Sistemas Computacionais

Tradução Capítulo 1

Bem-vindo a este livro! Estamos muito satisfeitos por ter essa oportunidade de transmitir a emoção do mundo dos sistemas de computador. Não é um campo seco e triste, onde o progresso é glacial e onde novas ideias se atrofiam da negligência. Não! os Computadores
são o produto da indústria de tecnologia da informação incrivelmente vibrante, todos aspetos dos quais são responsáveis por quase 10% do produto nacional bruto dos Estados Unidos e cuja economia se tornou dependente em parte da rápida melhoria na tecnologia da informação prometidas pela Lei de Moore. É incomum A indústria abraça a inovação a um ritmo de tirar o fôlego. Nos últimos 30 anos, houve vários novos computadores cuja introdução pareceu revolucionar a indústria de computação; essas revoluções foram interrompidas apenas porque alguém mais construiu um computador ainda melhor.
"A corrida para inovar levou a um progresso sem precedentes desde o início da computação eletrônica no final da década de 1940. A indústria de transporte manteve ritmo com a indústria de computadores, por exemplo, hoje poderíamos viajar de New York para Londres em um segundo por um centavo. Tome um momento para contemplar como essa melhoria mudaria a sociedade - vivendo no Taiti enquanto trabalhava em San Francisco, indo a Moscou para uma noite no Ballet Bolshoi - e você pode apreciar as implicações dessa mudança.

Os computadores levaram a uma terceira revolução para a civilização, com as informações revolução ocorrendo ao lado das revoluções agrícola e industrial. A multiplicação resultante da força intelectual e alcance da humanidade naturalmente afetou profundamente nossa vida cotidiana e mudou as maneiras pelas quais a busca por novos conhecimentos é realizada. "Agora existe uma nova veia da ciência investigação, com cientistas computacionais unindo conhecimentos teóricos e experimentais cientistas na exploração de novas fronteiras em astronomia, biologia, química e física, entre outros.

"A revolução dos computadores continua. Cada vez que o custo da computação melhora por outro fator de 10, as oportunidades para computadores se multiplicam. Aplicativos que economicamente inviáveis, de repente se tornam práticos. No passado recente, as seguintes aplicações foram "citação de ciência da computação".

Computadores em automóveis: até os microprocessadores melhorarem drasticamente em preço e desempenho no início dos anos 80, o controle de carros por computador era ridículo. Hoje, os computadores reduzem a poluição, melhoram a eficiência de combustível através de controles do motor e aumentar a segurança através de avisos de ponto cego, faixa avisos de partida, deteção de objetos em movimento e airbag (proteção para proteger ocupantes em um acidente.

Telefones celulares: quem teria sonhado que avança no computador sistemas levariam mais da metade do planeta a ter telefones celulares, permitindo a comunicação de pessoa para pessoa com quase qualquer pessoa em qualquer lugar do mundo?

Projeto do genoma humano: "e custo dos equipamentos de informática para mapear e analisar sequências de DNA humano foram centenas de milhões de dólares. É improvável que alguém teria considerado esse projeto se os custos com computadores fossem 10 100 vezes mais altos, como teriam sido 15 a 25 anos antes. Além disso, os custos continuam caindo; em breve você poderá adquirir seu próprio genoma, permitindo que os cuidados médicos sejam adaptados a você.

World Wide Web: Não existe na época da primeira edição deste livro, a web transformou nossa sociedade. Para muitos, a web substituiu as bibliotecas e jornais.

Mecanismos de pesquisa: à medida que o conteúdo da Web cresceu em tamanho e valor, informações relevantes se tornaram cada vez mais importantes. Hoje muitas pessoas dependem de mecanismos de busca para uma parte tão grande de suas vidas que seria uma dificuldade para ir sem eles.
Claramente, os avanços nessa tecnologia agora afetam quase todos os aspetos de nossa sociedade. Os avanços do hardware permitiram aos programadores criar maravilhosamente útil), o que explica por que os computadores são onipresentes.

A ciência de hoje sugere os aplicativos matadores de amanhã: já estão a caminho óculos que aumentar a realidade, a sociedade sem dinheiro e carros que podem se conduzir.

Classes de aplicativos de computação e suas Características

Embora um conjunto comum de tecnologias de hardware (consulte as Seções 1.4 e 1.5) seja usado em computadores que variam de eletrodomésticos inteligentes a telefones celulares até os maiores
supercomputadores, essas diferentes aplicações têm diferentes requisitos de design e empregar as principais tecnologias de hardware de diferentes maneiras. Em geral, computadores são usados em três classes diferentes de aplicativos.
Os computadores pessoais (PCs) são possivelmente a forma mais conhecida de computação, quais leitores deste livro provavelmente usaram extensivamente. Computadores pessoais enfatizar a entrega de bom desempenho para usuários únicos a baixo custo e geralmente
executar terceiros para que) mercadorias. "é uma classe de computação impulsionou a evolução de muitos tecnologias de computação, que tem apenas cerca de 35 anos! Servidores são a forma moderna do que antes eram computadores muito maiores, e geralmente são acessados apenas através de uma rede. Os servidores são orientados a transportar grandes cargas de trabalho, que podem consistir em aplicativos complexos únicos - geralmente uma
aplicação científica ou de engenharia - ou lidar com muitos pequenos trabalhos, como ocorrem na construção de um grande servidor web. "Esses aplicativos geralmente são baseados em então) de outra fonte (como uma base de dados ou sistema de simulação), mas são) modificados ou personalizados para uma função específica. Os servidores são criados a partir da mesma tecnologia básica dos computadores de mesa, mas proporcionam maior computação, armazenamento e capacidade de entrada / saída. Em geral, os servidores também enfatizam mais confiabilidade, já que uma falha é geralmente mais cara do que em um único utilizador de PC.

Os servidores abrangem a mais ampla faixa de custo e capacidade. Na extremidade inferior, um servidor pode ser pouco mais que um computador de mesa sem uma tela ou teclado e custa mil dólares. "ese servidores low-end são normalmente usados para # le storage, aplicativos para pequenas empresas ou serviço da web simples (consulte a Seção 6.10). No outro extremo são os supercomputadores, que atualmente consistem em dezenas de milhares de processadores e muitos terabytes de memória, e custam dezenas a centenas de milhões de dólares. Os supercomputadores geralmente são usados para ciência e engenharia de ponta cálculos, como previsão do tempo, exploração de petróleo, estrutura de proteínas determinação e outros problemas em larga escala. Embora esses supercomputadores representam o pico da capacidade de computação, eles representam uma fração relativamente pequenos servidores e uma fração relativamente pequena do mercado geral de computadores em termos da receita total.
O Computador embutido ou incorporados são a maior classe de computadores e abrangem a mais ampla variedade de aplicativos e desempenho. Os computadores incorporados incluem os microprocessadores encontrados em seu carro, os computadores numa televisão e os redes de processadores que controlam um avião ou navio de carga moderno. Os sistemas Embutidos de computação são projetados para executar um aplicativo ou um conjunto de aplicativos normalmente integrados ao hardware e entregues como um sistema único; portanto, apesar do grande número de computadores incorporados, a maioria dos utilizadores nunca realmente vejo que eles estão usando um computador.

Termo Decimal	Abreviação	Valor em	Termo binário	Abreviação Maior	Valor	%
kilobyte	KB	10^3	kibibyte	KiB	2^{10}	2%
megabyte	MB	10^6	mebibyte	MiB	2^{20}	5%
gigabyte	GB	10^9	gibibyte	GiB	2^{30}	7%
terabyte	TB	10^{12}	tebibyte	TiB	2^{40}	10%
petabyte	PB	10^{15}	pebibyte	PiB	2^{50}	13%
exabyte	EB	10^{18}	exbibyte	EiB	2^{60}	15%
zettabyte	ZB	10^{21}	zebibyte	ZiB	2^{70}	18%
yottabyte	YB	10^{24}	yobibyte	YiB	2^{80}	21%

FIGURA 1.1 A ambiguidade de 2X vs. 10Y bytes foi resolvida adicionando uma notação binária para todos os termos de tamanho comum. Na última coluna, observamos quanto maior o termo binário é do que o termo decimal correspondente, que é composto à medida que avançamos no gráfico.
Esses prefixos funcionam tanto para bits quanto para bytes; portanto, o gigabit (Gb) é de 109 bits, enquanto o gibibits (Gib) é de 230 bits.

