

Arquitetura de Computadores II

2026/1

Nicolas Ramos Carreira

Sumário

1	Importância da matéria	2
2	Processamento paralelo	3
2.1	Sobre	3
2.2	Classificação de Flynn	3
2.2.1	MIMD	4
2.3	Computação em Nuvem e Multithread	6
2.3.1	Computação em nuvem	6
2.3.2	Threading e Multithreading	7

Capítulo 1

Importância da matéria

Capítulo 2

Processamento paralelo

2.1 Sobre

O processamento paralelo é uma ideia que surgiu para aumentar a performance dos computadores.

Sem o processamento paralelo, o que acontece é que nós temos processos que serão executados pela CPU e o sistema operacional basicamente irá escolher os processos prioritários, ele manda começar e parar um processo. Essa troca de processos é extremamente rápida, então dá a impressão que a CPU está executando os processos ao mesmo tempo, mas não está. Isso é chamado de arquitetura sequencial, que executa os processos vez a vez. Veja uma imagem:

Dessa forma, para melhorar a performance e fazer a CPU executar mais de um processo ao mesmo tempo, surgiu a arquitetura paralela.

2.2 Classificação de Flynn

As arquiteturas de processamento foram classificadas por um cara chamado Flynn conforme a quantidade de fluxos de instruções e dados.

A arquitetura sequencial, por exemplo, só tinha um fluxo de instrução e dados, sendo classificada como SISD

Por outro lado, nós temos aquelas com múltiplos fluxos de instrução e dados, que são as arquiteturas paralelas e foram divididas em 3 categorias:

- SIMD: Unica instrução e múltiplo de dados. Dentro disso, temos processadores matriciais e processadores vetoriais
- MISD (não existe mais): Múltiplo de instrução e apenas um de dados
- MIMD: Múltiplo de instrução e dados. Dentro disso temos a memória compartilhada e a memória distribuída, sendo que se você está trabalhando com memória distribuída, você está trabalhando com clusters e se você está trabalhando em memória compartilhada, você pode estar trabalhando com SMP ou NUMA

2.2.1 MIMD

SMP

SMP, conhecido como múltiplo processador simétrico é basicamente a prática de colocar mais um processador, geralmente idêntico. Veja a imagem a seguir:

O SMP é memória compartilhada porque eles acessam o mesmo banco de memórias, como pode ser visto acima

A questão é que nós temos 2 problemas com o SMP. Veja a imagem abaixo antes de eu explicar:

Um dos problemas é que temos o engarrafamento no barramento. CPU-2, por exemplo, vai acessar a RAM, então outros não podem usar nesse meio tempo. Esse problema começa com um limite físico de 30 processadores.

Além desse problema, temos outro, onde se temos uma variável $x=0$ na RAM e a CPU-0 quer usar, ele lê a variável e coloca na memória cache. Se a CPU-2 ler essa variável também e tiver uma instrução $x++$ e mudar a variável para $x=1$, criamos um problema em CPU-0, pois essa CPU não pode fazer o cálculo com $x=0$. O nome desse problema é: problema de coerência de cache.

Para resolver o problema de coerência de cache, foram propostas duas soluções: uma utilizando software e outra utilizando hardware, sendo que a que foi pra frente mesmo foi utilizando hardware:

- Protocolo Mesi: Pequeno processador que tem a função de ficar monitorando as CPUs através do algoritmo mesi. Esse chip é EXTREMAMENTE CUSTOSO

Assim o Chip Mesi resolveu este que seria o pior dos problemas, mas ainda não resolve o engarrafamento no barramento. Para isso, saímos da arquitetura SMP e vamos para a NUMA.

NUMA

Na arquitetura NUMA, nós dividimos as conexões em outros barramentos e dividimos a memória RAM. Veja a imagem a seguir:

Como as memórias estão separadas, uma vai do endereço de 0 a 999, outra de 1000 a 1999.. e assim vai.

Um detalhe é que o acesso é NÃO UNIFORME, uma vez que dependendo do onde está o endereço de memória do que você queira acessar, uma CPU demoraria mais que outra.

Uma observação é que a arquitetura NUMA está morrendo, por ser muito cara.

Cluster

O cluster se parecerá com a arquitetura numa, porém lá a memória é compartilhada (mesmo sendo separadas, podemos acessar uma memória com qualquer CPU), já no cluster, são literalmente máquinas diferentes. Um processador não acessa a memória do outro. Veja a imagem abaixo:

Dessa forma, podemos resumir cluster em: computadores separados que trabalham como um só.

Um exemplo clássico da aplicação de cluster é o cálculo de π , onde temos que dividir a carga de trabalho para diferentes computadores para chegar no valor.

Existem alguns exemplos de clusters bem famosos. São os supercomputadores. Aqui no Brasil nós temos o Santos Dumont, que é composto por 246 nós, ou seja, 246 computadores.

Podemos ter duas categorias de clusters:

- HPC: Feito para alta performance
- HAC: Não foi feito para ter um alto poder de processamento, mas sim para não parar. É bastante usado por empresas de cartão de crédito, militares e etc. Dentro dos HAC temos duas categorias:
 - Failover: Quando um dos computadores dá defeito, outro computador assume.

- Fail Back: Quando um computador dá defeito, nós consertamos, colocamos de volta e o computador que tinha dado defeito assume de onde parou.

Algo que é importante de se pontuar é que é necessário uma área de storage para o cluster continuar a operação que um cluster estava fazendo quando deu defeito. Existem três formas de implementar isso:

PAS

NAS

SAN

Em cluster, você deve programar usando técnicas de programação paralela, você deve mandar seu programa para todos os clusters. O difícil disso é dividir qual tarefa vai para cada cluster

2.3 Computação em Nuvem e Multithread

2.3.1 Computação em nuvem

A computação em nuvem é basicamente usar o servidor de outra pessoa. Criar um servidor por si próprio custa uma grana e precisa de alguém para dar manutenção

Quando usamos a computação em nuvem o que fazemos é alugar uma máquina virtual em um grande cluster. Veja como ocorre isso na computação em nuvem por meio da imagem a seguir:

A computação em nuvem tem alguns mantras, sendo eles:

- Serviço mensurado
- Auto serviço sob demanda
-

Os tipos de nuvem que temos são:

- Nuvem pública: Disponível publicamente para você usar
- Nuvem privada: Nuvem que empresas criam para elas por conta de uma política de sigilo de dados
- Nuvem comunitária: Mistura da pública e privada. Toda on que participou da criação pode usar, sendo que cada ong tem sua nuvem privada

2.3.2 Threading e Multithreading

A prática de Threading permite a melhora do desempenho