UNIVERSIDAD DEL BÍO-BÍO
FACULTAD DE CIENCIAS EMPRESARIALES

Multi-core CPU
Computación Heterogénea

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

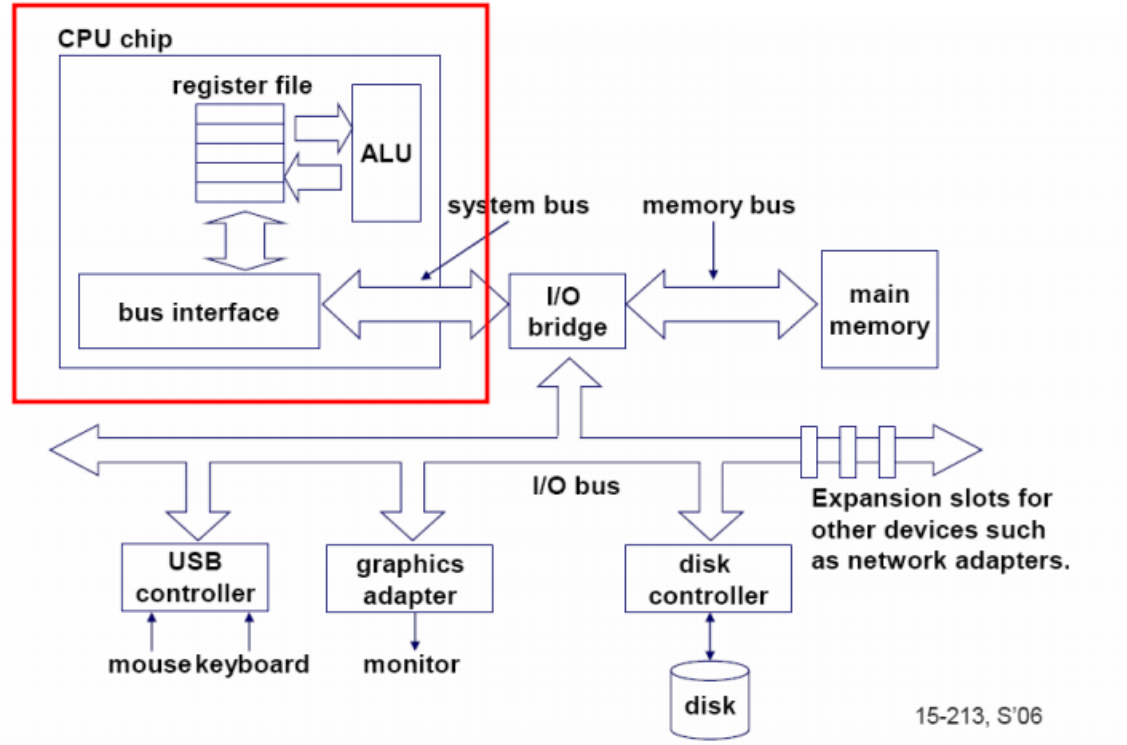
Ayudantes:
- Daniel López - daniel.lopez1701@alumnos.ubiobio.cl
- Sebastián González - sebastian.gonzalez1801@alumnos.ubiobio.cl

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

Contenidos

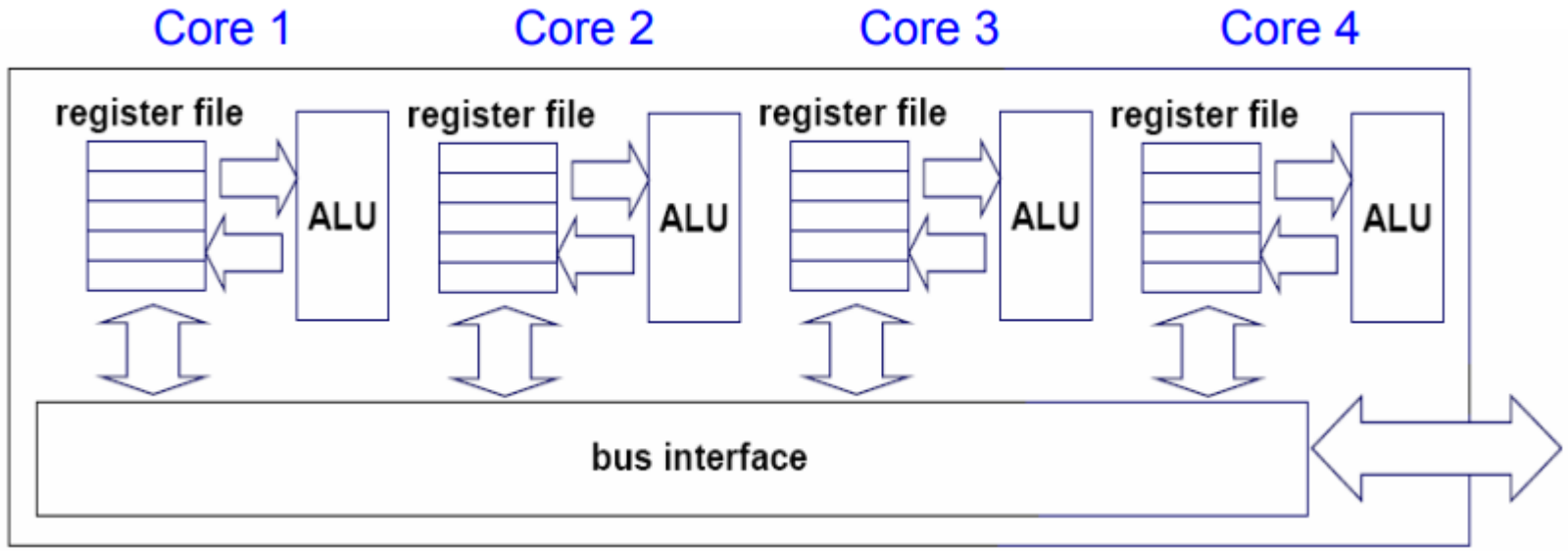
- Single-core CPU
- Multi-core CPU
- Coherencia de caché

