Protocols – Dash Protocols, NUMA Multiprocessors, NUMA Algorithms

Introduction to DASH Protocols

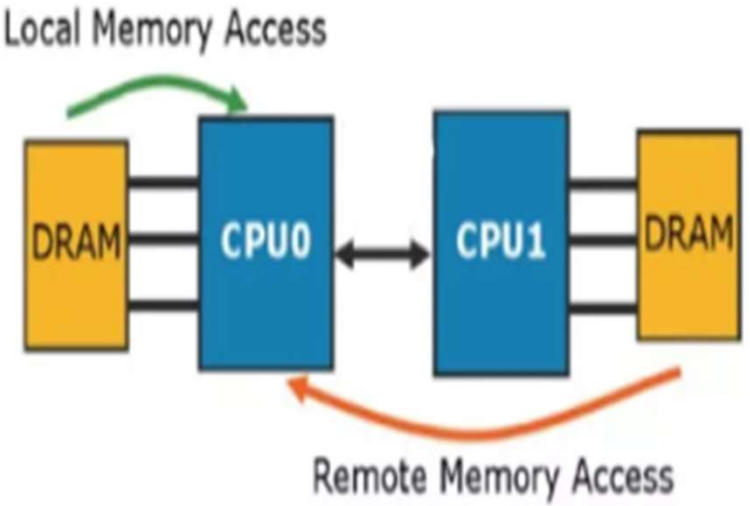
Definition: DASH, or Directory Architecture for Shared Memory, is a protocol used in distributed shared memory systems. It coordinates memory accesses across multiple processors by using a directory-based approach.

Purpose: DASH protocols aim to efficiently manage and maintain coherence across distributed memory in multiprocessor systems, allowing processors to share data as if they were accessing a single, unified memory.

Context in DSM: In DSM, memory is physically distributed among processors, but DASH protocols allow processes to access this memory seamlessly while keeping data consistent across nodes.

NUMA Multiprocessors

- NUMA stands for Non-Uniform Memory Access.
- In NUMA multiprocessors, memory is divided across different nodes, each with its own memory that it accesses faster than the memory of other nodes.



How NUMA Works

- Local Memory: Each processor has direct, faster access to its 'local' memory.
- Remote Memory: Processors can still access memory in other nodes (remote memory), but it’s slower than accessing local memory.

Why NUMA?

- As the number of processors increases, it's harder for all of them to share a single memory efficiently (like in Uniform Memory Access, or UMA).

- NUMA reduces memory access bottlenecks by keeping data closer to the processor that needs it most often. .

Properties of NUMA

- Non-Uniform Memory Access Time:
- Memory access time depends on whether a processor is accessing its local memory or remote memory. Access to local memory is faster, while access to remote memory is slower.
- Scalability:
- NUMA systems can scale more easily as they add more processors and memory because each processor doesn’t rely on a single memory bus.

Properties of NUMA

- Memory Affinity:
- Programs are often designed to keep frequently used data close to the processor that uses it (memory affinity). This helps to minimize remote memory accesses, making the system more efficient.

NUMA Algorithms

- First-Touch Allocation:
- Data is initially placed in the local memory of the first processor that accesses it, reducing remote access if that processor uses it frequently.
- Page Migration:
- Frequently accessed data by a particular processor is moved to its local memory, minimizing remote access for that processor.

NUMA Algorithms

- Dynamic Memory Mapping:
- Continuously adjusts the placement of memory pages based on recent access patterns, optimizing local access and adapting to changing workloads.
- Replication:
- Copies of commonly accessed data are placed in the local memory of each processor needing it, reducing remote access at the cost of additional memory usage.