

Máquinas Paralelas - Sistemas Multiprocessados

Igor Pereira Teixeira
João Vitor de Araújo Braga
Leonardo Miranda da Silva Paixão
Roberto Vítor Lima Gomes Rodrigues

**JUAZEIRO
2018**

Tradicionalmente, o computador tem sido visto como uma máquina sequencial. Esta visão do computador nunca foi totalmente verdadeira. Em nível de micro-operações, vários sinais de controle são gerados ao mesmo tempo. O pipeline de instruções, pelo menos quando há sobreposição de operações de leitura e execução, está presente há muito tempo. Ambos são exemplos de desempenho de processamentos em paralelo.

Taxonomia de Flynn

Uma taxonomia introduzida inicialmente por Michael J. Flynn é ainda a maneira mais comum de categorizar sistemas com capacidade de processamento paralelo. A taxonomia de Flynn se baseia na possível unicidade e multiplicidade dos fluxos de instruções e de dados para definir quatro tipos de arquiteturas.

- **SISD (Single Instruction, Single Data)**

É um dos tipos de arquitetura mais simples, já que opera apenas um dado a cada instrução. Um processador único executa uma única sequência de instruções para operar nos dados armazenados em uma única memória. Uniprocessadores enquadram-se nesta categoria

- **SIMD (Single Instruction, Multiple Data)**

Em arquiteturas SIMD um único fluxo de instruções opera sobre múltiplos fluxos de dados. Existe uma única UC e múltiplos EP. Todos os EP executam simultaneamente e sincronizadamente a mesma instrução sobre conjuntos de dados distintos.

Processadores vetoriais funcionam melhor apenas quando há grandes quantidades de dados a serem trabalhados. Por esta razão, estes tipos de CPUs foram encontrados principalmente em supercomputadores, como os supercomputadores próprios eram, em geral, encontrados em locais como centros de previsão do tempo e laboratórios de física, onde grandes quantidades de dados são processados.

- **MISD (Multiple Instruction, Single Data)**

É um tipo de arquitetura de computação paralela, onde uma sequência de dados é transmitida para um conjunto de processadores e cada um executa uma sequência de instruções diferente sobre o mesmo dado. Mas até o momento não há implementação comercial desse tipo de arquitetura.

- **MIMD (Multiple Instruction Multiple Data)**

Um conjunto de processadores que executam sequências de instruções diferentes simultaneamente em diferentes conjuntos de dados. Com a organização MIMD, os processadores são de uso geral; cada um é capaz de processar todas as instruções

necessárias para efetuar transformação de dados apropriada. MIMDs podem ser ainda divididos pelos meios de comunicação do processador

SMP - Simetric Multi-Processing

Até pouco tempo, as máquinas possuíam um processador único, com funcionalidade de propósito geral. A medida que os processos exigiam mais e mais desempenho do sistema, e o custo de produção de microprocessadores começou a diminuir, foi surgindo a necessidade do uso de máquinas com processos paralelos. O SMP (Simetric Multi-processing) então, é uma organização de hardware e também de sistema operacional, com as seguintes características:

- 1. Há dois ou mais processadores semelhantes de capacidade comparável.
- 2. Esses processadores compartilham a mesma memória principal e os recursos de E/S, e são interconectados por um barramento ou algum outro esquema de conexão interna, de tal forma que o tempo de acesso à memória é aproximadamente igual para cada processador.
- 3. Todos os processadores compartilham acesso aos dispositivos de E/S, ou pelos mesmos canais ou por canais diferentes que fornecem caminhos para o mesmo dispositivo.
- 4. Todos os processadores desempenham as mesmas funções (daí o termo simétrico).
- 5. O sistema é controlado por um sistema operacional integrado que fornece interação entre processadores e seus programas em nível de trabalhos, tarefas, arquivos ou elementos de dados.

O sistema de SMP tem várias vantagens em relação ao uniprocessador, dentre elas podemos destacar o desempenho, já que se for possível organizar para que processos possam ocorrer em paralelo, há um ganho substancial no tempo ao processar várias instruções simultaneamente, a disponibilidade, pois caso haja falha de um processador, a máquina continuará funcionando normalmente, o crescimento incremental, pois a melhora do sistema depende do número de processadores inseridos, e a escalabilidade, pois facilita a venda de processadores, mudando apenas os números de processadores disponíveis.