Single-core CPU

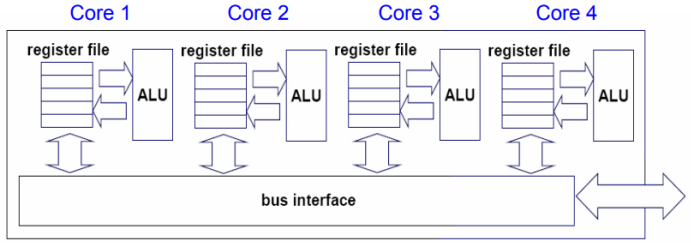


Arquitectura multi-core

- Idea principal: replicar múltiples cores en un mismo procesador



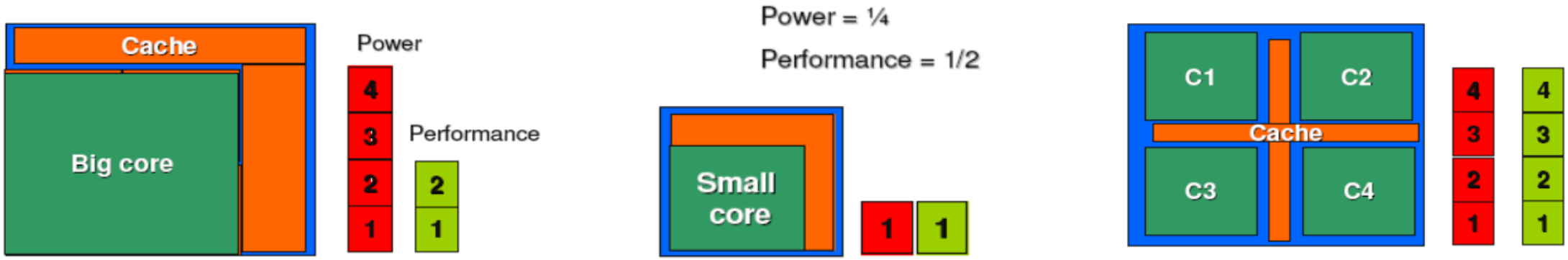
Arquitectura multi-core



- ¿Por qué multi-core es una buena idea?
  - Es difícil crear procesadores single-clore que trabaje a alta frecuencia
  - Problemas de calor son manejados de mejor manera en multi-core
  - Beneficios de paralelismo en muchas apps. Mejor rendimiento.
  - Al igual que single-core, multi-core también soporta ejecuciones multi-tarea

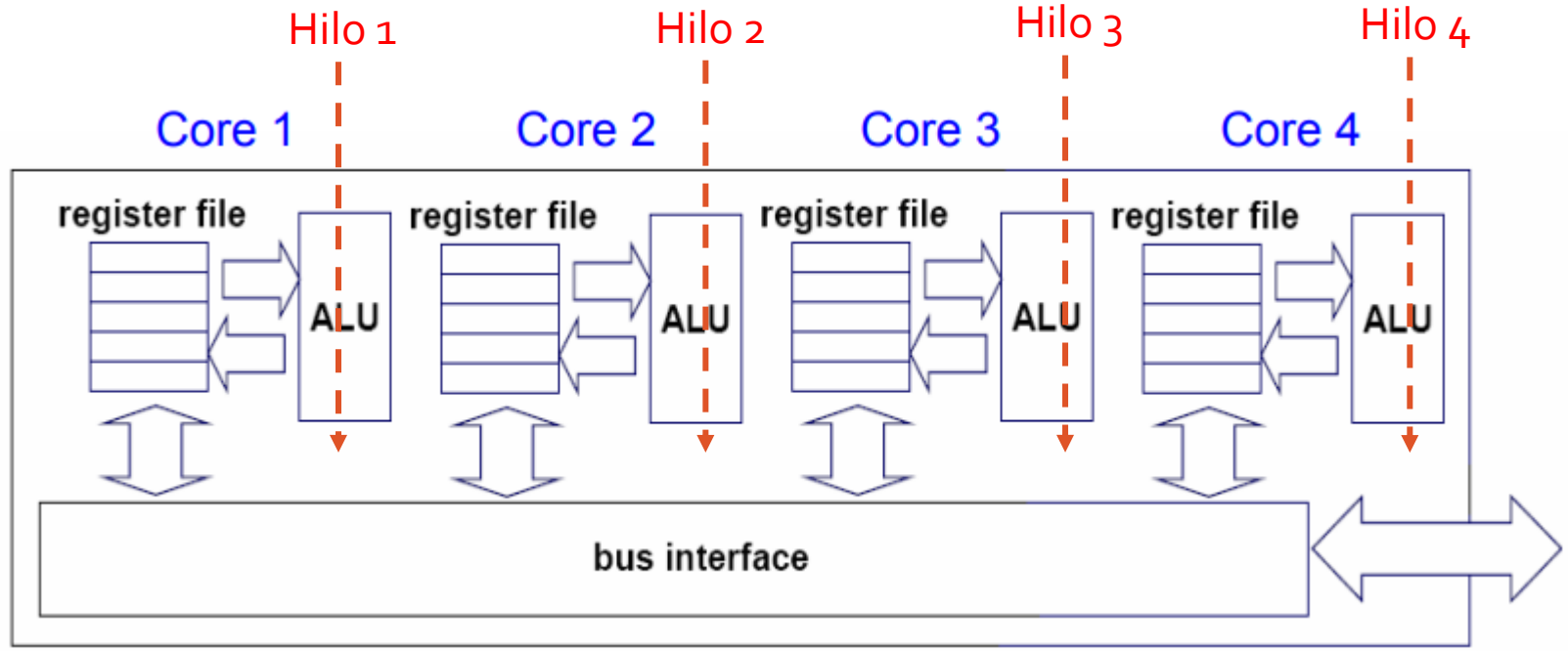
Arquitectura multi-core

- ¿Por qué multi-core es una buena idea?
  - Multi-core entrega mayor performance por watt



