Arquitetura de Computadores

Aula 6 – Arquiteturas Paralelas

Prof. André Roberto Guerra

Organização da Aula

- **Paralelismo**
  - Visão geral do paralelismo
  - Paralelismo no CHIP
  - Coprocessadores
  - Multiprocessadores *versus* multicomputadores
  - Escalonamento
  - Desempenho

Visão Geral do Paralelismo

- Embora os computadores continuem a ficar cada vez mais rápidos, as demandas impostas a eles estão crescendo no mínimo com a mesma rapidez

- Em suma, seja qual for a capacidade de computação disponível, para muitos usuários, em especial nas áreas da ciência, engenharia e industrial, ela nunca será suficiente

- Portanto, para enfrentar problemas cada vez maiores, os arquitetos de computadores estão recorrendo cada vez mais a computadores paralelos

- Apesar de talvez não ser possível construir uma máquina com uma única CPU é com tempo de ciclo de 0,001 ns, pode ser perfeitamente viável construir uma com 1.000 CPUs com um tempo de ciclo de 1 ns cada

- O paralelismo pode ser introduzido em vários níveis
- No nível mais baixo, ele pode ser adicionado ao CHIP da CPU por *pipeline* e projetos superescalares com várias unidades funcionais

- Também pode ser adicionado por meio de palavras de instrução muito longas com paralelismo implícito
- Várias CPUs podem ser reunidas no mesmo chip

- Juntas, essas características podem equivaler, talvez, a um fator de 10 vezes em desempenho em relação a projetos puramente sequenciais

- No nível seguinte, placas extras de CPU com capacidade de processamento adicional podem ser acrescentadas a um sistema
- Funções especializadas, como: processamento de rede, de multimídia ou criptografia

- Para conseguir um fator de cem, de mil, ou de milhão, é necessário replicar CPUs inteiras e fazer que todas elas funcionem juntas com eficiência
- Essa ideia leva a grandes multiprocessadores e multicomputadores (*clusters*)

- É possível envolver organizações inteiras pela Internet e formar grades de computação fracamente acopladas
- Esses sistemas estão apenas começando a surgir, mas têm um potencial interessante para o futuro

- Duas CPUs próximas, em termos computacionais, são **fortemente acopladas**
- Quando longe uma da outra, são **fracamente acopladas**

Paralelismo

- **Paralelismo no CHIP**
  - Um modo de aumentar a produtividade de um chip é conseguir que ele faça mais coisas ao mesmo tempo
  - Em outras palavras, explorar o paralelismo

- Alguns modos de aumentar a velocidade por paralelismo no chip, incluídos paralelismo no nível da instrução, *multithreading* e mais de uma CPU no Chip
- Técnicas diferentes, mas cada uma delas ajuda à sua própria maneira

Paralelismo no Nível da Instrução

- Um modo de paralelismo no nível mais baixo é emitir múltiplas instruções por ciclo de *clock*
- Há duas CPUs de emissão múltipla
  - Processadores superescalares
  - Processadores VLIW

Paralelismo

- CPUs superescalares são capazes de emitir múltiplas instruções para as unidades de execução em um único ciclo de *clock*
- O número real de instruções emitidas depende do projeto do processador, bem como das circunstâncias correntes

- O *hardware* determina o número máximo que pode ser emitido
- Em geral, duas a seis instruções
- Se a instrução precisa de unidade funcional não disponível ou um resultado ainda não foi calculado, ela não será emitida

- A outra forma de paralelismo no nível da instrução é encontrada em processadores VLIW (*Very Long Instruction Word*)
- Na forma original, máquinas VLIW tinham palavras longas que continham instruções que usavam múltiplas unidades funcionais

- Projeto muito rígido
- Nem toda instrução utiliza todas unidades funcionais, resultando em muitas NO-OP inúteis, usadas como filtro

- Modernas máquinas VLIW têm modo de marcar grupo de instruções que formam um conjunto com bit “final de grupo”
- O processador pode buscar o grupo inteiro e emiti-lo de uma vez só

