Parallel Computer Memory Architecture
CS 540- High Performance Computing

Spring 2017

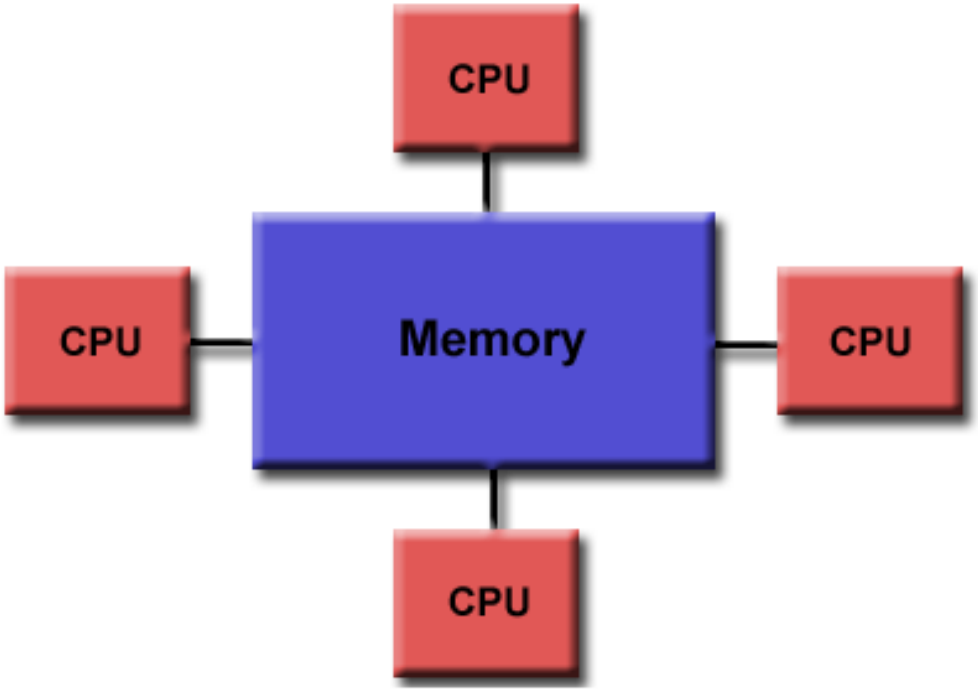
Shared Memory

- all processors/cores have access to the entire global address space.
- Multiple processors can operate independently of each other, however share the same memory resources.
- Changes in a memory location effected by one processor are visible to all other processors.
- Shared memory machines have traditionally been classified as **UMA** and **NUMA**.

Shared Memory, UMA

- **U**niform **M**emory **A**ccess
- Identical processors
- Equal access and access times to memory
- Also knows as, CC-UMA - Cache Coherent UMA; if one processor updates a location in shared memory, all the other processors know about the update.
- Cache coherency is accomplished at the hardware level.

