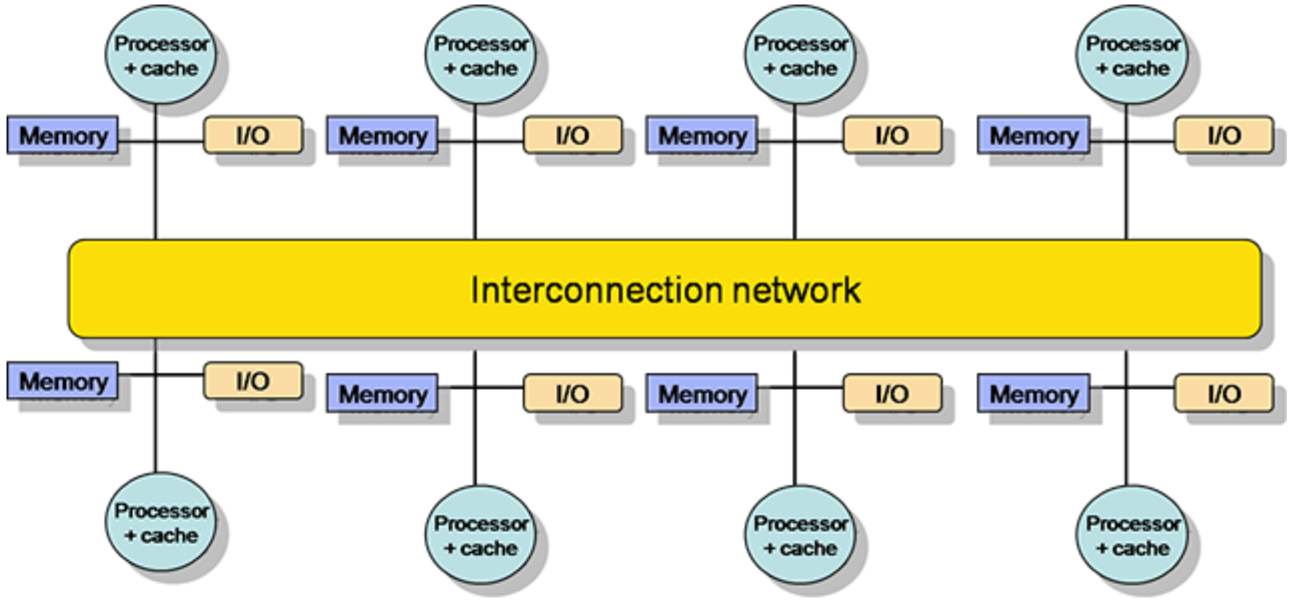
NORMA Systems

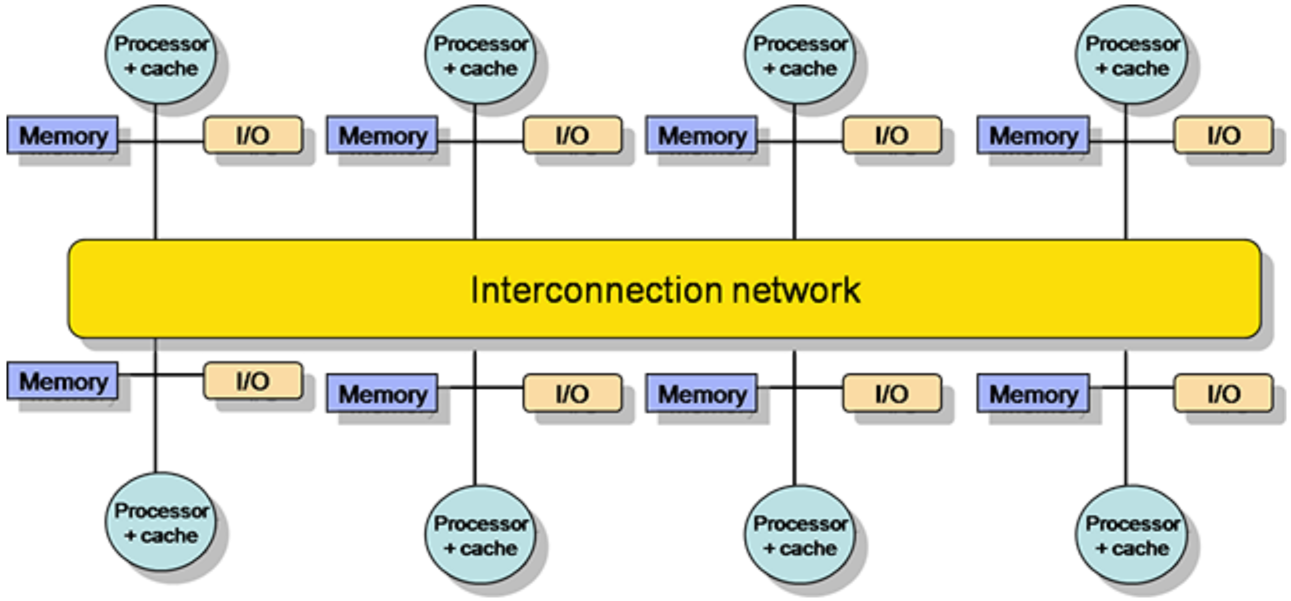
NORMA (no remote memory access) systems



Each processor has a separate private memory & address space

NORMA Systems

Processors only have direct access to their local memory



Message exchange is used to obtain data from remote modules

Clusters fit this loosely coupled model

