Arquitetura de Alto desempenho

Abstrações e tecnologia de computador

António Rui Borges

Resumo

• Sociedade da informação – Destaques da revolução da informação • Arquitetura de computadores versus organização de computadores • Estrutura e função • Perspectiva histórica • Classes de sistemas de computadores • Classes de paralelismo • Lei de Amdahl • Princípios quantitativos de projeto de computadores • Potência e energia em circuitos integrados • Confiabilidade • Medindo o desempenho • A equação de desempenho do processador • Leitura sugerida

Sociedade da informação - 1

A presença de sistemas computacionais na sociedade atual é onipresente.

Eles constituem o alicerce básico da *Internet* – a infra-estrutura de rede de comunicações que liga computadores em todo o mundo, dando origem ao conceito de *aldeia global* através do qual pessoas em qualquer lugar do planeta comunicam com pessoas em qualquer outro lugar, enviando e/ou recebendo mensagens a quase qualquer momento. tempo real.

Eles também são parte integrante dos smartphones e tablets que possuímos e representam os laptops e desktops que usamos em casa ou no local de trabalho. Através deles, é possível trabalhar em casa, acessar uma infinidade de serviços como participação em videoconferência, serviços bancários e contábeis, compras online, pagamento de contas remotamente e contato com órgãos governamentais.

A indústria beneficiou enormemente da sua utilização. A automação das linhas de produção melhorou a eficiência dos processos de fabricação e baixou os preços dos produtos. Ao incorporá-los, os próprios produtos tornaram-se mais adequados às necessidades e mais versáteis.

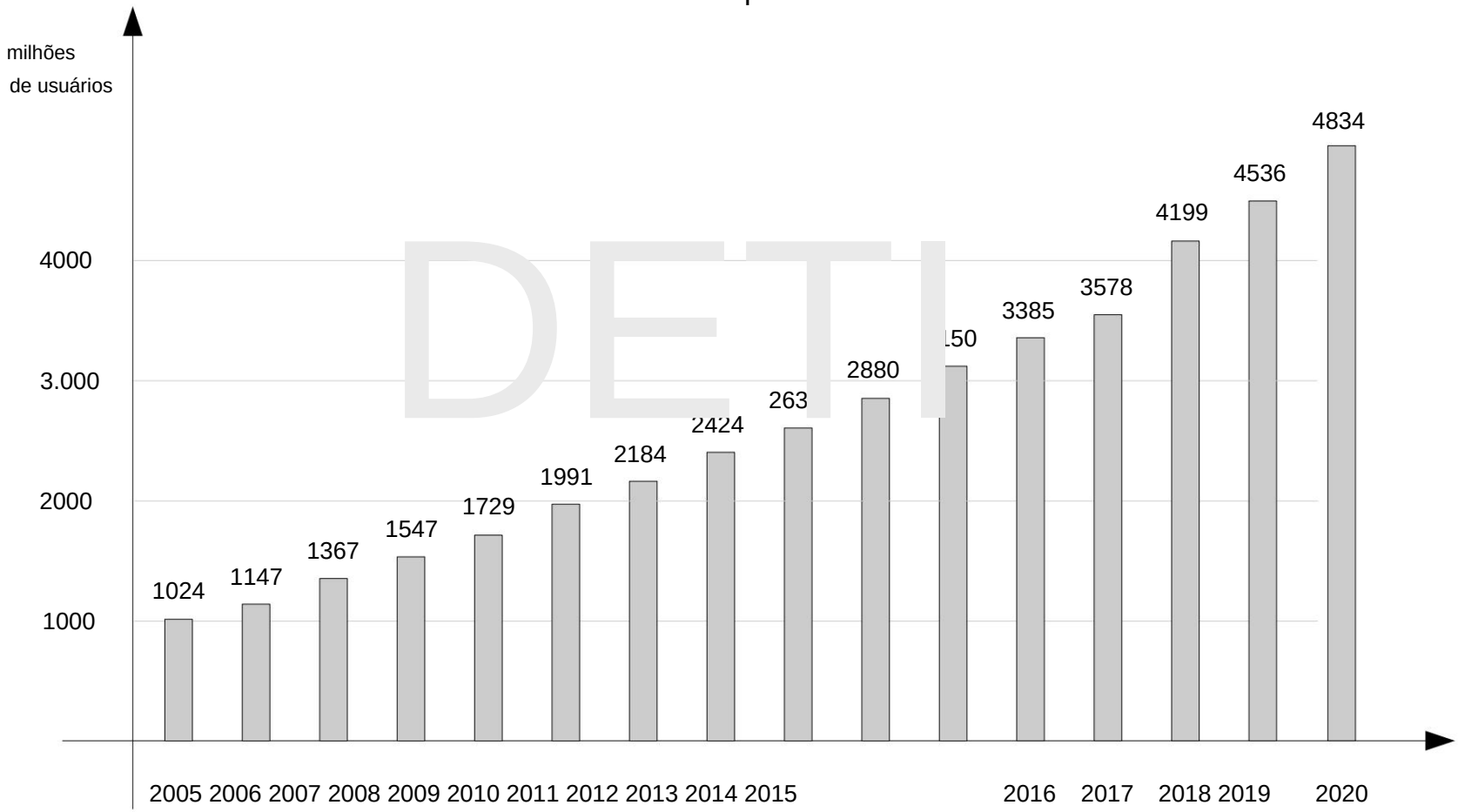
Sociedade da informação - 2

Os sistemas informáticos introduziram mudanças profundas na sociedade e o ritmo das mudanças está a acelerar. Por exemplo, prevê-se para os próximos anos a possibilidade de eliminar completamente a necessidade de dinheiro físico (moedas e notas) e substituí-lo por transacções puramente electrónicas, e de construir automóveis que se possam conduzir sozinhos por razões de segurança e conforto.

O seu impacto na vida quotidiana é tal que se diz frequentemente que a humanidade está actualmente a suportar outra revolução civilizacional – a *revolução da informação,* em comparação com a *revolução agrícola* que liderou a transição da caça e da recolha para a agricultura estabelecida há cerca de 12.000 anos, e a *revolução industrial* que deu origem a novos processos de produção mais eficientes na segunda metade do século XVIII.

