CAPÍTULO 1

Fundamentos do projeto e análise quantitativos

"Eu acho justo dizer que os computadores pessoais se tornaram a ferramenta mais poderosa que já criamos. Eles são ferramentas de comunicação, são ferramentas de criatividade e podem ser moldados por seu usuário."

Bill Gates, 24 de fevereiro de 2004

1.1 Introdução 1
1.2 Classes de computadores 4
1.3 Definição da arquitetura do computador 9
1.4 Tendências na tecnologia 14
1.5 Tendências na alimentação dos circuitos integrados 19
1.6 Tendências no custo 24
1.7 Dependência 30
1.8 Medição, relatório e resumo do desempenho 32
1.9 Princípios quantitativos do projeto de computadores 39
1.10 Juntando tudo: desempenho e preço-desempenho 46
1.11 Falácias e armadilhas 48
1.12 Comentários finais 52
1.13 Perspectivas históricas e referências 54
Estudos de caso e exercícios por Diana Franklin 54

1.1 INTRODUÇÃO

A tecnologia de computação fez um progresso incrível no decorrer dos últimos 65 anos, desde que foi criado o primeiro computador eletrônico de uso geral. Hoje, por menos de US$ 500 se compra um computador pessoal com mais desempenho, mais memória principal e mais armazenamento em disco do que um computador comprado em 1985 por US$ 1 milhão. Essa melhoria rápida vem tanto dos avanços na tecnologia usada para montar computadores quanto da inovação no projeto de computadores.

Embora as melhorias tecnológicas tenham sido bastante estáveis, o progresso advindo de arquiteturas de computador aperfeiçoadas tem sido muito menos consistente. Durante os primeiros 25 anos de existência dos computadores eletrônicos, ambas as forças fizeram uma importante contribuição, promovendo a melhoria de desempenho de cerca de 25% por ano. O final da década de 1970 viu o surgimento do microprocessador. A capacidade do microprocessador de acompanhar as melhorias na tecnologia de circuito integrado

levou a uma taxa de melhoria mais alta — aproximadamente 35% de crescimento por ano, em desempenho.

Essa taxa de crescimento, combinada com as vantagens do custo de um microprocessador produzido em massa, fez com que uma fração cada vez maior do setor de computação fosse baseada nos microprocessadores. Além disso, duas mudanças significativas no mercado de computadores facilitaram, mais do que em qualquer outra época, o sucesso comercial com uma nova arquitetura: 1) a eliminação virtual da programação em linguagem Assembly reduziu a necessidade de compatibilidade de código-objeto; 2) a criação de sistemas operacionais padronizados, independentes do fornecedor, como UNIX e seu clone, o Linux, reduziu o custo e o risco de surgimento de uma nova arquitetura.

Essas mudanças tornaram possível o desenvolvimento bem-sucedido de um novo conjunto de arquiteturas com instruções mais simples, chamadas arquiteturas RISC (Reduced Instruction Set Computer — computador de conjunto de instruções reduzido), no início da década de 1980. As máquinas baseadas em RISC chamaram a atenção dos projetistas para duas técnicas críticas para o desempenho: a exploração do *paralelismo em nível de instrução* (inicialmente por meio do pipelining e depois pela emissão de múltiplas instruções) e o uso de caches (inicialmente em formas simples e depois usando organizações e otimizações mais sofisticadas).

Os computadores baseados em RISC maximizaram o padrão de desempenho, forçando as arquiteturas anteriores a acompanhar esse padrão ou a desaparecer. O Vax da Digital Equipment não fez isso e, por essa razão, foi substituído por uma arquitetura RISC. A Intel acompanhou o desafio, principalmente traduzindo instruções 80x86 (ou IA-32) para instruções tipo RISC, internamente, permitindo a adoção de muitas das inovações pioneiras nos projetos RISC. À medida que a quantidade de transistores aumentava no final dos anos 1990, o overhead do hardware para traduzir a arquitetura x86 mais complexa tornava-se insignificante. Em aplicações específicas, como telefones celulares, o custo com relação à potência e à área de silício relativo ao overhead da tradução do x86 ajudou uma arquitetura RISC, a ARM, a se tornar dominante.

A Figura 1.1 mostra que a combinação de melhorias na organização e na arquitetura dos computadores fez com que o crescimento do desempenho fosse constante durante 17 anos, a uma taxa anual de mais de 50% — ritmo sem precedentes no setor de computação.

Quatro foram os impactos dessa notável taxa de crescimento no século XX. Primeiro, ela melhorou consideravelmente a capacidade disponível aos usuários de computador. Para muitas aplicações, os microprocessadores de desempenho mais alto de hoje ultrapassam o supercomputador de menos de 10 anos atrás.

