
CMPEN 431

Computer Architecture

Fall 2018

Caching: AMAT vs. CPI examples

Jack Sampson(www.cse.psu.edu/~sampson)

[Slides adapted from work by Mary Jane Irwin, in turn adapted from
Computer Organization and Design, Revised 4th Edition,
Patterson & Hennessy, © 2011, Morgan Kaufmann and from
Computer Organization and Design, 5th Edition,
Patterson & Hennessy, © 2014, Morgan Kaufmann]

Generalized AMAT

- ❑ AMAT is the **AVERAGE** memory access time
 - ❑ So... average of what, in what units?
 - ❑ Answer: Weighted average of all types of memory accesses, by frequency, in either explicit (seconds, picoseconds) or abstract (cycles) time

- ❑ How to calculate AMAT, generally:
 - ❑ For each type **T** of memory access (instruction, load, store, etc.)
 - Compute, for memory hierarchy **H**,
$$\text{AMAT}(H, T) = \text{AccessTime}(H[0], T) + \text{MissRate}(H[0], T) * (\text{AMAT}(H[1:], T));$$

where $\text{AMAT}(\{\}, T) = 0$
 - ❑ Take the weighted average over all **T** of $\text{AMAT}(H, T)$ by the relative frequency of each type **T**

Generalized AMAT, abstract example

- How to calculate AMAT, generally:

- For each type **T** of memory access (instruction, load, store, etc.)

- Compute, for memory hierarchy **H**,
$$\text{AMAT}(\mathbf{H}, \mathbf{T}) = \text{AccessTime}(\mathbf{H}[0], \mathbf{T}) + \text{MissRate}(\mathbf{H}[0], \mathbf{T}) * (\text{AMAT}(\mathbf{H}[1:], \mathbf{T}));$$

where $\text{AMAT}(\{\}, \mathbf{T}) = 0$

- Take the weighted average over all **T** of $\text{AMAT}(\mathbf{H}, \mathbf{T})$ by the relative frequency of each type **T**

- Assume $\mathbf{H} = \{\text{L1 (split), L2 (unified), Memory}\}$ and the relevant types are I-Fetch and LoadStore.

- $\text{AMAT}(\mathbf{H}, \text{I-Fetch}) =$

- $$\begin{aligned} &\text{AccessTime}(\text{L1-I}, \text{I-Fetch}) + \text{MissRate}(\text{L1-I}, \text{I-Fetch}) * (\\ &\quad \text{AccessTime}(\text{L2}, \text{I-Fetch}) + \text{MissRate}(\text{L2}, \text{I-Fetch}) * (\\ &\quad \quad \text{AccessTime}(\text{Memory}, \text{I-Fetch}) + \text{MissRate}(\text{Memory}, \text{I-Fetch}) * (\\ &\quad \quad \quad 0))) \end{aligned}$$

- $\text{AMAT}(\mathbf{H}, \text{LoadStore}) =$

- $$\begin{aligned} &\text{AccessTime}(\text{L1-D}, \text{LoadStore}) + \text{MissRate}(\text{L1-D}, \text{LoadStore}) * (\\ &\quad \text{AccessTime}(\text{L2}, \text{LoadStore}) + \text{MissRate}(\text{L2}, \text{LoadStore}) * (\\ &\quad \quad \text{AccessTime}(\text{Memory}, \text{LoadStore}) + \text{MissRate}(\text{Memory}, \text{LoadStore}) * (\\ &\quad \quad \quad 0))) \end{aligned}$$

- $$\text{AMAT} = (\text{I-Fetch}\% * \text{AMAT}(\mathbf{H}, \text{I-Fetch}) + \text{LoadStore}\% * \text{AMAT}(\mathbf{H}, \text{LoadStore})) / (\text{I-Fetch}\% + \text{LoadStore}\%)$$

Generalized AMAT, concrete example 1

- Assume $H = \{L1 \text{ (split)}, L2 \text{ (unified)}, \text{Memory}\}$ and the relevant types are I-Fetch and LoadStore and that access times in both L1-I and L1-D caches are 1 cycle, that L2 accesses are 10 cycles for all types, and that Memory accesses are 103 cycles for all types. Assume that L1-I hit rate is 98%, L1-D hit rate is 75%, L2 hit rate is 55% for all types, and that Memory hit rate is 100% for all types. Assume 31% of instructions are Loads or Stores.

- $$\text{AMAT}(H, \text{I-Fetch}) = 1 + (1 - 98\%) * (10 + (1 - 55\%) * (103)) \text{ [cycles]}$$

- $$\text{AMAT}(H, \text{LoadStore}) = 1 + (1 - 75\%) * (10 + (1 - 55\%) * (103)) \text{ [cycles]}$$

- $$\text{AMAT} = (100\% * \text{AMAT}(H, \text{I-Fetch}) + 31\% * \text{AMAT}(H, \text{LoadStore})) / (100\% + 31\%)$$

Generalized AMAT, concrete example 2

- Assume $H=\{L1 \text{ (split), } L2 \text{ (unified), Memory}\}$ and the relevant types are I-Fetch, Load, and Store and that access times in L1-I is 1 cycle, access time in L1-D cache is 2 cycles for loads and 1 cycle for stores, that L2 accesses are 10 cycles for all types, and that Memory accesses are 103 cycles for all types. Assume that L1-I hit rate is 98%, L1-D hit rate is 75% for loads and 90% for stores, L2 hit rate is 55% for all types, and that Memory hit rate is 100% for all types. Assume 20% of instructions are Loads and 11% are Stores.

