

Organização e Arquitetura de computadores

Multiprocessadores e clusters

Prof. Dr. Luciano José Senger

Introdução

- Os multiprocessadores e clusters normalmente definem alto desempenho como alta vazão para tarefas independentes.
 - Como os processadores paralelos compartilham dados?
 - Como os processadores paralelos se coordenam?
 - Quantos processadores?
- Compartilhamento de dados:
 - Os processadores com um único espaço de endereçamento, algumas vezes chamados de processadores de memória compartilhada, oferecem ao programador um único espaço de endereçamento de memória que todos os processadores compartilham. Os processadores se comunicam por meio de variáveis compartilhadas na memória, com todos os processadores sendo capazes de acessar qualquer local da memória por meio de loads e stores.

Memória compartilhada

- Espaço de endereçamento único:
 - O primeiro leva o mesmo tempo para acessar a memória principal independente de qual processador o requisita e de qual word é requisitada. Essas máquinas são chamadas **UMA** (*Uniform Memory Access*) ou **SMP** (*Symmetric Multiprocessors*)
 - No segundo estilo, alguns acessos à memória são mais rápidos do que outros, dependendo de que processador pede que word. Essas máquinas são chamadas **NUMA** (*Nonuniform Memory Access*)
- As dificuldades de programação em um multiprocessador **NUMA** são diferentes das dificuldades em um multiprocessador **UMA**, mas as máquinas **NUMA** podem escalar para tamanhos maiores e, portanto, potencialmente possuem desempenho mais alto.

Troca de mensagens

- A troca de mensagens é necessária para máquinas com memórias privadas, em contraste com a memória compartilhada.
 - Exemplo: arranjo computacional (*cluster*)
 - processadores em diferentes computadores desktop se comunicam trocando mensagens por meio de uma rede local.
 - Desde que o sistema tenha rotinas de envio e recepção de mensagens, a coordenação está embutida na troca de mensagens, já que um processador sabe quando uma mensagem é enviada e o processador receptor sabe quando uma mensagem chega.
 - O processador receptor, então, pode enviar uma mensagem de volta para o emissor dizendo que a mensagem chegou se o emissor precisar dessa confirmação.
- Além desses dois estilos de comunicação principais, os multiprocessadores são construídos em duas

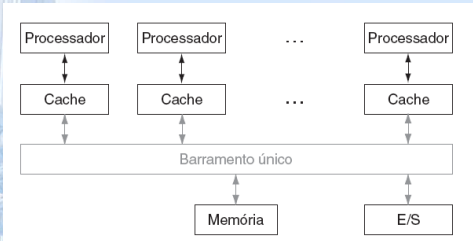
Programação Paralela

- Aplicações paralelas
 - Para que a computação paralela tenha bom desempenho, geralmente é necessário que sejam escritas aplicações paralelas que resolvem problemas que demandam de maior potência computacional
 - Objetivo: obter Speed-up linear: desempenho aumenta (tempo de execução diminui) linearmente na medida que aumenta a quantidade de processadores
 - Limitação: Lei de Amdahl – certas partes de um programa não podem ser paralelizáveis
 - $T_a = T_e/Q + T_n$
 - T_a : tempo de execução após a melhoria
 - T_e : tempo de execução antes da melhoria
 - Q : quantidade da melhoria (no caso número de processadores)
 - T_n : tempo de execução não afetado pela melhoria
 - Algumas vezes o speed-up pode ser superlinear: aplicações específicas que se beneficiam do paralelismo

Multiprocessadores conectados por um único barramento

- O alto desempenho e o baixo custo do microprocessador inspiraram um novo interesse nos multiprocessadores na década de 1980.
- microprocessadores podem ser convenientemente colocados em um barramento comum:
 - Como cada microprocessador é muito menor do que um processador multichip, mais processadores podem ser colocados em um barramento.
 - As caches podem reduzir o tráfego de barramento.
 - Foram inventados mecanismos para manter as caches e a memória consistentes para multiprocessadores, assim como as caches e a memória são mantidas consistentes para a E/S, simplificando, assim, a programação.

