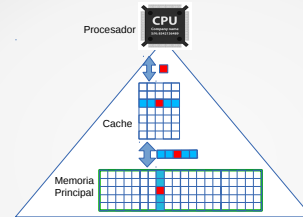


Memoria Cache Performance de Cache

Tiempo promedio de acceso a cache

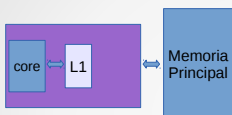


$$\text{Average Memory Access time} = \text{Hit time} + \text{Miss rate} * \text{Miss penalty}$$

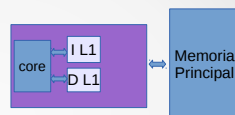
Tiempo de acceso al nivel más alto, incluido el tiempo para determinar si fue un miss o un hit

Contempla el tiempo de acceso al bloque y el tiempo de transferencia

Ejemplo: Cache unificada vs. Cache separada



- (+) Menor Miss Rate
- (-) Menor ancho de banda
- (-) Optimización única
- (+) Espacio total para D e I
- (-) Conflictos entre bloques de datos y bloques de instrucciones
- (-) Hazard estructural por F y M de Load/Store



- (-) Mayor Miss Rate
- (+) Mayor ancho de banda
- (+) Optimización individual
- (-) Tamaño fijo para D e I
- (+) Elimina conflictos entre bloques de datos y bloques de instrucciones
- (+) Elimina hazard estructural por F y M de Load/Store

Ejemplo: Cache unificada vs. Cache separada

- Comparamos Cache de Instrucciones de 16KB + Cache de Datos de 16KB con Cache Unificada de 32KB

| Cache size | Miss Rate | | |
|------------|-------------------|------------|---------------|
| | Instruction cache | Data cache | Unified cache |
| 8KB | 0.008 | 0.122 | 0.0463 |
| 16KB | 0.004 | 0.114 | 0.0375 |
| 32KB | 0.001 | 0.107 | 0.0318 |
| 64KB | 0.001 | 0.103 | 0.0290 |
| 128KB | 0.0003 | 0.098 | 0.0266 |
| 256KB | 0.00002 | 0.091 | 0.0242 |

74% de los accesos a memoria son referencias a Instrucciones

Ejemplo: Cache unificada vs. Cache separada

- Comparamos Cache de Instrucciones de 16KB + Cache de Datos de 16KB con Cache Unificada de 32KB

| Cache size | Miss Rate | | |
|------------|-------------------|------------|---------------|
| | Instruction cache | Data cache | Unified cache |
| 8KB | 0.008 | 0.122 | 0.0463 |
| 16KB | 0.004 | 0.114 | 0.0375 |
| 32KB | 0.001 | 0.107 | 0.0318 |
| 64KB | 0.001 | 0.103 | 0.0290 |
| 128KB | 0.0003 | 0.098 | 0.0266 |
| 256KB | 0.00002 | 0.091 | 0.0242 |

Para calcular el Miss Rate de la cache separada ¿puedo sumar los dos miss rates y listo?

74% de los accesos a memoria son referencias a Instrucciones

Ejemplo: Cache unificada vs. Cache separada

- Miss Rate para Caches separadas:

$$\text{Miss Rate}_s = \%AC_{\text{INST}} * \text{Miss Rate}_{\text{INST}} + \%AC_{\text{DATOS}} * \text{Miss Rate}_{\text{DATOS}}$$

$$\text{Miss Rate}_s = (0.74 * 0.004) + (0.26 * 0.114) = 0.0326$$

Ejemplo: Cache unificada vs. Cache separada

- Miss Rate para Caches separadas:

$$\text{Miss Rate}_s = (0.74 * 0.004) + (0.26 * 0.114) = 0.0326$$

| Miss Rate | | | |
|------------|-------------------|------------|---------------|
| Cache size | Instruction cache | Data cache | Unified cache |
| 8KB | 0.008 | 0.122 | 0.0463 |
| 16KB | 0.004 | 0.114 | 0.0375 |
| 32KB | 0.001 | 0.107 | 0.0318 |
| 64KB | 0.001 | 0.103 | 0.0290 |
| 128KB | 0.0003 | 0.098 | 0.0266 |
| 256KB | 0.00002 | 0.091 | 0.0242 |

Ejemplo: Cache unificada vs. Cache separada

- Miss Rate para Caches separadas:

$$\text{Miss Rate}_s = (0.74 * 0.004) + (0.26 * 0.114) = 0.0326$$

| Miss Rate | | | |
|------------|-------------------|------------|---------------|
| Cache size | Instruction cache | Data cache | Unified cache |
| 8KB | 0.008 | 0.122 | 0.0463 |
| 16KB | 0.004 | 0.114 | 0.0375 |
| 32KB | 0.001 | 0.107 | 0.0318 |
| 64KB | 0.001 | 0.103 | 0.0290 |
| 128KB | 0.0003 | 0.098 | 0.0266 |
| 256KB | 0.00002 | 0.091 | 0.0242 |

¿Tener caches separadas es peor?