- $AMAT(H, \text{I-Fetch}) = 1 + (1-98\%) * (10 + (1-55\%) * (103))$ [cycles]
- $AMAT(H, \text{Load}) = 2 + (1-75\%) * (10 + (1-55\%) * (103))$ [cycles]
- $AMAT(H, \text{Store}) = 1 + (1-90\%) * (10 + (1-55\%) * (103))$ [cycles]
- $AMAT = (100\% * AMAT(H, \text{I-Fetch}) + 20\% * AMAT(H, \text{Load}) + 11\% * AMAT(H, \text{Store})) / (100\% + 20\% + 11\%)$

Generalized AMAT, concrete example 3

- Assume $H=\{L1 \text{ (split), } L2 \text{ (unified), Memory}\}$ and the relevant types are I-Fetch, Load, and Store and that access times in L1-I is 1 cycle, access time in L1-D cache is 2 cycles for loads and 1 cycle for stores, that L2 accesses are 10 cycles for all types, and that Memory accesses are 110ns for all types. Assume that L1-I hit rate is 98%, L1-D hit rate is 75% for loads and 90% for stores, L2 hit rate is 55% for all types, and that Memory hit rate is 100% for all types. Assume 20% of instructions are Loads and 11% are Stores. Assume that the clock speed is 2 GHz

- $AMAT(H, \text{I-Fetch}) = 1 + (1-98\%) * (10 + (1-55\%) * (110\text{ns} * 2\text{GHz}))$ [cycles]
- $AMAT(H, \text{Load}) = 2 + (1-75\%) * (10 + (1-55\%) * (110\text{ns} * 2\text{GHz}))$ [cycles]
- $AMAT(H, \text{Store}) = 1 + (1-90\%) * (10 + (1-55\%) * (110\text{ns} * 2\text{GHz}))$ [cycles]
- $AMAT = (100\% * AMAT(H, \text{I-Fetch}) + 20\% * AMAT(H, \text{Load}) + 11\% * AMAT(H, \text{Store})) / (100\% + 20\% + 11\%)$

AMAT vs. CPI_{stall}

- ❑ AMAT accounts for the total end-to-end latency of a memory access (so, in terms of latency per-memory access)
- ❑ CPI_{stall} only considers the average latency overheads of events that cause non-ideal (stall) behavior, and is normalized per instruction, not per memory access
- ❑ Consider the following – in a 5 stage pipeline with a 100% hit rate for both I and D caches, the AMAT would be 1 (on average, an access to memory takes 1 cycle) but the CPI_{stall} would be 0 (on average, no stalls are caused by events that are **not already part of the ideal CPI**)

Calculating CPI with memory effects

- ❑ Recall that $CPI = CPI_{ideal} + \sum CPI_{stalls}(cause_i)$
where $cause_i \in \{\text{Non-ideal memory, non-ideal control prediction, data hazards, lions, tigers, bears, etc.}\}$

- ❑ For the moment, let's just look at the first of these causes for non-ideal CPI and abstract the rest as a base non-memory-stall CPI_{base} (that already accounts for the other sources of stalls) and compute our CPI in the presence on non-ideal memory = $CPI_{base} + CPI_{mem-stalls}$

- ❑ The important things to note are:
 - ❑ $CPI \neq AMAT$
 - ❑ $CPI_{mem-stalls} \neq AMAT$

Why/How do CPI and AMAT calculations differ?

- ❑ Base CPI already includes L1-I/L1-D hits
 - ❑ Only misses will cause memory-access stalls in a well-designed in-order pipeline (we will account for load-use stalls separately)
- ❑ Normalization is different:
 - ❑ Even assuming that L1 hit times were accounted for, CPI stalls is looking to calculate not just the average memory stall time/memory access, but will want to multiply that by memory accesses/instruction to get average memory stall time / instruction
 - ❑ More succinctly, weighted sum vs. weighted average

CPI vs AMAT calculation example

□ Recall AMAT example from slide 4:

Assume $H = \{L1 \text{ (split)}, L2 \text{ (unified)}, \text{Memory}\}$ and the relevant types are I-Fetch and LoadStore and that access times in both L1-I and L1-D caches are 1 cycle, that L2 accesses are 10 cycles for all types, and that Memory accesses are 103 cycles for all types. Assume that L1-I hit rate is 98%, L1-D hit rate is 75%, L2 hit rate is 55% for all types, and that Memory hit rate is 100% for all types. Assume 31% of instructions are Loads or Stores.

- $AMAT(H, \text{I-Fetch}) = 1 + (1-98\%) * (10 + (1-55\%) * (103))$ [cycles]
- $AMAT(H, \text{LoadStore}) = 1 + (1-75\%) * (10 + (1-55\%) * (103))$ [cycles]
- $AMAT = (100\% * AMAT(H, \text{I-Fetch}) + 31\% * AMAT(H, \text{LoadStore})) / (100\% + 31\%)$

□ Now, compare the calculation of CPI for the above assuming $CPI_{base} = 1$ and that CPI_{base} assumes (and therefore subsumes) L1 hits. Then

- $CPI = CPI_{base} + CPI_{mem-stalls} = 1 \text{ [cycle/instruction]} + (1 \text{ [fetch/instruction]} * (1-98\%) * (10 + (1-55\%) * (103)) \text{ [cycles/fetch]} + 0.31 \text{ [loadstore/instruction]} * (1-75\%) * (10 + (1-55\%) * (103)) \text{ [cycles/loadstore]})$