UNIVERSIDAD DEL BÍO-BÍO
FACULTAD DE CIENCIAS EMPRESARIALES

Jerarquía de memoria
Computación Heterogénea

Profesor: Dr. Joel Fuentes - jfuentes@ubiobio.cl

Ayudantes:
- Daniel López - daniel.lopez1701@alumnos.ubiobio.cl
- Sebastián González - sebastian.gonzalez1801@alumnos.ubiobio.cl

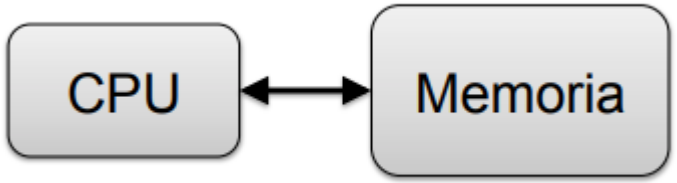
Página web del curso: http://www.face.ubiobio.cl/~jfuentes/classes/ch

Contenidos

- Conceptos elementales
- Jerarquía de memoria
- Caching

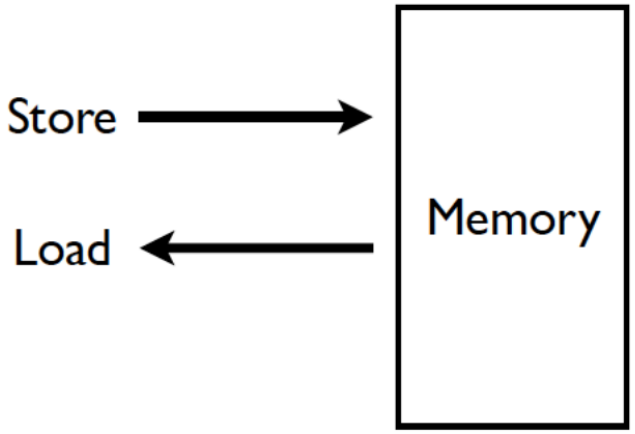
Conceptos elementales

- Limitaciones de eficiencia entre CPU y memoria suelen ser latencia y ancho de banda
- Latencia (latency): tiempo para un único acceso
  - Tiempo de acceso a memoria >> tiempo de ciclo del procesador
- Ancho de banda (bandwidth): números de accesos por unidad de tiempo

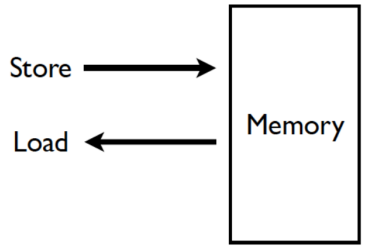


Conceptos elementales

- Lo que el programador ve:

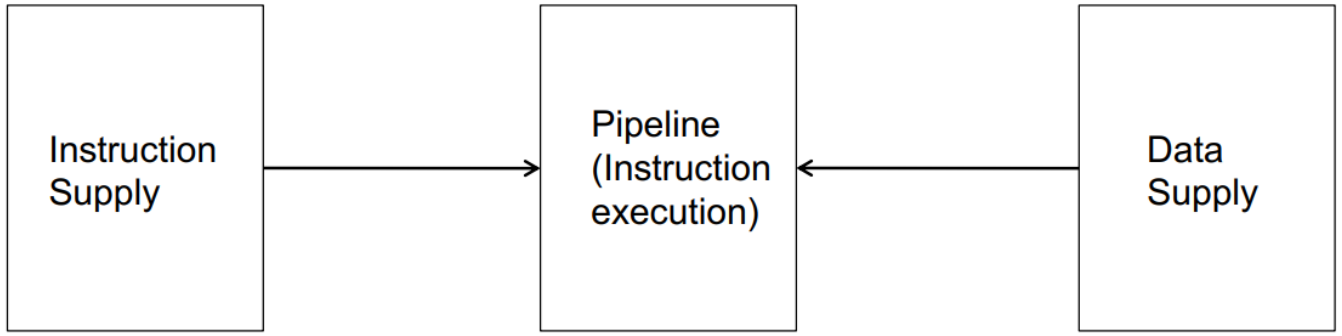


Memoria virtual vs memoria física



- El programador ve memoria virtual
  - Asume que la memoria es infinita
- Realidad: El tamaño de la memoria física es mucho mas pequeña que lo que el programador asume.
- El sistema (software y hardware) mapea direcciones de memoria virtual a memoria física.
  - Este mapeo es completamente transparente para el programador