Ejemplo: Cache unificada vs. Cache separada

- ¿Qué pasa con el **Average Memory Access Time**?
 - Suponiendo
 - Hit time de 1 ciclo en caches separadas
 - Miss penalty de 10 ciclos en ambas
 - Referencias a datos toman 1 ciclo más en cache unificada
 - Analizamos AMAT separando en Instrucciones y Datos

$$\text{Average Memory Access Time (AMAT)} = \text{\%Acc}_{\text{INST}}(\text{Hit time} + \text{Miss Rate}_I * \text{Miss Penalty}) + \text{\%Acc}_{\text{DATOS}}(\text{Hit time} + \text{Miss Rate}_D * \text{Miss Penalty})$$

Ejemplo: Cache unificada vs. Cache separada

- ¿Qué pasa con el **Average Memory Access Time**?
 - Suponiendo
 - Hit time de 1 ciclo en caches separadas
 - Miss penalty de 10 ciclos en ambas
 - Referencias a datos toman 1 ciclo más en cache unificada

$$\text{AMAT}_s = 0.74 * (1 + 0.004 * 10) + 0.26 * (1 + 0.114 * 10) = 1.326$$

$$\text{AMAT}_u = 0.74 * (1 + 0.0318 * 10) + 0.26 * (1 + 1 + 0.0318 * 10) = 1.578$$

$$\text{AMAT}_u = 1.19 * \text{AMAT}_s$$

Ejemplo: Dos Niveles de Cache

- Agregar un segundo nivel de cache apunta a reducir la disparidad entre el procesador y MP
 - Si bien la idea es simple, el análisis de performance se modifica

$$\text{Average Memory Access time} = \text{Hit time} + \text{Miss rate} * \text{Miss penalty}$$

Se convierte en:

$$\text{Average Memory Access time} = \text{Hit time}_{L1} + \text{Miss rate}_{L1} * \text{Miss penalty}_{L1}$$

Donde:

$$\text{Miss penalty}_{L1} = \text{Hit time}_{L2} + \text{Miss rate}_{L2} * \text{Miss penalty}_{L2}$$

Ejemplo: Dos Niveles de Cache

- ¿Cuál será el **Average Memory Access Time**? Si:
 - Cada 1000 referencias a memoria hay 40 misses en L1 y 20 misses en L2
 - Hit Time_{L1} = 1 ciclo
 - Hit Time_{L2} = 10 ciclos
 - Miss Penalty_{L2} = 200 ciclos

$$\begin{aligned} \text{AMAT}_{2\text{niveles}} &= \text{Hit time}_{L1} + \text{Miss rate}_{L1} * \text{Miss penalty}_{L1} \\ &= 1 + (40/1000) * (\text{Hit time}_{L2} + \text{Miss rate}_{L2} * \text{Miss penalty}_{L2}) \\ &= 1 + 0.04 * (10 + (20/40) * 200) \\ &= 1 + 0.04 * 110 = 5.4 \text{ ciclos} \end{aligned}$$

$$\begin{aligned} \text{AMAT}_{\text{sin L2}} &= 1 + (40/1000) * 200 \\ &= 1 + 0.04 * 200 = 9 \text{ ciclos} \end{aligned}$$

Impacto de la cache en la Performance del Procesador

• Performance de CPU

$$CPU_{time} = CPU_{ciclos} * Clock\ cycle\ time$$

• Si incluimos los ciclos de stall por accesos a memoria:

$$CPU_{time} = (CPU_{ciclos} + Memory\ stall\ cycles) * Clock\ cycle\ time$$

Incluye ciclos de Hit en cache

Impacto de la cache en la Performance del Procesador

• Performance de CPU

$$CPU_{time} = (CPU_{ciclos} + Memory\ stall\ cycles) * CC_{time}$$

Diagram showing the breakdown of the formula:

