

Memory System Performance

Memory system performance is largely captured by three parameters,

- Latency, Bandwidth, Average memory access time (AMAT).
- **Latency:** The time it takes from the issue of a memory request to the time the data is available at the processor.
- **Bandwidth:** The rate at which data can be pumped to the processor by the memory system.
- **AMAT:** The average time it takes for the processor to get a data item it requests.

Average Memory Access Time (AMAT)

- The time it takes to get requested data to the processor can vary: due to the memory hierarchy.
- Performance of a cache is largely determined by:
 - **Cache hit rate**: number of cache hits divided by number of accesses.
 - **Cache miss rate**: number of cache misses divided by number of accesses.
 - **Cache hit time**: the time between sending address and data returning from cache.
 - **Cache miss penalty**: the extra processor stall cycles caused by access to the next-level cache.

AMAT can be expressed as:

$$AMAT = \text{Cache hit time} + \text{Miss rate} \times \text{Miss penalty}$$

Example-1

- Main memory access time (T_m)=100nsec. Cache access time (T_c) is 5 times lower than main memory access time. Hit rate is 95%. Find out AMAT?
- $AMAT = h * T_c + \{(1-h) * (T_c + T_m)\}$

Example-2

- Cache hit time=10nsec. Miss penalty=150nsec. Hit rate is 98%. Find out AMAT?
- $AMAT = \text{Cache hit time} + \text{Miss rate} * \text{Miss penalty}$

CPU Performance

- Basic Terms related to CPU performance
- Frequency (f) or clock rate.(in HZ)
- Cycle Time (CT) ($=1/\text{Frequency}$)
- Cycle count (CC): Total no. of cycles required to execute a set of instructions or programs
- Instruction Count (IC): Total number of instructions in a set of instructions
- Cycles per instruction (CPI) = CC/IC
- $\text{CC} = \text{IC} * \text{CPI}$
- Execution Time = $\text{CC} * \text{CT} = \text{IC} * \text{CPI} * \text{CT} = \text{IC} * \text{CPI} / \text{F}$

Example-1

There are three processors. The processor P1 has 3GHz clock rate and a CPI of 1.5. The processor P2 has 2.5GHz clock rate and a CPI of 1.0. The processor P3 has 4GHz clock rate and a CPI of 2.2.

Which processor has the highest performance

Solution

Performance = $1/\text{Execution time}$

Execution time (ET) = Total CPU clock cycles * Cycle time

= total CPU clock cycles * (1/clock rate or frequency)

P1: ET = $(I * 1.5) / 3G$, Performance = $2G/I$

P2: ET = $(I * 1) / 2.5G$, Performance = $2.5G/I$

P3: ET = $(I * 2.2) / 4G$, Performance = $1.8G/I$

$P2 > P1 > P3$

Impact of Cache Memory System on Processor Performance

- Basic Terms related to Cache memory performance

CPU Time = Total number of cycles * clock cycle time

Total no. of cycles = CPU clock cycles + Memory stall cycles

CPU clock cycles = IC * CPI

Memory stall cycles = No. of misses * Miss Penalty

= (IC * Misses/Instruction) * Miss Penalty

= IC * (Memory accesses * Miss rate) / Instruction * Miss Penalty

= IC * (Memory accesses/Instruction) * Miss rate * Miss Penalty

= IC * Memory Stall CPI

[Miss rate = No. of misses / Memory accesses

⇒ No. of misses = Memory accesses * Miss rate]

Memory Stall CPI = (Memory accesses/Instruction) * Miss rate * Miss Penalty

Total no. of cycles= CPU clock cycles + Memory stall cycles

$$= (IC * CPI) + \{IC * (\text{Memory accesses/Instruction}) * \text{Miss rate} * \text{Miss Penalty}\}$$

Example-1:

Assume we have a computer where the cycle per instruction (CPI) is 1.0 when all memory accesses hit in the cache. The only data accesses are loads and stores, and these total 50% of the instructions. If the miss penalty is 25 clock cycles and the miss rate is 2%, how much faster would the computer be if all instructions were cache hits?