En un mundo ideal…



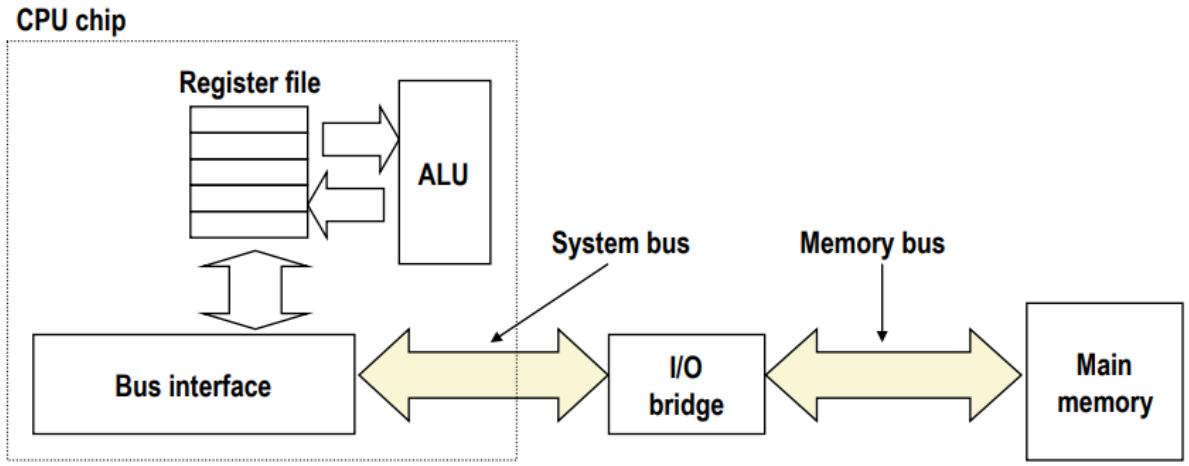
- Zero latency access
- Infinite capacity
- Zero cost
- Perfect control flow

- No pipeline stalls
- Perfect data flow (reg/memory dependencies)
- Zero-cycle interconnect (operand communication)
- Enough functional units
- Zero latency compute

- Zero latency access
- Infinite capacity
- Infinite bandwidth
- Zero cost

**Obtenido desde CSC 2224 University of Toronto, Prof. Gennady Pekhimenko.*

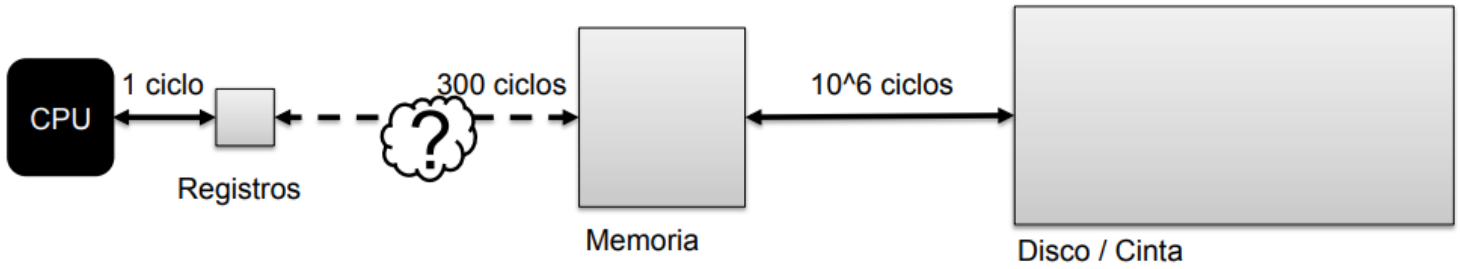
Nace la jerarquía de memoria



- Idea: Tener múltiples niveles de almacenamiento
- Progresivamente más grandes y más lentos a medida que se encuentran más alejados del procesador, pero asegurando que la mayor cantidad de datos que el procesador necesita están en los niveles más rápidos