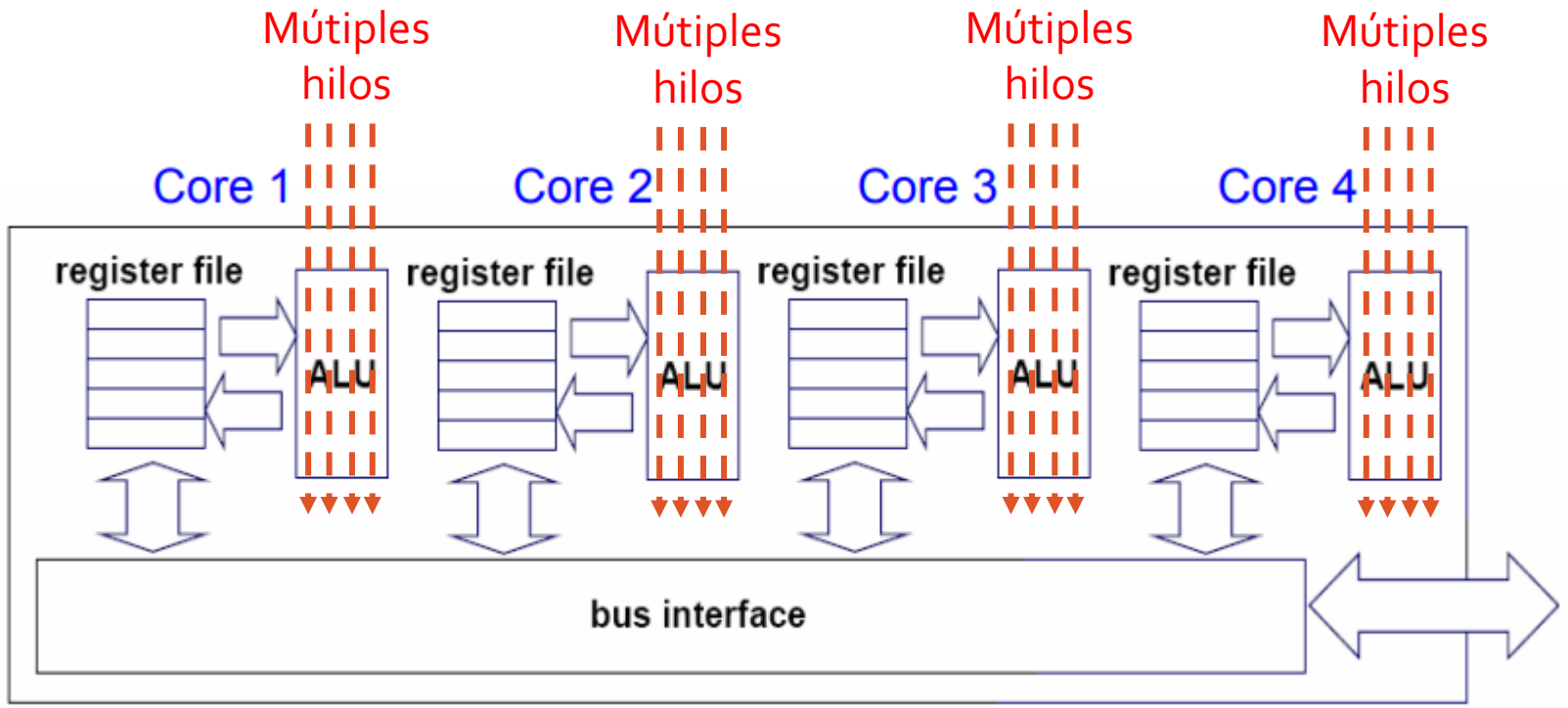
Arquitectura multi-core

- Cores ejecutan hilos en paralelo



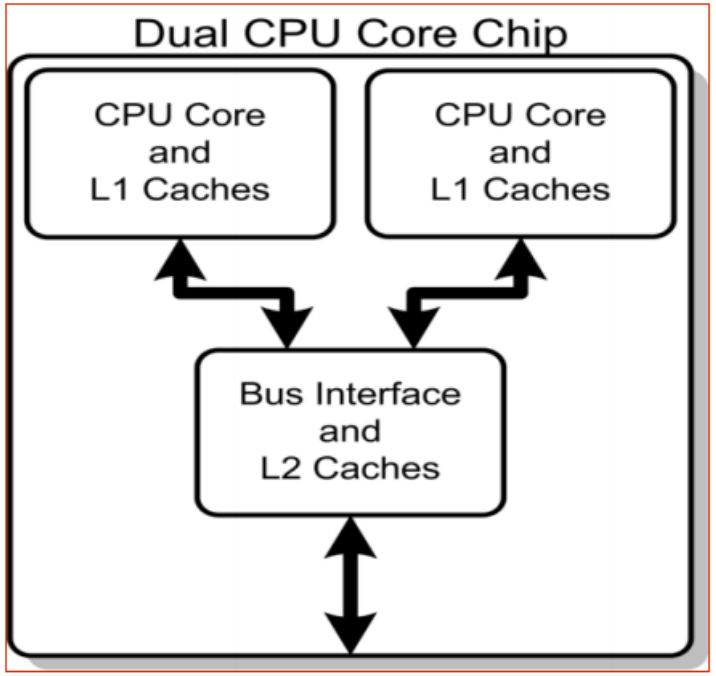
Arquitectura multi-core

- En cada core, múltiples hilos son ejecutados de forma de forma intercalada.