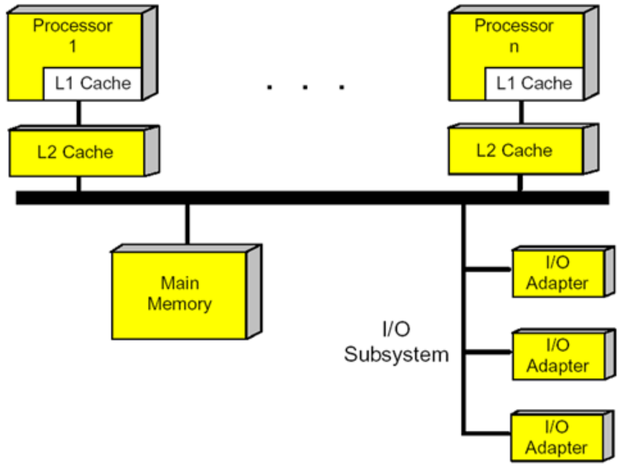
- These are easier to scale than the SMP tightly coupled counterpart

JOHNS HOPKINS UNIVERSITY | Engineering for Professionals

UMA on SMPs

Using a single global memory unit

- Uniform memory access (UMA) for all processors



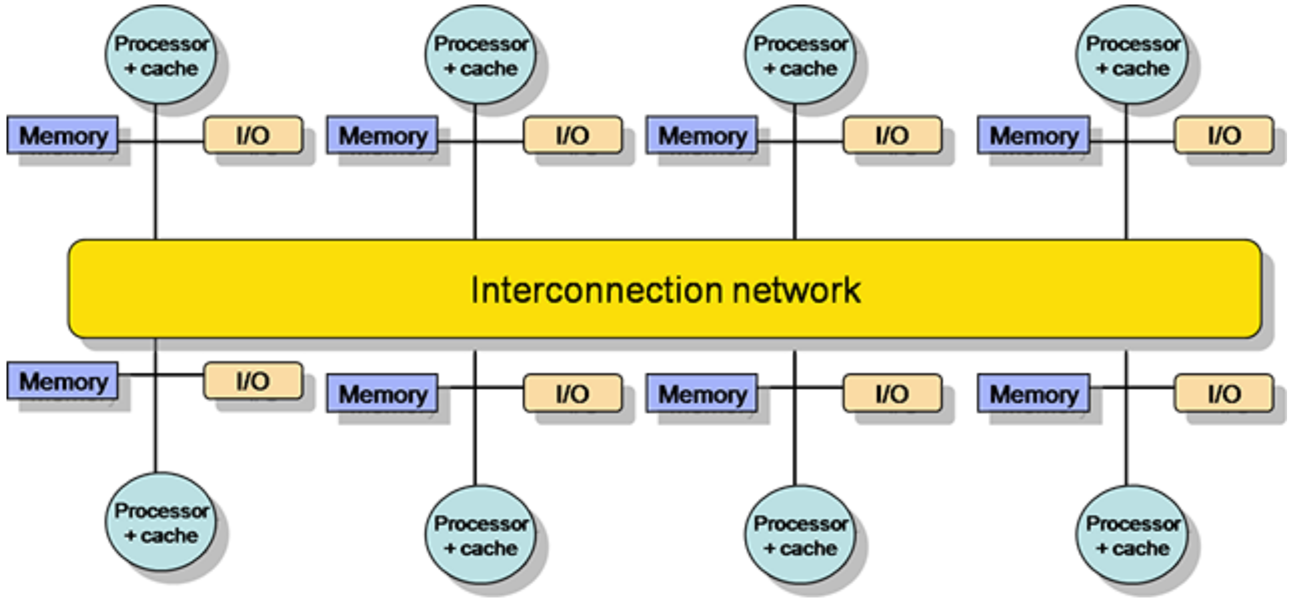
Only one processor at a time can access memory

