

# 6

*Combinar largura de banda e armazenamento... permite acesso veloz e confiável às trovas de conteúdo em expansão nos discos e... repositórios que se proliferam na Internet.*

**George Gilder.**  
*The End is Drawing Nigh*, 2000

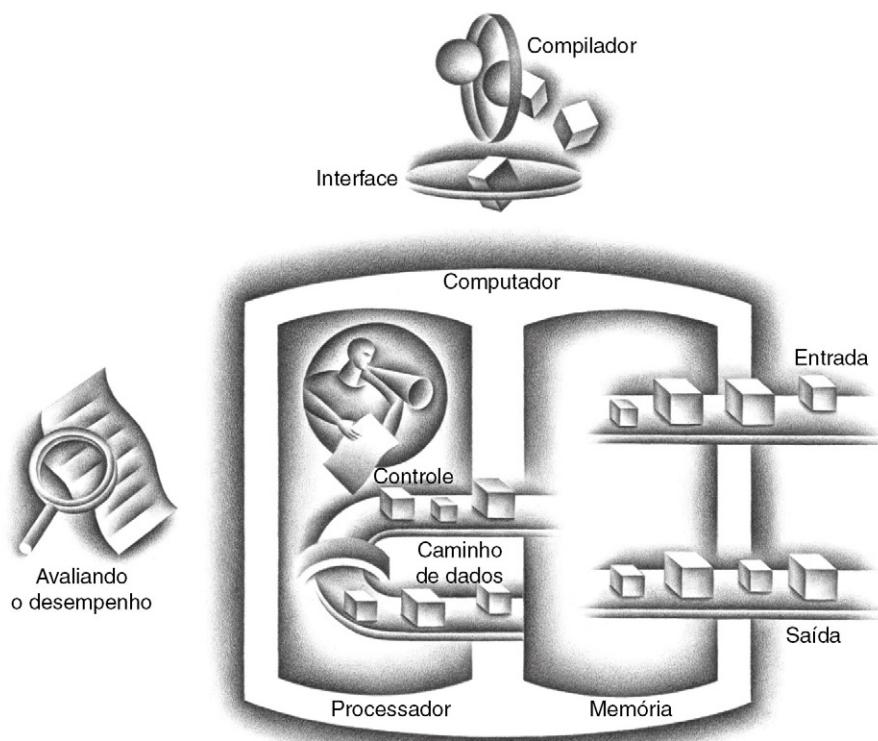
## **Armazenamento e outros tópicos de E/S**

- 6.1      Introdução    460**
- 6.2      Confiança, confiabilidade e disponibilidade    462**
- 6.3      Armazenamento em disco    464**
- 6.4      Armazenamento flash    468**
- 6.5      Conectando processadores, memória  
e dispositivos de E/S    469**
- 6.6      Interface dos dispositivos de E/S  
com processador, memória e sistema  
operacional    473**

- 6.7 Medidas de desempenho de E/S: exemplos de sistemas de disco e de arquivos 480**
- 6.8 Projetando um sistema de E/S 482**
- 6.9 Paralelismo e E/S: Redundant Arrays of Inexpensive Disks (RAID) 483**
- 6.10 Vida real: servidor Sun Fire x4150 488**
- 6.11 Tópicos avançados: redes 494**
- 6.12 Falácia e armadilhas 494**
- 6.13 Comentários finais 498**
- 6.14 Perspectiva histórica e leitura adicional 498**
- 6.15 Exercícios 499**