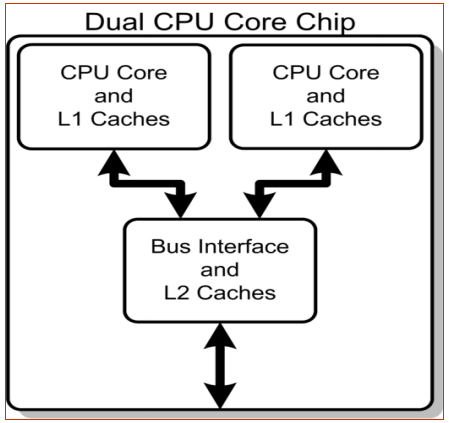
Arquitectura multi-core

- Diagrama conceptual de una CPU dual-core
  - Cada core cuenta con caché L1
  - Ambos cores están conectados con caché L2

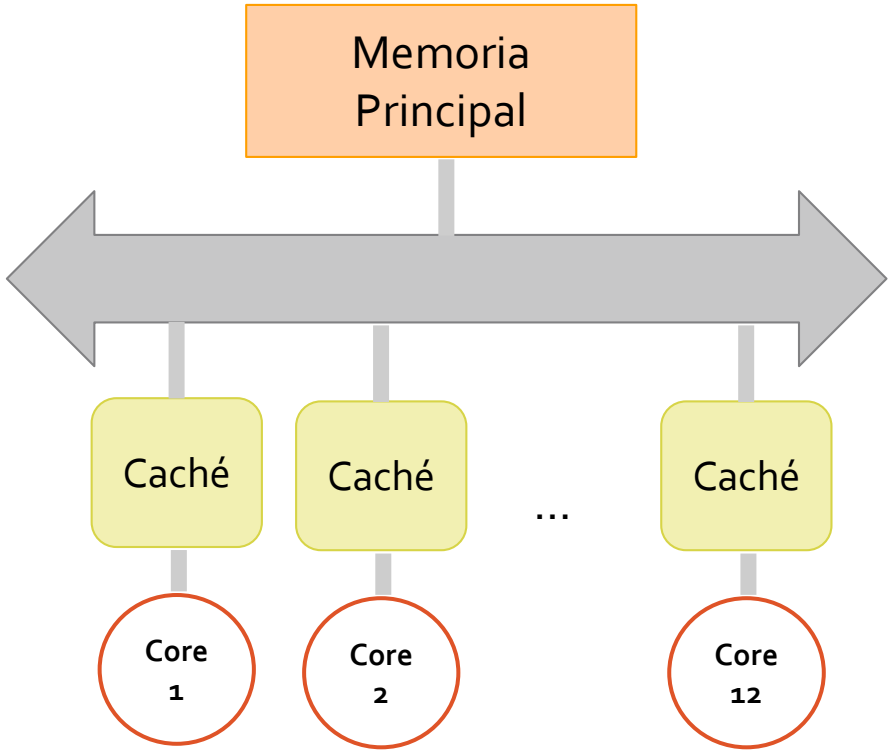


Arquitectura multi-core

- Problemas
  - Coherencia de memoria
  - Soporte del sistema operativo
  - Afinidad procesador-caché

