

Computer Abstractions and Technology – Powerpoint1

Sociedade da informação

A presença de sistemas informáticos na sociedade de hoje é onnipresente.

Eles constituem o elemento básico da Internet - a rede de comunicações infraestrutura que liga computadores em todo o mundo, dando origem ao conceito de global village através da qual as pessoas em qualquer parte do planeta comunicam com pessoas em qualquer outro lugar, enviando e/ou recebendo mensagens em tempo quase real.

São também parte integrante dos telefones inteligentes e comprimidos de que se é proprietário e de pé para os computadores portáteis e os computadores de secretária que se utilizam em casa ou no local de trabalho. Através deles, pode-se trabalhar a partir de casa, aceder a uma miríade de serviços como a participação num vídeo conferência, banca e contabilidade, compras on-line, pagamento de contas à distância e contactar agências governamentais.

A indústria tem beneficiado enormemente com a sua utilização. A automatização das linhas de produção tem melhorado a eficiência dos processos de fabrico e baixou os preços dos produtos. Ao incorporá-los, os próprios produtos tornaram-se mais adequados às suas necessidades e mais versátil.

Os sistemas informáticos introduziram alterações profundas na sociedade e a taxa da mudança está a acelerar. Por exemplo, está prevista para os próximos anos a possibilidade de eliminar completamente a necessidade de dinheiro físico (moedas e banco de notas) e a sua substituição por transações puramente eletrónicas, e de construção de automóveis que podem conduzir por si próprios por razões de segurança e conforto.

O seu impacto na vida quotidiana é tal que se diz frequentemente que a humanidade está atualmente a sofrer outra revolução civilizacional - a revolução da informação, em comparação com a revolução agrícola que liderou a transição de caça e recolha para a agricultura estabelecida há cerca de 12000 anos, e a revolução industrial que deu origem a novos processos de fabrico mais eficientes na segunda metade do século XVIII.

Arquitetura informática vs. organização informática

Arquitetura informática (Computer architecture) - refere-se aos atributos de um sistema informático que são visíveis para o programador, ou seja, para aqueles atributos que têm um impacto na execução lógica de um programa.

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

Organização informática (computer organization) - refere-se ao papel das unidades operacionais internas e às formas como estas se interligam para implementar a especificação arquitetónica.

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

A distinção entre arquitetura e organização tem sido, e ainda é, muito importante. Muitos fabricantes de computadores oferecem uma família de modelos de computadores, todos com a mesma arquitetura, mas com diferenças na organização, ou seja, representando diferentes implementações da mesma arquitetura, de modo que cada modelo tenha diferentes características de preço e desempenho. Assim, uma arquitetura particular pode abranger muitos anos e abrange uma série de diferentes modelos informáticos, a sua organização mudando com a mudança da tecnologia.

Um exemplo notável neste sentido é a arquitetura do Sistema IBM 370, que foi a primeira introduzido em 1970 e incluía uma série de modelos diferentes. Os clientes com requisitos modestos poderiam começar por comprar um modelo mais barato, mais lento e, se a procura aumentada, mais tarde, para um modelo mais caro e mais rápido sem ter de abandonar o software que já tinha sido desenvolvido. Ao longo dos anos, a IBM tem produzido muitos modelos novos com tecnologia melhorada para substituir os modelos mais antigos, oferecendo a maior rapidez do cliente, menor custo, ou ambos. Estes modelos mais recentes mantiveram os mesmos protegendo assim o investimento do cliente em software. Notavelmente, a arquitetura do Sistema 370, com algumas melhorias, tem sobrevivido até hoje como arquitetura da linha de produtos mainframes da IBM.

O z13 foi concebido para ser capaz de encriptar em tempo real para telemóveis transações, com processamento analítico em tempo real para identificar tendências e ajudar a detetar fraudes, enquanto o sistema é capaz de aumentar para processar um impressionante 2,5 mil milhões de transações por dia, de acordo com a IBM. (Fonte: <http://www.v3.co.uk>)

Para os microcomputadores, a relação entre arquitetura e organização é muito perto. As mudanças na tecnologia não só influenciam a organização como também resultam na introdução de arquiteturas mais poderosas e mais complexas. Em geral, há menos exigências de compatibilidade de geração para geração para estas máquinas mais pequenas.

Estrutura e função

Um sistema informático é um sistema muito complexo. O ponto-chave para conceber e/ou para descrever um sistema complexo é utilizar a abstração para expressar o sistema, ou uma das suas partes, em diferentes níveis de representação, de forma hierárquica. A um determinado nível, os detalhes dos níveis inferiores são ocultados para que a imagem que é apresentada seja com base no conceito do que se precisa de saber. Ao concentrarmo-nos no que é relevante em cada momento, menos detalhes têm de ser considerados e inter-relacionados e torna-se mais produtivo na conceção e/ou obtém-se uma imagem mais clara e uma melhor compreensão do que é descrito.

A cada nível, é necessário preocupar-se com a estrutura e a função:

- **Estrutura** - quantos componentes existem e a forma como são interligados

- **Função** - o funcionamento de cada componente individual como parte do todo.

As funções básicas de um sistema informático são: entrada de dados, saída de dados, processamento de dados, armazenamento de dados e controlo.