Em segundo lugar, essa melhoria drástica em custo/desempenho levou a novas classes de computadores. Os computadores pessoais e workstations emergiram nos anos 1980 com a disponibilidade do microprocessador. A última década viu o surgimento dos smartphones e tablets, que muitas pessoas estão usando como plataformas primárias de computação no lugar dos PCs. Esses dispositivos clientes móveis estão usando a internet cada vez mais para acessar depósitos contendo dezenas de milhares de servidores, que estão sendo projetados como se fossem um único gigantesco computador.

Em terceiro lugar, a melhoria contínua da fabricação de semicondutores, como previsto pela lei de Moore, levou à dominância de computadores baseados em microprocessadores por toda a gama de projetos de computador. Os minicomputadores, que tradicionalmente eram feitos a partir de lógica pronta ou de gate arrays, foram substituídos por servidores montados com microprocessadores. Os mainframes foram praticamente substituídos por

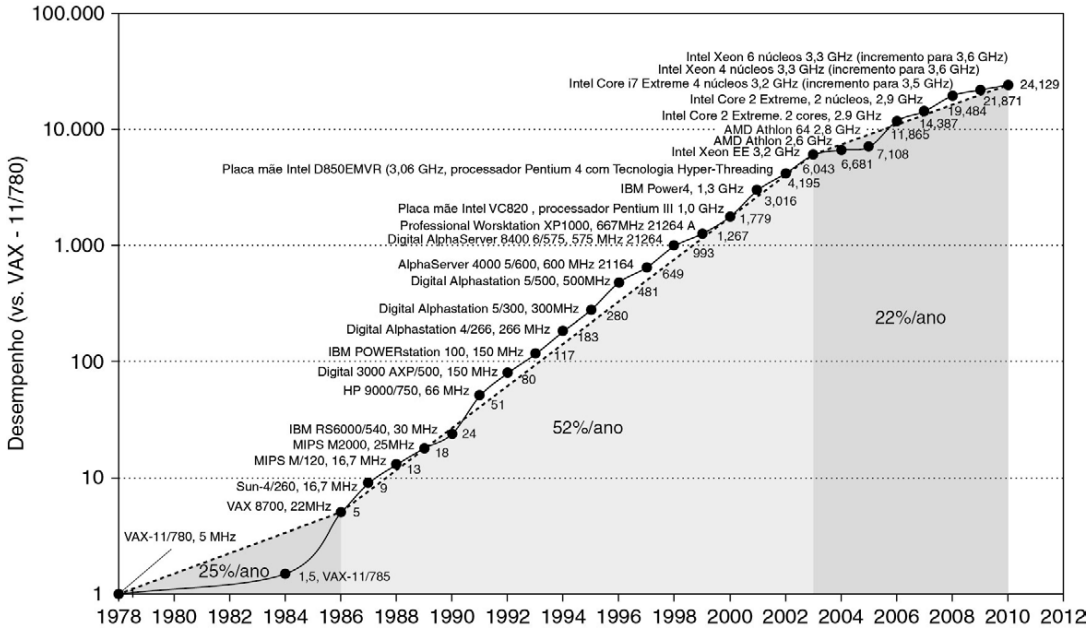

FIGURA 1.1 Crescimento no desempenho do processador desde o fim da década de 1970.

Este gráfico mostra o desempenho relativo ao VAX 11/780, medido pelos benchmarks SPECint (Seção 1.8). Antes de meados da década de 1980, o crescimento no desempenho do processador era, em grande parte, controlado pela tecnologia e, em média, era de 25% por ano. O aumento no crescimento, para cerca de 52% desde então, é atribuído a ideias arquitetônicas e organizacionais mais avançadas. Em 2003, esse crescimento levou a uma diferença no desempenho de cerca de um fator de 25 *versus* se tivéssemos continuado com a taxa de 25%. O desempenho para cálculos orientados a ponto flutuante aumentou ainda mais rapidamente. Desde 2003, os limites de potência, paralelismo disponível em nível de instrução e latência longa da memória reduziram o desempenho do uniprocessador para não mais de 22% por ano ou cerca de cinco vezes mais lento do que se tivéssemos continuado com 52% ao ano. (O desempenho SPEC mais rápido desde 2007 teve a paralelização automática ativada, com um número cada vez maior de núcleos por chip a cada ano, então a velocidade do uniprocessador é difícil de medir. Esses resultados se limitam a sistemas de soquete único para reduzir o impacto da paralelização automática.) A Figura 1.11, na página 22, mostra a melhoria nas taxas de clock para essas mesmas três eras. Como o SPEC foi alterado no decorrer dos anos, o desempenho das máquinas mais novas é estimado por um fator de escala que relaciona o desempenho para duas versões diferentes do SPEC (por exemplo, SPEC89, SPEC92, SPEC95, SPEC2000 e SPEC2006).

um pequeno número de microprocessadores encapsulados. Até mesmo os supercomputadores de ponta estão sendo montados com grupos de microprocessadores.