- Cabe ao compilador preparar grupos de instruções compatíveis
- VLIW transfere do tempo de execução para o tempo de compilação o trabalho de determinar quais instruções podem ser emitidas em conjunto

- Essa opção simplifica o *hardware* e o torna mais rápido
- Permite que se montem pacotes melhores do que o *hardware* poderia montar durante o tempo de execução

- O paralelismo no nível da instrução não é a única forma de paralelismo de baixo nível
- Outra forma é o paralelismo no nível da memória, no qual há múltiplas operações de memória no ar ao mesmo tempo

Multithreading, Multiprocessadores e Coprocessadores

- ***Multithreading* no chip**
- **Multiprocessadores com um único chip**
- **Coprocessadores**

- **Processadores de rede**
- **Processadores de mídia**
- **Criptoprocessadores**

Multiprocessadores e Multicomputadores

- **Multiprocessadores de memória compartilhada**
- **Multiprocessadores *versus* Multicomputadores**

- **Multiprocessadores**
  - Computador paralelo – todas CPUs compartilham memória comum
  - Todos os processos que funcionam juntos podem compartilhar um único espaço de endereço virtual mapeado para a memória comum

- Qualquer processo pode **ler/escrever** uma palavra de memória apenas executando uma instrução. Nada mais é preciso. O *hardware* faz todo resto
- Modelo de fácil entendimento pelos programadores e é aplicável a uma ampla faixa de problemas

- **Multicomputadores**
  - Arquitetura paralela – todas CPUs possuem sua própria memória privada, acessível somente por ela e nenhuma outra
  - Também denominado **sistema de memória distribuída**

- O aspecto fundamental que distingue um **multicomputador** de **multiprocessadores** é que a CPU de um multicomputador tem sua própria memória local privada, a qual pode acessar apenas executando *LOAD* e *STORE*

- CPUs em multicomputador não se comunicam lendo e escrevendo na memória comum
- Mensagens usando rede de interconexão
- Exemplos de **multicomputadores:** IBM BlueGene/P, Red Storm, *cluster* Google

Cluster, Escalonamento e Desempenho

- **Computação de *Cluster***
  - Outro estilo de multicomputador
  - Centenas de milhares de PCs ou estações de trabalho conectadas por uma placa de rede

- Dois tipos: o centralizado e o descentralizado
- O centralizado é um *cluster* de estações de trabalho ou PCs montado em uma grande estante em uma sala
- Máquinas homogêneas sem periféricos, exceto placas de rede

- *Clusters* descentralizados consistem em estações de trabalho ou PCs espalhados por um prédio ou *campus*
- Ociosos por muitas horas do dia
- Conectados por uma LAN

- Heterogêneos com conjunto completo de periféricos
- *Clusters* são conjuntos pequenos, com cerca de 500 PCs
- Contudo, também é possível construir *clusters* muito grandes com PCs de prateleira, como o Google faz

- **Escalonamento**
  - Programadores podem criar *jobs* com facilidade, requisitando várias CPUs e executando durante períodos substanciais de tempo

- Quando várias requisições independentes estão disponíveis vindas de diferentes usuários, cada uma necessitando um número diferente de CPUs por períodos de tempos diferentes, o *cluster* precisa de um escalonador para determinar qual *job* é executado

- Um algoritmo mais sofisticado requer que cada *job* apresentado especifique seu formato, isto é, quantas CPUs ele quer durante quantos minutos
- Esquema especialmente eficaz quando *jobs* são apresentados durante o dia para execução a noite, o escalonador tem as informações e pode executá-los na melhor ordem

- **Desempenho**
  - O ponto principal de um computador paralelo é a velocidade de execução – mais rápido que uma máquina com único processador

- Se não cumprir, não vale a pena o ter
- Deve ser eficiente em relação ao custo
- Para mensurar, são utilizadas métricas de *hardware* e *software*

Síntese

Referência de Apoio

- TANENBAUM, A. S. **Organização Estruturada de Computadores**. 6. ed. São Paulo: Prentice-Hall, 2013.