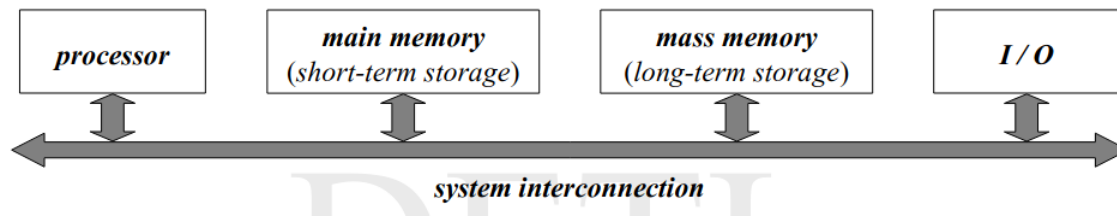
O sistema informático deve ser capaz de processar dados. Os dados requerem uma grande variedade de formas e a gama de requisitos de processamento é ampla. Existem, no entanto, apenas alguns métodos ou tipos fundamentais de processamento de dados.

O sistema informático deve também ser capaz de transferir dados entre si e o mundo exterior. O ambiente operacional consiste em dispositivos que ou servem como fontes ou destinos de dados. Quando os dados são recebidos de ou entregues a um dispositivo que está diretamente ligado a um sistema informático, o processo de o fazer é conhecido por executar entrada - saída (I/O) e o dispositivo é referido como um periférico. Quando os dados são transferidos a longas distâncias, de ou para um periférico o processo é conhecido como comunicação de dados.

É essencial que o sistema informático também armazene dados. Mesmo que o computador esteja a processar dados em tempo real (ou seja, os dados são introduzidos, processados e os resultados são imediatamente enviados), o sistema informático deve armazenar temporariamente pelo menos os dados que estão a ser trabalhados a qualquer momento. Assim, existe uma função de armazenamento de dados a 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, recuperados e atualizados sequencialmente.

Finalmente, estas funções devem ser controladas para se conseguir algum trabalho útil. Isto o controlo é exercido, em última análise, pelo indivíduo (programador) que fornece o sistema informático com instruções, mas dentro do próprio sistema informático um a unidade de controlo deve gerir os recursos disponíveis e orquestrar o espetáculo das suas partes funcionais, em resposta a essas instruções.

Ao nível superior, a estrutura de um sistema informático pode ser entendida como



- **Processador** (ou unidade central de processamento - CPU) - controla o funcionamento do computador e desempenha as suas funções de processamento de dados.
- **Memória principal** - armazena dados durante o processamento; tem um carácter volátil.
- **Memória de massa** - armazena dados entre as execuções de processamento e permite quantidades de dados a serem recuperadas e eventualmente atualizadas durante o processamento; tem uma natureza não volátil e é visto como um dispositivo de E/S especial.
- **E/S** - move dados entre o sistema informático e o seu ambiente externo.
- **Interconexão do sistema** - assegura que a transferência de dados se realiza entre os componentes; é normalmente implementado como um autocarro.

Para ter o sistema informático a realizar uma tarefa específica, é necessário fornecer um grupo de instruções que, no seu conjunto, constituem o programa a ser executados. A ideia chave aqui é como representar instruções?

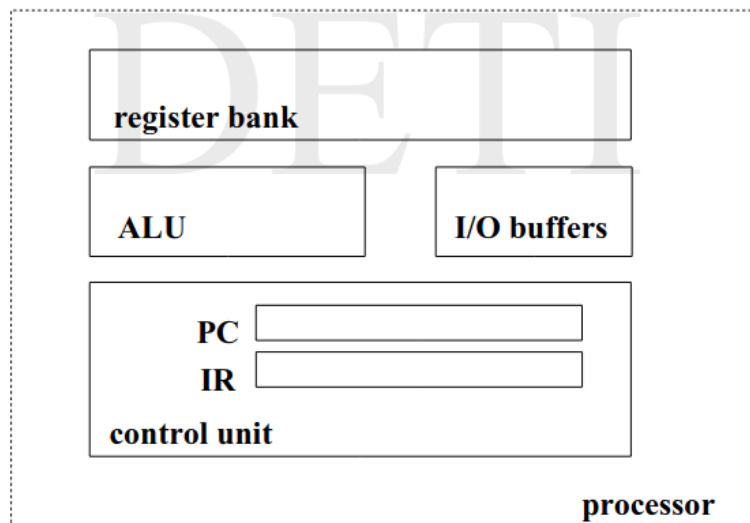
Suponha que poderiam ser representadas de uma forma adequada para serem armazenadas na memória principal juntamente com os dados, constituindo assim um tipo especial de dados em sentido abstrato. Então, um sistema informático poderia obter as suas instruções lendo-as da memória principal e um programa poderia ser definido ou alterado através da definição dos valores dessa porção da memória.

Esta ideia, desenvolvida independentemente por John von Neumann e Alan Turing, passou a ser conhecido como o conceito de programa armazenado e tem sido universalmente adotado pelos designers de computadores. Esta é a razão pela qual os sistemas informáticos são por vezes chamados máquinas von Neumann.

Neste sentido, embora um sistema informático seja um sistema digital muito complexo, pode ser entendido como um sistema que alterna continuamente entre dois sistemas internos básicos: **fetch de instruções**, onde o processador acede à memória para obter o próximo instrução, e **execução de instrução**, onde o processador descodifica o recém-recuperado e efetua a operação correspondente.



De uma forma simplificada, o próprio processador pode ser visto como sendo constituído por uma unidade de controlo, que se ocupa principalmente das fases de instrução e descodificação das instruções; por uma unidade aritmética/lógica, que executa as operações prescritas; por um registo banco, que armazena dados temporários; e por buffers de E/S, que permitem a comunicação com os outros componentes do sistema informático.



