

Arquitetura de computadores

Capítulo 01

Introdução

Walisson Pereira

walisson_pereira@uvanet.br

Universidade Estadual Vale do Acaraú

Apresentação

Tendências tecnológicas

Medindo desempenho

Aceleração

A lei de amdahl

Resumo

Referências

Apresentação

Conteúdo programático:

- Introdução.
- Representação de dados e aritmética de computadores; organização de computadores; e modelos de programação.
- Projetos de processadores; pipelining; e paralelismo no nível da instrução.
- Memória; memória virtual; e caches.
- Entrada e saída; e multiprocessadores.

Bibliografia básica

- Carter, N. Teoria e Problemas de Arquitetura de computadores. Coleção Schaum. Tradução Ralph Miller Jr. Porto Alegre: Bookman, 2003.

Tendências tecnológicas

Desde o início da década de 1980, o desempenho dos computadores tem sido impulsionado por aperfeiçoamentos nas capacidades dos circuitos integrados utilizados para implementar microprocessadores, nos chips de memória e em outros componentes de computadores.

O impressionante crescimento do desempenho de computadores ao longo das últimas duas décadas foi impulsionado pelo fato de que a velocidade e a densidade dos chips foram aperfeiçoados geometricamente, em vez de linearmente.

A observação de que o desempenho dos computadores melhora geometricamente e não linearmente é frequentemente citada como a **Lei de Moore**.

“A complexidade para componentes com custos mínimos tem aumentado em uma taxa de aproximadamente um fator de dois por ano... Certamente, em um curto prazo, pode-se esperar que esta taxa se mantenha, se não aumentar. A longo prazo, a taxa de aumento é um pouco mais incerta, embora não haja razões para se acreditar que ela não se manterá quase constante por pelo menos 10 anos. Isso significa que, em torno de 1975, o número de componentes por circuito integrado para um custo mínimo será 65.000 (65nM). Eu acredito que circuitos grandes como este poderão ser construídos em um único componente (pastilha)”. (Moore, 1965)

Tendências tecnológicas

Em resumo, Moore previu, baseado em suas observações da indústria, que o número de transistores em um processador dobraria, em média, a cada dois anos e mantendo o mesmo (ou menor) custo e o mesmo espaço.

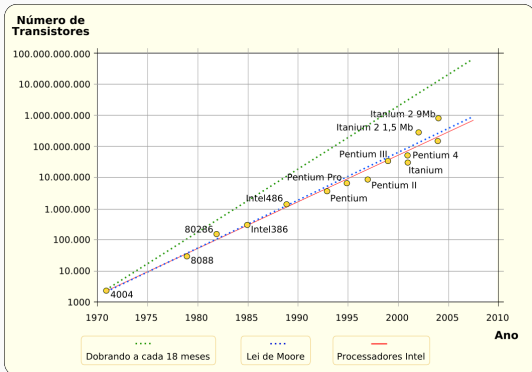


Figura 1: Lei de Moore x Produção

Em 1975 houve uma revisão dessa “lei” onde Moore redefiniu o período em que o número de transistores dobraria de dois anos para 18 meses. E ela tem se mostrado certa até hoje.

Inicialmente a lei de Moore não passava de uma observação, mas acabou tornando-se um objetivo para as indústrias de semicondutores, fazendo-as despenderem muitos recursos para poder alcançar as previsões de Moore no nível de desempenho e é isso que torna a Lei de Moore realmente importante, pois sem ela, **talvez não tivéssemos um desenvolvimento tão acelerado em nível de hardware e com custos cada vez mais acessíveis.**

O aumento da taxa de desempenho tem sido devido a **melhorias na arquitetura e na organização de computadores** - os projetistas de computadores têm sido capazes de tirar proveito da crescente densidade dos circuitos integrados para acrescentar recursos ao microprocessadores e aos sistemas de memória, os quais **proporcionam um desempenho acima do aumento da velocidade dos transistores** que os implementam.

Medindo desempenho

Desempenho descreve a rapidez com a qual um determinado sistema pode executar um programa ou programas. Sistemas que executam programas em menos tempo são ditos de melhor desempenho.

A melhor medida para o desempenho de um computador é o **tempo de execução de um programa**, ou programas, que o usuário deseja executar.

Geralmente é impraticável testar todos os programas que serão executados em um dado sistema antes de decidir qual computador comprar, ou quando se tomar decisões de projeto. Assim, os projetistas de computadores **produziram um certo número de unidades para escrever o desempenho de computadores**.

Além do desempenho, outros fatores podem influenciar decisões de projeto ou de compra.

Exemplos:

- Facilidade de programação
- Compatibilidade

Uma medida antiga do desempenho de computadores é a **taxa pela qual a máquina executa instruções**.

É calculado dividindo-se o número de instruções executadas em um programa pelo tempo necessário para executá-lo e, geralmente, expresso em **milhões de instruções por segundo (MIPS)**.