A organização mais comum para computadores pessoais, estações de trabalho e servidores é o barramento de tempo compartilhado. O barramento de tempo compartilhado é o mecanismo mais simples para construir um sistema multiprocessador. Para um sistema de compartilhamento de barramento, alguns recursos são fornecidos, entre eles: endereçamento: deve ser possível distinguir os módulos no barramento para determinar a origem e o destino dos dados. Arbitração: qualquer módulo de E/S pode funcionar temporariamente como "mestre". Um mecanismo é fornecido para arbitrar requisições concorrentes para o controle do barramento, usando algum tipo de esquema de prioridade. Tempo compartilhado: quando um módulo está controlando o barramento, outros módulos são bloqueados e devem, se necessário, suspender a operação até que o acesso

ao barramento seja possível. simplicidade: esta é a abordagem mais simples para organização de multiprocessadores. A interface física e lógica de endereçamento, arbitração e tempo compartilhado de cada processador permanecem as mesmas, como em um sistema de um único processador. Flexibilidade: normalmente é fácil expandir o sistema anexando mais processadores ao barramento. confiabilidade: o barramento é basicamente um meio passivo, e uma falha de qualquer dispositivo conectado não deve causar uma falha do sistema todo.

A principal desvantagem do barramento compartilhado é seu desempenho, visto que a velocidade máxima de troca de dados deste irá limitar a velocidade como um todo do sistema. Para resolver este problema, alguns tratamentos são utilizados, sendo o principal o uso de memória cache nos processadores, diminuindo o acesso constante ao barramento. tal solução apresenta alguns detalhes de cuidado com coerência de cache, que serão abordados mais á frente.

Uma outra organização possível é a de memórias multiportas. Nela uma memória pode ser ligada diretamente aos processadores sem a utilização de uma rede de conexão. A utilização de memórias multiportas é limitada por questões de custo e escalabilidade (o número de portas, e portanto de processadores, é limitado). Uma boa saída é ligação de cada memória a apenas um conjunto de processadores.

Sistemas operacionais para multiprocessadores

Um sistema operacional SMP gerencia processadores e outros recursos computacionais para que o usuário perceba um único sistema operacional controlando os recursos do sistema. Vários processos podem estar ativos ao mesmo tempo e é responsabilidade do sistema operacional escalonar a sua execução e alocar recursos. Um sistema operacional para multiprocessadores deve fornecer toda a funcionalidade de um sistema multiprogramado mais os recursos adicionais para acomodar múltiplos processadores.

- **Processos concorrentes simultâneos**

Rotinas do SO precisam ser reentrantes para permitir que vários processadores executem o mesmo código do SO simultaneamente. A execução das diferentes partes do SO, tabelas do SO e estruturas de gerenciamento devem ser gerenciadas de acordo para evitar deadlock ou operações inválidas.

- **Escalonamento**

Qualquer processador pode efetuar escalonamento, portanto os conflitos devem ser evitados. O escalonador deve atribuir processos prontos para processadores disponíveis.

- **Sincronização**

Com múltiplos processos ativos tendo acesso potencial a espaços da

memória compartilhada ou recursos de E/S compartilhados, cuidados devem ser tomados para fornecer sincronização eficiente. A sincronização é um recurso que reforça a exclusão mútua e ordenação de eventos.

- **Confiabilidade e tolerância a falhas**

O sistema operacional deve prover uma degradação sutil perante uma falha do processador. O escalonador e outras partes do sistema operacional devem reconhecer a perda de um processador e reestruturar as tabelas de gerenciamento de acordo.

- **Gerenciamento de memória**

O sistema operacional precisa explorar o paralelismo disponível no hardware, tais como memórias com múltiplas portas para alcançar o melhor desempenho. Os mecanismos de paginação em diferentes processadores devem ser coordenados para reforçar a consistência quando vários processadores compartilham uma página ou um segmento para decidir sobre substituição de página

Incoerência de cache

A moderna disposição das unidades de processamento favorece os sistemas multiprocessados com a disposição de múltiplos núcleos ligados a uma memória em comum. Em contrapartida, isso propicia também a geração da chamada incoerência de cachê, já que cada processador tem sua própria unidade de memória local (ou cache). O que acontece é que todo núcleo, ao eventualmente acionar suas políticas de (re)estrita dos dados da cache (write-back ou write through), fá-lo independentemente dos outros, o que pode decididamente invalidar as cópias do dado em todas as outras memória locais.