O conjunto de instruções de qualquer processador inclui sempre instruções do tipo

- **Movimento de dados** - transferência de dados entre algum registo do banco de registo e memória principal ou algum controlador de E/S.
- **Aritmética / lógica** - instruções aritméticas (add, subtract, multiply, divide), quer em formato de ponto fixo ou flutuante, instruções lógicas (not, and, or, x-or) e conteúdo do registo de deslocações / rotativas (shifting / rotating).
- **Ramificação** - modificando a sequência estrita da execução das instruções, quer incondicionalmente ou dependente de alguma condição.
- **Chamada de sub-rotina** - execução de subprogramas (segmentos de código autónomos dentro de todo o grupo de instruções fornecidas), quer incondicionalmente ou dependente de alguma condição.

Perspectiva histórica

O desempenho dos sistemas de computador tem feito um progresso incrível desde a época em que o primeiro computador eletrónico de uso geral foi construído. Esse ritmo acelerado 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 podiam ser colocados em um único chip dobrava a cada ano e previu corretamente que esse ritmo continuaria no futuro próximo. Essa observação ficou conhecida como Lei de Moore. O ritmo continuou ano após ano e década após década desde então. Ele desacelerou um pouco para dobrar a cada dois anos na década de 1970 e, mais recentemente, em 2015, a Intel anunciou que o ritmo atualmente está dobrando a cada dois anos e meio.

Em meados da década de 1970, surgiu o 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 uma fração crescente dos sistemas de computador 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 Unix e, posteriormente, seu clone Linux, reduziu o custo e o risco de lançar um novo hardware de processador.

Essas mudanças possibilitaram o desenvolvimento bem-sucedido de um novo conjunto de arquiteturas baseado em instruções mais simples, conhecido como RISC (Reduced Instruction Set Computers), no início da década de 1980. Os processadores RISC focaram a atenção dos designers em duas técnicas de desempenho: a exploração do paralelismo de nível de instrução (primeiro, usando pipelining e, posteriormente, usando o problema de instrução múltipla) e a aplicação sistemática de caches para acelerar o acesso do processador a instruções e dados (primeiro, em formulários simples e, posteriormente, usando organizações e otimizações mais sofisticadas).

O efeito da combinação destas melhorias arquitectónicas e organizacionais foi quatro vezes superior:

- melhorou significativamente o poder computacional posto à disposição dos utilizadores (os microprocessadores de maior desempenho do dia superaram os supercomputadores construído 10 anos antes).
- novas classes de sistemas informáticos surgiram devido à dramática melhoria em desempenho dos custos: os computadores pessoais e as estações de trabalho surgiram no 1980 e os telefones inteligentes e os computadores tablet tornaram-se os principais preferidos plataforma informática de muitas pessoas na última década.
- progresso contínuo no fabrico de semicondutores, como previsto pela Moore's O direito, levou ao domínio de sistemas informáticos baseados em microprocessadores em toda a gama de concepção de computadores.
- trouxe um profundo impacto no desenvolvimento de software: em primeiro lugar, ao permitir programadores para trocar desempenho para produtividade em muitas aplicações através a utilização do paradigma orientado para os objectos; em segundo lugar, o desempenho está a ser perseguido substituindo os intérpretes por compiladores just-in-time e compilação com base em vestígios para a abordagem tradicional de compilador e ligador; terceiro, a actual popularidade do Software como Serviço (SaaS- Software as a Service) através da Internet para a execução de aplicações.

Desde 2003, a taxa de melhoria do desempenho do processador único diminuiu devido à barreira imposta pela quantidade máxima permitida de dissipação de energia do ar chips arrefecidos e a falta de novas ideias para o paralelismo a nível de instrução a explorar eficazmente.

Em 2004, a Intel cancelou os seus projectos de alto desempenho não processados e juntou-se a outros fabricantes de chips ao declarar que o caminho para um maior desempenho seria através chips multinúcleos, em vez de através de uniprocessadores mais rápidos.

Isto significa que a ênfase agora não é colocada apenas no paralelismo a nível de instrução (ILP - instruction-level parallelism), mas também no paralelismo a nível de dados (DLP - data-level parallelism), no paralelismo a nível de fios (TLP - thread-level parallelism) e paralelismo de nível de exigência (RLP - request-level parallelism).

Considerando que as técnicas de hardware e compilador exploram ILP implicitamente sem a atenção do programador, DLP, TLP e RLP são explicitamente paralelamente e exigir a reestruturação dos pedidos a fim de ser eficiente exploradas.

Classes de sistemas informáticos

As alterações que estão a ocorrer na utilização de computadores levaram a cinco aplicações diferentes mercados, cada um caracterizado por requisitos e tecnologias específicas.

- **Dispositivos móveis pessoais (PMD - personal mobile devices)** - compreendem uma colecção de dispositivos sem fios com interfaces de utilizador multimédia, tais como telefones inteligentes e computadores tablet; 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 um ventilador para refrigeração também limitam o consumo total de energia; Os requisitos de energia, tamanho e peso levam à utilização de memória flash para armazenamento em massa; as aplicações são frequentemente baseadas na Web e orientadas para os meios de comunicação

- **Computação de secretária (desktop computing)** - abrangem desde computadores portáteis de gama baixa ou portáteis a estações de trabalho de gama alta e fortemente configuradas; o foco está na optimização do preço-desempenho, tanto na computação em bruto como nos gráficos; como resultado, os microprocessadores mais recentes e de maior desempenho e os microprocessadores de custo reduzido aparecem frequentemente em primeiro lugar nos sistemas de secretária; a computação tende a ser bem caracterizada em termos de aplicações e benchmarking, embora o uso crescente de aplicações interactivas centradas na Web coloque novos desafios.