Solution:

CPU execution time (in clock cycle terms) = (CPU Clock Cycles + Memory stall cycles) / Clock cycle

Computation of the performance of computer that always gives cache hits:

CPU execution time = $(IC * CPI + 0) / \text{Clock cycle} = IC * 1.0 / \text{Clock cycle}$

Performance₁ = $1 / (IC * 1.0)$

Computation of the performance of computer with the real cache:

Memory stall cycles = $IC * (\text{Memory access} / \text{Instruction}) * \text{Miss rate} * \text{Miss penalty}$
 $= IC * (1 + 0.5) * 0.02 * 25 = IC * 0.75 \text{ Clock cycles}$

CPU execution times = CPU cycles + Memory stall cycles
 $= (IC * CPI) + (IC * 0.75) = (IC * 1.0) + (IC * 0.75)$
 $= 1.75 * IC \text{ Clock cycle}$

Performance₁ = $1 / (IC * 1.75)$

Performance ratio = Performance₁ / Performance₂
 $= (IC * 1.75) / (IC * 1.0) = 1.75 \text{ times faster}$

Example 2

- Suppose:
 - Clock Rate = 200 MHz (5 ns per cycle), Ideal (no misses) CPI = 1.1
 - 50% arith/logic, 30% load/store, 20% control
 - 10% of data memory operations get 50 cycles miss penalty
 - 1% of instruction memory operations also get 50 cycles miss penalty
- Compute CPI and AMAT ?

Solution

$$\begin{aligned}\text{CPI} &= 1.1 + (0.30 * 0.10 * 50) + (1 * 0.01 * 50) \\ &= 1.1 + 1.5 + .5 \text{ cycle/ins} = 3.1\end{aligned}$$

(memory per instruction for data is 30% (due to load and store)
whereas for instruction is always 100%)

$$\begin{aligned}\text{AMAT} &= \text{Cache hit time} + \text{Miss rate} * \text{Miss penalty} \\ \text{Cache hit time} &= \text{Ideal CPI}\end{aligned}$$

$$\text{AMAT} = \{\% \text{instruction} * (\text{Cache hit time} + \text{Miss rate} * \text{Miss penalty})\} + \{\% \text{data} * (\text{Cache hit time} + \text{Miss rate} * \text{Miss penalty})\}$$

$$\text{AMAT} = (1/1.3) * [1.1 + 0.01 * 50] + (0.3/1.3) * [1.1 + 0.1 * 50] = 2.63$$

Example-3

- Assume that:
- Instruction miss rate 2%
- Data miss rate 4%
- CPI is 2 (without any memory stalls)
- Miss penalty 40 cycles
- 36% of instructions are load/store
- Determine how much faster a machine would run with a perfect cache that never missed.

Solution

Total memory stall cycles= $IC * [(1 * 0.02 * 40) + (0.36 * 0.04 * 40)] = 1.38 * IC$

CPU execution time with stall= $(IC * 2) + (IC * 1.38)$ Clock cycle= $3.38 * IC$

CPU execution time with perfect cache (in clock cycle)= $(IC * 2)$ Clock cycle

CPU time with stall/CPU time with perfect cache= $(3.38 * IC) / (2 * IC)$
 $= 1.69$

Example 4

Assume 20% Load/Store instructions

Assume CPI without memory stalls is 1

Cache miss penalty = 100 cycles

Miss rate = 1%

What is: memory stall cycles per instruction?

CPI with and without cache?

