ReTransformer

ReRAM-based Processing-in-Memory Architecture for Transformer Acceleration

Reza Adinepour

Amirkabir University of Technology (Tehran Polytechnic)

adinepour@aut.ac.ir

Computer Engineering Department

June 27, 2024

Presentation Overview

1. Memory Architecture Overview
   - Uniform Memory Access (UMA)
   - Non-uniform Memory Access (NUMA)
   - Cache-only Memory Access (COMA)
2. Code Snippets
3. Differences between UMA and NUMA

Memory Architecture Overview

What is this structure?

— Defines how computer memory is organized and accessed.

1. Uniform Memory Access (UMA)
2. Non-Uniform Memory A ccess (NUMA)
3. Cache-Only Memory Access (COMA)

— In this presentation we talk about UMA and NUMA Architecture

Uniform Memory Access (UMA)

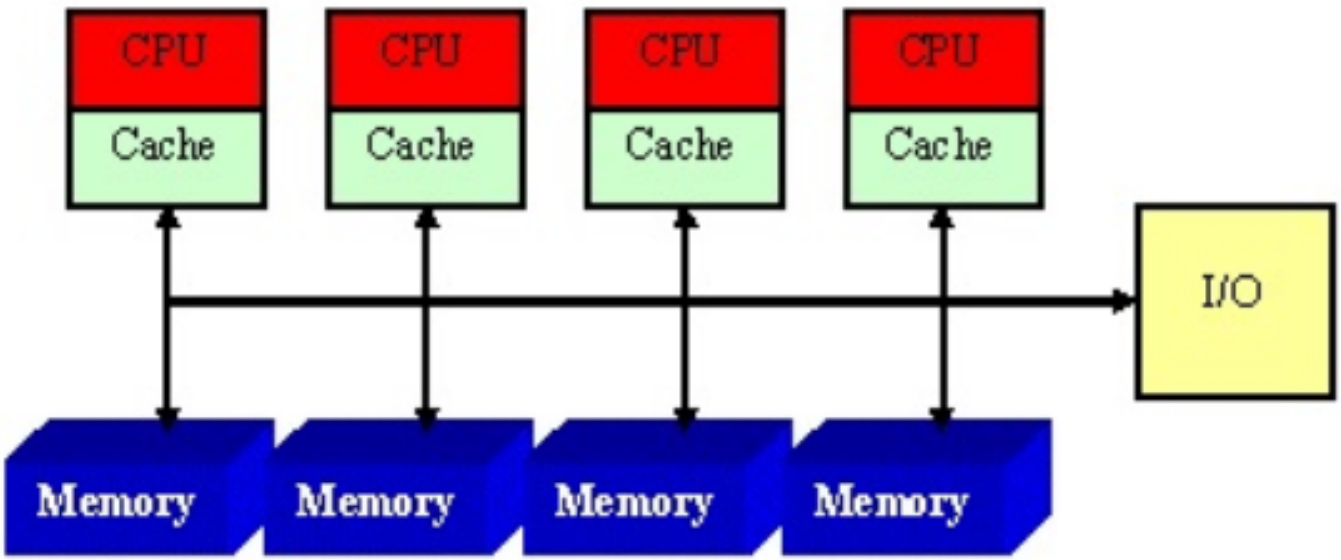

Figure: Processors with equal access to memory

1. Same latency for all processors to access memory.
2. Hardware cache typically present with each processor.

Uniform Memory Access (UMA) (Cont.)

1. Equal memory access for all processors.
2. Shared memory, any processor can access any part at any time.
3. Simple, cost-effective, highly scalable.

Advantages:

1. **Ease of Implementation:** Minimal hardware modifications, cost-effective.
2. **Scalability:** Easily scales with more processors without impacting access times.

Disadvantages:

1. **Memory Contention:** Increased processors may lead to slower access times.
2. **Limited Bandwidth:** Shared memory bus can become a bottleneck.

Uniform Memory Access (UMA) (Cont.)

Example System:

1. **Symmetric Multiprocessing (SMP) System:**
   1. Multiple processors share common memory.
   2. Controlled by a single operating system.
   3. Common in servers and high-performance computing.