- **Servidores (servers)** - representam a espinha dorsal da computação empresarial de grande escala, substituindo o mainframe tradicional como fornecedor de serviços informáticos e de ficheiros em maior escala e mais fiáveis; surgiram nos anos 80 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 dos serviços que suportam ou a um aumento dos requisitos funcionais (assim, a capacidade de aumentar a capacidade de computação, a memória principal, o armazenamento em massa e a largura de banda de E/S é crucial); os servidores são também concebidos para um rendimento eficiente - a capacidade de resposta aos pedidos individuais mantém-se importante, mas globalmente eficiente e eficaz em termos de custos, tal como definido pelo número de pedidos que podem ser tratados por unidade de tempo, são as principais métricas a serem tomadas em conta.

- **clusters / computadores à escala de armazém (clusters / warehouse-scale computers)** - os clusters são colecções de computadores de secretária, ou servidores, ligados por redes locais para actuarem como um único computador maior; cada nó de processamento gere o seu próprio sistema operativo [rede] e comunica com os outros através de um protocolo de rede; os maiores clusters são chamados de armazém computadores de escala (WSCs); o preço-desempenho e a potência são críticos; a disponibilidade também crítico, mas ao contrário dos servidores onde é assegurado por computador integrado hardware, as WSCs utilizam componentes redundantes de baixo custo como os blocos de construção, confiando numa camada de software para detectar e isolar as muitas falhas que ocorrem; o enfoque é colocado em aplicações interactivas, armazenamento em larga escala. Fiabilidade e largura de banda elevada da Internet - representam a base do que é agora chamado a nuvem supercomputadores e, em geral, computação de alto desempenho (HPC) estão relacionados com WSCs, mas diferem por enfatizarem o desempenho em ponto flutuante e por funcionarem em grande programas de lotes intensivos de comunicação cuja execução leva semanas de cada vez; as redes internas muito rápidas são aqui críticas.

- **computadores incorporados (embedded computers)** - encontram-se em máquinas do quotidiano, instrumentos 'inteligentes' e produtos de consumo; estão relacionados com computadores PMD, mas ao contrário deles, eles não executar software desenvolvido externamente; o preço é o factor chave - desempenho requisitos existem, mas o objectivo principal é satisfazer as necessidades de desempenho a um preço mínimo.

Classes of computer systems and their systemic characteristics

Source: Adapted from Computer Architecture: A Quantitative Approach

| <i>Feature</i> | <i>PMD</i> | <i>Desktop</i> | <i>Server</i> | <i>C/WSC</i> | <i>Embedded</i> |
|-------------------------------|---|---|---|---------------------------------------|---|
| <i>System Price</i> | \$100 - \$1 000 | \$300 - \$2 500 | \$5 000 - \$10 000 000 | \$100 000 - \$200 000 000 | \$10 - \$100 000 |
| <i>Microprocessor Price</i> | \$10 - \$100 | \$50 - \$500 | \$200 - \$2 000 | \$50 - \$250 | \$0.01 - \$100 |
| <i>Critical Design Issues</i> | cost, energy, performance, responsiveness | price-performance, energy, graphics-performance | throughput, availability, scalability, energy | price-performance, throughput, energy | price, energy, application-specific performance |

Classes de paralelismo

O paralelismo a vários níveis é a força motriz predominante da concepção de computadores através todas as classes de computadores, sendo a energia e o custo as principais restrições. Existem basicamente dois tipos de paralelismo nas aplicações:

- paralelismo de nível de dados (DLP - data-level parallelism) - surge quando existem múltiplos itens de dados que podem ser processados ao mesmo tempo.
- paralelismo a nível de tarefa (TLP - task-level parallelism) - surge quando a tarefa a ser executada pode ser dividida em subtarefas que operam independentemente.

O hardware de computador, por sua vez, pode explorar estes dois tipos de paralelismo de aplicação em quatro maneiras diferentes:

- paralelismo a nível de instrução (instruction-level parallelism) - o paralelismo a nível de dados é explorado com a ajuda do compilador usando ideias como a canalização, a execução especulativa e a questão múltipla.
- através de hardware especial (through special hardware) - arquiteturas vectoriais e unidades processadoras gráficas (GPUs) explorar o paralelismo ao nível dos dados, aplicando a mesma instrução a uma recolha de dados em paralelo.
- paralelismo ao nível dos fios (thread-level parallelism) - o paralelismo ao nível dos dados e o paralelismo ao nível das tarefas podem ser ambos explorados num modelo fortemente acoplado que permite a interacção entre threads.
- paralelismo de nível de exigência (request-level parallelism) - o paralelismo é explorado entre tarefas essencialmente pouco desenvolvidas especificado pelo programador ou pelo sistema operativo.

Na década de 1960, Michael Flynn estudou os esforços de computação paralela que foram feita até à data e encontrou uma classificação que ainda hoje é popular. Ele olhou para a paralelismo nas instruções e fluxos de dados exigidos pelas instruções no componente mais restrito do multiprocessador e colocou todos os computadores num só das quatro categorias

- instrução única - fluxos de dados únicos (SISD) (single instruction – single data streams) - corresponde ao uniprocessador; no entanto, o paralelismo a nível de instrução ainda pode ser explorado.

- instrução única - múltiplos fluxos de dados (SIMD) (single instruction – multiple data streams)- a mesma instrução é executada por múltiplas unidades de processamento que utilizam diferentes

fluxos de dados; esta categoria compreende arquiteturas vectoriais, extensões multimédia a conjuntos de instruções standard e GPUs.

