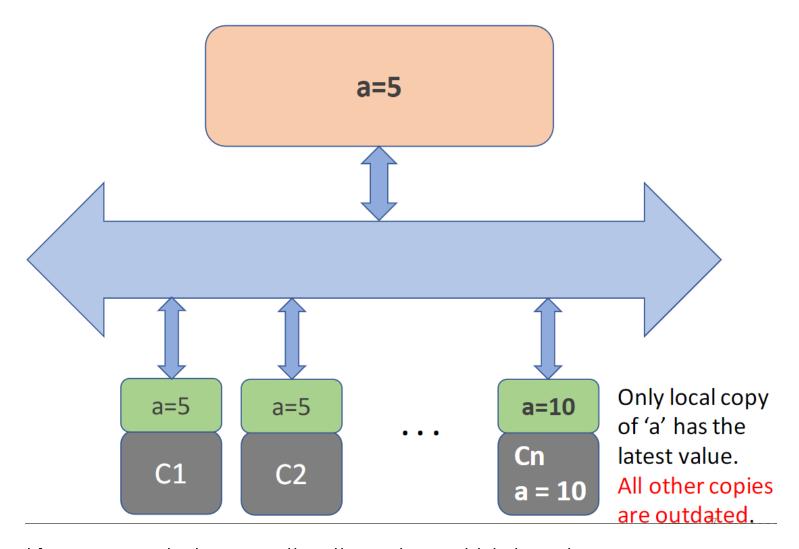
# Multicore architecture Multicore systems Multi core processors

- •Each core has their own L1 cache and sometimes even L2
- •All cores share L3 cache and share main memory

### Coherence of data

Coherence is the quality of being *logical* and *consistent* and as such the data must be consistent too

on a multicore system, a variable may have local versions in each core and causes coherence problems as they are updated only in each core.



if any core wants to load a then they get an outdated version

## Coherence mechanisms

To achieve coherence, any write to a mem loc must cause all copies of the loc to be removed from their caches. So in the **example** above, as soon as a=10 occurs, then invalidate or delete a in C1 and C2

#### Coherence Protocols

Coherence protocols apply cache coherence in multicore sys

Fin – no two cores see different value for shared data

To do this, additional information such as *is data stored in multiple caches* and has cache line been modified is stored and passed.

## M5I protocol

In this protocol, each cache line is labelled with a state

M : cache block modified
5 : cache block shared
I : cache block invalid

s: a=5
s: b=13
s: c=5

C1
a=10

d=2 s: b=13 s: c=5 I: a=5 S: b=13 S: c=5

In this example, when C1 does a = 10, the cache line holding a in Cn is labelled as I

M: a=10 S: b=13 S: c=5

d=2 s: b=13 s: c=5 I: a=5 S: b=13 S: c=5

and afterwards, the cache line in C1 is labelled as M and hardware updates this in shared memory. If Cn wants to load a or use it, then it loads it from shared memory

## Programming multicore platforms

Two options

- •Program directly targets processor cores
  - Programmer does synchronization
  - •Difficult and error-prone

•Uses a concurrency platform

- •Abstracts cores, handles synchronization and communication protocols and performs load balancing
- •Offers easier multicore programming environment

Examples of *concurrency platforms* are:

- •PThreads and WINAPI threads
- •OpenMP