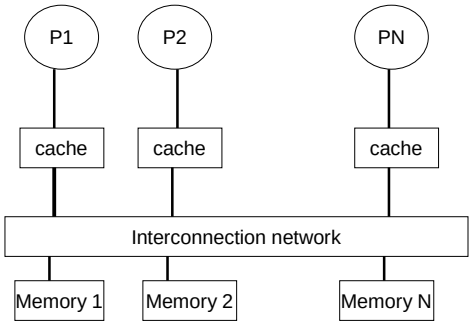
Multiprocessor Architectures: Shared memory and Message Passing

Shared Memory Architectures: (multiprocessors)
•Single address space shared by all processors
•Processors communicate through shared variables
•Common memory
•SMP and NUMA

Shared Memory Architectures



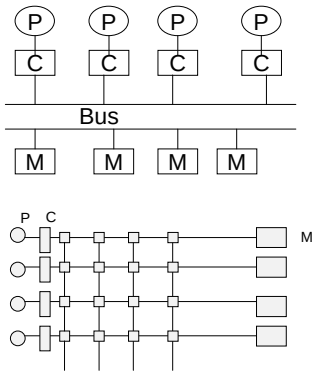
Shared Memory Multiprocessor Models

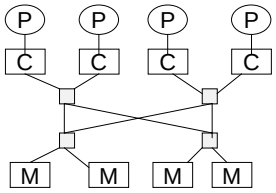
- 3 shared memory multiprocessor models:
  - UMA (Uniform Memory-Access)
  - NUMA (Non-Uniform Memory-Access)
  - COMA (Cache-Only Memory Architecture)

These models differ in how the memory and peripheral resources are shared and distributed

UMA- SMP(SymmetricMultiprocessors)

- Existed since 1960s
- Very popular now
- SMPs may have 2 or 64 processors
- Interconnection is a bus, crossbar or network





Network based: Sparse network

SMP Characteristics

- Have a single address space
- Processors communicate via reading.writing into shared memory.
- Relation between processors and memory is *symmetric*
  - Access time to the memory cell is the same for all processors
  - UMA
- Stores a single copy of each program,single copy of OS
- Looks like a single system image
- To a user, looks like a sequential machine with multiple processes running in parallel

Disadvantage

- Do not scale well: why?
  - Bus-based : bottleneck
  - Crossbar: No contention if they go to different memory modules. Hardware costs. P processors and P memory modules, P^2 switches
  - Network based:Cheaper to build; but maybe memory contention due to multiple switches to route. They are scalable but memory access time increases with number of processors.

Caches

- Caches are used to reduce access time to memory.
- As in uniprocessors, caches are used to hold frequently accessed data during a period of execution. Data is replicated from shared memory to cache whey accessed and overwritten when not accessed for a while.
- Data is transferred in cache lines.
- Problem: coherence! Large research area.

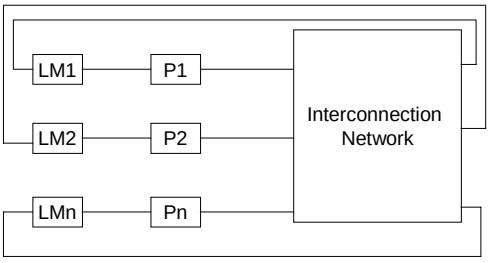
UMA

- UMA is suitable for general-purpose and time-sharing applications by multiple users
- Can be used to speedup the execution of a single large program in time-critical applications

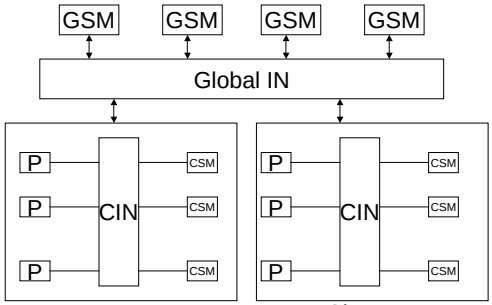
NUMA

- Each memory is closer to some processors than others
- Each processor has local memory accessed faster than remote memory.
- Shared memory physically distributed to all processors called *local memories.*
- Collection of all local memories forms a global address space accessible by all processors
- "Distributed Shared Memory"
- Interconnection is a grid or hypercube
- Locality is important (harder to program)
  - Faster to access local memory than remote memory through IN
- Scales to more processors

BBN Butterfly



Cluster Model (Cedar System at Univ. of Illinois)



CC-NUMA

- In NUMA, access time between local and remote memory grows with number of processors.
  - Size of interconnection grows
  - CC induces more overhead
    - Cache coherence protocols more complicated for CC-NUMA architectures

CC-NUMA

- When a processor reads data from remote memory, data is placed in processor's cache for fast access.
- To replace it back, depends on OS.
- Maybe written back to remote memory where they came from or may migrate data to local memory assuming it will be used again.
- Or it may even replicate data for concurrent accesses.

COMA

- Special case of NUMA model
- Distributed main memories are converted to caches
- All caches form a global address space
- Remote cache access assisted by distributed cache directories (D)
- Eg: KSR-1 (Kendall Square Research)
- CC-NUMA (distributed shared memory and cache directories)- Stanford DASH, MIT Alewife