- instrução múltipla - são executados fluxos de dados únicos (MISD) (multiple instruction – single data streams) - instruções múltiplas sobre o mesmo pedaço de dados; nenhum multiprocessador comercial deste tipo foi construído ainda (embora se o pedaço de dados for considerado como representando um vector de dados, então as arraias sistólicas podem ser raciocinadas para se enquadrarem nesta categoria).

- instrução múltipla - múltiplos fluxos de dados (MIMD) (multiple instruction – multiple data streams) - visa o paralelismo a nível de tarefas onde cada processador executa o seu próprio programa com os seus próprios dados; paralelismo a nível de dados podem também ser exploradas, embora seja provável que as despesas gerais sejam mais elevadas do que na SIMD; os processadores multicondutores enquadram-se nesta categoria.

Amdahl's Law (Lei de Amdahl) -> [fiabilidade do sistema/fração da taxa de falha](#)



execution time for the entire task without using the improvement

tempo de execução para toda a tarefa sem utilizar a melhoria



execution time for the entire task using the improvement

tempo de execução para toda a tarefa utilizando a melhoria

O ganho de desempenho que pode ser obtido melhorando alguma característica de um sistema informático pode ser estimado pela Lei de Amdahl. Amdahl declarou 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 toda a operação que pode ser melhorada e é expresso pela fórmula.

$$\begin{aligned} \text{speedup}_{\text{overall}} &= \frac{\text{execution time for the entire task without using the improvement}}{\text{execution time for the entire task using the improvement}} = \\ &= \frac{1}{(1 - \text{frac}_{\text{enhanc}}) + \frac{\text{frac}_{\text{enhanc}}}{\text{speedup}_{\text{enhanc}}}}, \end{aligned}$$

onde $\text{frac}_{\text{enhanc}}$ é a fração de tempo no sistema informático original que pode ser convertido para tirar partido do modo de execução mais rápido e $\text{speedup}_{\text{enhanc}}$ é a velocidade a ser ganha localmente pela adoção do modo mais rápido de execução.

O processador utilizado num servidor web deve ser mudado de modo a acelerar operações. O novo processador é 10 vezes mais rápido do que o original. Assumindo que presentemente o processador está ocupado 40% do tempo e está à espera de E/S 60% da época, qual é a velocidade global ganha se a substituição se realizar?

$$\begin{aligned} \text{frac}_{\text{enhanc}} &= 0.4 \quad \wedge \quad \text{speedup}_{\text{enhanc}} = 10 \quad \Rightarrow \\ \Rightarrow \text{speedup}_{\text{overall}} &= \frac{1}{0.6 + \frac{0.4}{10}} = \frac{1}{0.64} = 1.56 . \end{aligned}$$

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

A Lei de Amdahl é particularmente útil na comparação do sistema global performance de diferentes alternativas.

Uma operação comum requerida nos processadores gráficos é a raiz quadrada. Suponha-se que a raiz quadrada de ponto flutuante (FPSQRT) é responsável por 20% do tempo de execução de uma referência gráfica crítica. Uma proposta é melhorar o hardware do FPSQRT e acelerar 10 vezes esta operação. A outra alternativa é apenas fazer com que todas as operações de ponto flutuante sejam executadas mais rapidamente por um fator de 1,6. Note-se que as operações de ponto flutuante são responsáveis por metade do tempo de execução da aplicação.

Comparar estas duas alternativas de concepção

$$\text{speedup}_{\text{FPSQRT}} = \frac{1}{0.8 + \frac{0.2}{10}} = \frac{1}{0.82} = 1.22$$

$$\text{speedup}_{\text{FP}} = \frac{1}{0.5 + \frac{0.5}{1.6}} = \frac{1}{0.81} = 1.23$$

Princípios quantitativos de concepção de computadores

Algumas diretrizes são úteis na concepção de sistemas informáticos

- Tirar partido do paralelismo - o paralelismo é um dos métodos mais importantes para melhorar o desempenho; é possível recorrer à redundância para aumentar a dependência capacidade e/ou às vezes fazer as operações irem mais depressa, simplesmente adicionando mais recursos – redimensionabilidade (scalability), ou através da obtenção da concorrência subjacente; isto pode ser feito a vários níveis de abstração: sistema, processador individual ou baixo nível desenho digital.
- Tirar partido do princípio da localidade - os programas tendem a reutilizar instruções e dados que utilizaram recentemente.
- Concentrar-se no caso comum - ao fazer um compromisso de design, favorecer o caso frequente sobre a infrequente; o caso frequente é frequentemente mais simples de otimizar e pode ser mais gratificante; funciona bem não só para o desempenho, mas também para atribuição de recursos e poder.