Solution

Total memory accesses per instruction = $1 + 0.2 = 1.2$

Memory Stall cycle per instruction =

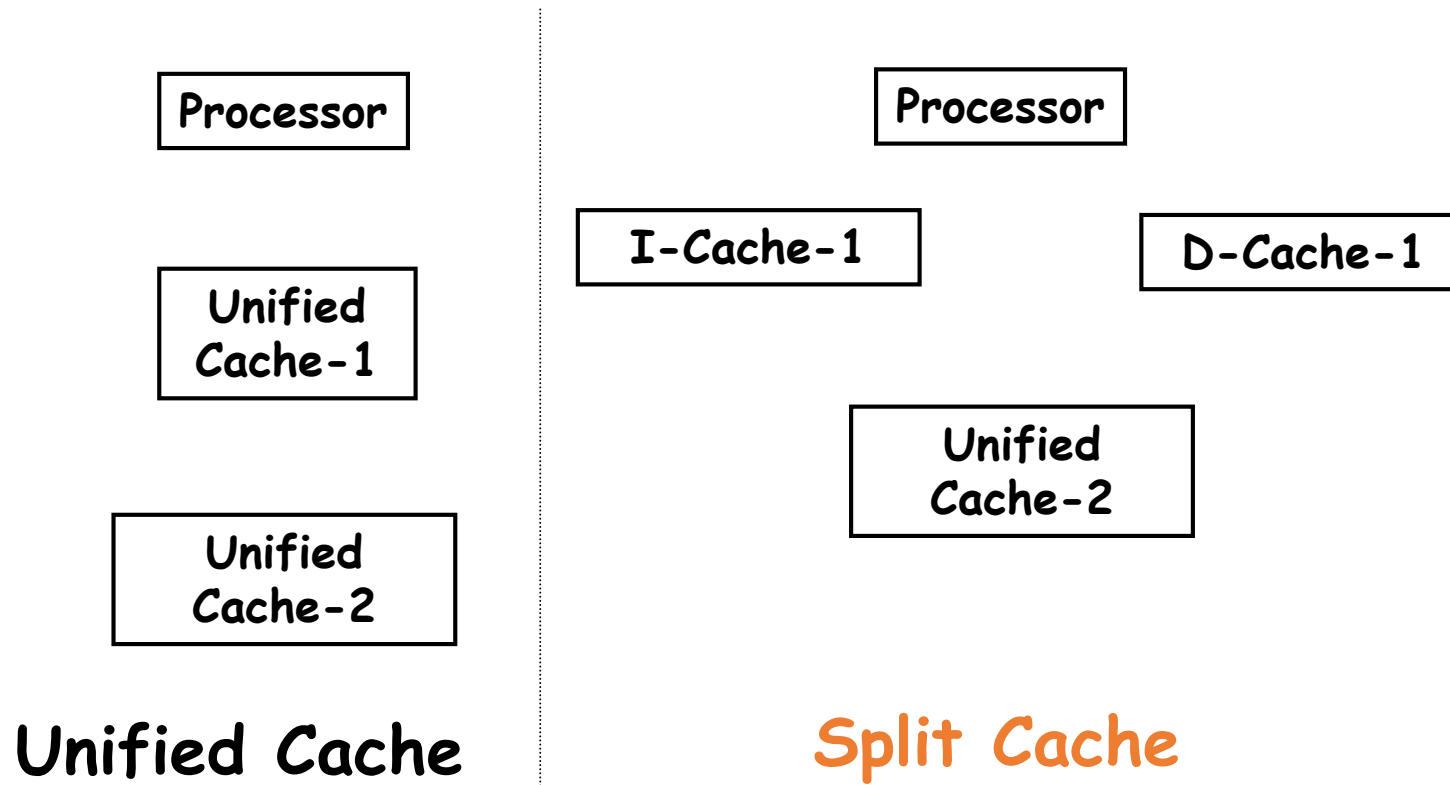
$$\begin{aligned} & \text{Total memory accesses per instruction} * \text{Miss rate} * \text{Miss Penalty} \\ & = (1 + 0.2) * 0.01 * 100 = 1.2 \text{ cycles} \end{aligned}$$

CPI with cache = Ideal CPI + Memory Stall cycle per instruction = $1 + 1.2 = 2.2$

CPI without cache = Ideal CPI + (Total memory access per instruction * Miss Penalty) = $1 + (1.2 * 100) = 121$

Unified vs Split Caches

- Separate Instruction and Data caches:
 - Avoids structural hazard



Example-1 (Split vs Unified)

Total instruction=1000	Miss per 1000 instruction
Instruction cache size=16 KB	3.82
Data cache size=16 KB	40.9
Unified cache size=32 KB	43.3

36% of the instructions are data transfers

74% of the memory accesses are instruction reference

A hit takes 1 clock cycles and a miss penalty is 100 clock cycles.

A load/store hit takes extra 1 clock cycle on a unified cache (because of structural hazard).

Which has lower miss rate? Obtains AMAT in each case.

Solution

Miss rate= Misses/memory access=(Misses/Instruction)/(Memory access/Instruction)

Total instruction=1000	Miss per instruction
Instruction cache size=16 KB	3.82/1000
Data cache size=16 KB	40.9/1000
Unified cache size=32 KB	43.3/1000

Miss rate (of 16KB instruction)= $(3.82/1000)/1.00=0.004$

Miss rate (of 16 KB data) = $(40.9/1000)/0.36=0.114$ (36% of the instructions are data transfer instruction)

Overall miss rate for the split cache= $(74\% * 0.004) + (26\% * 0.114)=0.0326$

- Miss rate (of the 32 KB unified Cache)= $(43.3/1000)/(1.00+0.36)=0.0318$
- 32KB of unified cache has slower miss rate than two 16 KB split cache.
- Average memory access time=%instruction * (Hit time + Miss rate* Miss penalty)+ %data * (Hit time + Miss rate* Miss penalty)
- AMAT (split)= $74\%*(1+0.004*200)+ 26\%*(1+0.114*200)=7.52$
- AMAT (unified)= $74\%*(1+0.0318*200)+ 26\%*(1+1+0.0318*200)=7.62$
- Split cache have better memory access time than Unified cache.