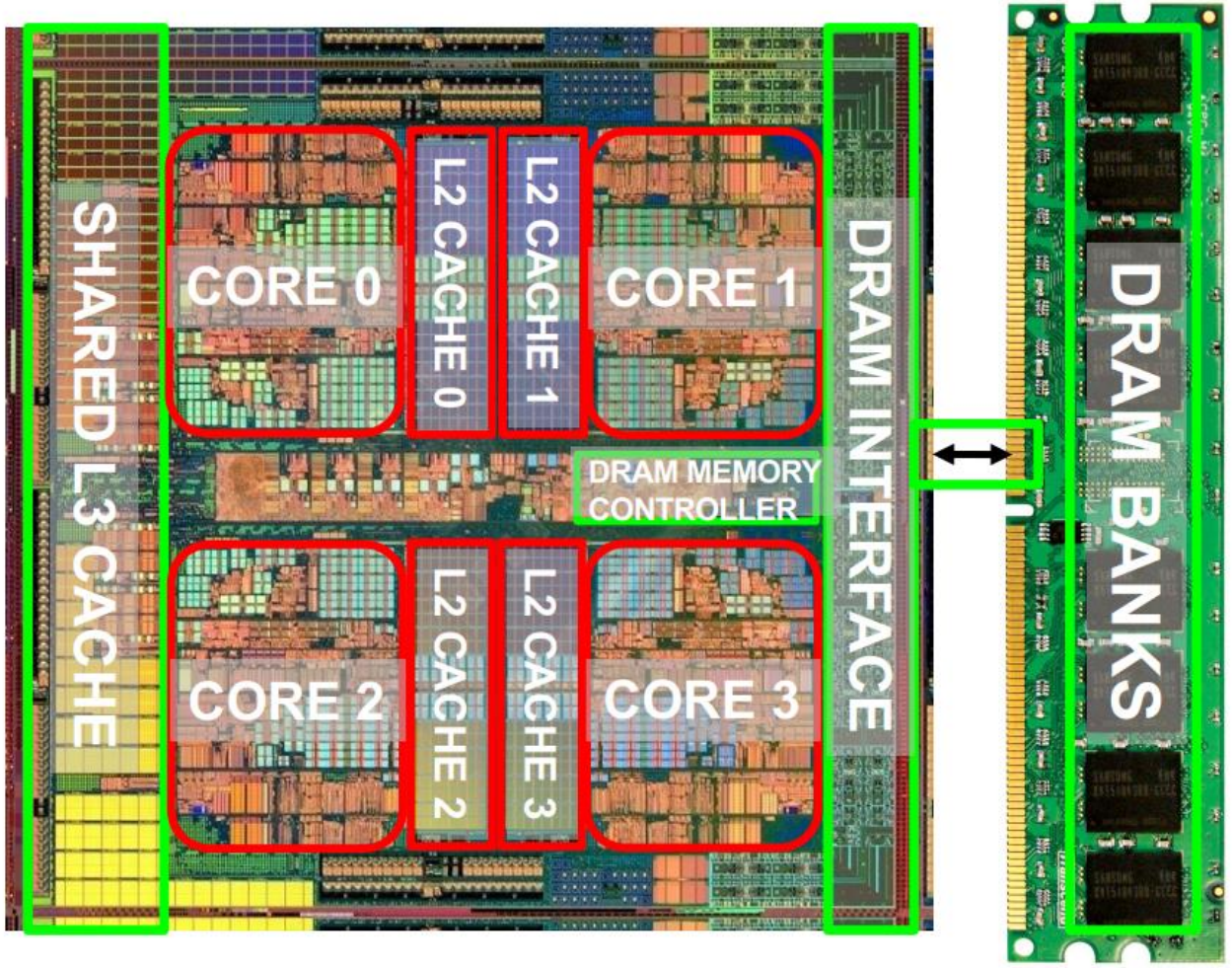
Nace la jerarquía de memoria

- Existe una gran camino (número de ciclos) para rescata/escribir datos entre diferentes memorias.



