

SSC0611

Arquitetura de Computadores

20ª Aula – Arquiteturas Paralelas

Arquitetura MIMD com Memória Compartilhada

Profa. Sarita Mazzini Bruschi

sarita@icmc.usp.br

Arquiteturas MIMD

- As arquiteturas MIMD dividem-se em dois grandes grupos:
 - Memória Compartilhada
 - Cada processador pode endereçar toda a memória do sistema
 - Memória Distribuída
 - Cada processador endereça somente a própria memória local

Arquiteturas MIMD com memória compartilhada

- Vantagens
 - A comunicação entre os processos é bastante eficiente, pois os dados não precisam se movimentar fisicamente
 - A programação não difere muita da programação para um único processador, não necessitando particionar código nem dados

Arquiteturas MIMD com memória compartilhada

- Desvantagens
 - Primitivas de sincronização são necessárias para acesso às regiões compartilhadas
 - Em algumas linguagens isso fica a cargo do programador
 - Linguagens mais novas escondem alguns detalhes do programador
 - Esse tipo de arquitetura não é muito escalável, devido ao limite da memória

Arquiteturas MIMD com memória distribuída

- Vantanges
 - Altamente escalável, permitindo a construção de MPPs (computadores massivamente paralelos)
 - A forma de comunicação (através de troca de mensagens) resolve tanto os problemas de comunicação como sincronização

Arquiteturas MIMD com memória distribuída

- Desvantagens
 - A programação exige que os problemas possuam uma natureza paralela
 - É necessária a distribuição da carga entre os processadores, seja de maneira automática ou manual