Aplicativos incorporados o) possuem requisitos de aplicativos exclusivos que combine um desempenho mínimo com limitações rigorosas de custo ou energia. Por exemplo, considere um music player: o processador precisa ser tão rápido quanto necessário para lidar com sua função limitada e, além disso, minimizar custos e energia é o objetivo mais importante. Apesar do baixo custo, os computadores embarcados o) têm menor tolerância a falhas, pois os resultados podem variar de perturbadores (quando o novas falhas na televisão) a devastadoras (como pode ocorrer quando o computador em um
acidente de avião ou navio de carga).

Nos aplicativos incorporados orientados ao consumidor, como um eletrodoméstico digital, a confiabilidade é alcançada principalmente pela simplicidade - a ênfase está em realizar uma função da maneira mais perfeita possível. Em grandes sistemas, técnicas de redundância do mundo dos servidores são o) empregadas.

Embora este livro se concentre em computadores de uso geral, a maioria dos conceitos se aplica diretamente, ou com pequenas modificações, aos computadores incorporados.

Elaboração: Elaborações são seções curtas usadas ao longo do texto para fornecer mais detalhes sobre um assunto específico que possa ser de seu interesse. Leitores desinteressados podem pular sobre uma elaboração, uma vez que o material subsequente nunca dependerá do conteúdo
da elaboração.
Muitos processadores incorporados são projetados usando núcleos de processador, uma versão de um processador escrito em uma linguagem de descrição de hardware, como Verilog ou VHDL (consulte
Capítulo 4). O núcleo permite que um designer integre outras especificações de aplicativos! c hardware com o núcleo do processador para fabricação em um único chip.
Bem-vindo à era PostPC "A marcha contínua da tecnologia traz mudanças geracionais em hardware de computador que abala toda a indústria de tecnologia da informação.
Desde a última edição do livro, passamos por essa mudança, pois é significativo no passado, como a mudança que começou há 30 anos para computadores pessoais. Substituindo o PC é o dispositivo móvel pessoal (PMD). Os PMDs são operados por bateria com conexão sem fio
conectividade à Internet e normalmente custam centenas de dólares e, como PCs, os utilizadores podem fazer o download para que o ware ("aplicativos") seja executado neles. Ao contrário dos PCs, eles não
tem teclado e mouse e é mais provável que confie em uma tela sensível ao toque ou mesmo entrada de fala. O PMD de hoje é um smartphone ou tablet, mas amanhã pode incluir óculos eletrônicos. A Figura 1.2 mostra o rápido tempo de crescimento de tablets e smartphones versus o de PCs e celulares tradicionais.
A substituição do servidor tradicional é a Cloud Computing, que depende de data centers gigantes que agora são conhecidos como Warehouse Scale Computers (WSCs).
Empresas como Amazon e Google constroem essas WSCs contendo 100.000 servidores e, em seguida, deixe que as empresas aluguem partes delas para que possam fornecer) mercadorias serviços para PMDs sem a necessidade de criar WSCs. De facto, então! mercadorias como
um serviço (SaaS) implantado através da nuvem está revolucionando o setor de mercadorias como PMDs e WSCs estão revolucionando a indústria de hardware. Hoje é assim) mercadorias os desenvolvedores o) terão uma parte de seu aplicativo que é executada no PMD e uma parte que é executada na nuvem.

FIGURA 1.2 O número fabricado por ano de tablets e smartphones, que refletem a era PostPC, versus computadores pessoais e telefones celulares tradicionais. Telefones inteligentes representam o crescimento recente do setor de telefonia celular e passaram nos PCs em 2011. Os tablets são os mais rápidos crescentes, quase dobrando entre 2011 e 2012. PCs recentes e categorias tradicionais de celulares são relativamente (em declínio.

1.6 Desempenho

Avaliar o desempenho dos computadores pode ser bastante desafiador. A escala e complexidade dos modernos sistemas de software, juntamente com a ampla gama de desempenho técnicas de aprimoramento empregadas pelos projetistas de hardware, fizeram desempenho avaliação muito mais difícil.
Ao tentar escolher entre diferentes computadores, o desempenho é um importante atributo. Medir e comparar com precisão computadores diferentes é essencial para compradores e, portanto, para designers. As pessoas que vendem computadores sabem disso como bem. Frequentemente, os vendedores gostariam que você visse o computador deles da melhor maneira possível luz, independentemente de refletir com precisão as necessidades do comprador inscrição. Portanto, entender a melhor forma de medir o desempenho e as limitações das medições de desempenho são importantes na seleção de um computador.
O restante desta seção descreve maneiras diferentes pelas quais o desempenho pode ser determinado; em seguida, descrevemos as métricas para medir o desempenho do ponto de vista de um usuário de computador e um designer. Também analisamos como essas métricas estão relacionados e apresentam a equação clássica de desempenho do processador, que iremos usar ao longo do texto.

Definindo o desempenho

Quando dizemos que um computador tem melhor desempenho que outro, o que fazemos significar? Embora essa pergunta possa parecer simples, uma analogia com os passageiros aviões mostra quão sutil a questão do desempenho pode ser. A figura 1.14 lista alguns aviões de passageiros típicos, juntamente com a sua velocidade de cruzeiro, alcancem a capacidade. Se quiséssemos saber qual dos aviões nesta tabela tinha as melhores primeiro precisamos definir o desempenho. Por exemplo, considerando diferentes medidas de desempenho, vemos que o avião com o maior cruzeiro de velocidade foi o Concorde (aposentado de serviço em 2003), o avião com o maior o alcance é o DC-8, e o avião com maior capacidade é o 747.

Vamos supor que definimos o desempenho em termos de velocidade. Th ainda deixa duas possíveis definições. Você poderia definir o plano mais rápido como o de maior velocidade de cruzeiro, levando um único passageiro de um ponto a outro no menor tempo possível.
Se você estava interessado em transportar 450 passageiros de um ponto para outro, no entanto, o 747 seria claramente o mais rápido, como mostra a última coluna da figura.
Da mesma forma, podemos definir o desempenho do computador de várias maneiras diferentes.
Se você estiver executando um programa em dois computadores desktop diferentes, diria que o mais rápido é o computador de mesa que realiza o trabalho primeiro. Se você fosse executando um datacenter que tinha vários servidores executando tarefas enviadas por muitos utilizadores, diria que o computador mais rápido foi o que completou mais empregos durante um dia. Como utilizador individual de computador, você está interessado em reduzir o tempo de resposta - o tempo entre o início e a conclusão de uma tarefa - também referido

para como tempo de execução. Os gerentes de datacenter geralmente estão interessados em aumentar taxa de transferência ou largura de banda - a quantidade total de trabalho realizado em um determinado período. Consequentemente, na maioria dos casos, precisaremos de métricas de desempenho diferentes e de conjuntos diferentes aplicativos para comparar dispositivos móveis pessoais, mais focados em tempo de resposta versus servidores, mais focados na taxa de transferência.

Rendimento e tempo de resposta

As seguintes alterações em um sistema de computador aumentam a taxa de transferência, diminuem tempo de resposta, ou ambos?
1. Substituindo o processador em um computador por uma versão mais rápida
2. Adicionando processadores adicionais a um sistema que usa vários processadores para tarefas separadas - por exemplo, pesquisando na web
A redução do tempo de resposta quase sempre melhora a taxa de transferência. Portanto, no caso 1, o tempo de resposta e o rendimento são aprimorados. No caso 2, nenhuma tarefa é executada trabalho feito mais rapidamente, portanto, apenas a taxa de transferência aumenta.
Se, no entanto, a demanda por processamento no segundo caso fosse quase Tão grande quanto a taxa de transferência, o sistema pode forçar as solicitações de enfileiramento.