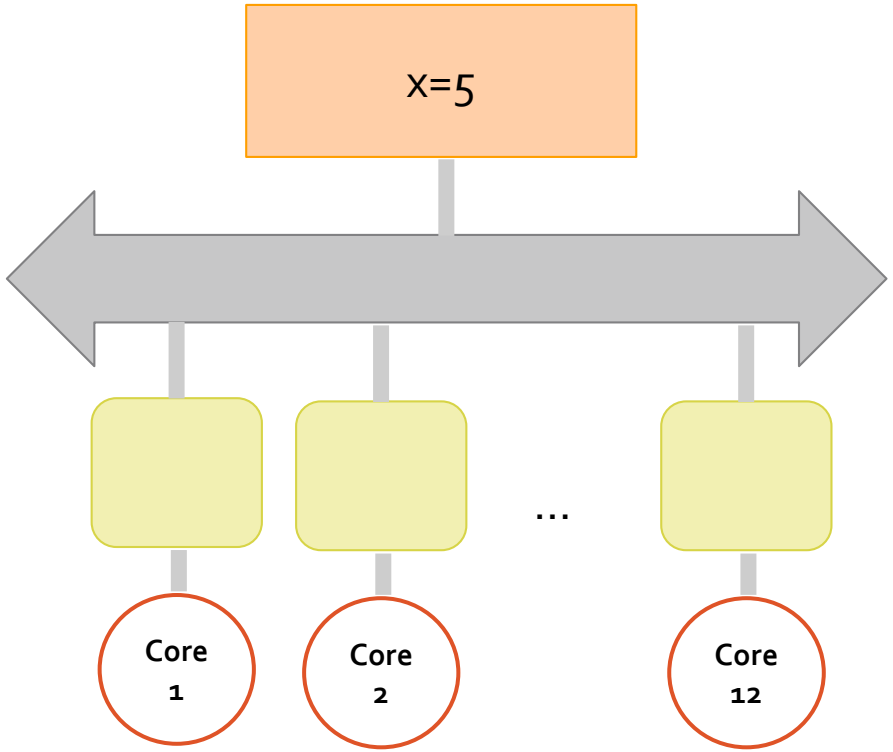


Coherencia de caché



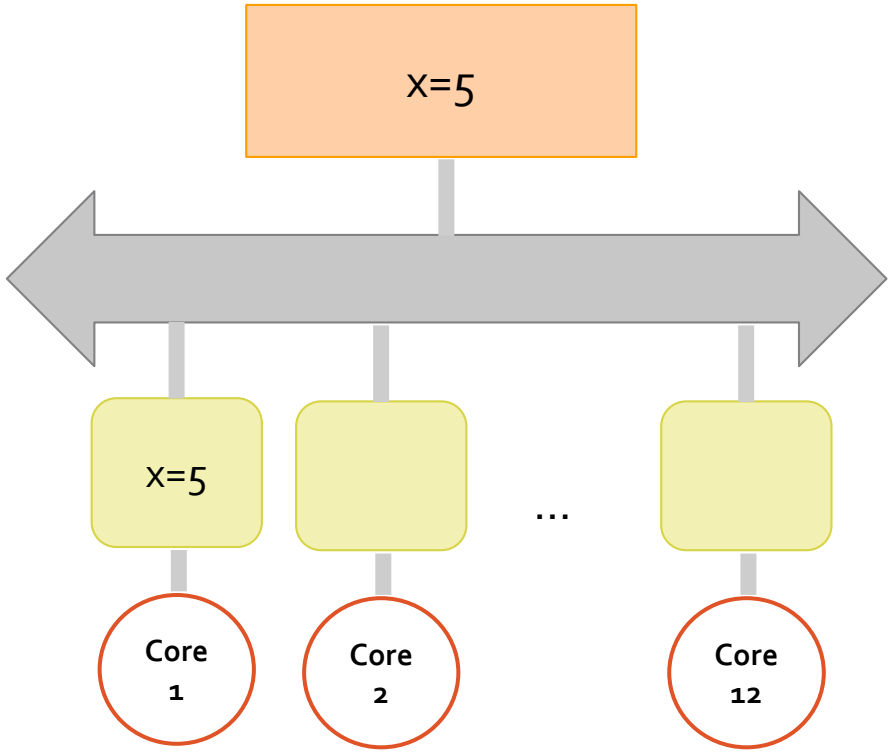
Coherencia de caché

- X tiene un valor 5 en memoria principal



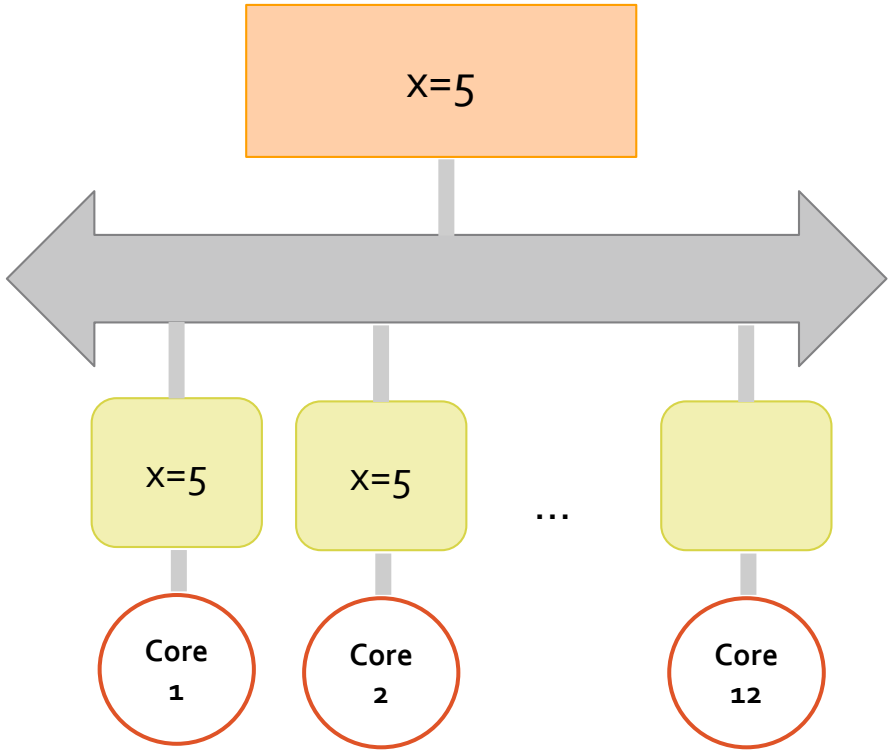
Coherencia de caché

- Core 1 lee x



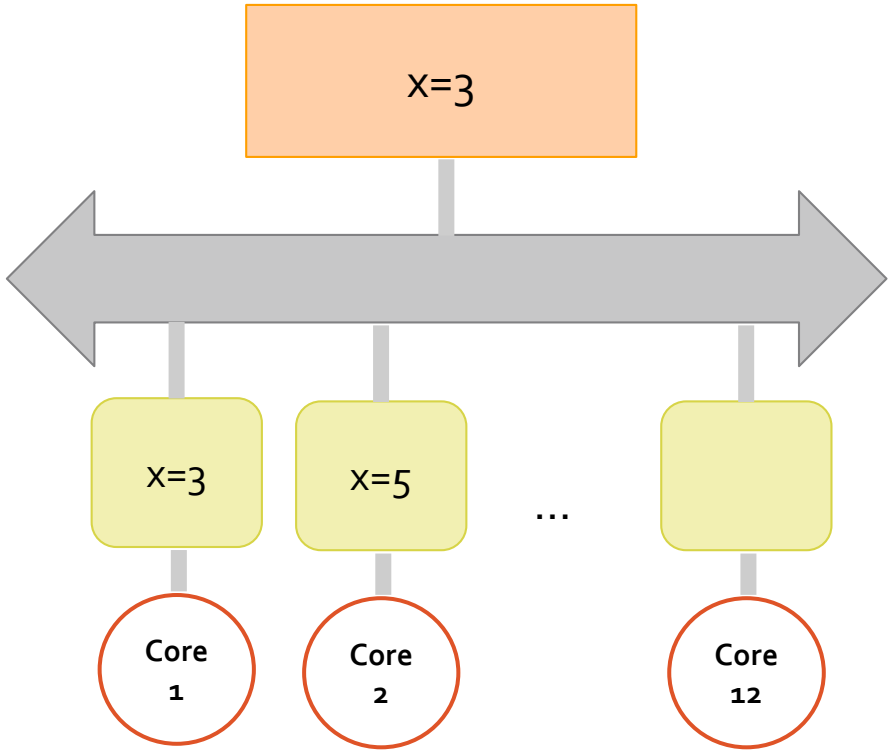
Coherencia de caché