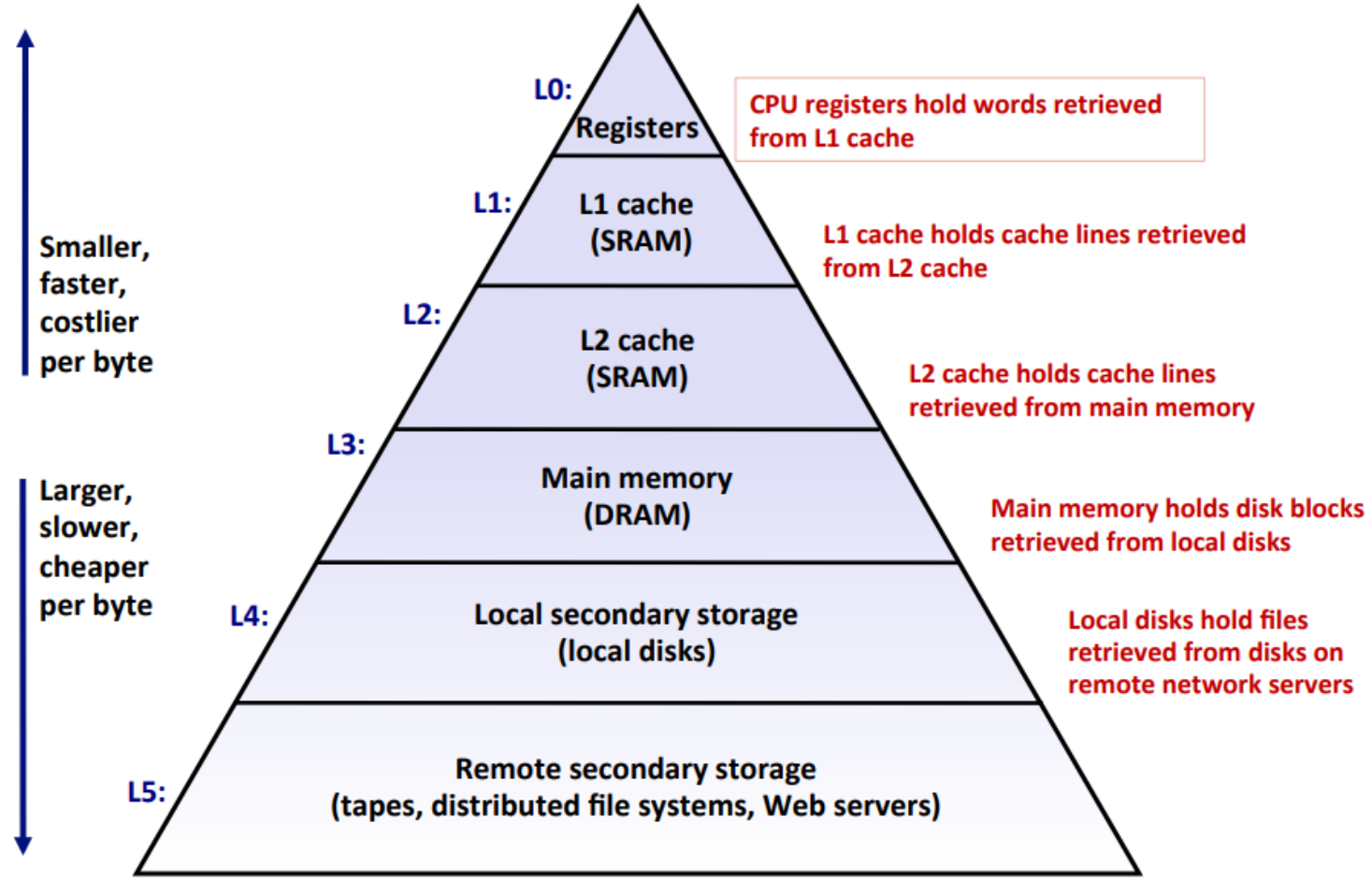
- ¿Qué falta para mejorar este proceso?