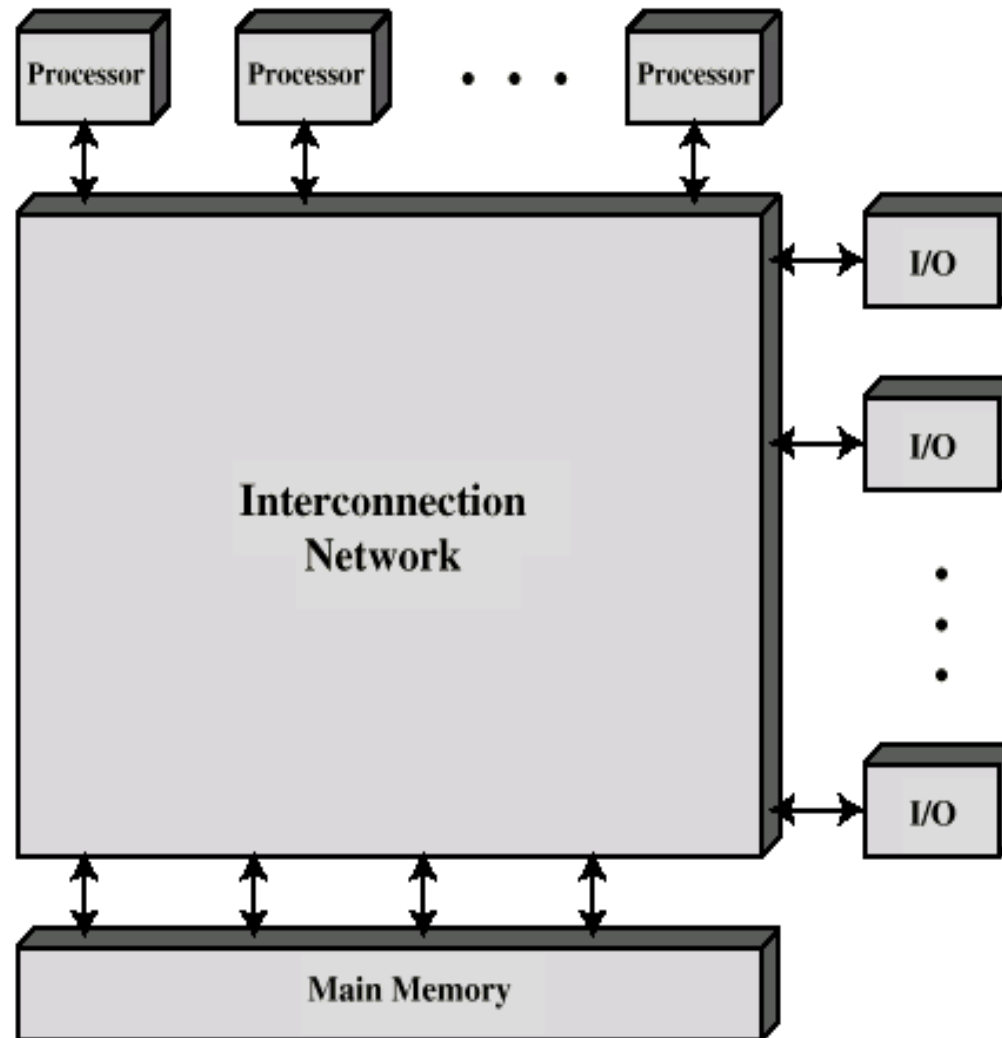
SMP – *Symmetric MultiProcessors*

- Computador com as seguintes características:
 - dois ou mais processadores com capacidade semelhante
 - processadores compartilham a mesma memória e I/O
 - processadores conectados por um barramento ou outra conexão interna
 - tempo de acesso à memória é aproximadamente o mesmo para todos os processadores
 - Também denominado por arquitetura **UMA (*Uniform Memory Access*)**
 - processadores compartilham acesso a I/O
 - podem usar o mesmo canal ou possuírem caminhos independentes para cada dispositivo
 - processadores podem fazer as mesmas funções (são simétricos)
 - sistema operacional integrado controla a arquitetura
 - fornece interação entre processadores, *jobs*, tarefas, *threads*, arquivos e níveis de elementos de dados

SMP - Vantagens

- Desempenho, caso algum trabalho possa ser feito em paralelo
- Disponibilidade de recursos
 - todos os processadores podem fazer as mesmas funções, falhas de um processador não param a máquina
- Aumento incremental
 - usuário pode aumentar o desempenho adicionando novos processadores, mas sempre limitado pela memória
- Diferentes faixas de equipamentos (*scaling*)
 - fornecedores oferecem diferentes produtos, baseados no número de processadores