Destaques da revolução da informação - 1

Usuários da Internet em todo o mundo Fonte: http://www.statista.com



Destaques da revolução da informação - 2

Usuários da Internet em todo o mundo (junho de 2022)

Fonte: http://internetworldstats.com

Regiões mundiais	*População (est.)*	*Usuários de Internet*	*Frac. População*
África	1 394 588 547	652 865 628	46,8%
Ásia	4 35 16 960	2 934 86 678	67,4%
Europa	83 47 045	750 45 495	89,6%
Portugal	**10 148 962**	**8 841 100**	**87,1%**
Médio Oriente	268 302 801	211 796 760	78,9%
América do Norte	374 226 482	349 572 583	93,4%
América Latina/Caribe	664 099 841	543 396 621	81,8%
Oceania / Austrália	43 602 955	31 191 971	71,5%
Total mundial	*7 934 462 631*	*5 473 055 736*	*69,0%*

Destaques da revolução da informação - 3

Número de usuários ativos mensais do Facebook em todo o mundo Fonte: http://www.statista.com

Destaques da revolução da informação - 4

Tipo de acesso à internet

Fonte: http://gs.statcounter.com

Destaques da revolução da informação - 5

Descrição do domínio ECU

Fonte: *A Europa está no comando? A Competitividade da Indústria Europeia de Sistemas Embarcados Automotivos,* Juliussen E. e Robinsonson R., Instituto de Estudos Tecnológicos Prospectivos, 2010, EUR 24601 EN

Destaques da revolução da informação - 6

Arquitetura do controlador de domínio (de 2010 em diante)

Fonte: *A Europa está no comando? A Competitividade da Indústria Europeia de Sistemas Embarcados Automotivos,* Juliussen E. e Robinsonson R., Instituto de Estudos Tecnológicos Prospectivos, 2010, EUR 24601 EN

Destaques da revolução da informação - 7

Evolução do microcontrolador (2000 a 2020)

Fonte: *A Europa está no comando? A Competitividade da Indústria Europeia de Sistemas Embarcados Automotivos,* Juliussen E. e Robinsonson R., Instituto de Estudos Tecnológicos Prospectivos, 2010, EUR 24601 EN

Arquitetura de computadores vs. organização de computadores - 1

Arquitetura de computador - refere-se aos atributos de um sistema de computador que são visíveis ao programador, ou seja, aos atributos que têm impacto direto na execução lógica de um programa.

Os atributos arquitetônicos incluem o conjunto de instruções, o número e o tamanho dos registros internos do processador, o formato dos diferentes tipos de dados, os modos de endereçamento da memória e os mecanismos de E/S.

Organização informática – refere-se ao papel das unidades operacionais internas e à as maneiras como eles se interconectam para implementar a especificação arquitetônica.

Os atributos organizacionais incluem os detalhes de hardware que são transparentes para o programador, como sinais de controle, interfaces entre processador e memória e entre o sistema de computador e dispositivos de E/S (periféricos) e a tecnologia de memória utilizada.

Arquitetura de computadores vs. organização de computadores - 2

A distinção entre arquitetura e organização foi, e ainda é, muito importante. Muitos fabricantes de computadores oferecem uma família de modelos de computadores, todos com a mesma arquitetura, mas com diferenças de organização, ou seja, representando diferentes implementações da mesma arquitetura, de modo que cada modelo possui características de preço e desempenho diferentes. Assim, uma arquitetura específica pode abranger muitos anos e abranger vários modelos de computadores diferentes, e sua organização muda com a mudança da tecnologia.

Um exemplo notável nesse sentido é a arquitetura IBM System 370, que foi introduzida pela primeira vez em 1970 e incluía vários modelos diferentes. O cliente com requisitos modestos poderia começar comprando um modelo mais barato e mais lento e, se a demanda aumentasse, posteriormente atualizar para um modelo mais caro e mais rápido, sem ter que abandonar o software já desenvolvido. Ao longo dos anos, a IBM produziu muitos modelos novos com tecnologia aprimorada para substituir os modelos mais antigos, oferecendo ao cliente maior velocidade, menor custo ou ambos. Esses modelos mais novos mantiveram a mesma arquitetura, protegendo assim o investimento em software do cliente. Notavelmente, a arquitetura System 370, com algumas melhorias, sobreviveu até hoje como a arquitetura da linha de produtos de mainframes da IBM.

Arquitetura de computadores vs. organização de computadores - 3

Mainframe z13 da IBM (Augusto Menezes/Feature Photo Service/IBM)

O z13 foi projetado para ser capaz de criptografia em tempo real para transações móveis, com processamento analítico em tempo real para identificar tendências e ajudar a detectar fraudes, enquanto o sistema é capaz de ser dimensionado para processar impressionantes 2,5 bilhões de transações por dia, de acordo com a IBM. (Fonte: http://www.v3.co.uk)

Arquitetura de computadores vs. organização de computadores - 4

Para microcomputadores, a relação entre arquitetura e organização é muito estreita. As mudanças na tecnologia não apenas influenciam a organização, mas também resultam na introdução de arquiteturas mais poderosas e complexas. Geralmente, há menos requisitos de compatibilidade geração a geração para essas máquinas menores.

Estrutura e função - 1

Um sistema de computador é um sistema muito complexo. O ponto chave para projetar e/ou descrever um sistema complexo é usar *a abstração* para expressar o sistema, ou uma de suas partes, em diferentes níveis de representação de maneira hierárquica. Num determinado nível, os detalhes dos níveis inferiores ficam ocultos para que a imagem apresentada se baseie no conceito do que *se precisa saber.* Ao concentrar-se no que é relevante em cada momento, menos detalhes têm de ser considerados e inter-relacionados e torna-se mais produtivo na concepção e/ou obtém-se uma imagem mais clara e uma melhor compreensão do que está descrito.

Em cada nível, é preciso preocupar-se com a estrutura e a função

- ***estrutura*** – quantos componentes existem e como eles são interligados
- ***função*** – a operação de cada componente individual como parte do todo.

Estrutura e função - 2

As funções básicas de um sistema de computador são: *entrada de dados, saída de dados, processamento de dados, armazenamento de dados* e *controle.*

O sistema informático deve ser capaz de *processar dados.* Os dados assumem uma ampla variedade de formas e a gama de requisitos de processamento é ampla. Existem, no entanto, apenas alguns métodos ou tipos fundamentais de processamento de dados.

O sistema informático também deve ser capaz de *transferir dados* entre si e o mundo exterior. O ambiente operacional consiste em dispositivos que servem como fontes ou destinos de dados. Quando os dados são recebidos ou entregues a um dispositivo que está diretamente conectado a um sistema de computador, o processo de fazer isso é conhecido como execução de *entrada-saída* (E/S) e o dispositivo é chamado de *periférico.* Quando os dados são transferidos por distâncias maiores, de ou para um dispositivo remoto, o processo é conhecido como *comunicação de dados.*