- Core 2 lee x



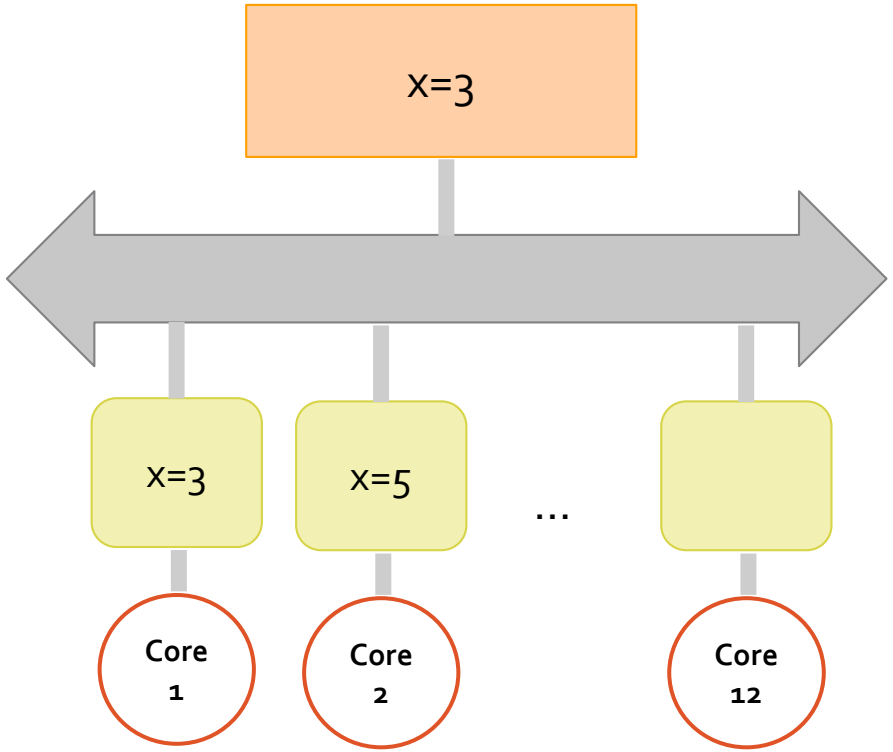
Coherencia de caché

- Core 1 escribe x y le asigna valor 3
- X en memoria principal es actualizado



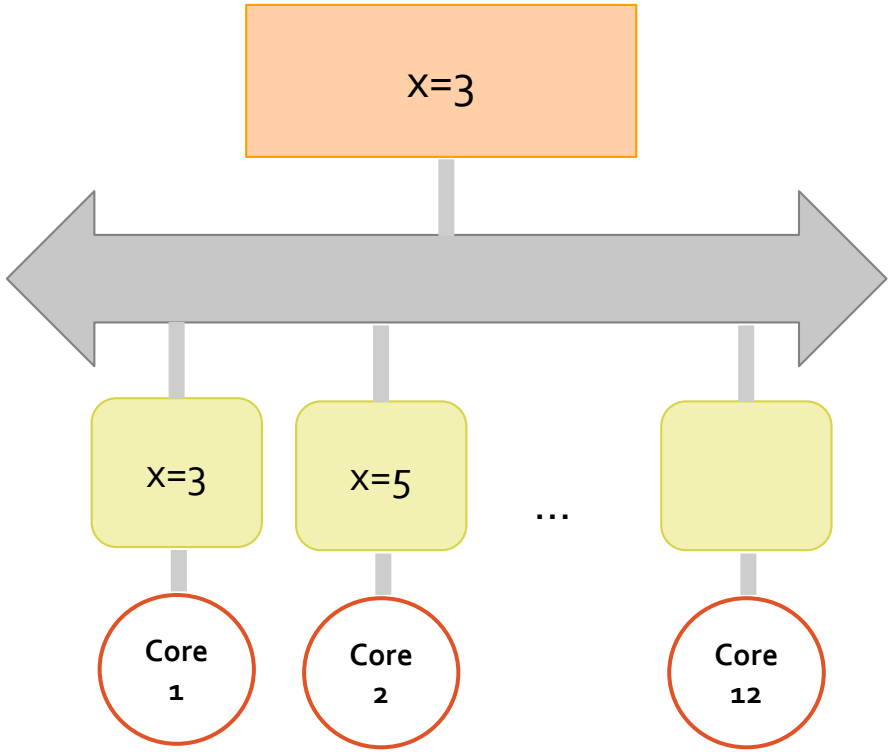
Coherencia de caché

- Core 2 lee nuevamente x pero tiene una copia desactualizada
- Problema de coherencia de datos en caché