Jerarquía de memoria



**Obtenido desde CSC 2224 University of Toronto, Prof. Gennady Pekhimenko.*

Jerarquía de memoria



**Obtenido desde 15-213/18-243 Carnegie Mellon University, G. Kesden and A. Rowe.*

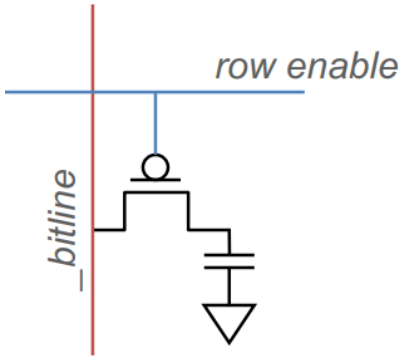
Principales problemas

- Entre más grande la memoria más lenta
  - Entre más grande es más complejo determinar la ubicación (dirección de memoria física)
- Memorias más rápidas son más caras
  - Tecnologías: SRAM vs DRAM vs Disk
- Mayor ancho de banda es más caro
  - Se necesitan más bancos de memoria, puertos, frecuencia más alta, tecnología más rápida, etc.

**Obtenido desde 15-213/18-243 Carnegie Mellon University, G. Kesden and A. Rowe.*

DRAM

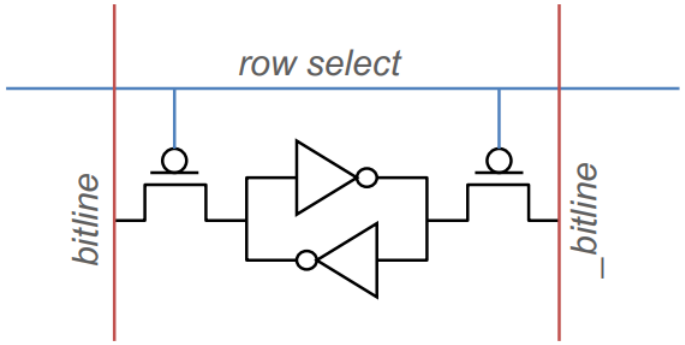
- Memoria dinámica de acceso aleatorio
- Estado de carga de capacitor indica valor almacenado
  - Si el capacitor está cargado o descargado indica almacenamiento de 1 o 0.
  - 1 capacitor
  - 1 transistor de acceso
- Capacitor se libera a través de la ruta RC
  - Celdas de DRAM pierden carga con el tiempo
  - Celdas de DRAM necesitan ser refrescadas



**Obtenido desde CSC 2224 University of Toronto, Prof. Gennady Pekhimenko.*

SRAM

- Memoria estática de acceso aleatorio
- Típicamente conocidas como memorias caché
- Dos inversores cruzados almacenan un bit individual
  - Ruta de retroalimentación permite que el valor almacenado persista en la celda
  - 4 transistores para almacenamiento
  - 2 transistores para acceso



*Obtenido desde CSC 2224 University of Toronto, Prof. Gennady Pekhimenko.