Shared Memory, UMA

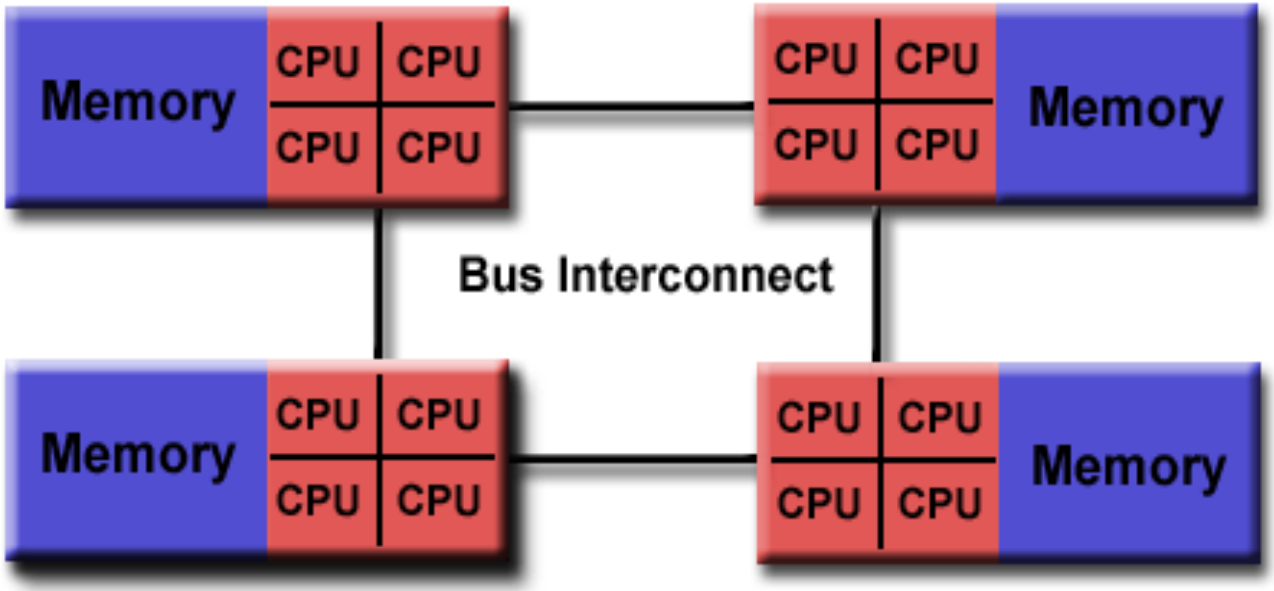


Source: Lawrence Livermore National Laboratory

Shared Memory, NUMA

- **N**on-**U**niform **M**emory **A**ccess
- Made by physically linking two or more SMPs (Symmetric Multiprocessor)
- One SMP can directly access memory of another SMP
- Not all processors have equal access time to all memories
- Memory access across link is reduced
- If cache coherency is maintained, then may also be called CC-NUMA - Cache Coherent NUMA

Shared Memory, NUMA



Source: Lawrence Livermore National Laboratory

Pros

- Pros
- Global address space provides a “user-friendly” programming standpoint to memory
- Data sharing between tasks is both fast and uniform due to the proximity of memory to CPUs

Cons

- Cons
- The lack of scalability between memory and CPUs. Adding more CPUs increases traffic on the shared memory-CPU bus
- It’s the responsibility of the developer to ensure synchronization constructs ensure proper access of global memory.

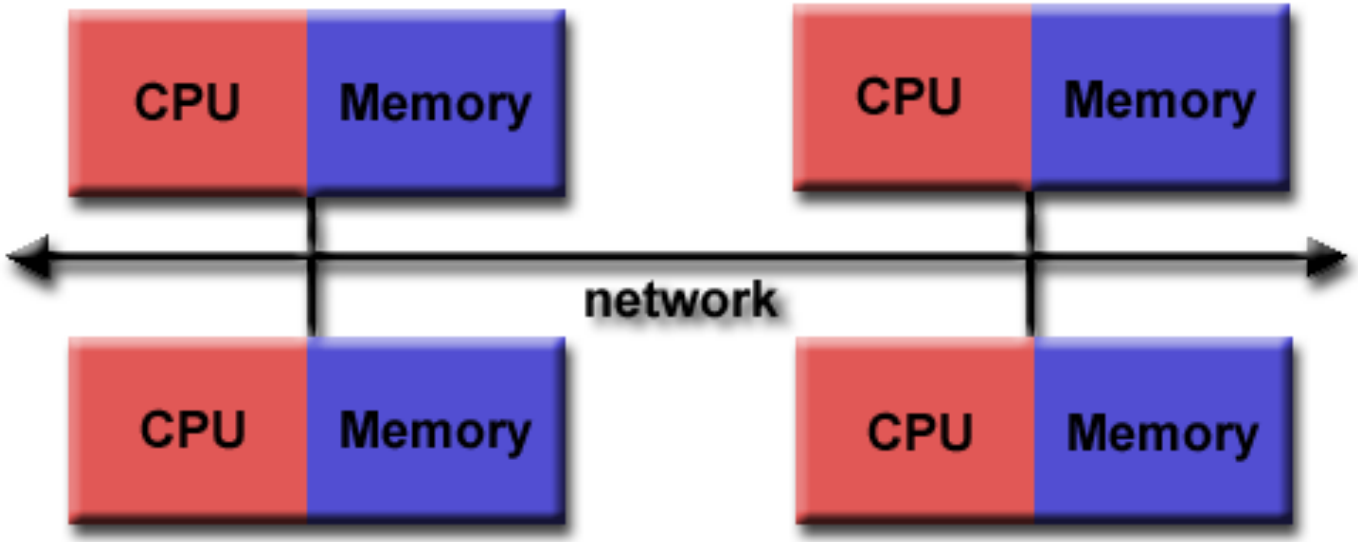
Distributed Memory

- Distributed memory systems require a communication network to connect inter-processor memory.
- Processors have their own local memory and memory addresses in one processor are not shared with another processor, so the concept of a global address space across all processing units is non-existent

Distributed Memory

- Since each processor has its own local memory, it operates independently.
- When a processor needs access to data in another processor/node, it is the responsibility of the developer to explicitly define how and when data is communicated.
- The network media used for data transfer varies widely, though it can be as simple as Ethernet.