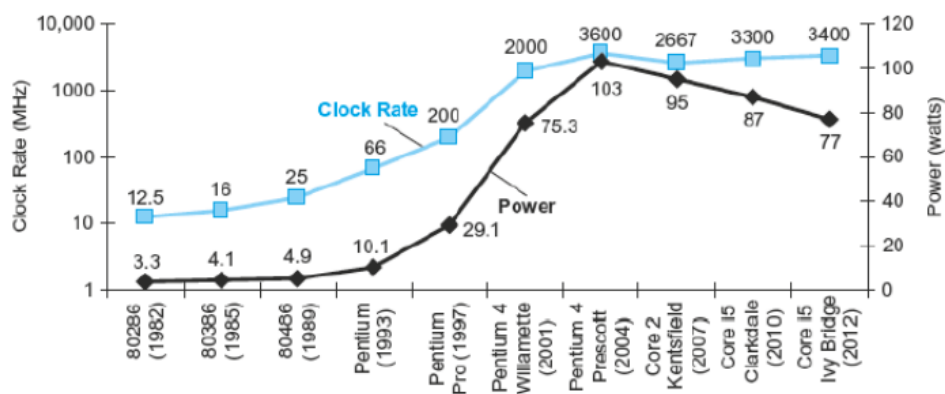
Potência e energia em circuitos integrados

O poder é o maior desafio que o designer de computadores enfrenta hoje em dia. Primeiro, a energia deve ser trazida e distribuída em torno do chip e os microprocessadores modernos utilizam centenas de pinos e múltiplas camadas de interligação apenas para energia e terra. Em segundo lugar, a energia é dissipada como calor e deve ser removida.

Tanto a velocidade do relógio como a potência aumentaram rapidamente durante décadas e depois aplanaram-se recentemente. A razão é que os projetistas têm chegado ao limite prático de potência para o arrefecimento dos microprocessadores.

Taxa de relógio e potência para microprocessadores Intel x86 durante oito gerações

Source: Computer Organization and Design: The Hardware / Software Interface



Para os chips CMOS, o consumo tradicional de energia tem sido na comutação transístores, normalmente chamados energia dinâmica. A energia necessária por transístor é proporcional ao produto da carga capacitiva impulsionada pelo transístor e pelo quadrado da voltagem.

$$\text{energy}_{\text{dynamic}} \cong \text{capacitive load} \cdot \text{voltage}^2.$$

A carga capacitiva é uma função do número de transístores ligados a um de saída e da tecnologia, que por si só determina a capacidade dos fios e os transístores.

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

$$\text{power}_{\text{dynamic}} \triangleq \text{capacitive load} \cdot \text{voltage}^2 \cdot \text{switching frequency} .$$

A potência dinâmica e a energia são grandemente reduzidas ao baixar a tensão, pelo que as tensões desceram de 5V para pouco menos de 1V em 20 anos para permitir o aumento da frequência sem afetar demasiado a potência. O problema hoje em dia é que uma maior redução da voltagem parece tornar os transístores demasiado estanques.

Para além do consumo dinâmico de energia, há que considerar também hoje em dia o consumo de energia estática devido à corrente de fuga que flui até quando o transistor é desligado. Nos servidores, por exemplo, a corrente de fuga é tipicamente responsável por 40% do consumo total de energia. Assim, aumentando o número de transístores aumenta a necessidade de mais dissipação de energia, mesmo quando todos os transístores são desligados. Os chips do servidor podem consumir mais de 100W e arrefecer o chip e o sistema circundante é uma grande despesa em computadores à escala de armazém.

Os microprocessadores modernos oferecem várias técnicas para melhorar a eficiência energética apesar de taxas de relógio planas e tensões de fornecimento constantes.

- Manter um registo das operações - a maioria dos microprocessadores desliga hoje o relógio de módulos inativos para poupar energia e energia dinâmica.
- Escala dinâmica de tensão-frequência (DVFS) - a maioria dos microprocessadores oferecem hoje uma poucas frequências e tensões de relógio em que funcionar para uma potência mais baixa e consumo de energia.
- Desenho para casos típicos - os microprocessadores a utilizar na informática de secretária são concebidos para um caso típico de utilização intensiva a altas temperaturas de funcionamento, confiar em sensores de temperatura on-chip para detetar quando a atividade deve ser reduzida automaticamente para evitar o sobreaquecimento
- over clocking - A Intel começou a oferecer o modo Turbo em 2008, onde o chip decide por si só, se for seguro funcionar a uma taxa de relógio mais elevada, normalmente 10% sobre a taxa nominal do relógio, durante um curto período de tempo, possivelmente apenas em alguns núcleos, até a temperatura começar a subir.

Fiabilidade

Historicamente, os circuitos integrados eram um dos componentes mais fiáveis de um sistema informático. Essa sabedoria convencional, no entanto, começou a mudar como os tamanhos das características dos transístores tornaram-se inferiores a 32 nm. Tanto transitórios como permanentes as falhas estão a tornar-se mais comuns, pelo que os arquitetos informáticos devem conceber sistemas para lidar com estes desafios.

A fiabilidade do módulo é uma medida de operação contínua do módulo, conforme especificado, ou por outras palavras, é uma medida do tempo que leva um módulo a falhar contados a partir de um instante inicial de referência. Assim, o tempo médio até ao fracasso (MTTF) é uma medida de fiabilidade. A reciprocidade do MTTF é a taxa de falhas, geralmente reportadas como falhas por bilião de horas de funcionamento, ou falhas no tempo (FIT).

A interrupção do serviço é medida pelo tempo médio para reparação (MTTR). Tempo médio entre falhas (MTBF) é apenas a soma de MTTF e MTTR.

Se uma coleção de módulos tiver vidas exponencialmente distribuídas - o que significa que a idade de um módulo não é relevante para a sua probabilidade de falhar, e as suas falhas são independente, então a taxa global de insucesso da recolha é a soma do insucesso taxas dos módulos individuais.

A disponibilidade do módulo (module availability) é uma medida de funcionamento contínuo do módulo, conforme especificado com respeito à alternância entre o serviço e a interrupção.