- $Memory\ stall\ cycles = N^{\circ}\ Misses * Miss\ penalty$
- $N^{\circ}\ Misses = IC * CPI * \frac{Misses}{Instrucción}$
- $\frac{Misses}{Instrucción} = \frac{Accesos\ a\ Mem}{Instrucción} * Miss\ Rate$

Impacto de la cache en la Performance del Procesador

• Performance de CPU

$$CPU_{time} = (CPU_{ciclos} + Memory\ stall\ cycles) * CC_{time}$$

$$CPU_{ciclos} = IC * CPI$$

$$Memory\ stall\ cycles = IC * \frac{Accesos\ a\ Mem}{Instrucción} * Miss\ Rate$$

$$CPU_{time} = IC * (CPI + \frac{Accesos\ a\ Mem}{Instrucción} * Miss\ Rate * Miss\ Penalty) * CC_{time}$$

Impacto de la cache en la Performance del Procesador

• Ejemplo 1: Supongamos una arquitectura con

- CPI = 8,5
- Miss Rate = 11%
- Miss Penalty = 6 ciclos
- 3 Accesos a memoria / Instrucción

¿Cuál es el impacto de la caché en la performance?

$$CPU_{time} = IC * (CPI + \frac{Accesos\ a\ Mem}{Instrucción} * Miss\ Rate * Miss\ Penalty) * CC_{time}$$

Impacto de la cache en la Performance del Procesador

• Ejemplo 1: Supongamos una arquitectura con

- CPI = 8,5
- Miss Rate = 11%
- Miss Penalty = 6 ciclos
- 3 Accesos a memoria / Instrucción

¿Cuál es el impacto de la caché en la performance?

Sin considerar la cache (todos Hits):

$$CPU_{time} = IC * (8,5 + 0) * CCT$$

$$CPU_{time} = 8,5 * IC * CCT$$

24% de Incremento

Considerando la cache:

$$CPU_{time} = IC * (8,5 + 3 * 0,11 * 6) * CCT$$

$$CPU_{time} = 10,48 * IC * CCT$$

Impacto de la cache en la Performance del Procesador

• Ejemplo 2: una arquitectura con

- CPI = 1,5
- Miss Rate = 11%
- Miss Penalty = 10 ciclos (no es más lenta la memoria)
- 1,4 Accesos a memoria / Instrucción

¿Cuál es el impacto de la caché en la performance?

Sin considerar la cache (todos Hits):

$$CPU_{time} = IC * (1,5 + 0) * CCT$$

$$CPU_{time} = 1,5 * IC * CCT$$

100% de Incremento

Considerando la cache:

$$CPU_{time} = IC * (1,5 + 1,4 * 0,11 * 10) * CCT$$

$$CPU_{time} = 3,04 * IC * CCT$$

Impacto de la cache en la Performance del Procesador

| | Ej. 1 | Ej. 2 |
|---------------------------|-------|-------|
| CPI | 8,5 | 1,5 |
| Miss Rate | 11% | 11% |
| Miss Penalty | 6 | 10 |
| AaM/Inst | 3 | 1,4 |
| CPU _{time} (scc) | 8,5 | 1,5 |
| CPU _{time} (cc) | 10,48 | 3,04 |
| Incremento | 24% | 100% |

Impacto en la performance por considerar la caché

- **Doble impacto** en CPUs con CPI bajo y reloj rápido
 - A menor CPI → mayor impacto relativo
 - A mayor frecuencia de reloj → mayor miss penalty

$$CPU_{time} = IC * (CPI + \frac{Accesos\ a\ Mem}{Instrucción} * Miss\ Rate * Miss\ Penalty) * CC_{time}$$

Impacto de la cache en la Performance del Procesador

- **Ejemplo 3:** Supongamos que para dos organizaciones de cache, una **Mapeo directo** y la otra **2-way**, tenemos

- CPI = 2 (CCT = 2ns)
- Miss Penalty = 70ns (un nº entero de ciclos)
- 1,3 Accesos a memoria / Instrucción
- Cache size = 64KB (parte de datos)
- Bloques de 32B
- Hit time = 1 ciclo
- Miss Rate_{mapeo_directo} = 1,4%
- Miss Rate_{2-way} = 1%
- Reloj_{2-way} = 10% mayor que Reloj_{mapeo_directo}

- Determinar el **tiempo promedio de acceso**
- Determinar la **performance del procesador**