Neste caso, aumentar a taxa de transferência também pode melhorar o tempo de resposta, pois isso reduziria o tempo de espera na fila. Th nós, em muitos computadores reais sistemas, alterando o tempo de execução ou a taxa de transferência, muitas vezes afeta o outro.
Ao discutir o desempenho dos computadores, nos preocuparemos principalmente com tempo de resposta para os primeiros capítulos. Para maximizar o desempenho, queremos minimizar o tempo de resposta ou o tempo de execução de alguma tarefa. Contudo, podemos relacionar o desempenho e o tempo de execução de um computador X pela seguinte expressão matemática:

$$Perfomance_X = \frac{1}{Tempo\ de\ execução_X}$$

Isto significa que, para dois computadores X e Y, se o desempenho de X for maior que o desempenho de Y, então temos que:

$$Perfomance_X > Perfomance_Y$$

$$\frac{1}{Tempo\ de\ execução_X} > \frac{1}{Tempo\ de\ execução_Y}$$

$$Tempo\ de\ execução_Y > Tempo\ de\ execução_X$$

Assim, o tempo de execução em Y é maior que o de X, se X for mais rápido que Y.

Ao discutir um projeto de computador, muitas vezes queremos relacionar o desempenho de dois diferentes computadores quantitativamente. Usaremos a frase “X é n vezes mais rápido que Y ”-- ou equivalente“ X é n vezes mais rápido que Y ”-- pode significar que:

$$\frac{Perfomance_X}{Perfomance_Y} = n$$

Se X é n vezes mais rápido que Y, então o tempo de execução em Y é n vezes desde que seja em X:

$$\frac{Perfomance_X}{Perfomance_Y} = \frac{Tempo\ de\ execução_X}{Tempo\ de\ execução_Y} = n$$

Desempenho relativo

Se o computador A executar um programa em 10 segundos e o computador B executar o mesmo programa em 15 segundos, quanto mais rápido é A do que B? Sabemos que A é n vezes mais rápido que B se

$$\frac{Perfomance_A}{Perfomance_B} = \frac{Tempo\ de\ execução_A}{Tempo\ de\ execução_B} = n$$

A taxa de desempenho é:

$$\frac{15}{10} = 1.5$$

A é, portanto, 1,5 vezes mais rápido que B.
No exemplo acima, também podemos dizer que o computador B é 1,5 vezes mais lento que o computador A, já que

$$\frac{Perfomance_A}{Perfomance_B} = 1{,}5$$

Significa que:

$$\frac{Perfomance_A}{1{,}5} = Perfomance_B$$

Por uma questão de simplicidade, normalmente usaremos a terminologia tão rápido quanto quando tentamos compare computadores quantitativamente. Como desempenho e tempo de execução são recíprocos, aumentar o desempenho requer diminuir o tempo de execução. Evitar a confusão potencial entre os termos aumentando e diminuindo, geralmente diga "melhorar o desempenho" ou "melhorar o tempo de execução" quando queremos dizer "aumentar desempenho "e" diminuir o tempo de execução ".

Medindo o desempenho

Tempo é a medida do desempenho do computador: o computador que executa o mesma quantidade de trabalho em menos tempo é a mais rápida. O tempo de execução do programa é medido em segundos por programa. Contudo, o tempo pode ser definido de diferentes maneiras, dependendo do que contamos. A definição mais direta de tempo é chamada relógio de parede, tempo de resposta ou tempo decorrido. Estes termos significam o tempo total para concluir uma tarefa, incluindo acesso a disco, memória, entrada / saída (E / S) atividades, sobrecarga do sistema operacional - tudo.

Os computadores são frequentemente compartilhados, no entanto, e um processador pode funcionar em vários programas simultaneamente. Nesses casos, o sistema pode tentar otimizar a taxa de transferência em vez de tentar minimizar o tempo decorrido para um programa. Portanto, nós Muitas vezes, deseja distinguir entre o tempo decorrido e o tempo em que o processador está trabalhando em nosso nome. Tempo de execução da CPU ou simplesmente tempo da CPU, que reconhece essa distinção, é o tempo que a CPU gasta computando para essa tarefa e não inclui o tempo gasto aguardando E / S ou executando outros programas.
(Lembre-se, porém, que o tempo de resposta experimentado pelo usuário será o tempo decorrido do programa, não o tempo da CPU.) O tempo da CPU pode ser dividido no tempo de CPU gasto no programa, chamado tempo de CPU do usuário e tempo de CPU gasto no sistema operacional executando tarefas em nome do programa, chamado tempo de CPU do sistema. A diferença entre o tempo de CPU do sistema e do usuário é difícil de com precisão, porque muitas vezes é difícil atribuir responsabilidade pelo sistema operacional atividades para um programa de utilizador em vez de outro e por causa da funcionalidade diferenças entre sistemas operativos.
Por consistência, mantemos uma distinção entre desempenho com base em tempo decorrido e baseado no tempo de execução da CPU. Vamos usar o termo sistema desempenho para se referir ao tempo decorrido em um sistema descarregado e desempenho da CPU para se referir ao tempo de CPU do usuário. Vamos nos concentrar no desempenho da CPU neste capítulo, embora nossas discussões sobre como resumir o desempenho possam ser aplicadas a tempo decorrido ou medições de tempo da CPU.
Diferentes aplicações são sensíveis a diferentes aspetos do desempenho de um sistema de computador. Muitos aplicativos, especialmente aqueles executados em servidores, dependem tanto no desempenho de E / S, que, por sua vez, depende de hardware e software.
O tempo total decorrido medido por um relógio de parede é a medida do interesse.

Em alguns ambientes de aplicativos, o usuário pode se preocupar com taxa de transferência, resposta tempo ou uma combinação complexa dos dois (por exemplo, taxa de transferência máxima com um pior tempo de resposta). Para melhorar o desempenho de um programa, é preciso tenha uma definição clara de qual métrica de desempenho importa e, em seguida, prossiga para procure gargalos de desempenho medindo a execução do programa e procurando para os prováveis gargalos. Nos capítulos seguintes, descreveremos como pesquisar gargalos e melhorar o desempenho em várias partes do sistema.
Embora, como utilizadores de computador, nos preocupemos com o tempo, quando examinamos os detalhes de um computador, é conveniente pensar no desempenho em outras métricas.

Em particular, os designers de computadores podem querer pensar em um computador usando uma medida relacionada à rapidez com que o hardware pode executar funções básicas. Quase todos os computadores são construídos usando um relógio que determina quando os eventos ocorrem coloque no hardware. Esses intervalos de tempo discretos são chamados de ciclos de relógio (ou ticks, ticks, períodos do relógio, relógios, ciclos). Designers se referem ao comprimento de um período do relógio, como o tempo para um ciclo completo do relógio (por exemplo, 250 picossegundos ou 250 ps) e como a taxa de clock (por exemplo, 4 gigahertz ou 4 GHz), que é o inverso do período do relógio. Na próxima subseção, formalizaremos o relacionamento entre os ciclos do relógio do designer de hardware e os segundos do usuário do computador.

1. Suponha que saibamos que um aplicativo que usa dispositivos móveis pessoais
   dispositivos e a nuvem é limitada pelo desempenho da rede. Para o seguinte mudanças, indique se apenas o rendimento melhora, tanto o tempo de resposta
   e o rendimento melhoram, ou nenhum deles melhora.
   a. Um canal de rede extra é adicionado entre o PMD e a nuvem, aumentando a taxa de transferência total da rede e reduzindo o atraso para obter acesso à rede (já que agora existem dois canais).
   b. O software de rede é aprimorado, reduzindo assim a rede atraso de comunicação, mas não aumentando a taxa de transferência.
   c. Mais memória é adicionada ao computador.
2. 2. O desempenho do computador C é 4 vezes mais rápido que o desempenho do computador B, que executa um determinado aplicativo em 28 segundos. Quanto tempo o computador C levar para executar esse aplicativo?