Essas inovações de hardware levaram ao renascimento do projeto de computadores, que enfatizou tanto a inovação arquitetônica quanto o uso eficiente das melhorias da tecnologia. Essa taxa de crescimento foi aumentada de modo que, em 2003, os microprocessadores de alto desempenho eram cerca de 7,5 vezes mais rápidos do que teriam alcançado contando-se apenas com a tecnologia, incluindo a melhoria do projeto do circuito. Ou seja, 52% ao ano *versus* 35% ao ano.

O renascimento do hardware levou ao quarto impacto sobre o desenvolvimento de software. Essa melhoria de 25.000 vezes no desempenho desde 1978 (Fig. 1.1) permitiu aos programadores da atualidade trocar o desempenho pela produtividade. Em vez de utilizar linguagens orientadas ao desempenho, como C e C++, hoje as programações utilizam mais as linguagens, como Java e C#, chamadas de *managed programming languages*. Além do mais, linguagens script, como Python e Ruby, que são ainda mais produtivas, estão ganhando popularidade juntamente com frameworks de programação, como Ruby on Rails. Para manter a produtividade e tentar eliminar o problema do desempenho, os interpretadores com compiladores just-in-time e compilação trace-based estão substituindo os

compiladores e o linkers tradicionais do passado. A implementação de software também está mudando, com o *software como serviço* (Software as a Service — SaaS) usado na internet, substituindo os softwares comprados em uma mídia (shirink-wrapped software), que devem ser instalados e executados em um computador local.

A natureza das aplicações também muda. Fala, som, imagens e vídeo estão tornando-se cada vez mais importantes, juntamente com o tempo de resposta previsível, tão crítico para o usuário. Um exemplo inspirador é o Google Goggles. Esse aplicativo permite apontar a câmera do telefone celular para um objeto e enviar a imagem pela internet sem fio para um computador em escala wharehouse, que reconhece o objeto e dá informações interessantes sobre ele. O aplicativo pode traduzir textos do objeto para outro idioma, ler o código de barras da capa de um livro e dizer se ele está disponível on-line e qual é o seu preço ou, se fizer uma panorâmica com a câmera do celular, dizer quais empresas estão próximas a você, quais são seus sites, números telefônicos e endereços.

Porém, a Figura 1.1 também mostra que esse renascimento de 17 anos acabou. Desde 2003, a melhoria de desempenho dos uniprocessadores únicos caiu para cerca de 22% por ano, devido tanto à dissipação máxima de potência dos chips resfriados a ar como à falta de maior paralelismo no nível de instrução que resta para ser explorado com eficiência. Na realidade, em 2004, a Intel cancelou seus projetos de uniprocessadores de alto desempenho e juntou-se a outras empresas ao mostrar que o caminho para um desempenho mais alto seria através de vários processadores por chip, e não de uniprocessadores mais rápidos.

Isso sinaliza uma passagem histórica, de contar unicamente com o *paralelismo em nível de instrução* (Instruction-Level Parallelism — ILP), foco principal das três primeiras edições deste livro, para contar com o *paralelismo em nível de thread* (Thread-Level Parallelism — TLP) e o *paralelismo em nível de dados* (Data-Level Parallelism — DLP), que são abordados na quarta edição e expandidos nesta. Esta edição também inclui computadores em escala wharehouse. Embora o compilador e o hardware conspirem para explorar o ILP implicitamente sem a atenção do programador, DLP, TLP e RLP são explicitamente paralelos, exigindo a reestruturação do aplicativo para que ele possa explorar o paralelismo explícito. Em alguns casos, isso é fácil. Em muitos, é uma nova grande carga para os programadores.

Este capítulo focaliza as ideias arquitetônicas e as melhorias no compilador que as acompanham e que possibilitaram a incrível taxa de crescimento no século passado, além dos motivos para a surpreendente mudança e os desafios e enfoques promissores iniciais para as ideias arquitetônicas e compiladores para o século XXI. No centro está o enfoque quantitativo para o projeto e a análise de compilador, que usa observações empíricas dos programas, experimentação e simulação como ferramentas. Esse estilo e esse enfoque do projeto de computador são refletidos neste livro. O objetivo, aqui, é estabelecer a base quantitativa na qual os capítulos e apêndices a seguir se baseiam.

Este livro foi escrito não apenas para explorar esse estilo de projeto, mas também para estimulá-lo a contribuir para esse progresso. Acreditamos que essa técnica funcionará para computadores explicitamente paralelos do futuro, assim como funcionou para os computadores implicitamente paralelos do passado.

1.2 CLASSES DE COMPUTADORES

Essas alterações prepararam o palco para uma mudança surpreendente no modo como vemos a computação, nas aplicações computacionais e nos mercados de computadores, neste novo século. Nunca, desde a criação do computador pessoal, vimos mudanças tão notáveis em como os computadores se parecem e como são usados. Essas mudanças no uso