---

## Os cinco componentes clássicos de um computador



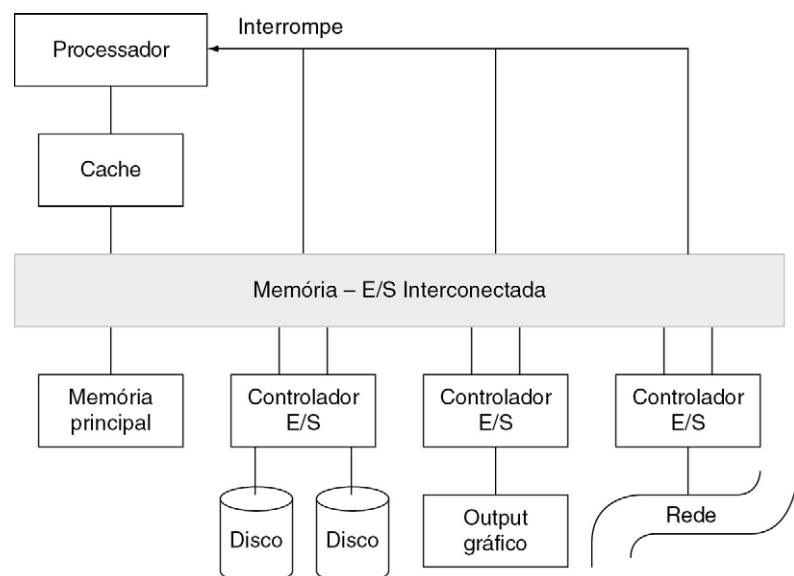
## 6.1 Introdução

Embora os usuários possam se frustrar se seus computadores travarem e tiverem de ser reinicializados, eles ficam irados se seu sistema de armazenamento falhar e informações forem perdidas. Assim, a tolerância à confiabilidade é muito mais alta em relação ao armazenamento do que à computação. As redes também são planejadas para tratar falhas na comunicação, incluindo diversos mecanismos para detectar e recuperar-se de tais falhas. Logo, os sistemas de E/S geralmente colocam muito mais ênfase sobre a confiabilidade e o custo, enquanto os processadores e a memória focalizam o desempenho e o custo.

Os sistemas de E/S também precisam planejar a facilidade de expansão e a diversidade de dispositivos, o que não é um problema para os processadores. A facilidade de expansão está relacionada à capacidade de armazenamento, que é outro parâmetro de projeto para os sistemas de E/S; os sistemas podem precisar de um limite inferior de capacidade de armazenamento a fim de cumprir seu papel.

Embora o desempenho tenha um papel secundário para E/S, ele é mais complexo. Por exemplo, com alguns dispositivos, precisamos cuidar principalmente da latência de acesso, enquanto em outros a vazão é fundamental. Além do mais, o desempenho depende de muitos aspectos do sistema, de características dos dispositivos, da conexão entre o dispositivo e o resto do sistema, da hierarquia de memória e do sistema operacional. Todos os componentes, dos dispositivos de E/S individuais ao processador e software de sistemas, afetarão a confiabilidade, a facilidade de expansão e o desempenho de tarefas que incluem E/S. A [Figura 6.1](#) mostra a estrutura de um sistema simples com sua E/S.

Os dispositivos de E/S são incrivelmente diversificados. Três características são úteis na organização dessa grande variedade:



**FIGURA 6.1 Uma coleção típica de dispositivos de E/S.** As conexões entre os dispositivos de E/S, processador e memória normalmente são chamadas de *barramentos*, embora o termo signifique fios paralelos compartilhados e a maioria das conexões de E/S hoje seja mais próxima de linhas seriais dedicadas. A comunicação entre os dispositivos e o processador utiliza interrupções e protocolos na interconexão, conforme veremos neste capítulo. A [Figura 6.9](#) mostra a organização para um PC desktop.

- *Comportamento*: entrada (somente leitura), saída (somente escrita, não pode ser lido) ou armazenamento (pode ser relido e normalmente reescrito).
- *Parceria*: um humano ou uma máquina está na outra extremidade do dispositivo de E/S, seja alimentando a entrada de dados ou lendo-os na saída.
- *Taxa de dados*: a taxa de pico em que os dados podem ser transferidos entre o dispositivo de E/S e a memória principal ou processador. É útil saber qual é a demanda máxima que o dispositivo pode gerar ao projetar um sistema de E/S.

Por exemplo, um teclado é um dispositivo de *entrada* usado por um *humano* com uma *taxa de dados máxima* de 10 bytes por segundo. A [Figura 6.2](#) mostra alguns dos dispositivos de E/S conectados aos computadores.

No Capítulo 1, vimos rapidamente quatro dispositivos de E/S importantes: mouses, monitores gráficos, discos e redes. Neste capítulo, vamos nos aprofundar no armazenamento e nos itens relacionados. No site, há uma seção de tópicos avançados sobre redes, que também são tratadas em outros livros.