- Competition for the CPU-to-memory bus will be high
- Cache can be used to reduce the conflicts
- This would be too limiting

NUMA Systems

Distributed memory modules work better

Non-uniform memory access (NUMA) system



Local module access is faster than to a remote module

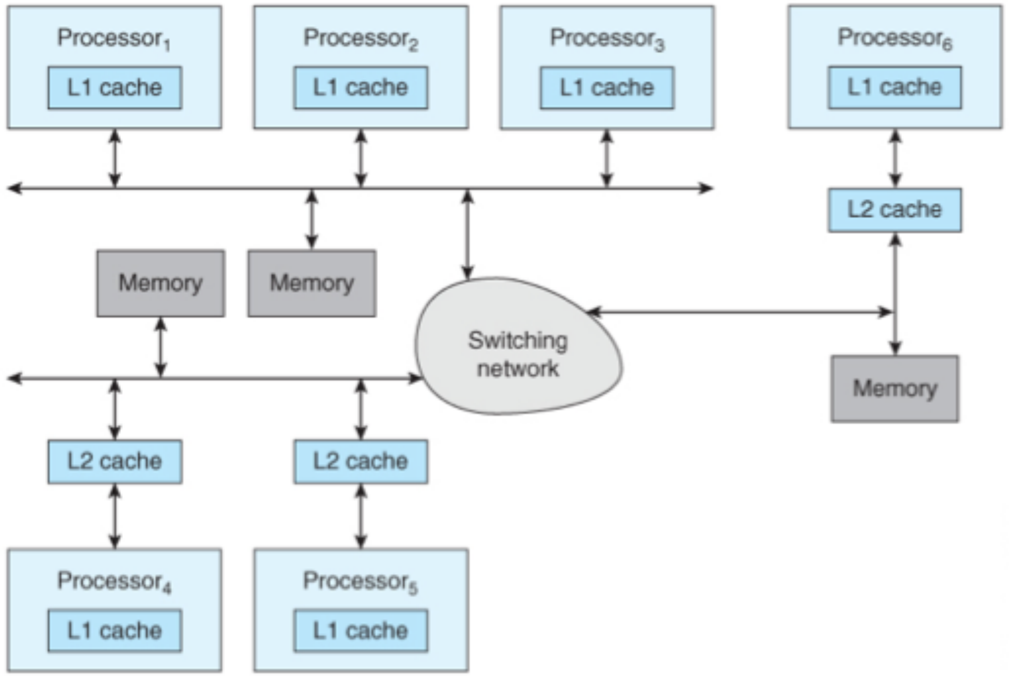
CPU addresses map to a particular memory module

Single shared address space spans all modules

Each processor has a cache system

NUMA Systems

Another design for a NUMA system is shown below:



Memory is accessed less often when cache is used

But shared data must be kept consistent (i.e., coherent)