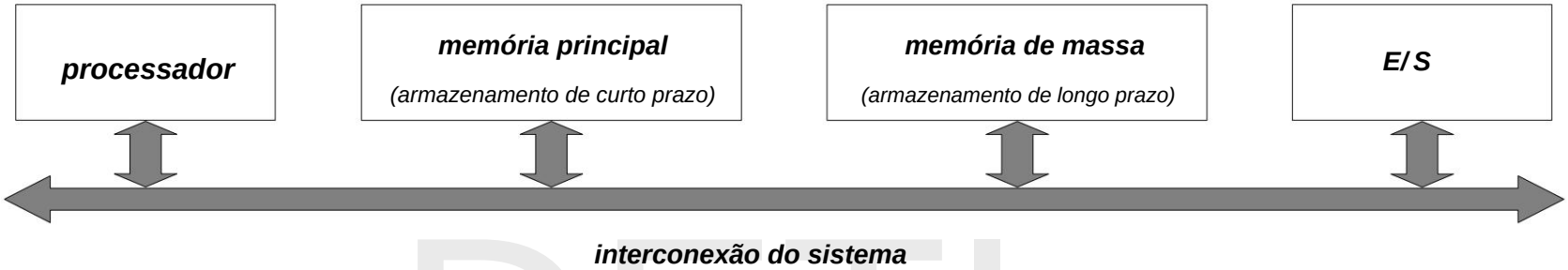
Estrutura e função - 3

É essencial que o sistema informático também *armazene dados .* Mesmo que o computador esteja processando dados em tempo real (ou seja, os dados são inseridos, processados e os resultados são imediatamente exibidos), o sistema de computador deve armazenar temporariamente pelo menos os dados que estão sendo trabalhados em um determinado momento. Assim, existe uma função *de armazenamento de dados de curto prazo .* Igualmente importante, o sistema informático requer uma função *de armazenamento de dados a longo prazo* para que os ficheiros de dados sejam armazenados e subsequentemente recuperados e actualizados.

Finalmente, estas funções devem ser *controladas* para realizar algum trabalho útil. Este controle é exercido em última análise pelo indivíduo *(programador)* que fornece *instruções ao sistema de computador,* mas dentro do próprio sistema de computador uma *unidade de controle* deve gerenciar os recursos disponíveis e orquestrar o desempenho de suas partes funcionais em resposta a essas instruções.

Estrutura e função - 4

No nível superior, a estrutura de um sistema de computador pode ser percebida como



- ***processador*** (ou ***unidade central de processamento*** – CPU) – controla a operação do computador e executa suas funções de processamento de dados
- ***memória principal*** – armazena dados durante o processamento; tem uma natureza *volátil*
- ***memória de massa*** – armazena dados entre execuções de processamento e permite que grandes quantidades de dados sejam recuperadas e eventualmente atualizadas durante o processamento; tem uma natureza *não volátil* e é percebido como um dispositivo de E/S especial
- ***E/S*** – move dados entre o sistema de computador e seu ambiente externo
- ***interconexão de sistemas*** – garante que a transferência de dados ocorra entre os componentes; geralmente é implementado como um *barramento.*

Estrutura e função - 5

Para que o sistema informático execute uma tarefa específica, é necessário fornecer um conjunto de instruções que, em conjunto, constituem o *programa* a ser executado. A ideia principal aqui é como representar instruções?

Suponhamos que eles pudessem ser representados de uma forma adequada para serem armazenados na memória principal junto com os dados, constituindo assim um tipo *especial* de dados em algum sentido abstrato. Então, um sistema de computador poderia obter suas instruções lendo-as da memória principal e um programa poderia ser configurado ou alterado definindo os valores daquela parte da memória.

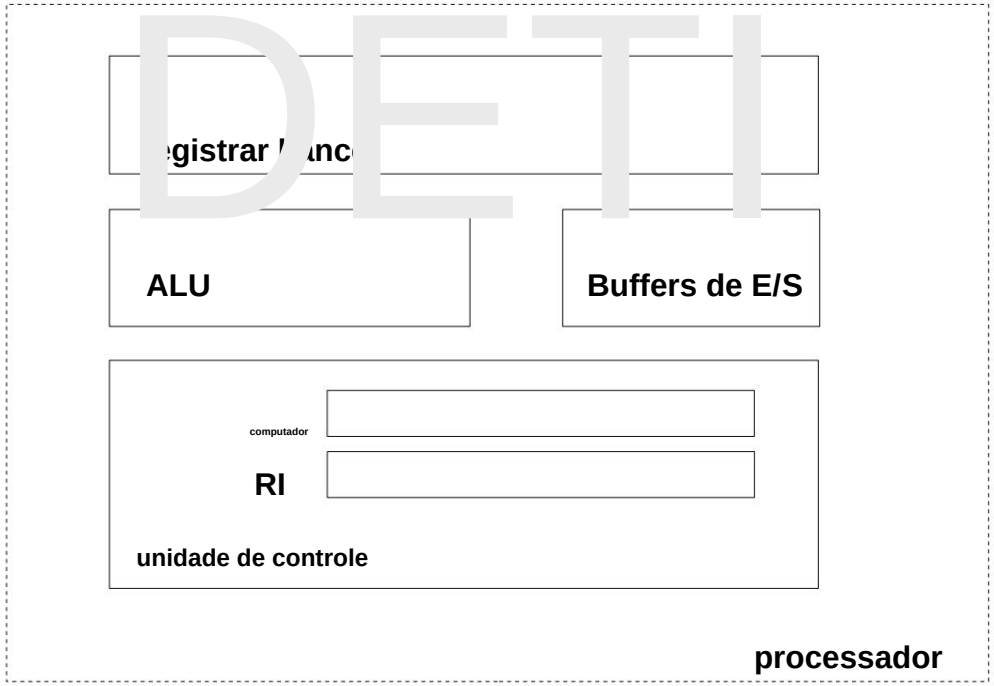
Esta ideia, desenvolvida independentemente por John von Neumann e Alan Turing, ficou conhecida como *conceito de programa armazenado* e foi universalmente adotada por projetistas de computadores. Esta é a razão pela qual os sistemas de computador são às vezes chamados de *máquinas de von Neumann.*

Estrutura e função - 6

Nesse sentido, embora um sistema computacional seja um sistema digital muito complexo, ele pode ser percebido como um sistema que alterna continuamente entre dois estados internos básicos: *busca de instrução,* onde o processador acessa a memória para obter a próxima instrução, e *execução de instrução,* onde o processador decodifica a instrução recém-recuperada e executa a operação correspondente.

Estrutura e função - 7

De forma simplificada, o próprio processador pode ser visto como constituído por uma *unidade de controle,* que cuida principalmente das fases *de busca* e *decodificação das instruções ;* de uma *unidade aritmética/ lógica ,* que realiza as operações prescritas; de um *banco de registro,* que armazena dados temporários; e de *buffers de E/ S,* que permitem a comunicação com os demais componentes do sistema computacional.



Estrutura e função - 8

O *conjunto de instruções* de qualquer processador sempre compreende instruções do tipo

- *movimentação de dados* – transferência de dados entre algum registro do banco de registro e memória principal ou algum controlador de E/S
- *aritmética / lógica* – instruções aritméticas (adicionar, subtrair, multiplicar, dividir), em formato de ponto fixo ou flutuante, instruções lógicas (não, e, ou, x-ou) e conteúdo de registro de deslocamento/rotação
- *ramificação* – modificando a sequencialidade estrita da execução da instrução, seja incondicionalmente ou dependente de alguma condição
- *chamada de subrotina* – execução de subprogramas (segmentos de código autônomos dentro de todo o grupo de instruções fornecidas), seja incondicionalmente ou dependente de alguma condição.