Caiu em desuso: não leva em conta o fato de que diferentes sistemas frequentemente precisam de números diferentes de instruções para implementar um dado programa.

O número de ciclos de relógio necessário para executar cada instrução, conhecido como **ciclos por instrução (CPI)**.

O CPI de um programa, em um sistema, é calculado dividindo-se o número de ciclos de relógio necessários para executar o programa pelo número de instruções executadas.

O número de **instruções executadas por ciclos (IPC)** é frequentemente utilizado para sistemas que podem executar mais de uma instrução por ciclo.

O IPC é calculado dividindo-se o número de instruções executadas, ao se executar um programa, pelo número de ciclos de relógio necessários para executar o programa.

Tanto o CPI quanto o IPC fornecem a mesma informação e a escolha de qual usar geralmente é baseada em qual dos valores é maior do que o número 1.

Valores altos de IPC indicam que o programa de referência demorou menos ciclos para ser executados do que valores baixos de IPC.

Valores altos de CPI indicam que foram necessários mais ciclos de que valores baixo de CPI.

Resumindo:

- IPC alto tende a indicar bom desempenho
- CPI alto indica um desempenho fraco.

Exemplo: Um dado programa consiste de um laço de 100 instruções que é executado 42 vezes. Se demora 16.000 ciclos para executar o programa em um dado sistema, quais são os valores de CPI e de IPC do sistema para este programa?

Exemplo: Um dado programa consiste de um laço de 100 instruções que é executado 42 vezes. Se demora 16.000 ciclos para executar o programa em um dado sistema, quais são os valores de CPI e de IPC do sistema para este programa?

Solução: O laço de 100 instruções é executado 42 vezes, de modo que o número total de instruções executara é $100 \times 42 = 4.200$

Demora 16.000 ciclos para executar o programa, de modo que o CPI é $16.000 \div 4.200 = 3,81$.

Para calcular do IPC, dividimos 4200 instruções por 16.000 ciclos, obtendo um IPC de 0,26.

Em geral, o IPC e o CPI são medidas ainda menos úteis do desempenho de sistemas atuais do que o MIPS, porque eles não contêm qualquer informação sobre:

- A frequência do relógio do sistema;
- Quantas instruções o sistema exige para executar uma tarefa.

CPI e IPC acabam sendo mais utilizadas em **pesquisas de arquitetura de computadores**, porque a maior parte desse tipo de pesquisa é feita utilizando programas que simulam uma arquitetura em especial, para estimar quantos ciclos um dado programa irá utilizar para ser executado naquela arquitetura.

Um **conjunto de benchmark** consiste de uma série de programas que acredita-se ser o correspondente típico de programas que serão executados no sistema.

A **pontuação** de um sistema no conjunto de benchmark é baseada em **quanto tempo o sistema demora para executar** todos os programas que o compõem.

Conjuntos de Benchmark

Um conjunto de benchmark fornecem várias vantagens sobre MIPS e CPI/IPC.

- Os resultados são baseados em tempos totais de execução, não na taxa de execução de instruções.
- Fazem a média do desempenho do sistema por vários programas, de modo a gerar uma estimativa da sua velocidade média.

Isto torna a avaliação geral do sistema em um conjunto de benchmark um indicador melhor do seu desempenho geral do que é a avaliação MIPS em qualquer programa isolado.

Média geométrica versus média aritmética

Muitos conjuntos de benchmark utilizam a **média geométrica**, em vez da **média aritmética**, para fazer a média dos resultados dos programas contidos no conjunto de benchmark porque um único valor extremo tem um impacto menor sobre a média geométrica de uma série do que sobre a média aritmética.

Média geométrica versus média aritmética

Exemplo: Quais são as médias aritméticas e geométrica dos valores 4, 2, 4, 82?

Média geométrica versus média aritmética

Exemplo: Quais são as médias aritméticas e geométrica dos valores 4, 2, 4, 82?

Solução:

A média aritmética é: $\frac{4+2+4+82}{4} = 23$

A média geométrica é: $\sqrt[4]{4 \times 2 \times 4 \times 82} = 7.16$

Note: a inclusão de um valor extremo na série teve um efeito muito maior sobre a média aritmética do que sobre a média geométrica.

Aceleração

Aceleração descreve como o desempenho de uma arquitetura muda à medida que diferentes melhoramentos são feitos naquela arquitetura.

A aceleração é simplesmente a **razão entre os tempos de execução antes e depois que a mudança** é feita, de modo que:

$$\text{Aceleração} = \frac{\text{Tempo de execução}_{\text{antes}}}{\text{Tempo de execução}_{\text{depois}}}$$

Exemplo: Um programa demora 25 segundos para ser executado e uma versão de uma arquitetura. Uma nova versão do programa roda em 15 segundos.

$$\text{Aceleração} = \frac{25 \text{ segundos}}{15 \text{ segundos}} = 1,67$$

A aceleração foi de 1,67.

A lei de amdahl

A lei de amdahl

A regra mais importante para projetar sistemas de computadores de alto desempenho é **faça com que o mais comum seja rápido**.