Coherency Requirements

Correct program order must be preserved

- Each processor must read the most recent version of data

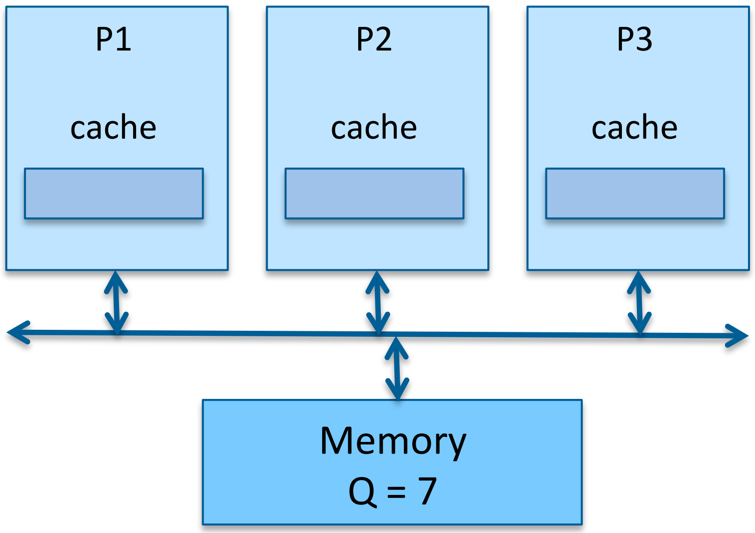
All writes must be visible to all processors

- If P1 writes x and then P2 reads x, P2 must obtain value written by P1

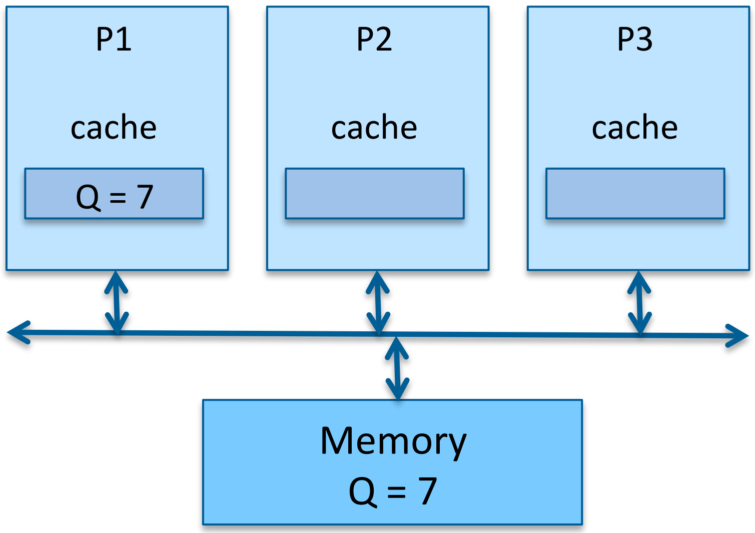
Order must be preserved

- Example: P1 sets x to 1, P2 reads x and if x==1, sets it to 2
- If P3 then reads x, it should see 2 and not the old value 1