Perspectiva histórica - 1

O desempenho dos sistemas de computador teve um progresso incrível desde a época em que o primeiro computador eletrônico de uso geral foi construído. Este rápido ritmo de melhoria deveu-se não apenas aos avanços na tecnologia de circuitos integrados, mas também aos avanços no design de computadores.

No lado tecnológico, Gordon Moore, cofundador da Fairchild Semiconductor Corporation e da Intel, observou em 1965 que o número de transistores que poderiam ser colocados em um único chip dobrava a cada ano e previu corretamente que esse ritmo continuaria no futuro próximo. . Esta observação ficou conhecida como *Lei de Moore.*

O ritmo continuou ano após ano e década após década desde então. Desacelerou um pouco para duplicar a cada dois anos na década de 1970 e, mais recentemente, em 2015, a Intel anunciou que o ritmo está actualmente a duplicar a cada dois anos e meio.

Perspectiva histórica - 2

Crescimento na contagem de transistores em circuitos integrados

Fonte: Organização e Arquitetura de Computadores: Projetando para Desempenho

Perspectiva histórica - 3

Densidade do transistor lógico

Fonte: Tecnologia de 10 nm da Intel (Intel News Fact Sheet)

Perspectiva histórica - 4

Em meados da década de 1970 viu o surgimento do microprocessador. A capacidade do microprocessador de aproveitar as melhorias na tecnologia de circuitos integrados e a vantagem de custo resultante de um microprocessador produzido em massa levaram a que uma fração cada vez maior dos sistemas de computador fossem baseados em microprocessadores. Além disso, duas mudanças significativas no mercado de computadores tornam mais fácil do que nunca o sucesso comercial com uma nova arquitetura

- a eliminação virtual da programação em linguagem assembly reduziu a necessidade de compatibilidade de código objeto
- a criação de sistemas operacionais padronizados e independentes de fornecedores, como o Unix e, mais tarde, seu clone Linux, reduziu o custo e o risco de lançar novo hardware de processador.

Essas mudanças permitiram o desenvolvimento bem-sucedido de um novo conjunto de arquiteturas baseadas em instruções mais simples, conhecidas como RISC (Reduced Instruction Set Computers), no início da década de 1980. Os processadores RISC focaram a atenção dos projetistas em duas técnicas de desempenho: a exploração do *paralelismo em nível de instrução* (primeiro, usando pipelining e, depois, usando emissão de múltiplas instruções) e a aplicação sistemática de caches para acelerar o acesso do processador a instruções e dados. (primeiro, de forma simples e, posteriormente, utilizando organizações e otimizações r

Perspectiva histórica - 5

O efeito da combinação dessas melhorias arquitetônicas e organizacionais foi quadruplicado

- melhorou significativamente o poder computacional disponibilizado aos usuários (os microprocessadores de maior desempenho da época superaram os supercomputadores construídos 10 anos antes)
- novas classes de sistemas de computador surgiram devido à melhoria drástica no custo-desempenho: computadores pessoais e estações de trabalho surgiram na década de 1980 e smartphones e tablets tornaram-se a principal plataforma de computação preferida de muitas pessoas na última década
- o progresso contínuo na fabricação de semicondutores, conforme previsto pela Lei de Moore, levou ao domínio de sistemas de computador baseados em microprocessadores em toda a gama de projetos de computadores
- trouxe um impacto profundo no desenvolvimento de software: primeiro, ao permitir que os programadores trocassem desempenho por produtividade em muitas aplicações através do uso do paradigma orientado a objetos; segundo, o desempenho está sendo buscado através da substituição de interpretadores por compiladores just-in-time e compilação baseada em rastreamento para a abordagem tradicional de compilador e vinculador; terceiro, a atual popularidade do software como serviço (SaaS) na Internet para a execução de aplicativos.

Perspectiva histórica - 6

Desde 2003, a taxa de melhoria de desempenho do processador único diminuiu devido à barreira imposta pela quantidade máxima permitida de dissipação de energia de chips resfriados a ar e à falta de novas idéias para que o paralelismo em nível de instrução seja explorado de forma eficiente.

Em 2004, a Intel cancelou seus projetos de uniprocessadores de alto desempenho e juntou-se a outros fabricantes de chips ao declarar que o caminho para um desempenho superior seria através de chips multicore, em vez de uniprocessadores mais rápidos.

Isto significa que a ênfase agora é colocada não apenas no *paralelismo ao nível da instrução* (ILP), mas também em *paralelismo em nível de dados* (DLP), *paralelismo em nível de thread* (TLP) e *paralelismo em nível de solicitação* (RLP). Enquanto as técnicas de hardware e compilador exploram o ILP implicitamente sem a atenção do programador, o DLP, o TLP e o RLP são explicitamente paralelos e requerem a reestruturação das aplicações para serem explorados de forma eficiente.

Perspectiva histórica - 7

Crescimento no desempenho do processador desde o final da década de 1970

Fonte: Arquitetura de Computadores: Uma Abordagem Quantitativa

Aulas de sistemas computacionais - 1

As mudanças que estão ocorrendo no uso do computador levaram a cinco aplicações diferentes. mercados, cada um caracterizado por requisitos e tecnologias específicas

- *dispositivos móveis pessoais (PMD)* – compreendem um conjunto de dispositivos sem fio com interfaces de usuário multimídia, como smartphones e tablets; o custo é uma preocupação primordial, uma vez que são principalmente produtos de consumo; embora a ênfase na eficiência energética seja frequentemente impulsionada pela utilização de baterias, a necessidade de utilizar embalagens menos dispendiosas e a ausência de uma ventoinha para arrefecimento também limitam o consumo total de energia; os requisitos de energia, tamanho e peso levam ao uso de memória flash para armazenamento em massa; os aplicativos geralmente são baseados na web e orientados à mídia
- *computação desktop* – abrangem desde laptops ou notebooks de baixo custo até estações de trabalho de alto desempenho e altamente configuradas; o foco está na *otimização de preço-desempenho,* tanto em computação bruta quanto em gráficos; como resultado, os microprocessadores mais novos e de maior desempenho e os microprocessadores de custo reduzido geralmente aparecem primeiro em sistemas desktop; a computação tende a ser bem caracterizada em termos de aplicativos e benchmarking, embora o uso crescente de aplicativos interativos centrados na web represente novos desafios