Distributed Memory



Source: Lawrence Livermore National Laboratory

Pros

- Memory is scalable with the number of processors.
- Each processor can rapidly access its own memory without interference and without the overhead of global cache coherency.
- Very cost effectiveness: can use commodity, off-the-shelf hardware (processors & network)

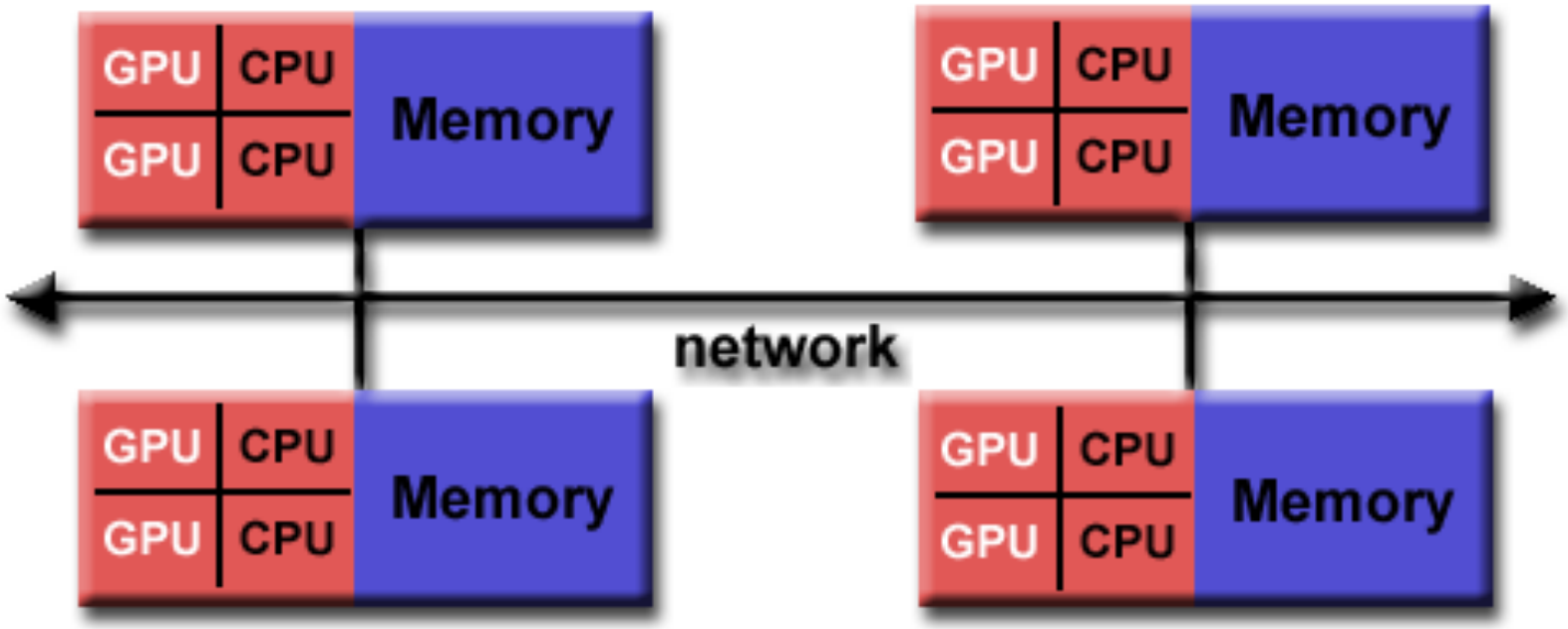
Cons

- The developer is responsible for various underlying details associated with data communication between processors/nodes.
- It may be difficult to map existing data structures, based on global memory, to this memory organization.
- Non-uniform memory access times - data existing on a remote node takes longer to access than node local data.

Hybrid Distributed -Shared Memory

- Some of the largest and fastest computers in the world today use both shared and distributed memory architecture
- The shared memory component can be a shared memory machine and/or graphics processing units (GPU).
- The distributed memory component is the networking of multiple shared memory/GPU machines

Hybrid Distributed -Shared Memory



Source: Lawrence Livermore National Laboratory

Pros/Cons

- Increased scalability is an important advantage
- Increased programmer complexity is an important disadvantage

References

1. Blaise Barney, Lawrence Livermore National Laboratory, https://computing.llnl.gov/tutorials/parallel_comp/#Overview
2. https://en.wikipedia.org/wiki/Flynn's_taxonomy
3. https://www.citutor.org/index.php, Parallel Computing Explained