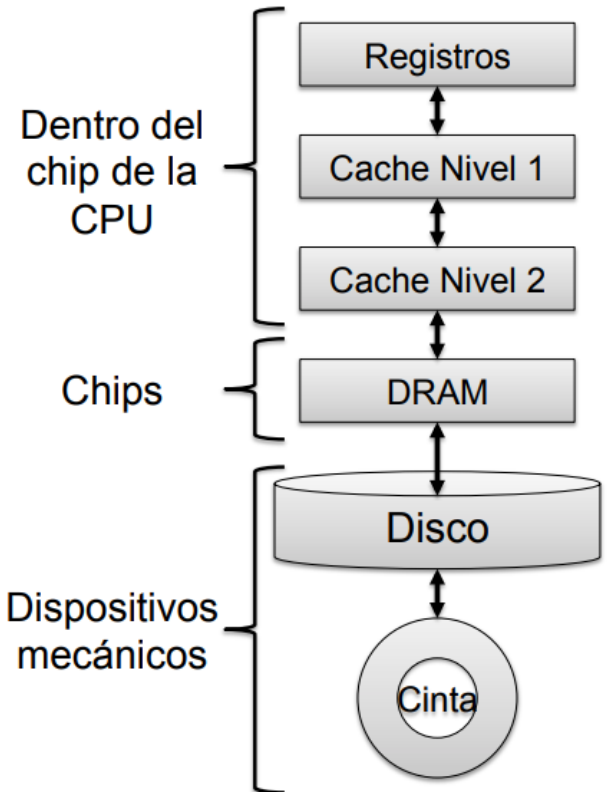
DRAM vs SRAM

- DRAM
  - Acceso más lento (capacitor)
  - Densidad más alta (1 Transistor 1 capacitor por celda)
  - Costo menor
  - Requiere refrescar memoria (energía, rendimiento, circuitería)
  - Manufacturación requiere poner capacitor y lógica junta

- SRAM
  - Acceso más rápido (sin capacitor)
  - Densidad más baja (6 transistores por celda)
  - Costo alto
  - No requiere refrescar memoria
  - Manufacturación compatible con proceso lógico (sin capacitor)

**Obtenido desde CSC 2224 University of Toronto, Prof. Gennady Pekhimenko.*

Jerarquía de memoria



Tiempo de acceso	Capacidad	Gestionado por
1 ciclo	~500 Bytes	software/compilador
1-3 ciclos	~64 KB	hardware
5-10 ciclos	1 – 10 MB	hardware
~100 ciclos	~10 GB	software/OS
10^6 – 10^7 ciclos	~100 GB	software/OS

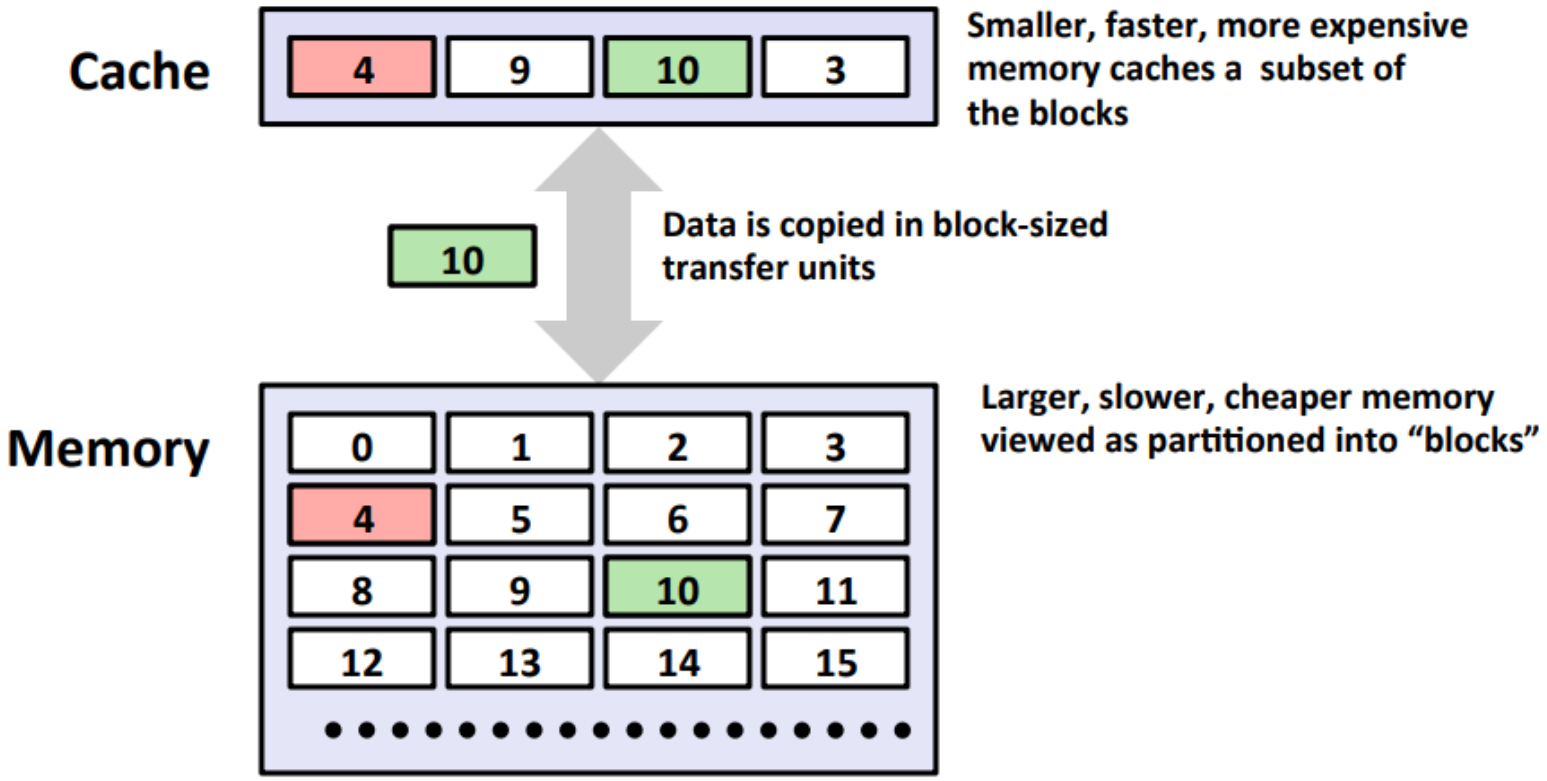
**Obtenido desde CSC 2224 University of Toronto, Prof. Gennady Pekhimenko.*