- *servidores* – representam a espinha dorsal da computação empresarial em grande escala, substituindo o mainframe tradicional como fornecedor de serviços de ficheiros e computação em maior escala e mais fiáveis; surgiram na década de 1980, quando os terminais foram substituídos por desktops como interface principal para serviços centrais; tanto a disponibilidade como a escalabilidade são críticas, porque os servidores devem estar sempre activos e crescer em resposta a uma procura crescente pelos serviços que suportam ou a um aumento nos requisitos funcionais (portanto, a capacidade de aumentar a capacidade computacional, a memória principal, o armazenamento em massa e a largura de banda de E/S são cruciais); os servidores também são projetados para um rendimento eficiente – a capacidade de resposta às solicitações individuais continua importante, mas a eficiência geral e a relação custo-benefício, definidas pelo número de solicitações que podem ser tratadas por unidade de tempo, são as principais métricas a serem levadas em consideração

Aulas de sistemas computacionais - 3

- *clusters/ computadores em escala de armazém – clusters* são coleções de computadores desktop, ou servidores, conectados por redes locais para atuar como um único computador maior; cada nó de processamento executa seu próprio sistema operacional [de rede] e se comunica com os demais por meio de um protocolo de rede; os maiores clusters são chamados de *computadores em escala de armazém* (WSCs); preço-desempenho e potência são críticos; a disponibilidade também é crítica, mas ao contrário dos servidores onde é garantida por hardware informático integrado, as WSC utilizam componentes redundantes e baratos como blocos de construção, contando com uma camada de software para detectar e isolar as muitas falhas que ocorrem; o foco é colocado em aplicativos interativos, armazenamento em grande escala, confiabilidade e alta largura de banda da Internet – eles representam o alicerce do que hoje é chamado de *nuvem*

  supercomputadores e, em geral, *computação de alto desempenho* (HPC) estão relacionados a WSCs, mas diferem por enfatizar o desempenho de ponto flutuante e por executar grandes programas em lote com uso intensivo de comunicação, cuja execução leva semanas; redes internas muito rápidas são críticas aqui

- *computadores incorporados* – são encontrados em máquinas de uso diário, instrumentos “inteligentes” e produtos de consumo; estão relacionados aos computadores PMD, mas, ao contrário deles, não executam software desenvolvido externamente; o preço é o factor chave – existem requisitos de desempenho, mas o objectivo principal é satisfazer as necessidades de desempenho a um preço mínimo.

Aulas de sistemas computacionais - 4

Classes de sistemas computacionais e suas características sistêmicas

Fonte: Adaptado de Arquitetura de Computadores: Uma Abordagem Quantitativa

Recurso	***PMD***	***Área de Trabalho***	***Servidor***	***C/ WSC***	***Integrado***
Sistema *Preço*	US$ 100 - US$ 1.000	US$ 300 - US$ 2.500	US$ 5.000 - US$ 10.000.000	US$ 100.000 - US$ 200.000.000	US$ 10 - US$ 100.000
Microprocessador *Preço*	US$ 10 - US$ 100	US$ 50 - US$ 500	US$ 200 - US$ 2.000	US$ 50 - US$ 250	US$ 0,01 - US$ 100
Crítico *Projeto* *Problemas*	custo, energia, desempenho, capacidade de resposta	preço-desempenho, energia, desempenho gráfico	rendimento, disponibilidade, escalabilidade, energia	preço-desempenho, rendimento, energia	preço, energia, desempenho específico da aplicação

Classes de paralelismo - 1

O paralelismo em vários níveis é a força motriz predominante do projeto de computadores em todas as classes de computadores, sendo a energia e o custo as principais restrições. Existem basicamente dois tipos de paralelismo em aplicações

- *paralelismo em nível de dados* (DLP) – surge quando há vários itens de dados que podem ser processado ao mesmo tempo
- *paralelismo em nível de tarefa* (TLP) – surge quando a tarefa a ser executada pode ser dividida em subtarefas que operam [illegible] independentemente.

O hardware do computador, por sua vez, pode explorar esses dois tipos de paralelismo de aplicações em quatro maneiras diferentes

- *paralelismo em nível de instrução* – o paralelismo em nível de dados é explorado com a ajuda do compilador usando ideias como pipeline, execução especulativa e problemas múltiplos
- *através de hardware especial* – arquiteturas vetoriais e unidades de processamento gráfico (GPUs) exploram o paralelismo em nível de dados aplicando a mesma instrução a uma coleção de dados em paralelo

- *Paralelismo em nível de thread* – o paralelismo em nível de dados e em nível de tarefa pode ser explorado em um modelo fortemente acoplado que permite a interação entre threads simultâneos
- *paralelismo em nível de solicitação* – o paralelismo é explorado entre tarefas essencialmente fracamente acopladas especificadas pelo programador ou pelo sistema operacional.

Classes de paralelismo - 2

Na década de 1960, Michael Flynn estudou os esforços de computação paralela feitos até agora e encontrou uma classificação que ainda hoje é popular. Ele observou o paralelismo nas instruções e nos fluxos de dados exigidos pelas instruções no componente mais restrito do multiprocessador e colocou todos os computadores em uma das quatro categorias

- *instrução única – single data streams* (SISD) – corresponde ao uniprocessador; no entanto, o paralelismo de nível de instrução ainda pode ser explorado
- *instrução única – múltiplos fluxos de dados* (SIMD) – a mesma instrução é executada por múltiplas unidades de processamento usando diferentes fluxos de dados; esta categoria compreende arquiteturas vetoriais, extensões multimédia para conjuntos de instruções padrão e GPUs
- *múltiplas instruções – fluxos de dados únicos* (MISD) – múltiplas instruções são executadas no mesmo dado; nenhum multiprocessador comercial deste tipo foi construído ainda (embora se o *dado* for considerado como representando um *vetor de dados,* então *as matrizes sistólicas* podem ser consideradas nesta categoria)
- *instrução múltipla – múltiplos fluxos de dados* (MIMD) – visa o paralelismo em nível de tarefa, onde cada processador executa seu próprio programa com seus próprios dados; o paralelismo em nível de dados também pode ser explorado, embora a sobrecarga seja provavelmente maior do que no SIMD; multicores de processador se enquadram nesta categoria.

Lei de Amdahl - 1

tempo de execução para toda a tarefa sem usar a melhoria

tempo de execução para toda a tarefa usando a melhoria

Lei de Amdahl - 2

O ganho de desempenho que pode ser obtido melhorando alguma característica de um sistema computacional pode ser estimado pela *Lei de Amdahl.* Amdahl afirmou em 1967 que *a velocidade a ser obtida com a adoção de algum modo de execução mais rápido é limitada pela fração de tempo de todas as operações que não podem ser melhoradas.* e é expresso pela fórmula

$$\text{aceleração geral} = \frac{\text{tempo de execução para toda a tarefa sem usar a melhoria}}{\text{tempo de execução para toda a tarefa usando a melhoria}} =$$

$$= \frac{1}{(1 - \text{fracenhanc}) + \frac{\text{fracenhanc}}{\text{speedupenhanc}}},$$

onde *fracenhanc* é a fração de tempo no sistema de computador original que pode ser convertida para aproveitar o modo de execução mais rápido e speedupenhanc é a velocidade a ser obtida localmente pela adoção do modo de execução mais rápido.

