CMPEN 431
Computer Architecture
Fall 2019

Caching: AMAT vs. CPI examples

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

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 fetch, load, store, etc.)
    - Compute, for memory hierarchy H,
      AMAT(H,T) = AccessTime(H[0],T) + MissRate(H[0],T) * (AMAT(H[1:],T));
      where AMAT ({},T) = 0
  - Take the weighted average over all T of 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,
      AMAT(H,T) = AccessTime(H[0],T) + MissRate(H[0],T) * (AMAT(H[1:],T));
      where AMAT ({},T) = 0
  - Take the weighted average over all T of AMAT(H,T) by the relative frequency of each type T

- Assume H={L1 (split), L2 (unified), Memory} and the relevant types are I-Fetch and LoadStore.
  - AMAT(H, I-Fetch) =
    AccessTime(L1-I,I-Fetch) + MissRate(L1-I, I-Fetch) * (
    AccessTime(L2, I-Fetch) + MissRate(L2, I-Fetch) * (
    AccessTime(Memory, I-Fetch) + MissRate(Memory, I-Fetch) * (
    0)))
  - AMAT(H, LoadStore) =
    AccessTime(L1-D,LoadStore) + MissRate(L1-D, LoadStore) * (
    AccessTime(L2, LoadStore) + MissRate(L2, LoadStore) * (
    AccessTime(Memory, LoadStore) + MissRate(Memory, LoadStore) * (
    0)))
  - AMAT = (I-Fetch% * AMAT(H, I-Fetch) + LoadStore% * AMAT (H, LoadStore)) / (I-Fetch% + LoadStore%)

Generalized AMAT, concrete example 1

- Assume H={L1 (split), L2 (unified), 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, I-Fetch) =
    1 + (1-98%) * (
    10 + (1-55%) * (
    103)) [cycles]
  - AMAT(H, LoadStore) =
    1 + (1-75%) * (
    10 + (1-55%) * (
    103)) [cycles]
  - AMAT = (100% * AMAT(H, I-Fetch) + 31% * AMAT (H, LoadStore)) / (100% + 31%)

Generalized AMAT, concrete example 2

- Assume H={L1 (split), L2 (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, I-Fetch) = 1 + (1-98%) * (10 + (1-55%) * (103)) [cycles]
  - AMAT(H, Load) = 2 + (1-75%) * (10 + (1-55%) * (103)) [cycles]
  - AMAT(H, Store) = 1 + (1-90%) * (10 + (1-55%) * (103)) [cycles]
  - AMAT = (100% * AMAT(H, I-Fetch) + 20% * AMAT (H, Load) + 11% * AMAT(H,Store)) /
    (100% + 20% + 11%)

Generalized AMAT, concrete example 3

- Assume H={L1 (split), L2 (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, I-Fetch) = 1 + (1-98%) * (10 + (1-55%) * (110ns*2GHz)) [cycles]
  - AMAT(H, Load) = 2 + (1-75%) * (10 + (1-55%) * (110ns*2GHz)) [cycles]
  - AMAT(H, Store) = 1 + (1-90%) * (10 + (1-55%) * (110ns*2GHz)) [cycles]
  - AMAT = (100% * AMAT(H, I-Fetch) + 20% * AMAT (H, Load) + 11% * AMAT(H,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 $cause_i \in$ {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 mempory $= CPI_{base} + CPI_{mem\text{-}stalls}$

- The important things to note are:
  - CPI != AMAT
  - $CPI_{mem\text{-}stalls}$!= 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 (split), L2 (unified), 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, I-Fetch) = 1 + (1-98%) * (10 + (1-55%) * (103)) [cycles]
  - AMAT(H, LoadStore) = 1 + (1-75%) * (10 + (1-55%) * (103)) [cycles]
  - AMAT = (100% * AMAT(H, I-Fetch) + 31% * AMAT (H, 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} + $CPI_{mem\text{-}stalls}$ = 1 [cycle/instruction] + (
    1 [fetch/instruction] * (1-98%) * (10 + (1-55%) * (103)) [cycles/fetch] +
    0.31 [loadstore/instruction] * (1-75%) * (10 + (1-55%) * (103)) [cycles/loadstore]
)