Para sistemas não redundantes com reparação, a disponibilidade de módulos é dada por

$$\text{module availability} = \frac{\text{MTTF}}{\text{MTTF} + \text{MTTR}} .$$

Assumir um subsistema de disco com os seguintes componentes e MTTFs

- 10 discos, cada um classificado a 1 000 000 horas MTTF
- 1 controlador ATA avaliado em 500 000 horas MTTF
- 1 fonte de alimentação nominal a 200 000 horas MTTF
- 1 ventilador avaliado a 200 000 horas MTTF
- 1 cabo ATA avaliado em 1 000 000 horas MTTF.

Utilizando as hipóteses simplificadoras sobre as distribuições de probabilidade dos tempos de vida e a independência das falhas, calcular o MTTF de todo o subsistema.

$$\begin{aligned} \text{failure rate}_{\text{subsys}} &= 10 \cdot \frac{1}{1000000} + \frac{1}{500000} + \frac{1}{200000} + \frac{1}{200000} + \frac{1}{1000000} = \\ &= \frac{23}{1000000 \text{ hours}} . \\ \text{MTTF}_{\text{subsys}} &= \frac{1}{\text{failure rate}_{\text{subsys}}} = \frac{1000000}{23} \approx 43\,500 \text{ hours} . \end{aligned}$$

Os subsistemas de disco têm frequentemente fontes de alimentação redundantes para melhorar a fiabilidade. Assumir a alimentação elétrica a partir do último exemplo, com um MTTR de 24 horas, e calcular a fiabilidade de um sistema redundante com duas fontes de alimentação.

$$\begin{aligned} \text{MTTF}_{\text{power supply pair}} &= \frac{\frac{\text{MTTF}_{\text{power supply}}}{2}}{\frac{\text{MTTR}_{\text{power supply}}}{\text{MTTF}_{\text{power supply}}}} = \frac{\text{MTTF}_{\text{power supply}}^2}{2 \cdot \text{MTTR}_{\text{power supply}}} = \\ &= \frac{200000^2}{2 \cdot 24} \approx 830\,000\,000 \text{ hours} . \end{aligned}$$

A Lei de Amdahl é também aplicável para além do desempenho.

Considere o exemplo de fiabilidade discutido anteriormente e assuma o poder a fiabilidade do fornecimento foi melhorada a partir das 200 000 horas MTTF anteriores para o impressionante

830 000 000 horas MTTF, ou seja, uma melhoria de 4 150 vezes. Como isto afeta a fiabilidade do sistema no seu todo?

$$\begin{aligned} \text{failure rate}_{\text{orig sys}} &= 10 \cdot \frac{1}{1000000} + \frac{1}{500000} + \frac{1}{200000} + \frac{1}{200000} + \frac{1}{1000000} = \\ &= \frac{23}{1000000 \text{ hours}} \end{aligned}$$

Portanto, a fração da taxa de falha devido ao fornecimento de energia 5 por 1 000 000 horas em 23 por 1 000 000 horas, ou aproximadamente 22%. Assim, tem-se

$$\text{reliability improv}_{\text{power supply}} = \frac{1}{0.78 + \frac{0.22}{4150}} = 1.28 \text{ .}$$

Medir o desempenho

Ao comparar alternativas de desenho, pretende-se frequentemente relacionar o desempenho de dois sistemas informáticos diferentes, digamos X e Y. A frase X é mais rápida do que Y significa que o tempo de execução de algum programa executado em X é menor do que o tempo de execução do mesmo programa executado em Y.

Mas o problema é como escolher um programa, ou um grupo de programas, normalmente chamado benchmark, ou benchmark suite (conjunto de referência), que produzirá resultados significativos em gerais e que não se baseiam em características específicas de nenhuma das alternativas.

Além disso, há que considerar que as aplicações no mundo real funcionam nos sistemas informáticos são tão diferentes em tamanho e complexidade que a escolha dum sistema informático para uma área específica não é uma tarefa fácil.

Uma forma de proceder é executar programas que são mais simples do que aplicações reais, mas que, ao mesmo tempo, pode ser considerado representativo

- kernels - pequenas porções chave de aplicações reais
- Programas de brinquedos (toy programs)- pequenos programas, geralmente com um comprimento até 100 linhas de código, do tipo que os alunos são convidados a escrever nas aulas de programação

- Código sintético (synthetic code) - programas falsos que foram escritos com a intenção de tentar e corresponder ao comportamento de aplicações reais.

Esta abordagem já não é popular, principalmente porque é possível escrever compiladores que levam em conta o conhecimento de programas específicos, de modo a que um sistema informático específico pareça executar estes programas mais rapidamente do que quando está realmente a executar aplicações reais.

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

A Standard Performance Evaluation Corporation (SPEC) é uma empresa sem fins lucrativos formada para estabelecer, manter e endossar um conjunto padronizado de referências relevantes que podem ser aplicadas à mais recente geração de computadores de alto desempenho. A SPEC desenvolve conjuntos de referência e também revê e publica os resultados apresentados pelas organizações membros e outros parâmetros de referência licenciados.