Diagrama de um bloco multiprocessador com memória compartilhada



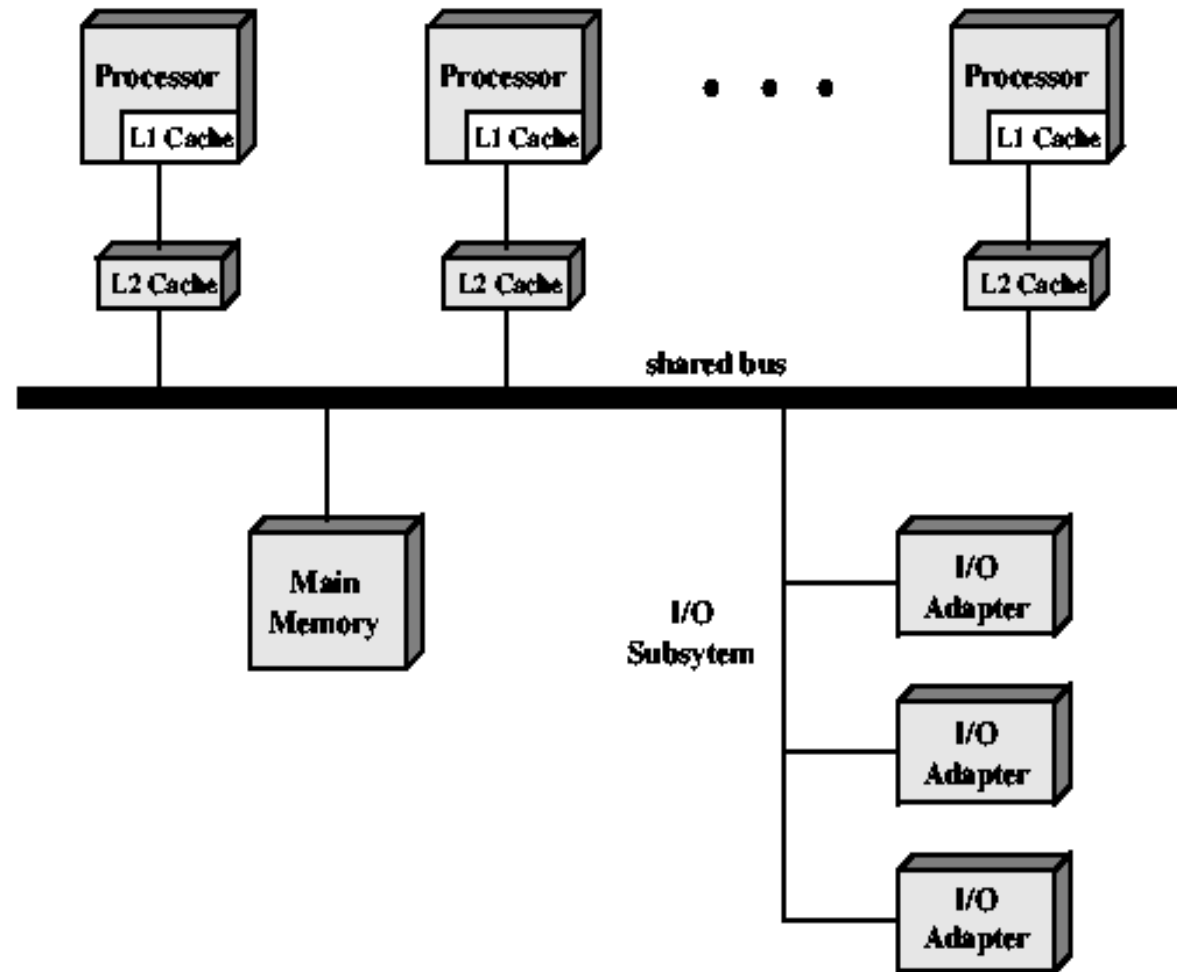
SMP - Organização

- Barramento de tempo compartilhado ou comum
- Memória multiportas (ou multiportos)

SMP – Barramento Compartilhado

- Forma mais simples
- Estrutura e interface similares às arquiteturas monoprocessadas
 - endereçamento – distingue módulos no barramento
 - arbitragem – qualquer módulo pode ser o mestre temporariamente
 - *time sharing* – se um módulo tem o barramento, outros dispositivos esperam
- Diferença em relação aos monoprocessados:
 - múltiplos processadores além de múltiplos dispositivos de I/O

SMP – Barramento Compartilhado

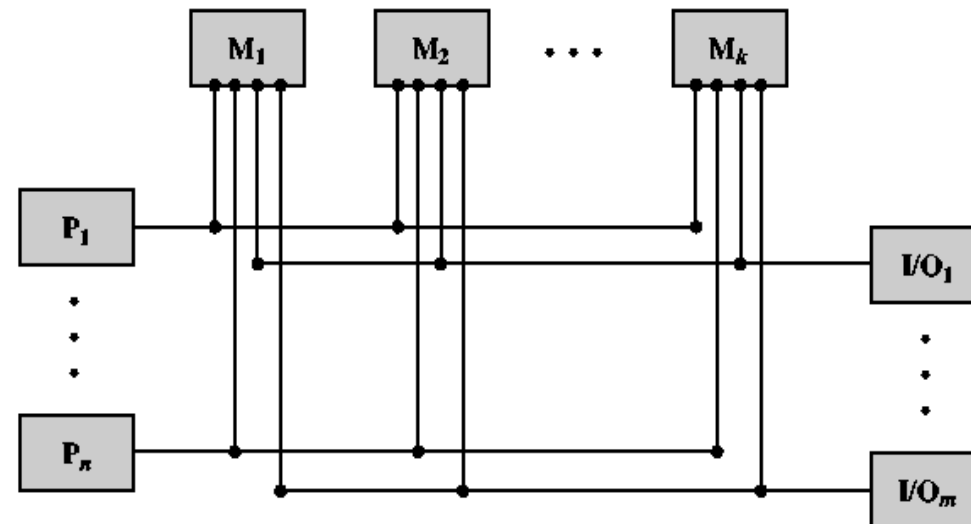


SMP – Barramento Compartilhado

- Vantagens
 - simplicidade
 - flexibilidade
 - confiabilidade
- Desvantagens:
 - desempenho limitado pelo *clock* do barramento
 - cada processador deveria ter *cache* local
 - reduz número de acessos ao barramento
 - gera problemas com a coerência da *cache*