Qualitativamente, isto significa que o **impacto de um dado aperfeiçoamento sobre o desempenho geral depende tanto de quanto o aperfeiçoamento melhora o desempenho quando ele é utilizado**, como de com que frequência esse aperfeiçoamento é utilizado.

Quantitativamente, esta regra foi expressa pela **Lei de Amdahl**, que define.

A lei de amdahl

$$TE_{novo} = TE_{antigo} \times \left[P_{\text{n\~ao usada}} + \frac{P_{\text{usada}}}{\text{Acelera\~c\~ao}_{\text{usada}}} \right]$$

Onde:

- TE_{novo} é o novo tempo de execução
- TE_{antigo} é o antigo tempo de execução
- $P_{\text{n\~ao usado}}$ é a parcela de tempo na qual o aperfeiçoamento não está em uso
- P_{usado} é a parcela de tempo na qual o aperfeiçoamento está em uso
- $\text{Acelera\~c\~ao}_{\text{usada}}$ é a aceleração que acontece quando o aperfeiçoamento é usado.

A lei de amdahl

A lei de Amdahl pode ser reescrita utilizando a definição de aceleração:

$$\text{Aceleração} = \frac{TE_{antigo}}{TE_{novo}} = \frac{1}{P_{\text{não usada}} + \frac{P_{\text{usada}}}{\text{Aceleração}_{\text{usada}}}}$$

Exemplo: Suponha que uma dada arquitetura não tenha suporte de hardware para multiplicação, de modo que as multiplicações tenham que ser feitas por meio de adições repetidas (este é o caso de alguns dos primeiros microprocessadores). Se demora 200 ciclos para executar uma multiplicação por software e quatro ciclos para executar a multiplicação por hardware, qual é a aceleração geral produzida pelo hardware para suporte de multiplicações, se um programa gasta 10% de seu tempo fazendo multiplicações? E com um programa que gasta 40% do seu tempo fazendo multiplicações?

A lei de amdahl

Solução:

Quando utilizado o hardware para multiplicação:

$$\text{Aceleração} = \frac{200}{4} = 50.$$

Se o programa gasta 10% do seu tempo fazendo multiplicações, então $P_{\text{não usada}} = 0,9$ e $P_{\text{usada}} = 0,1$:

$$\text{Aceleração} = \frac{1}{0,9 + \frac{0,1}{50}} = 1,11$$

Se o programa gasta 40% do seu tempo fazendo multiplicações, antes que o hardware para multiplicações seja acrescentado, então $P_{\text{não usada}} = 0,6$ e $P_{\text{usada}} = 0,4$:

$$\text{Aceleração} = \frac{1}{0,6 + \frac{0,4}{50}} = 1,64$$

A lei de amdahl

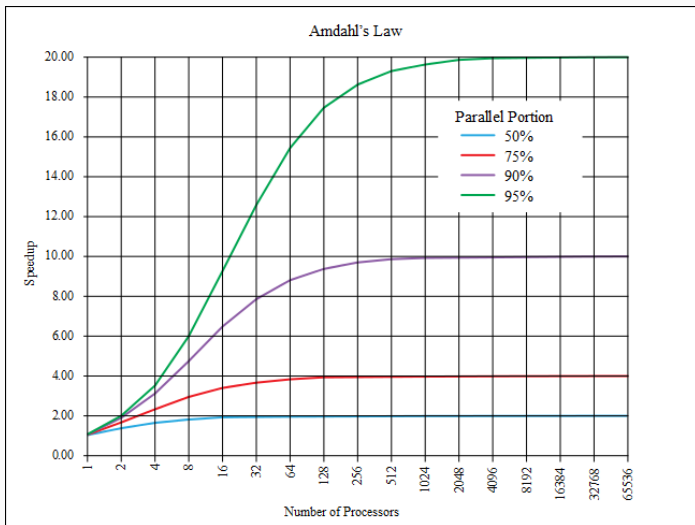


Figura 2: Lei de Amdahl: Aceleração x Paralelismo

Resumo

- A tecnologia de computadores é impulsionada por aperfeiçoamento na tecnologia de fabricação de semicondutores;
- Existem várias formas de medir o desempenho de computadores e os **benchmark** tendem a ser os mais efetivos;
- É importante compreender como uma dada unidade de desempenho é gerado, de modo a entender qual é a sua utilidade para prever o desempenho de um sistema, em uma determinada aplicação.
- O impacto de uma mudança em uma arquitetura tem sobre o desempenho geral depende não apenas de quanto esta mudança melhora o desempenho quando ela é utilizada, mas com qual frequência esta modificação é útil.

Referências

- Carter, N. Teoria e Problemas de Arquitetura de computadores. Coleção Schaum. Tradução Ralph Miller Jr. Porto Alegre: Bookman, 2003.
- Wikipédia. Lei de Moore.
https://pt.wikipedia.org/wiki/Lei_de_Moore