Conceptos sobre Caching

- Bloque (block line): Unidad de almacenamiento en caché
  - La memoria es lógicamente divida en bloques que son mapeados a ubicaciones en caché.
- Hit: Si dato está en caché, usar dato en caché en vez de accesar memoria
- Miss o falla: Si dato no está en caché, traer bloque a caché
- Colocación: Dónde y cómo ubicar un bloque en cache
- Reemplazo: Qué dato remover para generar espacio en caché

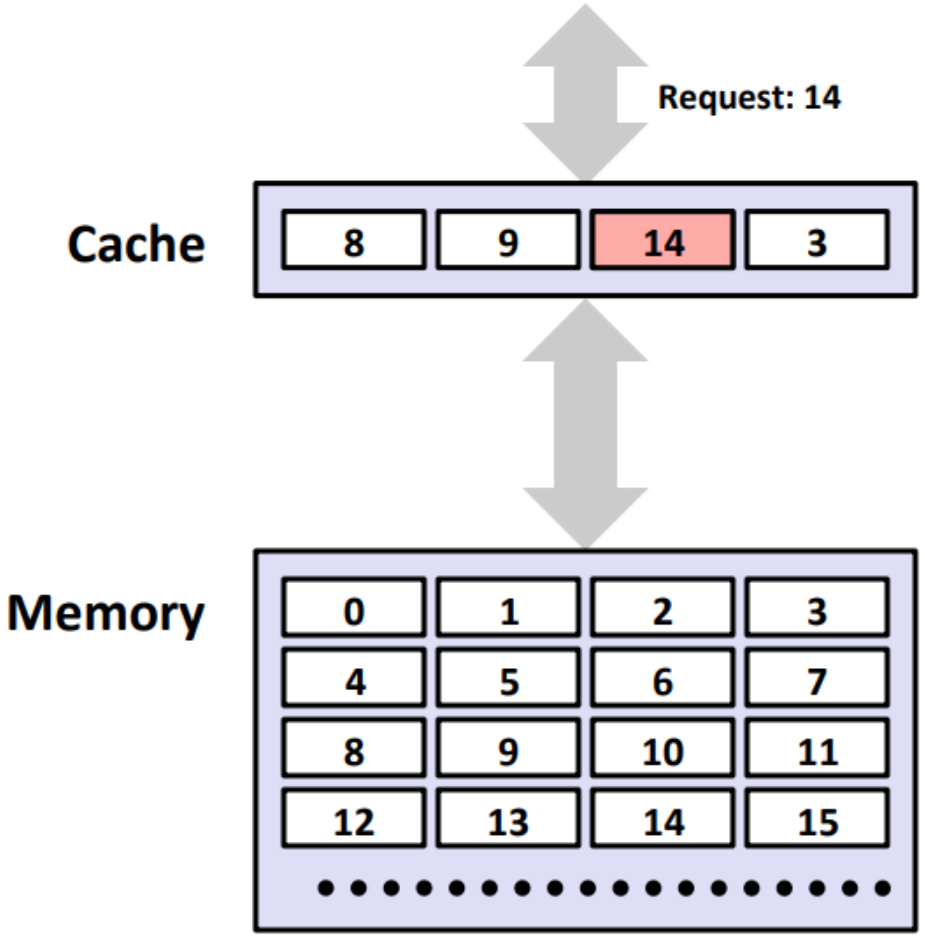
**Obtenido desde CSC 2224 University of Toronto, Prof. Gennady Pekhimenko.*