SMP – Memória Multiportas

- Acesso direto e independente dos módulos de memória pelos processadores
- Lógica necessária para solucionar conflitos
- Pouca ou nenhuma modificação nos processadores ou módulos de memória



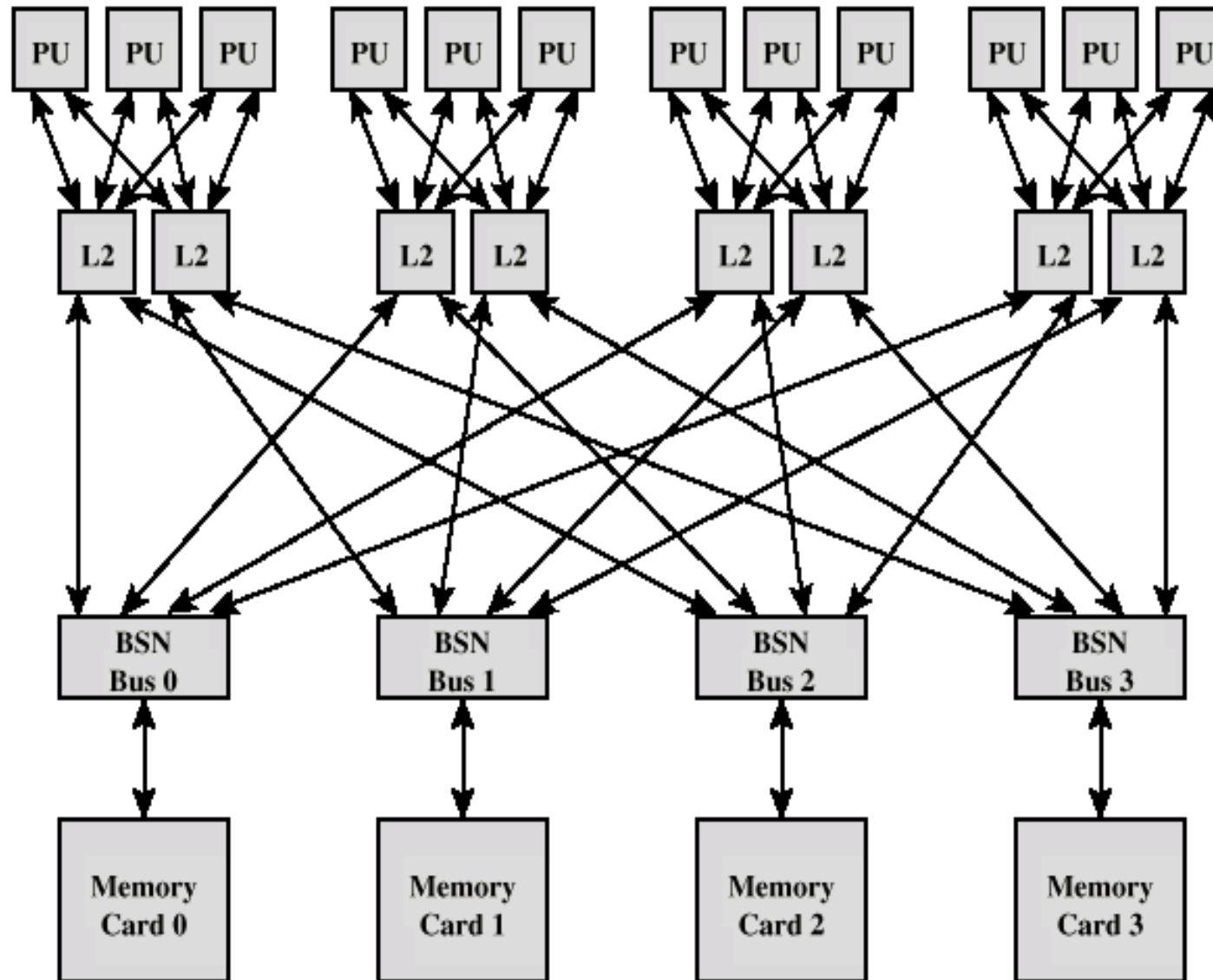
SMP – Memória Multiportas

- Mais complexa
 - precisa de um *login* extra no sistema de memória
- Desempenho melhor
 - cada processador tem o seu próprio caminho aos módulos de memória
- Pode configurar porções da memória como “privada” para um ou mais processadores
 - aumenta a segurança
- Política *write through* para atualização da cache

