR} \times \text{MP}$$

$$= 1\, \text{ns} + 0.01(100\, \text{ns})$$

$$= 1\, \text{ns} + 1\, \text{ns}$$

$$\text{AMAT} = 2\, \text{ns}$$
Polling Questions (2/2)

- Processor specs: 200 ps clock, MP of 50 clock cycles, MR of 0.02 misses/instruction, and HT of 1 clock cycle

\[ AMAT = HT + MR \times MP = 1 + 0.02 \times 50 = 2 \text{ clock cycles} = 400 \text{ ps} \]

- Which improvement would be best?
  A. 190 ps clock (overclocking, faster CPU)
     \[ 2 \text{ clock cycles} = 380 \text{ ps} \]
  B. Miss penalty of 40 clock cycles (reduced Mem size)
     \[ 1 + 0.02 \times 40 = 1.8 \text{ clock cycles} = 360 \text{ ps} \]
  C. MR of 0.015 misses/instruction (write better code)
     \[ 1 + 0.015 \times 50 = 1.75 \text{ clock cycles} = 350 \text{ ps} \]
Caches I – Context
Amat, Revisited

- *Average Memory Access Time (AMAT):* average time to access memory considering both hits and misses
  \[
  \text{AMAT} = \text{Hit time} + \text{Miss rate} \times \text{Miss penalty}
  \]
  (abbreviated AMAT = HT + MR × MP)

- We called this a *cache performance metric*
  - This isn’t the only metric we could have used!
Metrics in Computing

❖ Generally, folks care most about **performance**
  ▪ Energy-efficiency is more important now since the plateau in 2004/2005
  ▪ This is why we have so many specialized chips nowadays

❖ Really, this is just **efficiency** – making efficient use of the resources that we have
  ▪ Performance: cycles/instruction, seconds/program
  ▪ Energy efficiency: performance/watt
  ▪ Memory: bytes/program, bytes/data structure
Metrics

❖ What do we do with metrics?
  ▪ We tend to optimize along them!
  ▪ Especially when jobs/funding depend on better performance along some metric
    • See all of Intel under “Moore’s Law”

❖ Sometimes, strange incentives emerge
  ▪ “Minimize the number of bugs on our dashboard”
    • Does it count if we make the bugs invisible?
  ▪ “Make this faster for our demo in a week”
    • Shortcuts might hurt performance at scale
  ▪ “Minimize our average memory access time”
    • What if we add more memory accesses that we know will hit?
Metrics and Success

❖ Success is defined along metrics
  ▪ This affects how we measure and optimize

❖ Let’s say that we choose performance/program or performance/program set (i.e., benchmarks):
  1. Measure existing performance
  2. Come up with a bunch of optimizations that would improve performance
  3. Select a few to build into the “next version”
Metrics and Success

❖ Success is *defined along metrics*
  - This affects how we measure and optimize

❖ Let’s say that we choose **profit/year** or **stock price**:  
  - Success means earning more profit than last year  
  - Improvement or optimizations might include:
    - Reduce expenses, cut staff  
    - Sell more things or fancier things (e.g., in-app purchases)  
    - Make people pay monthly for things they could get for free  
    - Increase advertising revenue:

---

*The New York Times*

**Whistle-Blower Says Facebook ‘Chooses Profits Over Safety’**

Frances Haugen, a Facebook product manager who left the company in May, revealed that she had provided internal documents to journalists and others.
Metrics and Success

❖ Success is *defined along metrics*
  ▪ This affects how we measure and optimize

❖ Let’s say that we choose *minoritized participation in computing*:
  ▪ What does success/participation mean (and dangers)?
    • Women? BIPOC? All minoritized lumped together?
      – Might optimize for one group at the expense of others
    • Taking intro? Passing intro? Getting a degree? Getting a job?
      – Says nothing about retention or participation/decision-making level
Design Considerations

❖ Regardless of what we build, the way that we define success shapes the systems we build
  ▪ Choose your metrics carefully
  ▪ There’s more to choose from than performance (e.g., usability, access, simplicity, agency)

❖ Metrics are a “heading” (in the navigational sense)
  ▪ Best to reevaluate from time to time in case you’re off course or your destination changes
Discussion Questions

❖ Discuss the following question(s) in groups of 3-4 students
   ▪ I will call on a few groups afterwards so please be prepared to share out
   ▪ Be respectful of others’ opinions and experiences

❖ Let’s say your (main) metric for college is to get a 4.0 overall GPA.
   ▪ What are some potential unintended consequences of this metric?
   ▪ What are some other potential metrics you could use for college?
Group Work Time

❖ During this time, you are encouraged to work on the following:
   1) If desired, continue your discussion
   2) Work on the homework problems
   3) Work on the lab (if applicable)

❖ Resources:
   ▪ You can revisit the lesson material
   ▪ Work together in groups and help each other out
   ▪ Course staff will circle around to provide support