Em detrimento desse novo problema, precisou-se criar protocolos de gerenciamento das memórias caches para evitar a incoerência e garantir o fluido desempenho dos sistemas multiprocessados. Tais protocolos envolvem implementações em hardware ou software. Estes, no entanto, apesar de dispensar hardware e circuitos adicionais, gera alguns problemas de ineficiência no uso da cache, em parte por ter de tomar escolhas conservadoras demais, como identificar tipos de dados que poderiam vir a ser problemas e evitar de guardá-los na cache.

Firmou-se a protocolação de resolução à incoerência a nível de hardware, dentre os quais popularizaram-se os protocolos de monitoramento, dentro dos quais, por sua vez, destaca-se o protocolo MESI. O paradigma de monitoração fundamenta-se na atualização de "estados" para cada cópia de dados na cache, ao qual as outras caches deverão responder por meio do mecanismo de difusão nos barramentos. Duas técnicas são utilizadas em maioria pelos protocolos de monitoramento, são chamadas de write-invalidate e write-through, entre as quais a única diferença é a prioridade de escrita -- na write-invalidate, apenas uma das caches é apta a escrever por vez. O protocolo MESI utiliza-se da técnica write-invalidate, discriminando quatro estados para cada dado na cache: M - modified; E - exclusive; S - shared e I - invalid. Isso é feito com a análise dos bits de validade e limpeza da cache. O estado modificado representa um bloco de dados válido, entretanto mais atualizado do que as cópias noutras caches, o exclusivo representa um dado válido, porém presente apenas numa das caches, o compartilhado representa um dado válido,

atualizado e presente noutras caches e o inválido representa um dado inválido. Toda vez que um núcleo requisita acesso ao dado para ler e/ou modificar, a sequência de passos executados por cada controlador de cache verifica se foi um hit ou miss e o tipo de requisição, lendo subseqüentemente os estados dos blocos para tomar a devida ação.

- **Sistemas NUMA (Non- Uniform Memory Access)**

Este sistema tem alguns termos que são comumente associados a ele, sendo :

- **Acesso Uniforme à Memória (UMA):** Tem como exemplo as SMPs, onde todos os processadores têm acesso a todas as partes da memória, com tempo de acesso à memória por um ou mais processadores iguais;
- **Acesso Não-Uniforme à Memória(NUMA):** Aqui, o tempo de acesso à memória por um processador muda de acordo com a região da memória que está sendo utilizada, o que vale para todos os processadores;
- **NUMA com coerência de cache(CC-NUMA):** É um sistema NUMA com coerência de cache entre as caches dos vários processadores. É mais ou menos equivalente a um cluster.

Numa organização CC-NUMA típica (esta é a mais utilizada dentre as três comentadas), existem vários nós independentes, cada qual, uma organização SMP, com diversos processadores cada um com suas próprias caches L1 e L2, além da memória principal. Este nó é o bloco básico da organização CC-NUMA como um todo. Os nós são conectados por meio de algum sistema de comunicação, por exemplo, uma rede ou um mecanismo de comutação. Mesmo com essa formatação dos nós, existe uma única área de memória endereçável, com um endereço único em todo o sistema.

Quando um processador inicia um acesso à memória, se a posição de memória requerida não está disponível na cache do processador, a cache L2 inicia uma busca. Se a linha requerida está na porção local memória principal ela é obtida pelo barramento local; se está numa porção remota da memória principal, é automaticamente enviada uma requisição para buscar a linha pela rede de interconexão do sistema e entregue ao barramento local, que então repassa a linha à cache conectada à ele. A operação é feita de forma automática e transparente para o processador e sua cache.

Possui uma visão de uma grande e única área de memória no sistema, permitindo, ao mesmo tempo, vários nós multiprocessadores cada qual com seu barramento próprio.