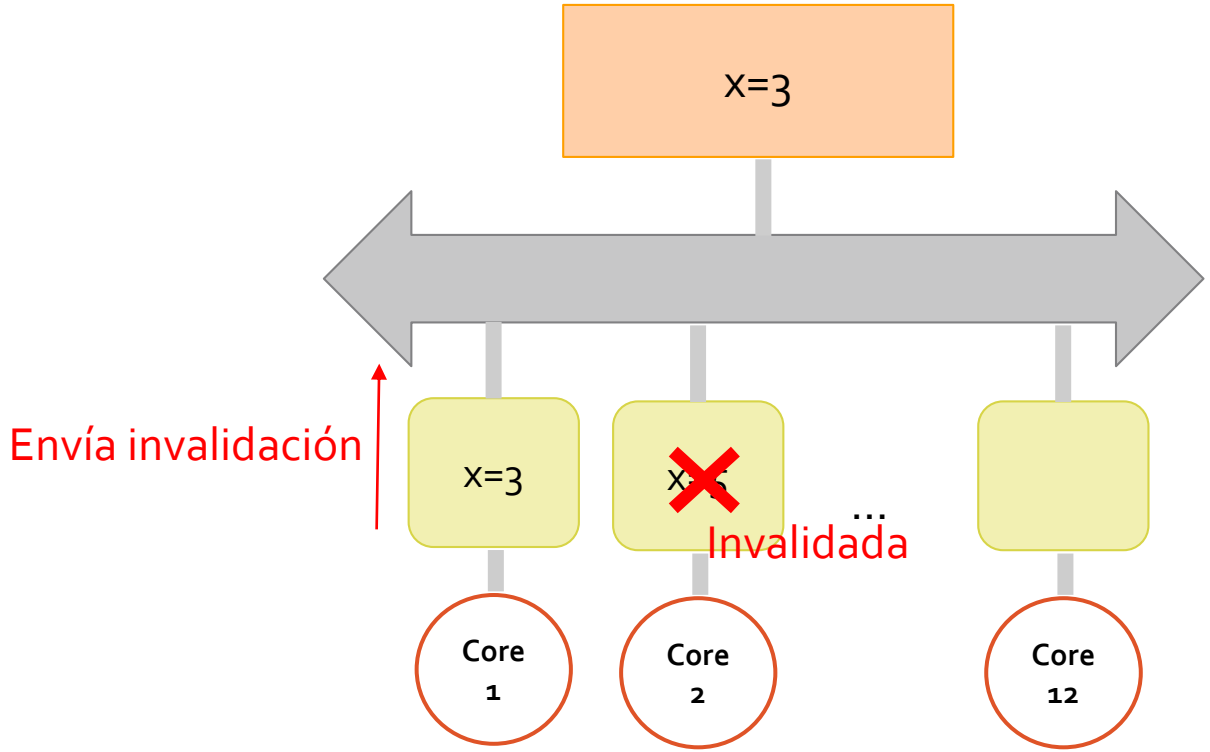
Coherencia de caché

- Posibles soluciones:
  - Utilizar protocolos de coherencia: invalidación, snooping, actualización



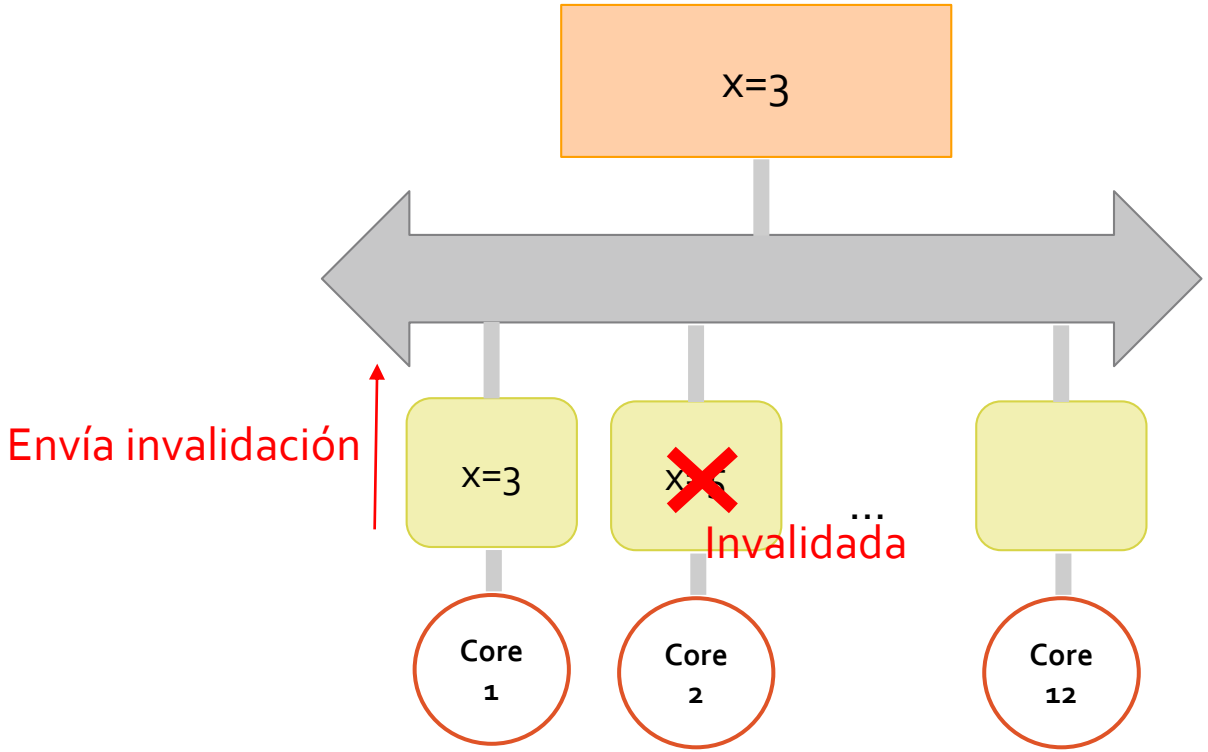
Coherencia de caché

- Protocolo de invalidación: Si un core escribe en un elemento, todas las otras copias en otras cachés de mismo elemento quedan invalidadas



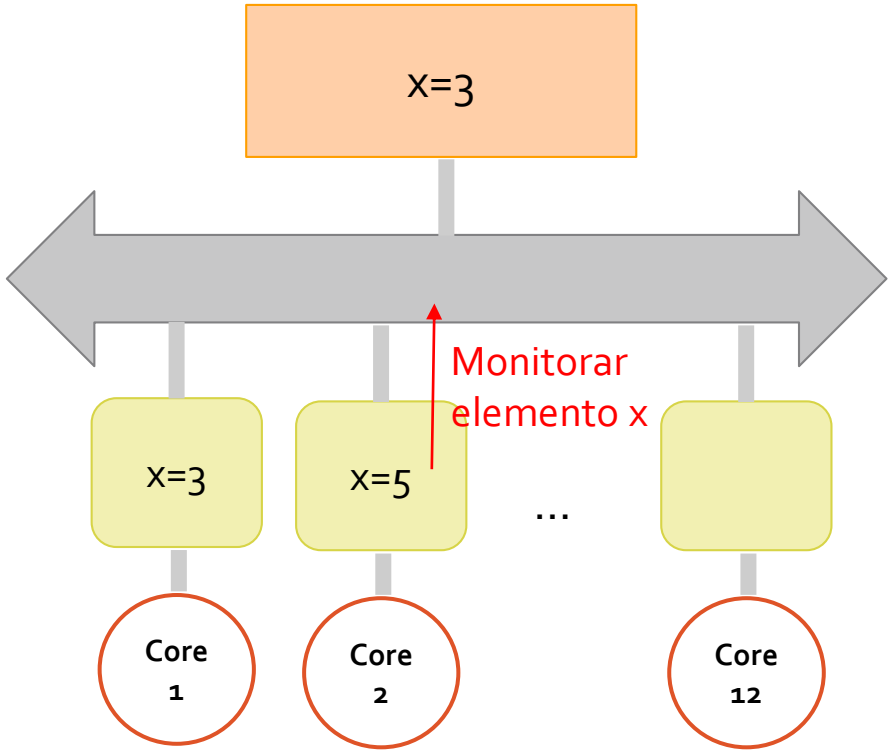
Coherencia de caché

- Protocolo de invalidación: Si un core escribe en un elemento, todas las otras copias en otras cachés de mismo elemento quedan invalidadas.
- Copias invalidadas deben leer valor actualizado desde memoria principal