Multiprocessadores conectados por um barramento único



Multiprocessadores conectados por um único barramento

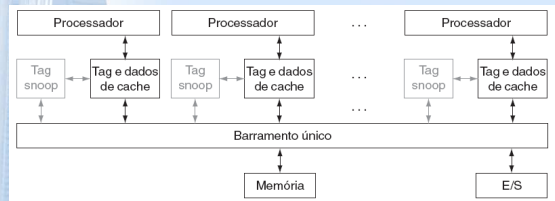
- O tráfego por processador e a largura de banda de barramento determinam o número útil de processadores nesse multiprocessador
 - As caches duplicam os dados em suas memórias mais rápidas para reduzir a latência para os dados e para reduzir o tráfego de memória no barramento
- Operações de E/S pode experimentar inconsistências no valor dos dados entre a versão na memória e a versão na cache. Esse problema de coerência de cache se aplica tanto a multiprocessadores quanto a E/S.
 - Múltiplos processadores normalmente requerem cópias dos mesmos dados em múltiplas caches.
 - Acessos a dados compartilhados podem ser forçados a sempre passar da cache para a memória (*write-through*), mas isso seria muito lento e exigiria muita largura de banda do barramento; o desempenho de um programa

Multiprocessadores conectados por um único barramento

- Os protocolos para manter coerência em múltiplos processadores são chamados **protocolos de coerência de cache**
- O protocolo mais comum para manter a coerência de cache é chamado **snooping**
 - Todos os controladores de cache monitoram o barramento para determinar se possuem uma cópia do bloco compartilhado
 - A manutenção da coerência possui dois componentes: leituras e escritas.
 - Múltiplas cópias não são um problema durante a leitura, mas um processador precisa ter acesso exclusivo para escrever uma word.
 - Os processadores também precisam ter a cópia mais recente quando estão lendo um objeto e, então, todos os processadores precisam obter valores novos após uma escrita. Portanto, os protocolos de snooping precisam localizar todas as caches que compartilham um objeto a

Multiprocessadores conectados por um único barramento

- Os protocolos de *snooping* precisam localizar todas as caches que compartilham um objeto a ser escrito. A consequência de uma escrita em dados compartilhados é invalidar todas as outras cópias ou atualizar as cópias compartilhadas com o valor sendo escrito.



Multiprocessadores conectados por um único barramento

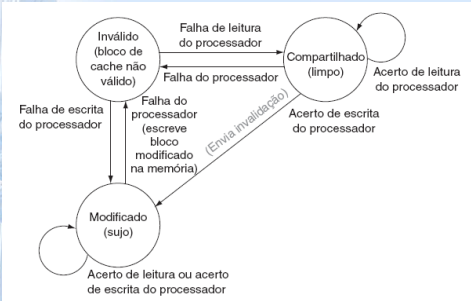
- Os bits de status já existentes em um bloco de cache são expandidos para protocolos de *snooping*, e essas informações são usadas no monitoramento de atividades de barramento:
- Em uma falha de leitura, todas as caches verificam se elas possuem uma cópia do bloco requisitado e, depois, tomam a ação apropriada, como fornecer os dados para a cache que falhou.
- Em uma escrita, todas as caches verificam se possuem uma cópia e, então, agem, invalidando ou atualizando sua cópia com o novo valor

Multiprocessadores conectados por um único barramento

- Para preservar a largura de banda de comunicações, todas as máquinas usam **write-invalidate** como o protocolo de coerência padrão:
 - o processador de escrita faz com que todas as cópias em outras caches sejam invalidadas antes de mudar sua cópia local; depois, eles estão liberados para atualizarem os dados locais até que outro processador os solicite
 - O processador de escrita ativa um sinal de invalidação no barramento e todas as caches verificam se elas possuem uma cópia; se possuírem, elas precisam invalidar o bloco contendo a word.
 - Esse esquema permite diversos leitores mas apenas um escritor
- Variações
 - MESI, um protocolo de invalidação de escrita cujo nome é um acrônimo para os quatro estados do protocolo: Modified (Modificado), Exclusive (Exclusivo), Shared (Compartilhado), Invalid (Inválido).

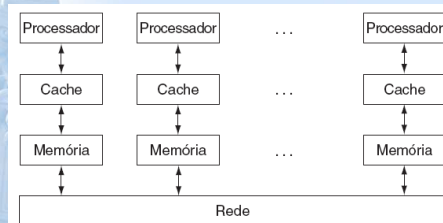
Multiprocessadores conectados por um único barramento

- Transições de estado da cache utilizando sinais do processador



Multiprocessadores conectados por uma rede

- Redes de comunicação permitem conectar muito mais processadores em relação a utilização de barramentos
- Ao invés de *loads/stores*, emprega-se primitivas *send/receive* para a troca de dados e sincronização
- Chamados de multicomputadores



Clusters

- Multiprocessadores construídos empregando redes locais de alta velocidade
- Utilização uma granulação de instruções em nível de processos ou também *threads*
- Uso de software especializado para implementar a imagem de uma máquina paralela
 - PVM - Parallel Virtual Machine
 - MPI - Message-Passing Interface
 - Bibliotecas de DSM - *Distributed Shared memory*