Lei de Amdahl - 3

O processador usado em um servidor web deve ser alterado para agilizar as operações. O novo processador é 10 vezes mais rápido que o original. Supondo que atualmente o processador esteja ocupado 40% do tempo e aguardando E/S 60% do tempo, qual será a aceleração geral obtida se a substituição ocorrer?

fracenhanc = 0,4 ÿ speedupenhanc = 10 ÿ

$$\text{ÿ aceleração geral} = \frac{1}{0,6 + \frac{0,4}{10}} = \frac{1}{0,64} = 1,56.$$

Assim, a Lei de Amdahl é uma lei dos rendimentos decrescentes!

$$\lim_{\text{speedupenhanc ÿ} \to +\text{ÿ}} \text{speedupoverall} = \frac{1}{1 \text{ ÿ fracenhanc}} .$$

Lei de Amdahl - 4

A Lei de Amdahl é particularmente útil na comparação do desempenho geral do sistema presença de diferentes alternativas.

Uma operação comum necessária em processadores gráficos é *a raiz quadrada.* Suponha que a raiz quadrada de ponto flutuante (FPSQRT) seja responsável por 20% do tempo de execução de um benchmark gráfico crítico. Uma proposta é aprimorar o hardware do FPSQRT e acelerar em 10 vezes essa operação. A outra alternativa é fazer com que todas as operações de ponto flutuante sejam executadas mais rapidamente por um fator de 1,6. Observe que as operações de ponto flutuante representam metade do tempo de execução da aplicação.
Compare essas duas alternativas de design.

$$\text{acelerarFPSQRT} = \frac{1}{0{,}8 + \frac{0{,}2}{10}} = \frac{1}{0{,}82} = 1{,}22$$

$$\text{acelerarFP} = \frac{1}{0{,}5 + \frac{0{,}5}{1.6}} = \frac{1}{0{,}81} = 1{,}23.$$

Princípios quantitativos de design de computador

Algumas diretrizes são úteis no projeto de sistemas de computador

- *tirar vantagem do paralelismo* – o paralelismo é um dos métodos mais importantes para melhorar o desempenho; pode-se recorrer à redundância para aumentar *a confiabilidade* e/ou às vezes tornar as operações mais rápidas simplesmente adicionando mais recursos – *escalabilidade,* ou obtendo a *simultaneidade subjacente;* isso pode ser feito em vários níveis de abstração: sistema, processador individual ou design digital de baixo nível

- *aproveitar o princípio da localidade* – os programas tendem a reutilizar instruções e dados que eles usaram recentemente

- *concentrar-se no caso comum* – ao fazer uma compensação no design, favoreça o caso frequente em detrimento do pouco frequente; o caso frequente costuma ser mais simples de otimizar e pode ser mais gratificante; funciona bem não apenas em termos de desempenho, mas também em termos de alocação de recursos e energia.

Potência e energia em circuitos integrados - 1

O poder é o maior desafio enfrentado hoje pelo projetista de computadores. Primeiro, a energia deve ser trazida e distribuída ao redor do chip e os microprocessadores modernos usam centenas de pinos e múltiplas camadas de interconexão apenas para alimentação e aterramento. Segundo, a energia é dissipada como calor e deve ser removida.

Tanto a frequência quanto a potência aumentaram rapidamente durante décadas e depois se estabilizaram recentemente. A razão é que os projetistas atingiram o limite prático de potência para resfriamento de microprocessadores.

Taxa de clock e potência para microprocessadores Intel x86 ao longo de oito gerações

Fonte: Organização e Design de Computadores: A Interface Hardware/Software

Potência e energia em circuitos integrados - 2

Para chips CMOS, o consumo tradicional de energia tem sido na comutação de transistores, normalmente chamado de *energia dinâmica.* A energia necessária por transistor é proporcional ao produto da carga capacitiva acionada pelo transistor e ao quadrado da tensão

ener mico ÿ carga capacitiva ÿtensão ^{2} .

A carga capacitiva é função da quantidade de transistores conectados a uma saída e da tecnologia, que por si só determina a capacitância dos fios e dos transistores.

Por outro lado, a *potência dinâmica* necessária por transistor é o produto de a energia de uma transição multiplicada pela frequência de transições

ÿ carga capacitiva ÿtensão powerdynamic ^{2} ÿ frequência de comutação.

Potência e energia em circuitos integrados - 3

A potência dinâmica e a energia são bastante reduzidas com a redução da tensão, de modo que as tensões caíram de 5 V para pouco menos de 1 V em 20 anos para permitir o aumento da frequência sem afetar muito a potência. O problema hoje é que uma redução adicional da tensão parece tornar os transistores com muito *vazamento.*

Além do consumo dinâmico de energia, hoje em dia deve-se considerar também o consumo *de energia estática* devido à corrente de fuga que flui mesmo quando o transistor está desligado. Em servidores, por exemplo, a corrente de fuga é normalmente responsável por 40% do consumo total de energia. Assim, aumentar o número de transistores aumenta a necessidade de mais dissipação de potência mesmo quando todos os transistores estão desligados. Os chips de servidor podem consumir mais de 100 W e o resfriamento do chip e do sistema circundante é uma despesa importante em computadores em escala de armazém.

Potência e energia em circuitos integrados - 4

Os microprocessadores modernos oferecem diversas técnicas para melhorar a eficiência energética apesar das taxas de clock planas e das tensões de alimentação constantes

- *acompanhar as operações* – a maioria dos microprocessadores atuais desliga o relógio dos módulos inativos para economizar energia e potência dinâmica
- *escalonamento dinâmico de tensão-frequência* (DVFS) – a maioria dos microprocessadores hoje oferece algumas frequências de clock e tensões nas quais operar para menor consumo de energia e consumo de energia
- *design para casos típicos* – os microprocessadores a serem usados em computação desktop são projetados para um caso típico de uso pesado em altas temperaturas operacionais, contando com sensores de temperatura no chip para detectar automaticamente quando a atividade deve ser reduzida para evitar superaquecimento
- *overclocking* – a Intel começou a oferecer *o modo Turbo* em 2008, onde o chip decide por si mesmo se é seguro operar em uma frequência de clock mais alta, normalmente 10% acima da taxa de clock nominal, por um curto período de tempo, possivelmente em apenas um alguns núcleos, até que a temperatura comece a subir.