Impacto de la cache en la Performance del Procesador

- **Ejemplo 3:**
 - Tiempo promedio de acceso

$$Average\ Memory\ Access\ time = Hit\ time + Miss\ rate * Miss\ penalty$$

$$AMAT_{MD} = 1 * 2.0ns + (0.014 * 70ns) = 2.98ns$$

$$AMAT_{2-way} = 1 * 2.0ns * 1.1 + (0.01 * 70ns) = 2.90ns$$

| | |
|-----------------------|--------|
| CPI | 2 |
| Miss P | 70ns |
| %AM/I | 1.3 |
| Hit time | 1ciclo |
| Miss R _{MD} | 1.4 |
| Miss R _{2-w} | 1.0 |
| Reloj _{2-w} | 10%+ |

Impacto de la cache en la Performance del Procesador

- **Ejemplo 3:**
 - Performance del procesador

$$CPU_{time} = IC * (CPI + \frac{Accesos\ a\ Mem}{Instrucción} * Miss\ Rate * Miss\ Penalty) * CC_{time}$$

$$CPU_{time} = IC * CPI * CC_{time} + IC * \frac{Accesos\ a\ Mem}{Instrucción} * Miss\ Rate * Miss\ Penalty * CC_{time}$$

$$CPU_{t_{MD}} = 2 * 2ns * IC + 1.3 * 0.014 * 70ns * IC = 5.27 * IC$$

$$CPU_{2-way} = 2 * 2ns * 1.1 * IC + 1.3 * 0.01 * 70ns * IC = 5.31 * IC$$

| | |
|-----------------------|--------|
| CPI | 2.0 |
| Miss P | 70ns |
| %AM/I | 1.3 |
| Hit time | 1ciclo |
| Miss R _{MD} | 1.4 |
| Miss R _{2-w} | 1.0 |
| Reloj _{2-w} | 10%+ |

Impacto de la cache en la Performance del Procesador

- **Ejemplo 3:**
 - Performance del procesador

$$CPU_{t_{MD}} = 2 * 2ns * IC + 1.3 * 0.014 * 70ns * IC = 5.27 * IC$$

$$CPU_{2-way} = 2 * 2ns * 1.1 * IC + 1.3 * 0.01 * 70ns * IC = 5.31 * IC$$

- Relación

$$\frac{CPU_{2-way}}{CPU_{t_{MD}}} = \frac{5.31 * IC}{5.27 * IC} = 1.007$$

| | |
|-----------------------|--------|
| CPI | 2.0 |
| Miss P | 70ns |
| %AM/I | 1.3 |
| Hit time | 1ciclo |
| Miss R _{MD} | 1.4 |
| Miss R _{2-w} | 1.0 |
| Reloj _{2-w} | 10%+ |

Impacto de la cache en la Performance del Procesador

- **Ejemplo 3:**
 - Performance del procesador

$$CPU_{t_{MD}} = 2 * 2ns * IC + 1.3 * 0.014 * 70ns * IC = 5.27 * IC$$

$$CPU_{2-way} = 2 * 2ns * 1.1 * IC + 1.3 * 0.01 * 70ns * IC = 5.31 * IC$$

- Relación

$$\frac{CPU_{2-way}}{CPU_{t_{MD}}} = \frac{5.31 * IC}{5.27 * IC} = 1.007$$

Si miramos el CPU time, vemos que mapeo directo parece tener una performance levemente mejor a pesar de tener mayor miss rate.

| | |
|-----------------------|--------|
| CPI | 2.0 |
| Miss P | 70ns |
| %AM/I | 1.3 |
| Hit time | 1ciclo |
| Miss R _{MD} | 1.4 |
| Miss R _{2-w} | 1.0 |
| Reloj _{2-w} | 10%+ |

Tipos de misses

- **Compulsivos:** son los que se ocasionan en la primera vez que un bloque es referenciado.
- **Capacitivos:** son los que se producen si la cache no puede contener a todos los bloques que se necesitan para la ejecución de un programa.
- **Conflictivos:** si se utiliza mapeo directo o set asociativo, se producen (además) misses debido a que muchos bloques mapean al mismo set, lo que genera que se descarten bloques que son requeridos posteriormente.

Referencias

Hennessey, J., and Patterson, D. Computer Architecture, second ed. Morgan Kaufmann, 1996. (Capítulo 5)

Hennessey, J., and Patterson, D. Computer Architecture, fourth ed. Morgan Kaufmann, 2006. (Apéndice C)