SMP – Considerações sobre o SO

- Encapsula detalhes, fornece visão de uma arquitetura monoprocessada
- Trabalha com:
 - processos concorrentes
 - escalonamento
 - sincronização
 - gerência de memória
 - confiabilidade e tolerância a falhas

Exemplo – IBM S/390 *mainframe*



Memória Compartilhada Distribuída

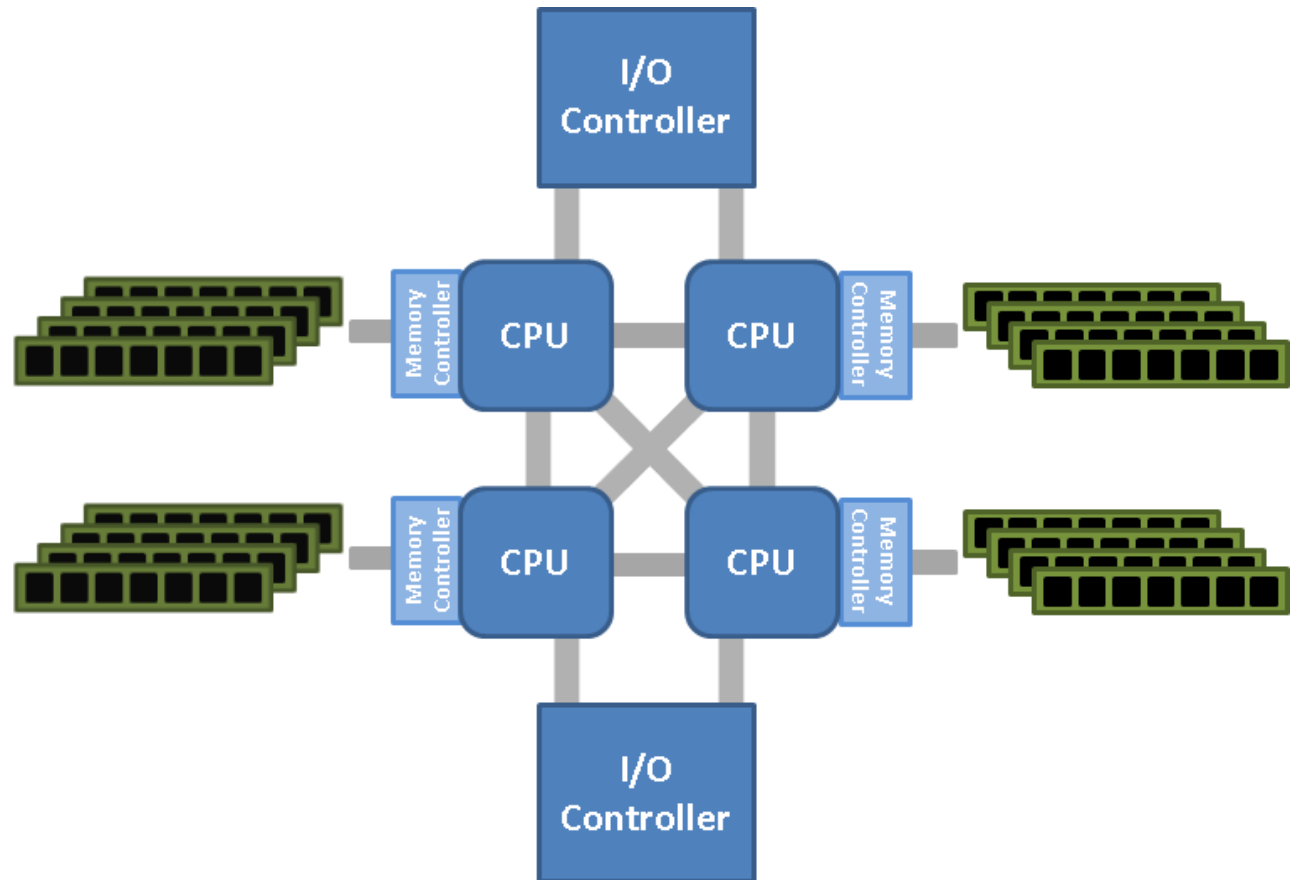
- Implementação de uma memória que é logicamente compartilhada mas implementada com o uso de um conjunto de memórias locais
- Pode ser de três classes:
 - NUMA (*Non-uniform Memory Access*)
 - COMA (*Cache-Only Memory Access*)
 - CC-NUMA (*Cache Coherent Non-Uniform Memory Access*)