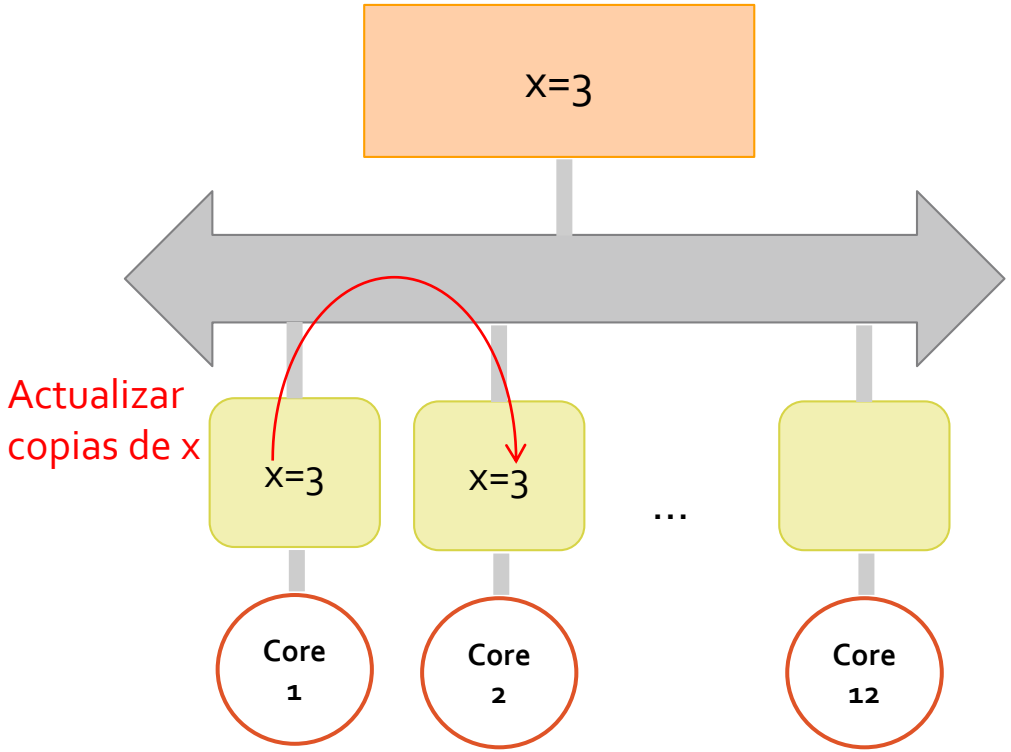
Coherencia de caché

- Protocolo de snooping: Todos los cores monitorean continuamente el bus de comunicación. Si detectan que algún valor ha cambiado lo actualizan.

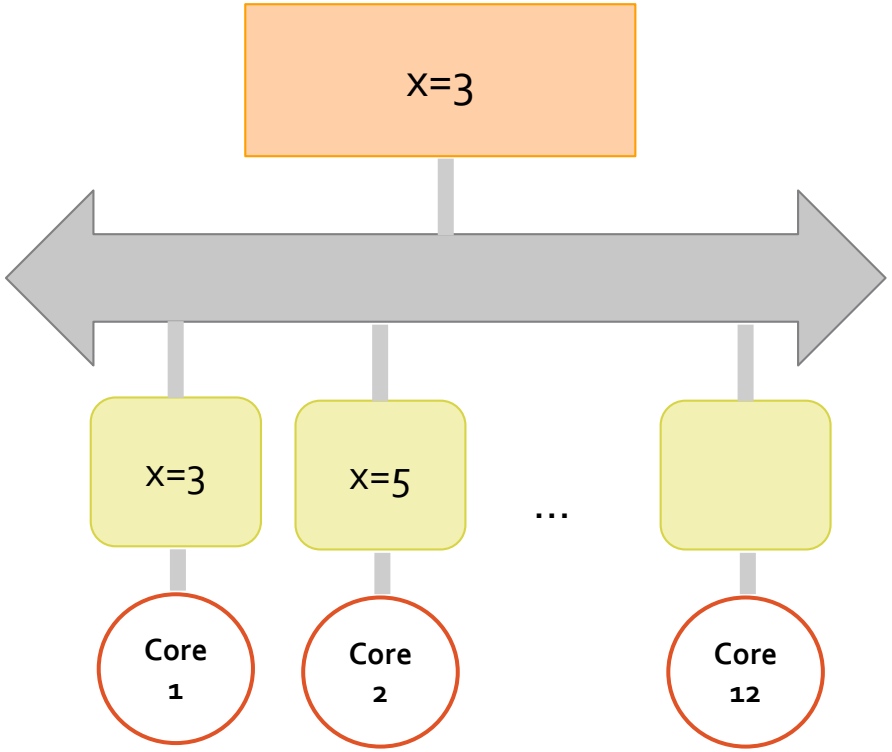


Coherencia de caché

- Protocolo de actualización: Cuando un core actualiza un valor en su caché, debe actualizarlo en todas las otras cachés de cores que existan copias.



Coherencia de caché



- Cuál es la mejor solución?
  - invalidación, snooping, actualización?
- Escrituras múltiples:
  - Invalidación: sólo la primera vez
  - Actualización: debe transmitir nuevas escrituras cada vez
  - Snooping: complejo de implementar
- En general invalidación tiene mejor rendimiento, ya que genera menor tráfico
- Más información sobre protocolos: mirar MSI, MESI (Modified, Exclusive, Shared, Invalid)