Confiabilidade - 1

Historicamente, os circuitos integrados foram um dos componentes mais confiáveis de um sistema de computador. Essa sabedoria convencional, no entanto, começou a mudar à medida que os tamanhos dos transistores se tornaram menores que 32 nm. Tanto as falhas transitórias como as permanentes estão se tornando mais comuns, por isso os arquitetos de computadores devem projetar sistemas para lidar com esses desafios.

A confiabilidade do módulo é uma medida da operação contínua do módulo conforme especificado ou, dito de outra forma, é uma medida do tempo que um módulo leva para falhar, contado a partir de um instante inicial de referência. Portanto, o *tempo médio até a falha* (MTTF) é uma medida de confiabilidade. A recíproca do MTTF é a taxa de falhas, geralmente relatada como falhas por bilhão de horas de operação, ou *falhas no tempo* (FIT).
A interrupção do serviço é medida pelo *tempo médio para reparo* (MTTR). *O tempo médio entre falhas* (MTBF) é apenas a soma do MTTF e do MTTR.

Se uma coleção de módulos tiver tempos de vida distribuídos exponencialmente – o que significa que a idade de um módulo não é relevante para sua probabilidade de falhar, e suas falhas são independentes, então a taxa de falha global da coleção é a soma das taxas de falha do indivíduo módulos.

Confiabilidade - 2

A disponibilidade do módulo é uma medida da operação contínua do módulo conforme especificado em relação à alternância entre serviço e interrupção.

Para sistemas não redundantes com reparo, *a disponibilidade do módulo* é dada por

$$\text{disponibilidade do módulo} = \frac{\text{MTTF}}{\text{MTTF} + \text{MTTR}} .$$

Confiabilidade - 3

Suponha um subsistema de disco com os seguintes componentes e MTTFs

- 10 discos, cada um com classificação MTTF de 1.000.000 horas
- 1 controlador ATA com classificação MTTF de 500.000 horas
- 1 fonte de alimentação com MTTF de 200.000 horas
- 1 ventoinha com MTTF de 200.000 horas
- 1 cabo ATA com capacidade MTTF de 1.000.000 horas .

Usando as suposições simplificadoras sobre as distribuições de probabilidade dos tempos de vida e a independência das falhas, calcule o MTTF de todo o subsistema.

$$\text{taxa de falhasubsys} = 10\ddot{y}\ \frac{1}{1.000.000} + \frac{1}{500.000} + \frac{1}{200.000} + \frac{1}{200.000} + \frac{1}{1.000.000} =$$

$$= \frac{23}{1 \text{ milhão de horas}} .$$

$$\text{MTTFsubsys} = \frac{1}{\text{taxa de falhasubsys}} = \frac{1.000.000}{23}\ \ddot{y}\ 43.500 \text{ horas.}$$

Confiabilidade - 4

Os subsistemas de disco geralmente possuem fontes de alimentação redundantes para melhorar a confiabilidade. Suponha a fonte de alimentação do último exemplo, com MTTR de 24 horas, e calcule a confiabilidade de um sistema redundante com duas fontes de alimentação.

$$\text{Par de fontes de alimentação MTTF} = \frac{\text{Fonte de alimentação MTTF}}{\frac{\text{Fonte de alimentação MTTR}}{\text{Fonte de alimentação MTTF}}} = \frac{\text{Fonte de alimentação MTTF}^2}{2 \text{ÿ} \text{MTTR}_{\text{fonte de alimentação}}} =$$

$$= \frac{200.000^2}{2 \text{ÿ} 24} \text{ ÿ } 830.000.000 \text{ horas.}$$

Confiabilidade - 5

A Lei de Amdahl também é aplicável além do desempenho.

Considere o exemplo de confiabilidade discutido anteriormente e suponha que a confiabilidade da fonte de alimentação melhorou do MTTF anterior de 200.000 horas para o impressionante MTTF de 830.000.000 horas, ou seja, uma melhoria de 4.150 vezes. Como isso afeta a confiabilidade do sistema geral?

$$\text{taxa de falhaorig sys} = 10ÿ \frac{1}{1.000.000} + \frac{1}{500.000} + \frac{1}{200.000} + \frac{1}{200.000} + \frac{1}{1.000.000} =$$

$$= \frac{23}{1 \text{ milhão de horas}}.$$

Portanto, a fração da taxa de falhas devido à fonte de alimentação é de 5 por 1.000.000 horas em 23 por 1.000.000 de horas, ou aproximadamente 22%. Portanto, tem-se

$$\text{confiabilidade melhorar fonte de alimentação} = \frac{1}{0{,}78 + \frac{0{,}22}{4150}} = 1{,}28.$$

Medindo o desempenho - 1

Ao comparar alternativas de design, muitas vezes queremos relacionar o *desempenho* de dois sistemas de computador diferentes, digamos X e Y. A *sentença X é mais rápida que Y* significa que o *tempo de execução* de algum programa executado em X é menor que o tempo de execução do mesmo programa executado em Y.

Mas o problema é como escolher um programa, ou um grupo de programas, comumente chamado de *benchmark,* ou *conjunto de benchmarks,* que produzirá resultados significativos em geral e que não dependa de características específicas de nenhuma das alternativas.

Além disso, deve-se considerar que as aplicações no mundo real executadas em sistemas computacionais são tão diferentes em tamanho e complexidade que a escolha de um sistema computacional para uma área específica não é uma tarefa fácil.

Medindo o desempenho - 2

Uma forma de proceder é executar programas que sejam mais simples que as aplicações reais, mas que ao mesmo tempo possam ser considerados representativos.

- *kernels* – porções pequenas e importantes de aplicativos reais
- *programas de brinquedo* – pequenos programas, geralmente com até 100 linhas de código, do tipo que os alunos são solicitados a escrever nas aulas de programação
- *código sintético* – programas falsos que foram escritos com a intenção de tentar corresponder ao comportamento de aplicativos reais.

Essa abordagem não é mais popular, principalmente porque é possível escrever compiladores que levam em conta o conhecimento de programas específicos, de modo que um sistema de computador específico pareça executar esses programas mais rapidamente do que quando realmente está executando aplicativos reais.

A abordagem preferida hoje em dia é considerar conjuntos de programas cuja representatividade é geralmente aceite por um público vasto e utilizá-los para caracterizar o desempenho relativo de dois sistemas informáticos, sendo um deles o computador de referência.

Medindo o desempenho - 3

A *Standard Performance Evaluation Corporation* (SPEC) é uma corporação sem fins lucrativos formada para estabelecer, manter e endossar um conjunto padronizado de benchmarks relevantes que podem ser aplicados à mais nova geração de computadores de alto desempenho. A SPEC desenvolve conjuntos de benchmark e também analisa e publica os resultados enviados por organizações membros e outros licenciados de benchmark.