Coherency Example

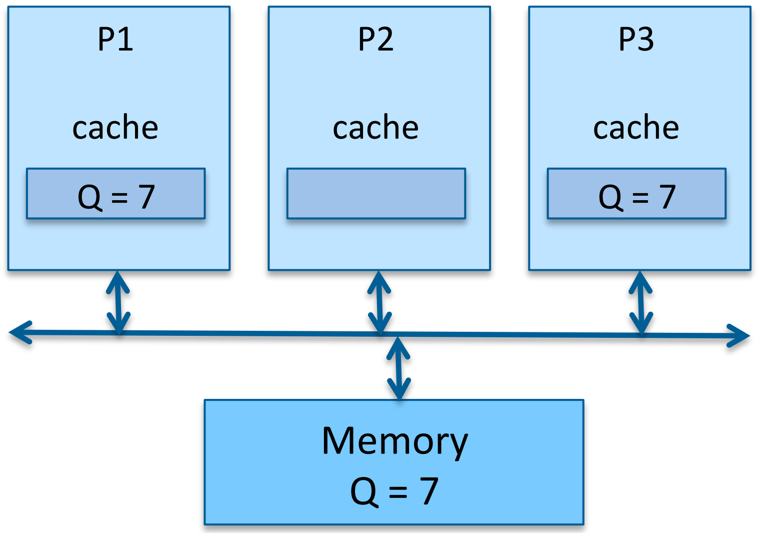


a) Initially Q = 7 in memory

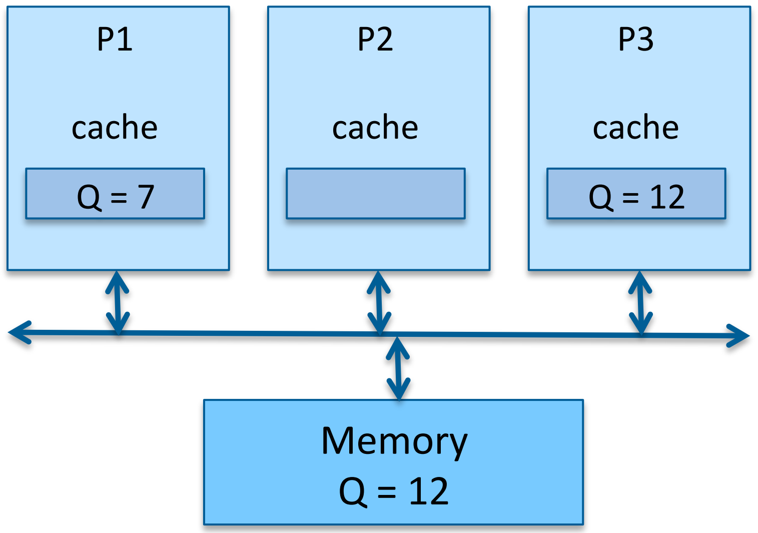


b) P1 reads Q and caches it locally

Coherency Example

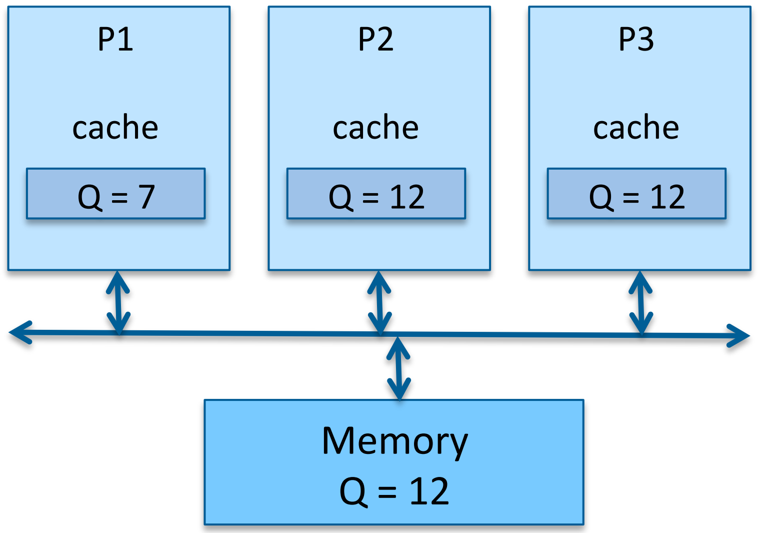


c) P3 reads Q from memory and caches it.

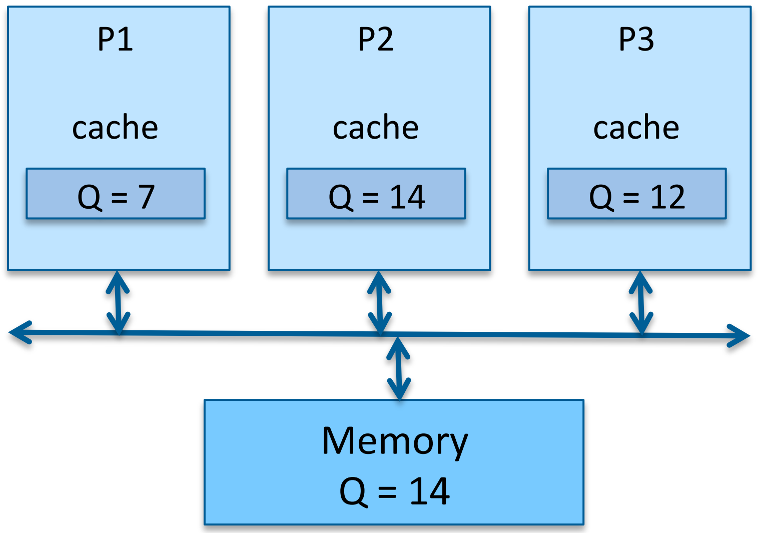


d) P3 writes Q=12 to cache and to memory

Coherency Example



e) P2 reads Q from memory and caches it



f) P2 computes Q + 2, caches it locally, and writes it back to memory

There are now three different cached values of Q
This happened because cache coherency was not maintained.