Unidad 5: Algoritmos de memoria externa

Modelo inconsciente de caché

Gibran Fuentes Pineda

Mayo 2020

Caché

- Almacena elementos (por ej. resultados anteriores o datos) que van a ser utilizados por el CPU para que su acceso sea más rápido
- Asociatividad limitada
  - Cada bloque se puede almacenar en solo *c* de *M* partes
  - Valores típicos de *c* son 1 y 2, aunque algunas cachés pueden tener 4 u 8

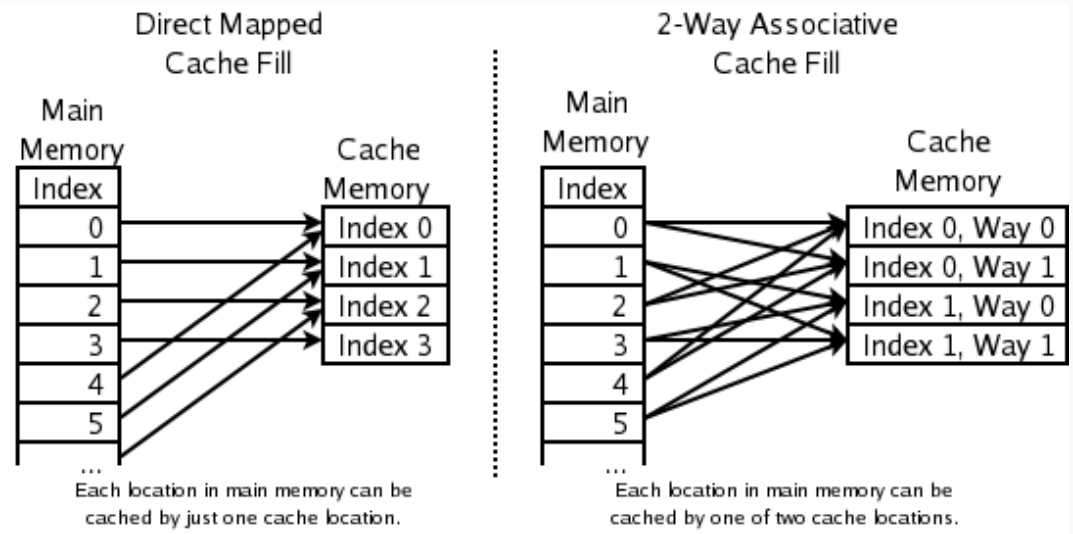


Imagen tomada de entrada de Wikipedia Cache (Computiong)

Localidad en acceso a memoria

- Caché aprovecha la localidad en el acceso a memoria
  - **Temporal**: elementos tienden a ser utilizados múltiples veces en tiempos cercanos
  - **Espacial**: elementos requeridos tienden a estar localizados en partes cercanas

- La jerarquía completa de memoria se puede ver como distintos niveles de caché, cada uno transfiriendo datos a sus niveles adyacentes en bloques

Acierto y fallo de caché

- Se llama *acierto de caché* si un bloque se encuentra disponible en caché cuando es requerido

- El *fallo de caché* ocurre si un bloque no puede ser accedido cuando se necesita y este se tenga que copiar de la memoria interna, provocando un tiempo de espera

Estrategias de reemplazo

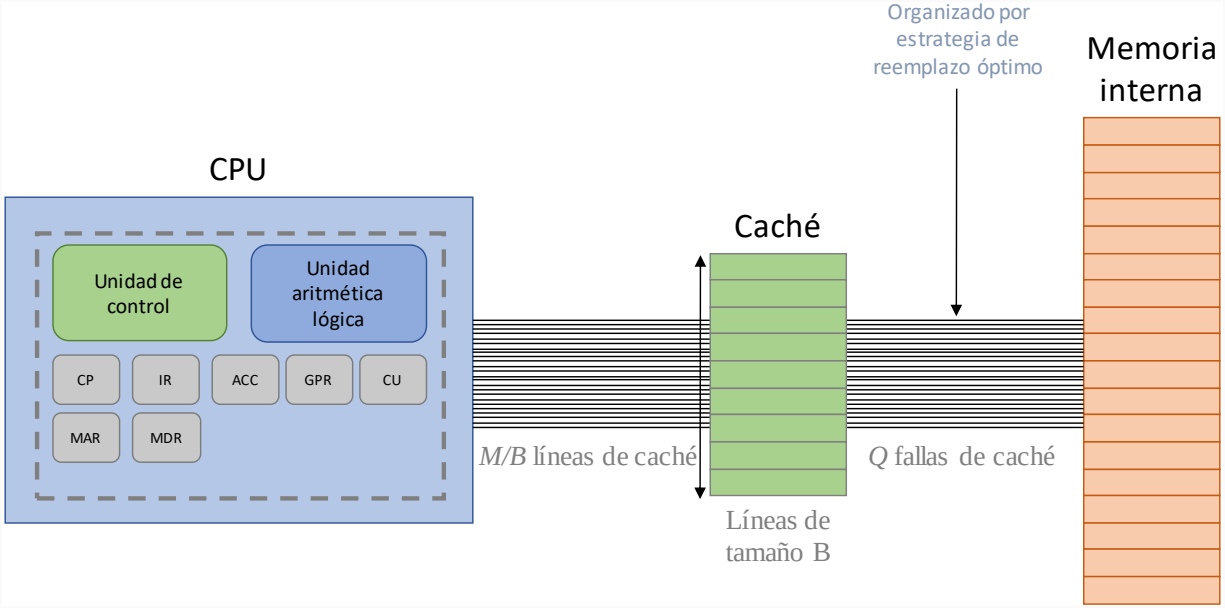
- Cuando ocurre un fallo y la caché está llena, se reemplaza un bloque