Desempenho da CPU e os seus fatores

Utilizadores e designers frequentemente examinam o desempenho usando métricas diferentes. Se nós pudéssemos relacionar essas métricas diferentes, poderíamos determinar o efeito de uma alteração no projeto no desempenho conforme experimentado pelo usuário. Desde que estamos nos confiando ao desempenho da CPU neste momento, a medida de desempenho final é tempo de execução. Uma fórmula simples relaciona as métricas mais básicas (ciclos de relógio e tempo de ciclo do clock) para o tempo da CPU:

Tempo de execução da CPU para um programa= Ciclos de clock da CPU X Tempo do ciclo do relógio

Como alternativa, como a taxa e o tempo do ciclo do relógio são inversos,

Tempo de execução da CPU para um programa

$$Tempo\ de\ execução\ da\ CPU = \frac{\text{Ciclos de clock da CPU}}{\text{Tempo do ciclo do relógio}}$$

Esta fórmula deixa claro que o designer de hardware pode melhorar o desempenho reduzindo o número de ciclos de clock necessários para um programa ou a duração do ciclo do relógio. Como veremos nos próximos capítulos, o designer muitas vezes enfrenta uma troca entre o número de ciclos de clock necessários para um programa e a duração de cada ciclo. Muitas técnicas que diminuem o número de ciclos de clock também podem aumentar o tempo do ciclo do relógio.

Melhorando a performance

Nosso programa favorito é executado em 10 segundos no computador A, que possui 2 GHz relógio. Estamos tentando ajudar um projetista de computadores a construir um computador, B, que execute este programa em 6 segundos. O designer determinou que uma quantidade substancial é possível aumentar a taxa de clock, mas esse aumento afetará o restante do Design da CPU, fazendo com que o computador B exija 1,2 vezes mais ciclos de clock computador A para este programa. Que frequência de relógio devemos dizer ao designer para o alvo?

Vamos primeiro encontrar o número de ciclos de relógio necessários para o programa em A:

$$CPU\ Tempo_B = \frac{1.2\ X\ CPU\ cilo\ de\ relegio_A}{\text{Taxa de relógio}_B}$$

$$6\ segundos = \frac{1.2X20X10^9\ ciclos}{\text{Taxa de relógio}_B}$$

$$\text{Taxa de relógio}_B = \frac{1.2X20X10^9\ ciclos}{6\ segundos} = \frac{0.2X20X10^9\ ciclos}{segundo}$$
$$= \frac{4X10^9\ ciclos}{segundo} = 4GH$$

Para executar o programa em 6 segundos, B deve ter o dobro da taxa de clock de A.

Desempenho das instruções

As equações de desempenho acima não incluíram nenhuma referência ao número de instruções necessárias para o programa. No entanto, como o compilador gerou claramente instruções para executar e o computador teve que executar as instruções para executar programa, o tempo de execução deve depender do número de instruções em um programa. Uma maneira de pensar sobre o tempo de execução é que ele é igual ao número de instruções executadas multiplicadas pelo tempo médio por instrução. Portanto, o número de ciclos de clock necessários para um programa pode ser gravado como

Ciclos de clock da CPU = Instruções para um programa X Ciclos médios de clock por instrução

O ciclo do relógio a termo por instrução, que é o número médio de relógios ciclos que cada instrução leva para executar, muitas vezes é abreviado como CPI. Dado que diferentes, as instruções podem levar quantidades diferentes de tempo, dependendo do que elas fazem, a CPI é uma média de todas as instruções executadas no programa. O CPI fornece uma maneira de comparando duas implementações diferentes da mesma arquitetura de conjunto de instruções, como o número de instruções executadas para um programa será, obviamente, o mesmo.

Usando a equação de desempenho

Suponha que tenhamos duas implementações da mesma arquitetura de conjunto de instruções.
O computador A possui um tempo de ciclo de clock de 250 ps e um CPI de 2,0 para alguns programas e o computador B tem um tempo de ciclo de clock de 500 ps e um CPI de 1,2 para o mesmo programa. Qual computador é mais rápido para este programa e por quanto?

Sabemos que cada computador executa o mesmo número de instruções para o programa; vamos ligar para esse número I. Primeiro, encontre o número de clock do processador ciclo para cada computador:

$$CPU\ ciclo\ de\ relogio_A = IX2.0$$

$$CPU\ ciclo\ de\ relogio_B = IX1.2$$

Agora podemos calcular o tempo de CPU para cada computador:

$$CPU\ tempo_A = CPU\ ciclo\ de\ relogio_A X\ tempo\ ciclo\ de\ relogio$$
$$= I\ X\ 2.0\ X\ 250\ ps = 500\ X\ I\ ps$$

Da mesma forma, para B:

$$CPU\ tempo_B = I\ X\ 1.2\ X\ 500\ ps = 600\ X\ I\ ps$$

Claramente, o computador A é mais rápido. A quantidade mais rápida é dada pela razão entre tempos de execução:

$$\frac{CPU\ Perfomance_A}{CPU\ Perfomance_B} = \frac{Tempo\ de\ execução_B}{Tempo\ de\ execução_A} = \frac{600X\ I\ ps}{500\ X\ I\ ps} = 1.2$$

Podemos concluir que o computador A é 1,2 vezes mais rápido que o computador B para este programa.

A equação clássica de desempenho da CPU

Agora podemos escrever esta equação básica de desempenho em termos de contagem de instruções (o número de instruções executadas pelo programa), CPI e tempo do ciclo do relógio:

$$\text{Tempo de CPU} = \text{Instruções de contagem X CPI X tempo do ciclo do relógio}$$

ou, como a taxa de clock é o inverso do tempo do ciclo do clock:

$$\text{Tempo de CPU} = \frac{\text{Instruções de contagem X CPI}}{\text{taxa de clock}}$$

Essas fórmulas são particularmente úteis porque separam os três fatores principais que afetam o desempenho. Podemos usar essas fórmulas para comparar duas diferenças implementações ou avaliar uma alternativa de design, se soubermos seu impacto sobre esses três parâmetros.

Comparando segmentos de código

Um designer de compilador está tentando decidir entre duas sequências de código para um computador particular. Os projetistas de hardware forneceram os seguintes fatos:

	CPI para cada classe de instrução		
	A	B	C
CPI	1	2	3

Para uma declaração específica de linguagem de alto nível, o escritor do compilador é considerando duas sequências de código que requerem as seguintes instruções de contagem:

1.9 Coisas reais: comparando o Intel Core i7

Cada capítulo tem uma seção intitulada “Coisas reais” que liga os conceitos do livro com um computador que você pode usar todos os dias. Essas seções cobrem a tecnologia computadores modernos subjacentes. Para esta primeira seção "Coisas reais", examinamos como os circuitos integrados são fabricados e como o desempenho e a potência são medidos, com o Intel Core i7 como exemplo.

Benchmark de CPU SPEC

Um utilizador de computador que executa os mesmos programas todos os dias seria o candidato perfeito para avaliar um novo computador. O conjunto de programas executados formaria uma carga de trabalho. Para avaliar dois sistemas de computador, um utilizador simplesmente compararia o tempo de execução da carga de trabalho nos dois computadores. A maioria dos utilizadores, no entanto, não estão nesta situação. Em vez disso, eles devem confiar em outros métodos que medem o desempenho de um computador candidato, esperando que os métodos reflitam quão bem o computador funcionará com a carga de trabalho do usuário. Th é alternativa é geralmente seguido pela avaliação do computador usando um conjunto de referências - programas
escolhido especificamente para medir o desempenho. Os benchmarks formam uma carga de trabalho que o utilizador espera prever o desempenho da carga de trabalho real.
Como observamos acima, para agilizar o caso comum, primeiro precisa de saber com precisão qual caso é comum, então os benchmarks desempenham um papel crítico na arquitetura do computador.
O SPEC (Cooperativa de Avaliação de Desempenho do Sistema) é um esforço e suportado por vários fornecedores de computadores para criar conjuntos padrão de benchmarks para sistemas de computador modernos. Em 1989, a SPEC criou originalmente uma referência conjunto com foco no desempenho do processador (agora chamado SPEC89), que evoluiu através de cinco gerações.
O mais recente é o SPEC CPU2006, que consiste em um conjunto de 12 benchmarks inteiros (CINT2006) e 17 benchmarks de ponto flutuante (CFP2006).