Example-2 (Split vs Unified)

Instruction cache size=16 KB

Data cache size=16 KB

Unified cache size=32 KB

Miss rates are as follows:

16KB instruction cache: 0.64%

16KB data cache: 6.47%

32KB unified cache: 1.99%

75% of the memory accesses are instruction reference

A hit takes 1 clock cycles and a miss penalty is 50 clock cycles.

A load/store hit takes extra 1 clock cycle on a unified cache

Which has lower miss rate? Obtains AMAT in each case.

Solution

$$\begin{aligned}\text{Overall Miss rate of Split cache} &= \{0.75 * (0.64/100)\} + \{0.25 * (6.47/100)\} \\ &= \{(0.75 * 0.64) + (0.25 * 6.47)\} / 100 \\ &= (0.48 + 1.61) / 100 = 2.09 / 100 = 2.09\%\end{aligned}$$

Miss rate of Unified Cache = 1.35%

Unified cache has slower miss rate.

- $\text{AMAT (split)} = 75\% * \{1 + (0.64/100) * 50\} + 25\% * \{1 + (6.47/100) * 50\}$
 $= 0.75(1 + 0.32) + 0.25(1 + 3.235) = 2.048$
- $\text{AMAT (unified)} = 75\% * \{1 + (1.99/100) * 50\} + 25\% * \{1 + 1 + (1.99/100) * 50\}$
 $= 2.246$
- Split cache have better (lower) memory access time than Unified cache.

Example 3

- Which has a lower miss rate?
 - A split cache (16KB instruction cache +16KB Data cache) or a 32 KB unified cache?
- Compute the respective AMAT also.
- 40% Load/Store instructions
- Hit time = 1 cycle
- Miss penalty = 100 cycles
- 70% of memory accesses are instruction references
- Simulator showed:
 - 40 misses per thousand instructions for data cache
 - 4 misses per thousand instr for instruction cache
 - 44 misses per thousand instr for unified cache

Solution:

- Miss rate = (misses/instructions)/(mem accesses/instruction)
- Instruction cache miss rate = $4/1000 = 0.004$
- Data cache miss rate = $(40/1000)/0.4 = 0.1$
- Unified cache miss rate = $(44/1000)/1.4 = 0.04$
- Overall miss rate for split cache = $0.3 * 0.1 + 0.7 * 0.004 = 0.0303$
- AMAT (split cache) = $0.7 * (1 + 0.004 * 100) + 0.3(1 + 0.1 * 100) = 4.3$
- AMAT (Unified) = $0.7(1 + 0.04 * 100) + 0.3(1 + 1 + 0.04 * 100) = 4.5$

Example 4

- Assume 16KB Instruction and Data Cache:
- Inst miss rate=0.64%, Data miss rate=6.47%
- 32KB unified cache: Aggregate miss rate=1.99%
- Assume 33% data ops (load/store)
- hit time=1, miss penalty=50
- Data hit has 1 additional stall for unified cache
- Which is better?

Solution

Assume 33% data ops \Rightarrow 75% accesses from instructions (1.0/1.33)

$$\begin{aligned} \text{AMAT}_{\text{Split}} &= 75\% \times (1 + 0.64\% \times 50) + 25\% \times (1 + 6.47\% \times 50) \\ &= 2.05 \end{aligned}$$

$$\begin{aligned} \text{AMAT}_{\text{Unified}} &= 75\% \times (1 + 1.99\% \times 50) + 25\% \times (1 + 1 + 1.99\% \times 50) \\ &= 2.24 \end{aligned}$$

Assignment -1

[3+3+4]

1. Draw and explain the Scoreboard architecture.
2. Draw and explain the Tomasulo architecture.
3. Differentiate between no-write allocate and write allocate method during cache miss.

Assume a fully associative write back cache with many cache entries that starts empty. Below is a sequence of memory operations (the address is in square brackets):

WriteMem[100];

WriteMem[200];

ReadMem[300];

WriteMem[200];

WriteMem[300].

What are the number of hits and misses when using no-write allocate VS write allocate?