- Estrategias para reemplazar un bloque
  - El menos utilizado recientemente (LRU, *Least Recently Used*)
  - El menos frecuentemente utilizado (LFU, *Least Frequently Used*)
  - Primero en entrar, primero en salir (FIFO, *First In First Out*)
  - OPT

Modelo inconsciente de caché (Frigo et al. 1999)

- Modelo de diseño de algoritmos para jerarquías de memoria multinivel que no requiere conocimiento del tamaño del caché *M* ni del bloque de transferencia *B*

- Presupone un *caché ideal*



Caché ideal

- Reemplazo de página óptimo: bloque reemplazado es aquel que se accederá al último en el futuro
- Caché completamente asociativo de M elementos: cuando se carga un bloque del nivel $\ell + 1$, este puede residir en cualquier segmento del nivel ℓ
- Cada bloque contiene B elementos
- Caché es más alto que ancho: el número de bloques M/B es más grande que el tamaño B de los bloques, esto es, $M = \Omega(B^2)$

Algoritmos inconscientes de caché

- Desempeño equivalente en cualquier par de niveles adyacentes de la jerarquía de memoria
- No gestionan de forma explícita las solicitudes de lectura y escritura de caché
- Tratan de evitar el número de fallos de caché: cada bloque que se copia contiene tanta información útil como sea posible
- Métricas de rendimiento
  - Número de fallas de caché ($Q(N; M, B)$)
  - Complejidad temporal

Propiedad de inclusión

- Modelo inconsciente de caché trabaja en dos niveles de memoria que tienen la propiedad de inclusión
  - Los datos no pueden estar en el nivel ℓ a menos que también estén presentes en el nivel $\ell + 1$
  - El tamaño del nivel ℓ en la jerarquía de memoria es estrictamente más pequeño que el nivel $\ell + 1$

Justificación del modelo inconsciente de caché

1. Si un algoritmo tiene $Q(N; M, B)$ fallos (T transferencias) con un caché ideal, entonces sufre a lo mucho $2Q(N; M, B)$ fallos ($2T$ transferencias) con reemplazo LRU o FIFO

2. Un caché con reemplazo LRU o FIFO se puede mantener usando una memoria de tamaño $O(M)$ tal que cualquier acceso a un bloque tome un tiempo esperado de $O(1)$

Condición de regularidad

- La condición de regularidad de una cota de complejidad de caché es $Q(N; M, B) = O(Q(N; 2M, B))$
  - Número de transferencias $T(M) = O(T(M/2))$

- Si un algoritmo con cota regular tiene $Q(N; M, B)$ fallos ($T(M)$ transferencias) usando reemplazo óptimo, entonces tiene $\Theta(Q(N; M, B))$ fallos ($\Theta(T(M))$ transferencias) usando LRU o FIFO

Escaneo en el modelo inconsciente de caché

- **Problema**: Recorrer todos los elementos de un arreglo
- Modelo de memoria externa: se almacenan los elementos en $\lceil N/B \rceil$ bloques de tamaño B y se recorren bloque por bloque
  - Se requieren $\lceil N/B \rceil$ transferencias
- Modelo inconsciente de caché: se almacenan los elementos en un segmento contiguo de memoria y se recorren uno por uno en el orden que fueron almacenados
  - Se requieren $\lceil N/B \rceil + 1$ transferencias

Divide y vencerás

- Diseño de algoritmos inconscientes de caché hace amplio uso de la estrategia *divide y vencerás*
- Varios algoritmos basados en esta estrategia se pueden considerar inconscientes de caché
- Al dividir el problema en subproblemas de forma recursiva, en algún nivel los subproblemas caben en caché (M) y en divisiones posteriores en B