Description	Nome	Instrução Conta x 10^9	CPI	Clock cycle time (seconds x 10^{-9})	Execution Time (seconds)	Reference Time (seconds)	SPECratio
Interpreted string processing	perl	2252	0.60	0.376	508	9770	19.2
Block-sorting	bzip2	2390	0.70	0.376	629	9650	15.4
GNU C compiler	gcc	794	1.20	0.376	358	8050	22.5
Combinatorial optimization	mcf	221	2.66	0.376	221	9120	41.2
Go game (AI)	go	1274	1.10	0.376	527	10490	19.9

Search gene sequence	hmmer	2616	0.60	0.376	590	9330	15.8
Chess game (AI)	sjeng	1948	0.80	0.376	586	12100	20.7
Quantum computer simulation	libquantum	659	0.44	0.376	109	20720	190.0
Video compression	h264av	3793	0.50	0.376	713	22130	31.0
Discrete event	omnetpp	367	2.10	0.376	290	6250	21.5
Games/path finding	astar	1250	1.00	0.376	470	7020	14.9
XML parsing	xalancbmk	1045	0.70	0.376	275	6900	25.1
Geometric mean	–	–	–	–	–	–	25.7

FIGURA 1.18 Pontos de referência SPECINTC2006 em execução no Intel Core i7 920 de 2,66 GHz.

Como a equação na página 35 explica, o tempo de execução é o produto dos três fatores nesta tabela: contagem de instruções em bilhões, relógios por instrução (CPI) e ciclo de clock tempo em nanossegundos. SPECratio é simplesmente o tempo de referência, fornecido pelo SPEC, dividido pelo tempo de execução medido. O número único citado como SPECINTC2006 é a média geométrica dos SPECratios.

Os benchmarks inteiros variam de parte de um compilador C a um programa de xadrez a uma simulação quântica por computador. Os pontos de referência de ponto flutuante incluem códigos de grade para modelagem de elementos finitos, códigos de método de partículas para dinâmica e códigos de álgebra linear esparsos para a dinâmica de fluidos.
A Figura 1.18 descreve os benchmarks inteiros do SPEC e seu tempo de execução no Intel Core i7 e mostra os fatores que explicam o tempo de execução: instrução contagem, CPI e tempo do ciclo do relógio. Observe que o CPI varia em mais de um fator de 5.
Para simplificar a comercialização de computadores, a SPEC decidiu reportar um único número para resumir todos os 12 valores de referência inteiros. Dividindo o tempo de execução de uma referência processador pelo tempo de execução do computador medido normaliza a execução

medições de tempo; essa normalização produz uma medida, chamada SPECratio, que tem a vantagem de que resultados numéricos maiores indicam desempenho mais rápido. Isso é, o SPECratio é o inverso do tempo de execução. Um resumo do CINT2006 ou CFP2006 a medição é obtida pela média geométrica dos SPECratios.

Elaboração: Ao comparar dois computadores usando SPECratios, use o comando geométrico significa que ele fornece a mesma resposta relativa, independentemente do computador usado para normalize os resultados. Se calcularmos a média dos valores do tempo de execução normalizado com uma aritmética, os resultados variariam dependendo do computador que escolhermos como referência.

A fórmula para a média geométrica é:

$$\sqrt[n]{\prod_{i=1}^{n} \text{Taxa de tempo de execução ratioi}}$$

em que tempo de execução ratioi é o tempo de execução normalizado para o computador de referência, para o i-ésimo programa de um total de n na carga de trabalho, e

$$\prod_{i=1}^{n} a_{i\ significa\ o\ produto\ de\ a1x\ a2x\ a\ldots an}$$

Referência de Potência SPEC

Dada a crescente importância de energia e potência, a SPEC acrescentou uma referência para medir o poder. Informa o consumo de energia dos servidores em diferentes cargas de trabalho níveis, divididos em incrementos de 10%, durante um período de tempo. A Figura 1.19 mostra os resultados para um servidor usando processadores Intel Nehalem semelhantes aos acima.

Target Load %	Performance (ssj_ops)	Average Power (watts)
100%	865,618	258
90%	786,688	242
80%	698,051	224
70%	607,826	204
60%	521,391	185
50%	436,757	170
40%	345,919	157
30%	262,071	146
20%	176,061	135
10%	86,784	121
0%	0	80
Overall Sum	4,787,166	1922
∑ssj_ops / ∑power =		2490

FIGURA 1.19 SPECpower_ssj2008 rodando em um soquete duplo 2,66 GHz Intel Xeon X565 com 16 GB de DRAM e um disco SSD de 100 GB.

O SPECpower começou com outro benchmark SPEC para aplicativos de negócios Java (SPECJBB2005), que exercita os processadores, caches e memória principal também como máquina virtual Java, compilador, coletor de lixo e partes do sistema operacional operativo. O desempenho é medido na taxa de transferência e as unidades são de negócios operações por segundo. Mais uma vez, para simplificar o marketing de computadores, o SPEC reduz esses números para um único número, chamado "ssj_ops gerais por watt".
A fórmula dessa métrica resumida única é:

$$\text{overall } ssj\_ops_i \text{ per watt} = \left(\sum_{i=0}^{10} ssj\_ops_i\right) / \left(\sum_{i=0}^{10} power_i\right)$$

onde ssj_ops_i é desempenho a cada incremento de 10% e $power_i$ é a potencia consumida em cada nível de desempenho.

Falácias e Armadilhas

O objetivo de uma seção sobre falácias e armadilhas, encontrada em todos os capítulo, é explicar alguns conceitos errôneos comuns de que você pode encontro. Nós os chamamos de falácias. Ao discutir uma falácia, tentamos dar uma contraexemplo. Também discutimos armadilhas ou erros facilmente cometidos. Muitas vezes, as armadilhas são generalizações de princípios que só são verdadeiras em um contexto limitado. O objetivo destas seções é para ajudá-lo a evitar esses erros nos computadores que você pode projetar ou usar. Falácias e armadilhas de custo / desempenho têm enredado muitos arquitetos de computadores, inclusive nós. Por conseguinte, esta seção não sofre escassez de
exemplos relevantes. Começamos com uma armadilha que prende muitos designers e revela um relacionamento importante no design de computadores.

Armadilha: Esperar que a melhoria de um aspecto de um computador aumente desempenho proporcional ao tamanho da melhoria.

A ótima idéia de acelerar o caso comum tem um corolário desmoralizante que atormentou os designers de hardware e software. Isso nos lembra que o A oportunidade de melhoria é afetada por quanto tempo o evento consome. Um problema de design simples ilustra bem isso. Suponha que um programa seja executado em 100 segundos em um computador, com operações de multiplicação responsáveis por 80 segundos desse Tempo. Quanto eu tenho para melhorar a velocidade de multiplicação se eu quiser o meu programa para rodar cinco vezes mais rápido?

O tempo de execução do programa depois de fazer a melhoria é dado por a seguinte equação simples conhecida como Lei de Amdahl:

$$\text{Tempo de execução após melhoria} = \frac{\text{Tempo de execução afetado pela melhoria}}{\text{Quantidade de melhorias}} + \text{Tempo de execução não afetado}$$

Para este problema temos:

$$\text{Tempo de execução após melhoria} = \frac{80\ segundos}{n} + (100 - 80\ segundos)$$

Como queremos que o desempenho seja cinco vezes mais rápido, o novo tempo de execução deve demorar 20 segundos, dando por:

$$20 \text{ segundos} = \frac{80\ segundos}{n} + 20\ segundos$$

$$0 = \frac{80\ segundos}{n}$$

Ou seja, não existe uma quantidade pela qual possamos aumentar ou multiplicar para atingir um nível de cinco vezes mais.
aumento no desempenho, se a multiplicação for responsável por apenas 80% da carga de trabalho. A melhoria de desempenho possível com uma melhoria dada é limitada pela quantidade que o recurso aprimorado é usado. Na vida cotidiana, esse conceito também produz o que chamamos a lei dos retornos decrescentes.
Podemos usar a Lei da Amdahl para estimar melhorias de desempenho quando conheça o tempo consumido para alguma função e seu potencial aumento de velocidade. Amdahl's A lei, juntamente com a equação de desempenho da CPU, é uma ferramenta útil para avaliar melhorias potenciais. A lei de Amdahl é explorada com mais detalhes nos exercícios.
A lei de Amdahl também é usada para defender limites práticos ao número de processadores. Examinamos esse argumento na seção Falácias e armadilhas do Capítulo 6.

Falácia: Os computadores com baixa utilização usam pouca energia.
A eficiência de energia é importante com baixas utilizações porque as cargas de trabalho do servidor variam.