NUMA (*Non-Uniform Memory Access*)

- NUMA - quando o acesso à memória NÃO é uniforme
 - todos os processadores têm acesso a toda memória
 - normalmente usando *load & store*
 - tempo de acesso à memória difere em função da região
 - diferentes processadores têm acesso às regiões da memória em diferentes velocidades, o que faz necessário um certo cuidado na hora de programar

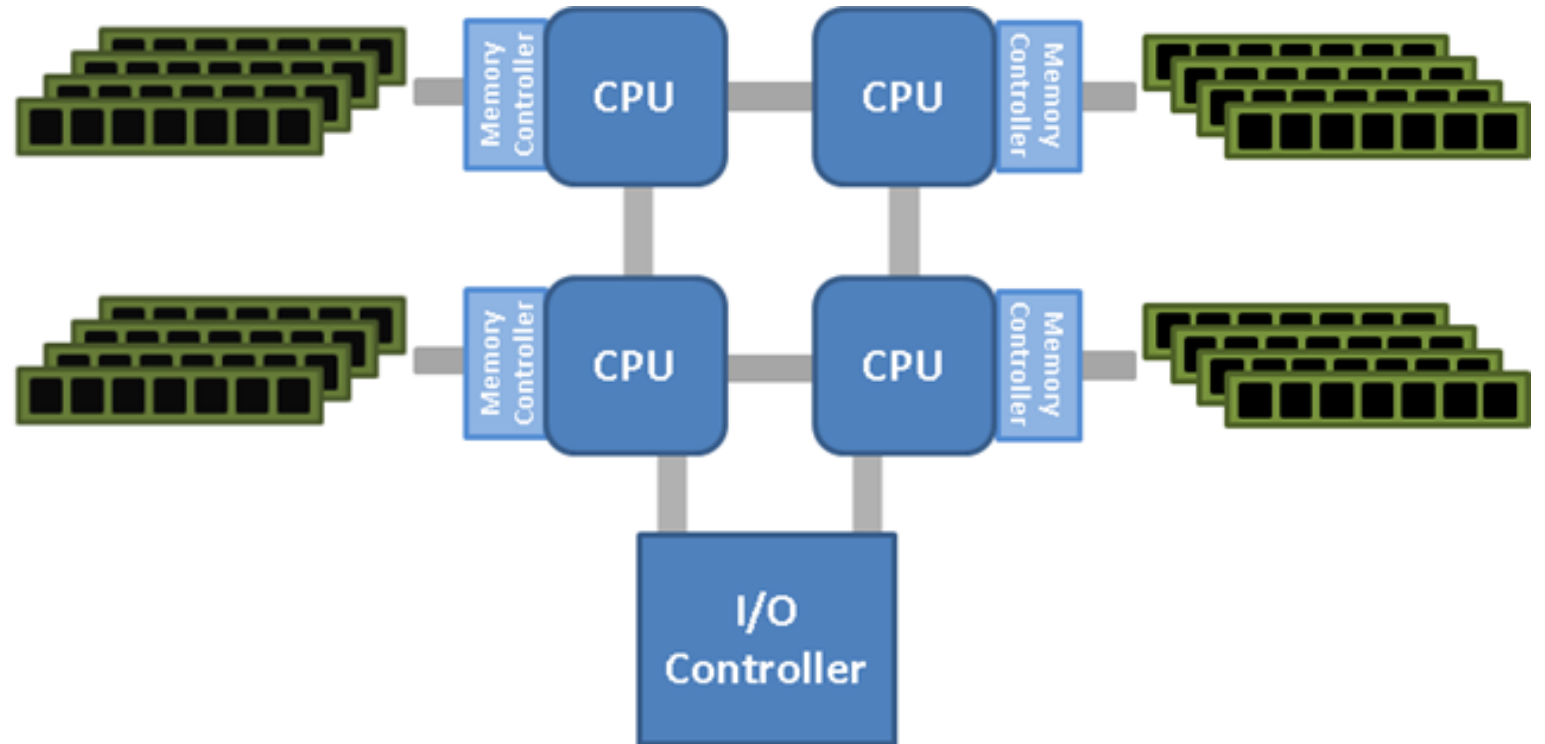
NUMA (*Non-Uniform Memory Access*)