Caching



*Obtenido desde 15-213/18-243 Carnegie Mellon University, G. Kesden and A. Rowe.

Caching: Hit

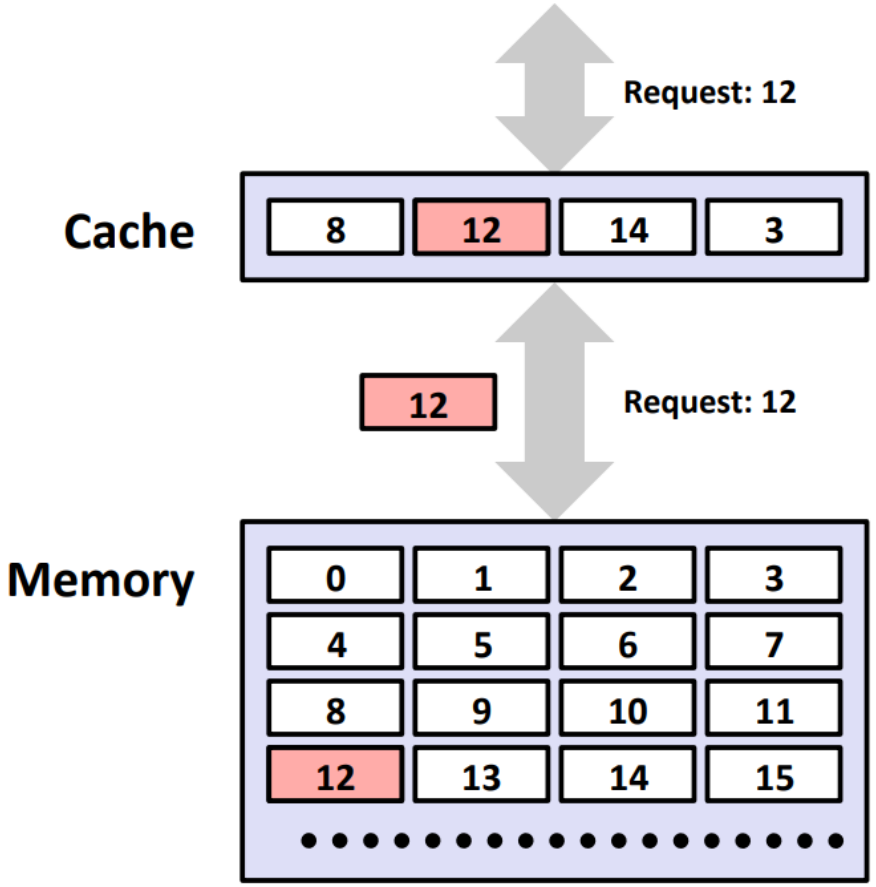


Data in block b is needed

Block b is in cache:
Hit!

**Obtenido desde 15-213/18-243 Carnegie Mellon University, G. Kesden and A. Rowe.*

Caching: Miss



Data in block b is needed

Block b is not in cache:
Miss!

Block b is fetched from memory

Block b is stored in cache
- Placement policy: determines where b goes
- Replacement policy: determines which block gets evicted (victim)

**Obtenido desde 15-213/18-243 Carnegie Mellon University, G. Kesden and A. Rowe.*