SPECfp2000 (Sun Ultra 5 as the reference computer)

Source: Computer Architecture: A Quantitative Approach

| Benchmark | Ultra 5 exec time (s) | Opteron exec time (s) | SPECRatio | Itanium 2 exec time (s) | SPECRatio | Opteron/Itanium 22 exec time (s) | Itanium 2 / Opteron SpecRatio |
|---------------------------|--------------------------|--------------------------|--------------|----------------------------|--------------|-------------------------------------|----------------------------------|
| wupwise | 1 600 | 51.5 | 31.06 | 56.1 | 28.53 | 0.92 | 0.92 |
| swim | 3 100 | 125.0 | 24.73 | 70.7 | 43.85 | 1.77 | 1.77 |
| mgrid | 1 800 | 98.0 | 18.37 | 65.8 | 27.36 | 1.49 | 1.49 |
| applu | 2 100 | 94.0 | 22.34 | 50.9 | 41.25 | 1.85 | 1.85 |
| mesa | 1 400 | 64.6 | 21.69 | 108.0 | 12.99 | 0.60 | 0.60 |
| galgel | 2 900 | 86.4 | 33.57 | 40.0 | 72.47 | 2.16 | 2.16 |
| art | 2 600 | 92.4 | 28.13 | 21.0 | 123.67 | 4.40 | 4.40 |
| equake | 1 300 | 72.6 | 17.92 | 36.3 | 35.78 | 2.00 | 2.00 |
| facerec | 1 900 | 73.6 | 25.80 | 86.9 | 21.86 | 0.85 | 0.85 |
| ampp | 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 | 17.48 | 131.0 | 16.09 | 0.92 | 0.92 |
| sixtrack | 1 100 | 123.0 | 8.95 | 68.8 | 15.99 | 1.79 | 1.79 |
| apsi | 2 600 | 150.0 | 17.36 | 231.0 | 11.27 | 0.65 | 0.65 |
| Geometric mean | | | 20.86 | | 27.12 | 1.30 | 1.30 |

$$\begin{aligned}
 \frac{\text{Geometric mean}_A}{\text{Geometric mean}_B} &= \frac{\sqrt[n]{\prod_{i=1}^n \text{SPECRatio } A_i}}{\sqrt[n]{\prod_{i=1}^n \text{SPECRatio } B_i}} = \sqrt[n]{\prod_{i=1}^n \frac{\text{SPECRatio } A_i}{\text{SPECRatio } B_i}} = \\
 &= \sqrt[n]{\prod_{i=1}^n \frac{\frac{\text{exec time Ref}_i}{\text{exec time } A_i}}{\frac{\text{exec time Ref}_i}{\text{exec time } B_i}}} = \sqrt[n]{\prod_{i=1}^n \frac{\text{exec time } B_i}{\text{exec time } A_i}} = \\
 &= \sqrt[n]{\prod_{i=1}^n \frac{\text{Performance } A_i}{\text{Performance } B_i}}
 \end{aligned}$$

A equação de desempenho do processador

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

$$\text{CPU execution time} = \text{CPU clock cycles} \cdot \text{clock cycle time} ,$$

onde a variável ciclos de relógio CPU representa o número total de ciclos de relógio para a execução do programa e a variável tempo de ciclo de relógio é o período do relógio CPU.

Esta expressão pode ser ainda expandida em

$$\text{CPU execution time} = \text{instruction count} \cdot \text{CPI} \cdot \text{clock cycle time} ,$$

onde a contagem das instruções variáveis representa o número total de instruções executado pelo programa e pode ser obtido de uma forma simples para organizações processadoras não sofisticadas se se tiver uma listagem do programa compilação em linguagem de montagem, e a variável CPI é a média do número de ciclos de relógio por instrução.

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 relógio por instrução e tempo de ciclo de relógio. Além disso, o tempo de execução da CPU está igualmente dependente destes três parâmetros: uma dada percentagem de alteração num dos resultarão na mesma alteração do tempo de execução da CPU.

Infelizmente, é difícil alterar um parâmetro em completo isolamento das outras como as tecnologias básicas envolvidas na mudança são interdependentes:

- De instrução - está dependente da arquitetura informática e da tecnologia de compilação.
- Ciclos de relógio por instrução - depende tanto da arquitetura informática como da organização informática.
- Tempo de ciclo de relógio - depende da tecnologia de hardware subjacente e da organização informática.

O tempo de execução da CPU para executar um programa pode ser ainda mais refinado se um tem em conta o CPI para cada tipo de instrução que está a ser executada

$$\text{CPU execution time} = \sum_i \left(\frac{\text{instruction count}_i}{\text{instruction count}} \cdot \text{CPI}_i \right) \cdot \text{clock cycle time} .$$

Suponha que as seguintes medições foram feitas na execução de um padrão de referência num dado sistema informático.

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

Assumir que estão a ser consideradas alternativas de conceção quer para diminuir o CPI para FPSQR para 2, ou para diminuir o CPI médio para todas as operações de PF para 2,5. Utilizar a equação de desempenho do processador para comparar estas alternativas.

Uma vez que o CPI é o único parâmetro que muda no desempenho do processador a resposta pode ser dada apenas através do cálculo do CPI global na equação caso original e para as duas alternativas.

Original situation

$$\begin{aligned}\text{overall CPI}_{\text{orig}} &= \sum_i \frac{\text{instruction count}_i}{\text{instruction count}} \cdot \text{CPI}_i = \\ &= 4.00 \cdot 0.25 + 1.33 \cdot 0.75 = 2.00\end{aligned}$$

Enhanced FPSQR

$$\begin{aligned}\text{overall CPI}_{\text{enFPSQR}} &= \text{overall CPI}_{\text{orig}} - 0.02 \cdot (\text{CPI}_{\text{origFPSQR}} - \text{CPI}_{\text{enFPSQR}}) = \\ &= 2.00 - 0.02 \cdot (20 - 2) = 1.64\end{aligned}$$

Enhanced FP

$$\begin{aligned}\text{overall CPI}_{\text{enFP}} &= \sum_i \frac{\text{instruction count}_i}{\text{instruction count}} \cdot \text{CPI}_i = \\ &= 2.50 \cdot 0.25 + 1.33 \cdot 0.75 = 1.63\end{aligned}$$