Summary:

1. **Strengths: Simplicity and scalability.**
2. **Weaknesses: Potential for memory contention and limited bandwidth in larger systems.**

Non-uniform Memory Access (NUMA)

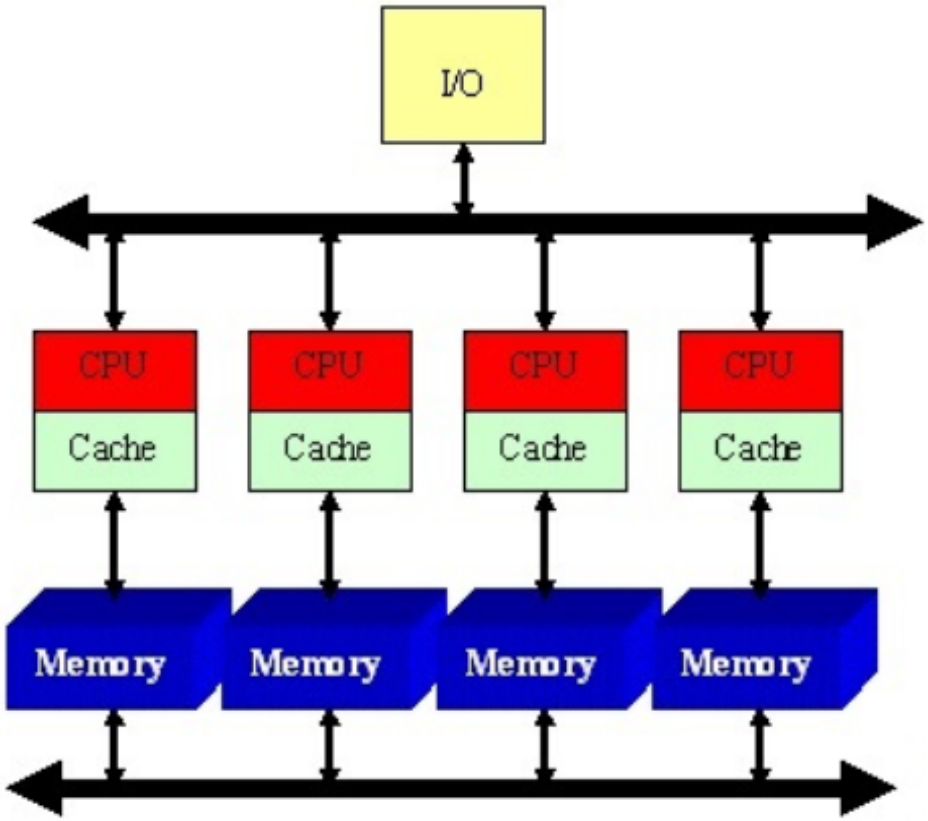

Figure: Processors with equal access to memory

1. Each processor has its local memory.

Non-uniform Memory Access (NUMA) (Cont.)

1. Memory divided into multiple banks, each processor has its local bank.
2. Processors can access other banks, but at higher latency than local access.
3. Efficient memory resource use, potential better performance than UMA for certain workloads.

Advantages:

1. **Reduced Memory Contention:** Each processor has its local memory, minimizing contention.
2. **Increased Memory Bandwidth:** Local memory banks lead to higher bandwidth than UMA.
3. **Efficient Memory Use:** Allocation based on processor needs enhances resource utilization.

Non-uniform Memory Access (NUMA) (Cont.)

Advantages:

1. **Higher Implementation Complexity:** Additional hardware and software complexity.
2. **Higher Latency for Remote Access:** Accessing remote memory incurs higher latency.

Example System:

1. **Multi-Socket Server:**
   1. Each socket has its processors and memory banks.
   2. Sockets connected via a high-speed interconnect.
   3. Commonly used in data centers, offers better performance for specific workloads.

Code Snippets

Run the code!

— This code is generated by ChatGP

Differences between UMA and NUMA

1. **Memory access time:**
   1. NUMA:
      Memory access time varies depending on the location of the data in memory. Accessing data in the local memory of a processor is faster than accessing data in the memory of a remote processor.
   2. UMA:
      Memory access time is uniform across all processors since they share the same memory pool.
2. **Scalability:**
   1. NUMA architecture is highly scalable and can support a large number of processors.
   2. UMA architecture is not as scalable as NUMA and may face performance issues when used with a large number of processors.

The End

Questions? Comments?

You can find this slides here:
github.com/M-Sc-AUT/M.Sc-Computer-Architecture/Memory Technologies