A utilização de servidores no computador de escala de armazém do Google, por exemplo, é entre 10% e 50% na maioria das vezes e a 100% menos de 1% do tempo.
Até dados por cinco anos para aprender a executar bem o benchmark SPECpower, os computador configurado com os melhores resultados em 2012 ainda usa 33% do pico de potência
a 10% da carga. Sistemas no campo que não são configurados para o SPECpower referência são certamente piores.
Como as cargas de trabalho dos servidores variam, mas usam uma grande fração do pico de potência, Luiz Barroso e Urs Hölzle [2007] argumentam que devemos redesenhar o hardware para alcançar “Computação proporcional à energia”. Se os servidores futuros usarem, digamos, 10% do pico de potência em 10% da carga de

trabalho, podemos reduzir a conta de eletricidade dos datacenters e nos tornarmos bons cidadãos corporativos em uma era de crescente preocupação com as emissões de CO2.
Falácia: projetar para desempenho e projetar para eficiência energética são objetivos não relacionados.
Como a energia é poder ao longo do tempo, é comum que hardware ou software otimizações que levam menos tempo economizam energia em geral, mesmo que a otimização demore um pouco mais de energia quando usado. Uma razão é que todo o resto do computador é consumir energia enquanto o programa estiver em execução, mesmo que a parte otimizada usa um pouco mais de energia, o tempo reduzido pode economizar a energia de todo o sistema.
Armadilha: Usando um subconjunto da equação de desempenho como uma métrica de desempenho.
Já alertamos sobre o perigo de prever o desempenho com base em simplesmente uma taxa de relógio, contagem de instruções ou CPI.

Outro erro comum, é usar apenas dois dos três fatores para comparar o desempenho. Embora usando dois dos três fatores podem ser válidos em um contexto limitado, o conceito também é facilmente mal utilizado. De facto, quase todas as alternativas propostas ao uso do tempo como métrica de desempenho levaram eventualmente a declarações enganosas, resultados distorcidos ou interpretações incorretas.
Uma alternativa ao tempo é o MIPS (milhões de instruções por segundo). Para um dado programa, o MIPS é simplesmente

$$MIPS = \frac{\text{Contagem de instruções}}{\text{Tempo de execução X } 10^6}$$

Como o MIPS é uma taxa de execução de instruções, o MIPS especifica o desempenho inversamente ao tempo de execução; computadores mais rápidos têm uma classificação MIPS mais alta. As boas notícias o MIPS é fácil de entender, e computadores mais rápidos significam maior MIPS, que corresponde à intuição.
Existem três problemas com o uso do MIPS como uma medida para comparar

computadores.
Primeiro, o MIPS especifica a taxa de execução da instrução, mas não leva em consideração os recursos das instruções. Não podemos comparar computadores com diferentes conjuntos de instruções usando o MIPS, pois a contagem de instruções certamente será diferente.
Segundo, o MIPS varia entre os programas no mesmo computador; assim, um computador não pode ter uma única classificação MIPS. Por exemplo, substituindo o tempo de execução, vemos a relação entre MIPS, taxa de clock e CPI:

$$MIPS = \frac{\text{Contagem de instruções}}{\frac{Contagem\ de\ instruções\ X\ CPI}{\text{Taxa de relogio}}\ X\ 10^6} = \frac{Taxa\ de\ relogio}{CPI\ X\ 10^6}$$

O CPI variou por um fator de 5 para o SPEC CPU2006 num computador Intel Core i7 na Figura 1.18, o MIPS também. Finalmente, e mais importante, se um novo programa executa mais instruções, mas cada instrução é mais rápida, o MIPS pode variar independentemente do desempenho!

Considere as seguintes medidas de desempenho para um programa:

Medição	Computador A	Computador B
Contagem de instruções	10 billion	8 billion
Taxa de relógio	4 GHz	4 GHz
CPI	1.0	1.1

a. Qual o computador tem a classificação MIPS mais alta?
b. Qual o computador é mais rápido?

1.11 Observações finais

Embora seja difícil prever exatamente qual o nível de custo / desempenho dos computadores terá no futuro, é uma aposta segura que eles serão muito melhores do que são hoje. Para participar desses avanços, designers e programadores de computador deve entender uma variedade maior de questões.
Os designers de hardware e software constroem sistemas de computador de maneira hierárquica camadas, com cada camada inferior ocultando detalhes do nível acima. Th é uma ótima idéia abstração é fundamental para entender os sistemas de computadores atuais, mas não significa que os designers possam limitar-se a conhecer uma única abstração.
Talvez o exemplo mais importante de abstração seja a interface entre hardware e software de baixo nível, chamado arquitetura do conjunto de instruções. A Manutenção da arquitetura do conjunto de instruções como constante permite muitas implementações de que a arquitetura - presumivelmente variando em custo e desempenho - funcione de maneira idêntica aos Programas. Por outro lado, a arquitetura pode impedir a introdução de inovações que exigem que a interface mude.

Existe um método confiável para determinar e relatar o desempenho usando o tempo de execução de programas reais como a métrica. O tempo de execução está relacionado a outras medidas importantes que podemos fazer pela seguinte equação:

$$\frac{Segundos}{Programa} = \frac{Instruções}{Programa} X \frac{Ciclos\ de\ relogio}{Instrução} X \frac{Segundos}{Ciclos\ de\ relogio}$$

Usaremos esta equação e os fatores constituintes muitas vezes. Devemos lembrar, entretanto, que individualmente os fatores não determinam o desempenho: somente o produto, que é igual ao tempo de execução, é uma medida confiável de desempenho.

O tempo de execução é a única medida válida e inatacável de desempenho. Muitas outras métricas foram propostas e consideradas necessárias.

Às vezes, essas métricas são defeituosas desde o início por não refletir o tempo de execução; outras vezes, uma métrica válida noutro contexto limitado é estendido e usado além desse contexto ou sem a necessidade adicional esclarecimentos necessários para torná-lo válido.

A principal tecnologia de hardware para processadores modernos é o silício. De Igual em importância para a compreensão da tecnologia de circuitos integrados é uma compreensão das taxas esperadas de mudança tecnológica, conforme previsto pela Lei de Moore. Enquanto o silício alimenta o rápido avanço do hardware, as novas idéias na organização de computadores melhoraram preço / desempenho. Duas das idéias principais estão explorando paralelismo no programa, normalmente hoje através de múltiplos processadores, e explorando localidade de acessos a uma hierarquia de memória, geralmente via caches.
A eficiência energética substituiu a área da matriz como o recurso mais crítico de design do microprocessador. Economizando energia ao tentar aumentar o desempenho forçou a indústria de hardware a mudar para microprocessadores multicore, forçando a indústria de software a mudar para a programação de hardware paralelo.
Agora o paralelismo é necessário para o desempenho.
Os projetos de computadores sempre foram medidos pelo custo e desempenho, bem como outros fatores importantes, como energia, confiabilidade, custo de propriedade e escalabilidade. Embora este capítulo tenha se concentrado em custo, desempenho e energia, os melhores projetos atingirão o equilíbrio apropriado para um determinado mercado entre todos os fatores.

Roteiro para este livro

Na parte inferior dessas abstrações estão os cinco componentes clássicos de um computador:
caminho de dados, controle, memória, entrada e saída (consulte a Figura 1.5). Estes cinco componentes também servem como estrutura para o restante dos capítulos deste livro:
■ Caminho de dados: Capítulo 3, Capítulo 4, Capítulo 6 e Apêndice C
■ Controle: Capítulo 4, Capítulo 6 e Apêndice C
■ Memória: Capítulo 5
■ Entrada: Capítulos 5 e 6
■ Resultado: capítulos 5 e 6
Como mencionado acima, o Capítulo 4 descreve como os processadores exploram implícitos paralelismo, o capítulo 6 descreve os microprocessadores multicore

explicitamente paralelos que estão no centro da revolução paralela, e o Apêndice C descreve o chip do processador gráfico altamente paralelo. O capítulo 5 descreve como uma memória hierarquia explora localidade. O capítulo 2 descreve os conjuntos de instruções - a interface entre compiladores e o computador - e enfatiza o papel dos compiladores e linguagens de programação no uso dos recursos do conjunto de instruções.
O Apêndice. A fornece uma referência para o conjunto de instruções do Capítulo 2. O Capítulo 3 descreve como computadores lidam com dados aritméticos. O Apêndice B apresenta o design lógico.