- Intel Quick-Path Interconnect (QPI)



NUMA (*Non-Uniform Memory Access*)

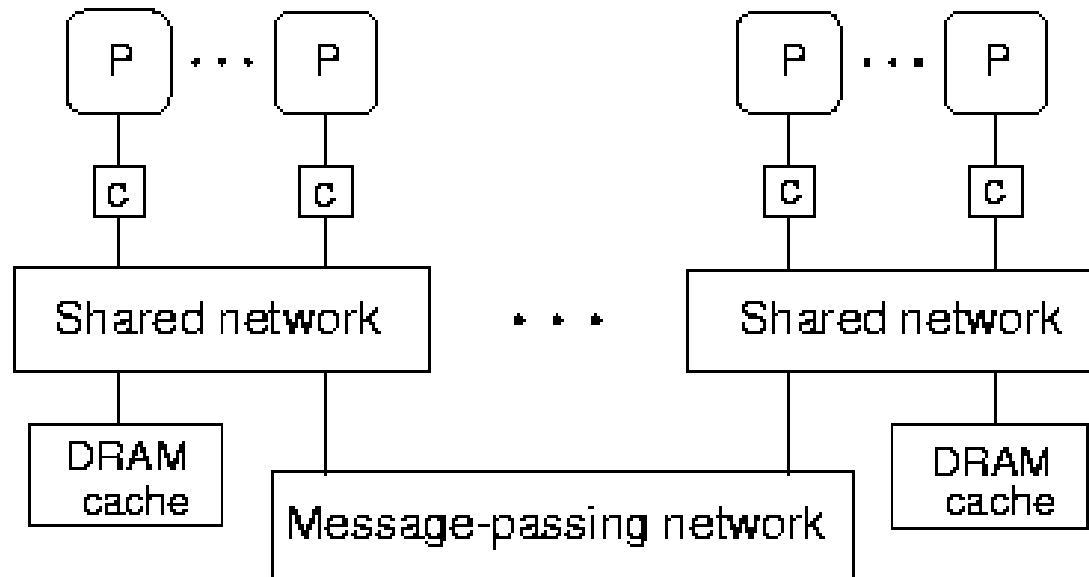
- AMD Hyper-Transport (HT)



COMA (*Cache-Only Memory Access*)

- Assemelham-se a uma arquitetura NUMA, onde cada nó de processamento possui uma parte da memória global.
- O particionamento dos dados entre as memórias de cada nó não é estático -> as memórias funcionam como caches de nível 3.
- O problema de partição de dados e balanceamento dinâmico de carga é realizado automaticamente.
- Conforme o algoritmo de coerência utilizado, os dados migram automaticamente para as caches locais dos processadores onde é mais necessária.

COMA (*Cache-Only Memory Access*)



CC-NUMA (*Cache Coherent Non-Uniform Memory Access*)

- Solução de compromisso entre as arquiteturas NUMA e COMA.
- Cada nó processador possui uma cache local para reduzir o tráfego na rede de interconexão.
- O balanceamento de carga é realizado dinamicamente pelos protocolos de coerência das caches.

CC-NUMA (*Cache Coherent Non-Uniform Memory Access*)