SPECfp2000 (Sun Ultra 5 como computador de referência)

Fonte: Arquitetura de Computadores: Uma Abordagem Quantitativa

Referência	Ultra5 tempo(s) de execução	Opteron tempo(s) de execução	Razão ESPECÍFICA	Itânio 2 tempo(s) de execução	Razão ESPECÍFICA	Opteron/Itânio 22 tempo(s) de execução	Itânio 2 / Opteron Taxa de especificação
wupwise	1.600	51,5	31.06	56,1	28,53	0,92	0,92
nadar	3 100	125,0	24,73	70,7	43,85	1,77	1,77
mgrid	1.800	98,0	18h37	65,8	27.36	1,49	1,49
aplicativo	2 100	94,0	22h34	50,9	41,25	1,85	1,85
mesa	1.400	64,6	21.69	108,0	12,99	0,60	0,60
arte	2 900	86,4	33,57	40,0	72,47	2,16	2.16
galgel	2 600	92,4	28,13	21,0	123,67	4,40	4h40
igualar	1 300	72,6	17,92	36,3	35,78	2h00	2h00
facerec	1 900	73,6	25,80	86,9	21,86	0,85	0,85
ammp	2 200	136,0	16,14	132,0	16,63	1,03	1.03
lucas	2 000	88,8	22,52	107,0	18,76	0,83	0,83
fma3d	2 100	120,0	17h48	131,0	16.09	0,92	0,92
apsi de seis	1 100	123,0	8,95	68,8	15,99	1,79	1,79
trilhas	2 600	150,0	17h36	231,0	11.27	0,65	0,65
Geométrico significar			**20,86**		**27.12**	**13h30**	**13h30**

Medindo o desempenho - 4

$$\frac{\text{Média geométricaA}}{\text{Média geométricaB}} = \frac{\sqrt[n]{\prod_{i=1}^{n} \text{ESPECRatio Ai}}}{\sqrt[n]{\prod_{i=1}^{n} \text{ESPRatio Bi}}} = \sqrt[n]{\prod_{i=1}^{n} \frac{\text{ESPECRatio Ai}}{\text{ESPRatio Bi}}} =$$

$$= \sqrt[n]{\prod_{i=1}^{n} \frac{\text{tempo exec Ai}}{\text{tempo exec Ref}_i} \Big/ \frac{\text{tempo exec Bi}}{\text{tempo exec Ref}_i}} = \sqrt[n]{\prod_{i=1}^{n} \frac{\text{tempo executivo Bi}}{\text{tempo executivo Ai}}} =$$

$$= \sqrt[n]{\prod_{i=1}^{n} \frac{\text{Desempenho IA}}{\text{Desempenho Bi}}}$$

A equação de desempenho do processador - 1

O tempo de execução da CPU para executar um programa pode ser expresso por

Tempo de execução da CPU = ciclos de clock da CPUÿtempo de ciclo do clock ,

onde os *ciclos de clock variáveis da CPU* representam o número total de ciclos de clock para a execução do programa e o *tempo de ciclo de clock* variável é o período do clock da CPU.

Esta expressão pode ser expandida ainda mais para

Tempo de execução da CPU = contagem de instruções ÿCPIÿtempo de ciclo do relógio ,

onde a variável *contagem de instruções* representa o número total de instruções executadas pelo programa e pode ser obtida de maneira direta para organizações de processadores não sofisticadas se houver uma listagem da compilação do programa em linguagem assembly, e a variável CPI for *simplesmente* o número médio de ciclos de clock por instrução.

A equação de desempenho do processador - 2

O tempo de execução da CPU para executar um programa é, portanto, dependente de três características: contagem de instruções para a execução do programa, número médio de ciclos de clock por instrução e tempo de ciclo de clock. Além disso, *o tempo de execução da CPU* depende *igualmente* destes três parâmetros: uma determinada alteração percentual em um deles resultará na mesma alteração no *tempo de execução da CPU.*

Infelizmente, é difícil alterar um parâmetro em completo isolamento do outros, pois as tecnologias básicas envolvidas na mudança são interdependentes

- *contagem de instruções* – depende da arquitetura do computador e do tecnologia de compilador
- *ciclos de clock por instrução* – depende tanto da arquitetura do computador e a organização informática
- *tempo de ciclo de clock* – depende da tecnologia de hardware subjacente e na organização do computador.

A equação de desempenho do processador - 3

O tempo de execução da CPU para executar um programa ainda pode ser refinado se leva em consideração o *CPI* de cada tipo de instrução que está sendo executada

$$\text{Tempo de execução da CPU} = \text{ontagem de instruções}_{eu} \text{ ÿ} \left(\frac{\text{ontagem de instruções}_{eu}}{\text{contagem de instruções}} \text{ ÿIPC}_{eu} \right) \text{ÿ}$$

ÿtempo de ciclo do relógio.

A equação de desempenho do processador - 4

Suponha que as seguintes medições foram feitas ao executar um benchmark em um determinado sistema de computador

- frequência das operações de FP = 25%
- IPC médio para operações de PF = 4,00
- IPC médio para todas as outras operações = 1,33
- frequência de FPSQR = 2%
- IPC do FPSQR = 20

Suponha que alternativas de projeto estejam sendo consideradas para diminuir o IPC do FPSQR para 2 ou para diminuir o IPC médio de todas as operações de FP para 2,5. Use a equação de desempenho do processador para comparar essas alternativas.

Como o IPC é o único parâmetro que muda na equação de desempenho do processador, a resposta pode ser dada apenas calculando o IPC geral no caso original e para as duas alternativas.

A equação de desempenho do processador - 5

Situação original

$$\text{CPIorig geral} = \sum_{i} \frac{\text{contagem de instruções}_i}{\text{contagem de instruções}} \times \text{CPI}_i =$$

$$= 4{,}00\times 0{,}25 + 1{,}33\times 0{,}75 = 2{,}00$$

FPSQR aprimorado

$$\text{CPIenFPSQR geral} = \text{CPIorig geral} - 0{,}02\times(\text{CPIorigFPSQR} - \text{CPIenFPSQR}) = = 2{,}00 - 0{,}02\times(20 - 2) = 1{,}64$$

PF aprimorado

$$\text{IPC geral}_{\text{enFP}} = \sum_{i} \frac{\text{contagem de instruções}_i}{\text{contagem de instruções}} \times \text{CPI}_i =$$

$$= 2{,}50\times 0{,}25 + 1{,}33\times 0{,}75 = 1{,}63$$

Leitura sugerida

- *Arquitetura de Computadores: Uma Abordagem Quantitativa,* Hennessy JL, Patterson DA, 6ª Edição, Morgan Kaufmann, 2017
  - Capítulo 1: *Fundamentos de Design e Análise Quantitativa*
  - Apêndice M: *Perspectivas Históricas e Referências*
- *Organização e Arquitetura de Computadores: Projetando para Desempenho,* Stallings W., 10ª Edição, Pearson Education, 2016
  - Capítulo 1: *Conceitos Básicos e Evolução do Computador*
  - Capítulo 2: *Problemas de desempenho*