1.12 Perspectiva histórica para leitura adicional

Para cada capítulo do texto, uma seção dedicada a uma perspetiva histórica pode ser encontrado on-line em um site que acompanha este livro. Podemos procurar o desenvolvimento de uma ideia através de uma série de computadores ou descrever alguns projetos importantes, e fornecemos referências caso você esteja interessado em investigar mais.
A perspetiva histórica deste capítulo fornece um pano de fundo para algumas das ideias-chave apresentadas neste capítulo de abertura. Seu objetivo é dar-lhe o humano história por trás dos avanços tecnológicos e colocar conquistas em seu histórico contexto. Ao entender o passado, você poderá entender melhor as forças que moldará a computação no futuro. Cada seção Perspetiva Histórica on-line termina com sugestões para leitura adicional, que também são coletadas separadamente online na seção “Leitura adicional”. O restante da Seção 1.12 é encontrado online.

1.13 Exercícios

As classificações de tempo relativo dos exercícios são mostradas entre colchetes depois de cada número do exercício. Em média, um exercício avaliado em [10] levará o dobro do tempo um classificado [5]. Seções do texto que devem ser lidas antes de tentar um exercício será dado entre colchetes; por exemplo, <§1.4> significa que você deveria ter lido a Seção 1.4, rever a matéria, para ajudá-lo a resolver este exercício.

1.1 [2] <§1.1> Além dos telefones celulares inteligentes usados por um bilhão de pessoas, liste e descreva quatro outros tipos de computadores.
1.2 [5] <§1.2> As oito grandes ideias em arquitetura de computadores são semelhantes às ideias de outros campos. Combine as oito ideias da arquitetura de

computadores, “Design for Lei de Moore "," Use a abstração para simplificar o design "," Crie o caso comum Rápido ”,“ Desempenho via paralelismo ”,“ Desempenho via pipelining ”,“ Desempenho via Prediction "," Hierarchy of Memories "e" Confiabilidade via Redundância "para as seguintes ideias de outros campos:

a. Linhas de montagem na contrução de automóveis
b. Cabos de ponte suspensa
c. Aeronaves e sistemas de navegação marítima que incorporam informações sobre o vento
d. Elevadores expressos em edifícios
e. Balcão de reservas para bibliotecas
f. Aumentando a área da porta em um transistor CMOS para diminuir seu tempo de comutação.
g. Adição de catapultas eletromagnéticas de aeronaves (que são acionadas eletricamente em oposição aos atuais modelos movidos a vapor), permitidos pelo aumento da potência geração oferecida pela nova tecnologia de reatores.
h. Construção de carros autônomos cujos sistemas de controle dependem parcialmente do sensor existente sistemas já instalados no veículo base, como sistemas de saída de faixas e sistemas inteligentes de controle de cruzeiro.

1.3 [2] <§1.3> Descreva as etapas que transformam um programa escrito em um nível superior linguagem como C em uma representação que é executada diretamente por um computador processador.

1,4 [2] <§1.4> Suponha uma exibição colorida usando 8 bits para cada uma das cores primárias (vermelho, verde, azul) por pixel e um tamanho de quadro de 1280 × 1024.

a. Qual é o tamanho mínimo em bytes do buffer de quadros para armazenar um quadro?
b. Quanto tempo levaria, no mínimo, para que o quadro fosse enviado acima de 100 Rede Mbit / s?

1.5 [4] <§1.6> Considere três processadores P1, P2 e P3 diferentes executando o mesmo conjunto de instruções. P1 tem uma taxa de clock de 3 GHz e um CPI de 1,5. P2 tem um Clock de 2,5 GHz e um CPI de 1,0. O P3 possui uma taxa de clock de 4,0 GHz e um CPI de 2.2.

a. Qual processador tem o maior desempenho expresso em instruções por segundo?
b. Se cada um dos processadores executar um programa em 10 segundos, encontre o número de ciclos e o número de instruções.
c. Estamos tentando reduzir o tempo de execução em 30%, mas isso leva a um aumento de 20% no IPC. Que taxa de relógio devemos ter para obter essa redução de tempo?

1.6 [20] <§1.6> Considere duas implementações diferentes da mesma instrução arquitetura de conjunto. As instruções podem ser divididas em quatro classes, de acordo com CPI (classe A, B, C e D). P1 com uma taxa de clock de 2,5 GHz e CPIs de 1, 2, 3, e 3 e P2 com uma taxa de clock de 3 GHz e CPIs de 2, 2, 2 e 2.
Dado um programa com uma contagem dinâmica de instruções de 1.0E6 instruções divididas nas classes a seguir: 10% classe A, 20% classe B, 50% classe C e 20% classe

D, qual implementação é mais rápida?
a. Qual é o CPI global para cada implementação?
b. Encontre os ciclos de relógio necessários nos dois casos.

1.7 [15] <§1.6> Os compiladores podem ter um impacto profundo no desempenho de um aplicativo. Suponha que, para um programa, o compilador A resulte em uma dinâmica contagem de instruções de 1.0E9 e tem um tempo de execução de 1,1 s, enquanto o compilador B resulta em uma contagem dinâmica de instruções de 1.2E9 e um tempo de execução de 1.5 s.

a. Encontre o CPI médio de cada programa, considerando que o processador possui um ciclo de clock tempo de 1 ns.
b. Suponha que os programas compilados sejam executados em dois processadores diferentes. Se a execução os tempos nos dois processadores são iguais, quanto mais rápido é o relógio do processador executando o código do compilador A em relação ao relógio do processador executando código do compilador B?
c. Um novo compilador é desenvolvido que usa apenas instruções 6.0E8 e possui um CPI médio de 1,1. Qual é a velocidade de usar esse novo compilador versus usar compilador A ou B no processador original?

1.8 O processador Pentium 4 Prescott, lançado em 2004, tinha uma taxa de clock de 3,6 GHz e tensão de 1,25 V. Suponha que, em média, consumisse 10 W de eletricidade estática potência e 90 W de potência dinâmica.

O Core i5 Ivy Bridge, lançado em 2012, tinha uma taxa de clock de 3,4 GHz e voltagem de 0,9 V. Suponha que, em média, consumisse 30 W de energia estática e 40 W de poder dinâmico.

1.8.1 [5] <§1.7> Para cada processador, encontra as cargas capacitivas médias.
1.8.2 [5] <§1.7> Encontre a percentagem da potência total dissipada composta por potência estática e a razão entre potência estática e potência dinâmica para cada tecnologia.

1.8.3 [15] <§1.7> Se a potência total dissipada for reduzida em 10%, quanto a tensão deve ser reduzida para manter a mesma corrente de fuga? Nota: poder é definido como o produto de tensão e corrente.

1.9 Suponha que, para instruções aritméticas, de carregamento / armazenamento e ramificação, um processador tenha CPIs de 1, 12 e 5, respetivamente. Suponha também que, em um único processador, um programa requer a execução de instruções aritméticas 2.56E9, carga / armazenamento 1.28E9 instruções e 256 milhões de instruções de ramificação. Suponha que cada processador tenha uma frequência de clock de 2 GHz.
Suponha que, como o programa é paralelo para rodar em múltiplos núcleos, o número de instruções aritméticas e de carga / armazenamento por processador é dividido por 0,7 x p (onde p é o número de processadores), mas o número de instruções de ramificação por processador continua o mesmo.

1.9.1 [5] <§1.7> Encontre o tempo total de execução deste programa em 1, 2, 4 e 8 processadores e mostram a aceleração relativa dos resultados de 2, 4 e 8 para o resultado do processador único.

1.9.2 [10] <§§1.6, 1.8> Se o CPI das instruções aritméticas foi duplicado, qual seria o impacto no tempo de execução do programa em 1, 2, 4 ou 8 processadores?

1.9.3 [10] <§§1.6, 1.8> Qual deve ser o CPI das instruções de carga / armazenamento reduzido para que um único processador corresponda ao desempenho de quatro processadores usando os valores originais de CPI?
1.10 Suponha que uma bolacha de 15 cm de diâmetro tenha um custo de 12, contenha

84 matrizes e 0,020 defeitos / cm2. Suponha que uma bolacha de 20 cm de diâmetro tenha um custo de 15, contenha 100 morre e possui 0,031 defeitos / cm2.

1.10.1 [10] <§1.5> Encontre o rendimento para ambas as bolachas.
1.10.2 [5] <§1.5> Encontre o custo por dado para as duas bolachas.
1.10.3 [5] <§1.5> Se o número de matrizes por bolacha for aumentado em 10% e a defeitos por unidade de área aumentam 15%, encontram a área da matriz e o rendimento.

1.10.4 [5] <§1.5> Suponha que um processo de fabricação melhore o rendimento de 0,92 a 0,95. Encontre os defeitos por unidade de área para cada versão da tecnologia dada uma matriz área de 200 mm2.

1.11 Os resultados do benchmark SPEC CPU2006 bzip2 em execução em um AMD Barcelona possui uma contagem de instruções de 2.389E12, um tempo de execução de 750 s e um tempo de referência de 9650 s.

1.11.1 [5] <§§1.6, 1.9> Encontre a CPI se o tempo do ciclo do relógio for 0,333 ns.
1.11.2 [5] <§1.9> Encontre o SPECratio.
1.11.3 [5] <§§1.6, 1.9> Encontre o aumento no tempo da CPU se o número de instruções do benchmark é aumentado em 10% sem afetar o IPC.

1.11.4 [5] <§§1.6, 1.9> Encontre o aumento no tempo da CPU se o número de instruções do valor de referência é aumentado em 10% e o CPI é aumentado em 5%.
1.11.5 [5] <§§1.6, 1.9> Encontre a alteração no SPECratio para esta alteração.
1.11.6 [10] <§1.6> Suponha que estamos desenvolvendo uma nova versão do AMD Processador de Barcelona com uma taxa de clock de 4 GHz. Adicionamos algumas instruções para o conjunto de instruções de forma que o número de instruções foi reduzido em 15%. O tempo de execução é reduzido para 700 se o novo SPECratio é 13.7. Encontre o novo CPI.

1.11.7 [10] <§1.6> O valor de CPI é maior que o obtido em 1.11.1 como o relógio A taxa foi aumentada de 3 GHz para 4 GHz. Determine se o aumento no O CPI é semelhante ao da taxa de clock. Se eles são diferentes, porquê?

1.11.8 [5] <§1.6> Em quanto tempo o CPU foi reduzido?

1.11.9 [10] <§1.6> Para um segundo benchmark, libquantum, assuma uma execução tempo de 960 ns, CPI de 1,61 e freqüência de 3 GHz. Se o tempo de execução for reduzido em 10% adicionais sem afetar o IPC e com uma taxa de clock de 4 GHz, determine o número de instruções.

1.11.10 [10] <§1.6> Determine a taxa de clock necessária para dar mais 10% redução no tempo da CPU, mantendo o número de instruções e com o CPI inalterado.

1.11.11 [10] <§1.6> Determine a taxa de clock se o CPI for reduzido em 15% e o tempo da CPU em 20%, enquanto o número de instruções permanece inalterado.
1.12 A Seção 1.10 cita como armadilha a utilização de um subconjunto do desempenho equação como uma métrica de desempenho. Para ilustrar isso, considere os dois seguintes processadores. P1 tem uma taxa de clock de 4 GHz, CPI médio de 0,9 e requer a execução de instruções 5.0E9. P2 tem uma taxa de clock de 3 GHz, um CPI médio de 0,75 e requer a execução de instruções 1.0E9.

1.12.1 [5] <§§1.6, 1.10> Uma falácia comum é considerar o computador com o maior taxa de clock como tendo o maior desempenho. Verifique se isso é verdade para P1 e P2

1.12.2 [10] <§§1.6, 1.10> Outra falácia é considerar que o processador que está executando o maior número de instruções precisará de um tempo de CPU maior. Considerando que O processador P1 está executando uma sequência de instruções 1.0E9 e que o CPI dos processadores P1 e P2 não mudam, determine o número de instruções que P2 pode executar ao mesmo tempo em que P1 precisa executar as instruções 1.0E9.

1.12.3 [10] <§§1.6, 1.10> Uma falácia comum é usar o MIPS (milhões de instruções por segundo) para comparar o desempenho de dois processadores diferentes,

e considere que o processador com o maior MIPS tem o maior desempenho. Verifique se isso é verdade para P1 e P2.

1.12.4 [10] <§1.10> Outro modelo de desempenho comum é o MFLOPS (milhões operações de ponto flutuante por segundo), definido como
MFLOPS = Nº operações FP / (tempo de execução × 1E6)
mas essa figura tem os mesmos problemas que o MIPS. Suponha que 40% das instruções executadas em P1 e P2 são instruções de ponto flutuante. Encontre os MFLOPS configurações para os programas.

1.13 Outra armadilha citada na Seção 1.10 espera melhorar o desempenho geral desempenho de um computador, melhorando apenas um aspecto do computador. Considerar um computador executando um programa que requer 250 s, com 70 s gastos executando FP instruções, 85 s instruções L / S executadas e 40 s gastos na execução de ramificações instruções.

1.13.1 [5] <§1.10> Por quanto é reduzido o tempo total se o tempo para FP operações é reduzida em 20%?

1.13.2 [5] <§1.10> Em quanto tempo as operações INT são reduzidas se o tempo total é reduzido em 20%?

1.13.3 [5] <§1.10> O tempo total pode ser reduzido em 20%, reduzindo apenas o tempo para instruções da filial?

1.14 Suponha que um programa exija a execução de instruções 50 × 106 FP, Instruções 110 × 106 INT, instruções 80 × 106 L / S e ramo 16 × 106 instruções. O CPI para cada tipo de instrução é 1, 1, 4 e 2, respectivamente. Suponha que o processador tenha uma taxa de clock de 2 GHz.

1.14.1 [10] <§1.10> Em quanto devemos melhorar o IPC das instruções de FP, se queremos que o programa seja executado duas vezes mais rápido?

1.14.2 [10] <§1.10> Em quanto devemos melhorar o CPI das instruções L / S se queremos que o programa seja executado duas vezes mais rápido?

1.14.3 [5] <§1.10> Em quanto é melhorado o tempo de execução do programa se o IPC das instruções INT e FP for reduzido em 40% e o IPC de L / S e Ramo é reduzido em 30%?

1.15 [5] <§1.8> Quando um programa é adaptado para ser executado em vários processadores em sistema multiprocessador, o tempo de execução em cada processador é composto por tempo de computação e o tempo de sobrecarga necessários para seções críticas bloqueadas e / ou para enviar dados de um processador para outro. Suponha que um programa exija t = 100 s de tempo de execução em um processador. Quando executado p, cada processador requer t / p s, além de 4 s adicionais de sobrecarga, independentemente do número de processadores. Calcular a execução por processador tempo para 2, 4, 8, 16, 32, 64 e 128 processadores. Para cada caso, liste os correspondentes aceleração em relação a um único processador e a razão entre a aceleração real versus aceleração ideal (aceleração se não houvesse sobrecarga).

§1.1, página 10: Perguntas para discussão: muitas respostas são aceitáveis.
§1.4, página 24: Memória DRAM: volátil, tempo de acesso curto de 50 a 70 nanossegundos, e o custo por GB é de US $ 5 a US $ 10. Memória em disco: não volátil, os tempos de acesso são 100.000 400.000 vezes mais lento que a DRAM, e o custo por GB é 100 vezes mais barato que DRAM. Memória Flash: não volátil, os tempos de acesso são 100 a 1000 vezes mais lentos do que DRAM, e o custo por GB é 7 a 10 vezes mais barato que DRAM.

§1.5, página 28: 1, 3 e 4 são razões válidas. A resposta 5 pode ser geralmente verdadeira porque alto volume pode fazer um investimento extra para reduzir o tamanho da matriz em, digamos, 10% decisão econômica, mas não precisa ser verdadeira.
§1.6, página 33: 1. a: ambos, b: latência, c: nenhum. 7 segundos.
§1.6, página 40: b.
§1.10, página 51: a. O computador. A possui a classificação MIPS mais alta. b. Computador B é mais rápido.