O modo como devemos avaliar o desempenho da E/S normalmente depende da aplicação. Em alguns ambientes, podemos nos importar principalmente com a vazão do sistema. Nesses casos, a largura de banda de E/S será mais importante. Até mesmo a largura de banda de E/S pode ser medida de duas maneiras diferentes:

1. Quantos dados podemos mover pelo sistema em determinado momento?
2. Quantas operações de E/S podemos realizar por unidade de tempo?

A decisão sobre a melhor medida de desempenho pode depender do ambiente. Por exemplo, em muitas aplicações de multimídia, a maioria das requisições de E/S é para fluxos de dados longos, e a largura de banda de transferência é a característica importante. Em outro ambiente, podemos querer processar um número maior de acessos pequenos e não relacionados a um dispositivo de E/S. Um exemplo desse ambiente poderia ser um escritório de processamento de impostos do National Income Tax Service (NITS). O NITS cuida principalmente do processamento de uma grande quantidade de formulários em determinado momento; cada formulário de imposto é armazenado separadamente

Dispositivo	Comportamento	Parceiro	Taxa de dados (Mbits/seg)
Teclado	Entrada	humano	0,0001
Mouse	Entrada	humano	0,0038
Entrada de voz	entrada	humano	0,2640
Entrada de som	entrada	máquina	3,0000
Scanner	entrada	humano	3,2000
Saída de voz	saída	humano	0,2640
Saída de som	saída	humano	8,0000
Impressora a laser	saída	humano	3,2000
Monitor gráfico	saída	humano	800,0000-8000,0000
Modem a cabo	Entrada ou saída	máquina	0,1280-6,0000
Rede/LAN	Entrada ou saída	máquina	100,0000-10000,0000
Rede/LAN sem fio	Entrada ou saída	máquina	11,0000-54,0000
Disco óptico	Armazenamento	máquina	80,0000-220,0000
Fita magnética	Armazenamento	máquina	5,0000-120,0000
Memória flash	Armazenamento	máquina	32,0000-200,0000
Disco magnético	Armazenamento	máquina	800,0000-3000,0000

**FIGURA 6.2 A diversidade de dispositivos de E/S.** Os dispositivos de E/S podem ser distinguidos analisando se servem como dispositivos de entrada, saída ou armazenamento; seu parceiro de comunicação (pessoas ou outros computadores); e suas taxas de comunicação máximas. As taxas de dados se espalham por oito ordens de grandeza. Observe que uma rede pode ser um dispositivo de entrada ou saída, mas não pode ser usada para armazenamento. As taxas de transferência dos dispositivos sempre são indicadas na base 10, de modo que 10 Mbits/seg = 10.000.000 bits/seg.

**requisições de E/S** Leituras ou escritas em dispositivos de E/S.

e é muito pequeno. Um sistema orientado para transferência de arquivos grandes pode ser satisfatório, mas um sistema de E/S que possa admitir a transferência simultânea de muitos arquivos pequenos pode ser mais barato e mais rápido para processar milhões de formulários de imposto.

Em outras aplicações, importamo-nos principalmente com o tempo de resposta, que, como você deve se lembrar, é o tempo total gasto para realizar uma tarefa em particular. Se as **requisições de E/S** forem extremamente grandes, o tempo de resposta dependerá muito da largura de banda, mas em muitos ambientes a maioria dos acessos será pequena, e o sistema de E/S com a menor latência por acesso oferecerá o melhor tempo de resposta. Em máquinas de monousuário, como computadores desktop e laptops, o tempo de resposta é a principal característica do desempenho.

Uma grande quantidade de aplicações, especialmente no vasto mercado comercial para a computação, exige alta vazão e pouco tempo de resposta. Alguns exemplos incluem caixas eletrônicos de banco, sistemas de entrada de pedidos e acompanhamento de estoque, servidores de arquivos e servidores Web. Nesses ambientes, preocupamo-nos com o tempo usado para cada tarefa e quantas tarefas podemos processar em um segundo. A quantidade de solicitações de caixas eletrônicos que você pode processar por hora não importa se cada uma exige 15 minutos – você ficará sem clientes! De modo semelhante, se você puder processar cada solicitação dos caixas eletrônicos rapidamente, mas só pode lidar com uma pequena quantidade de requisições ao mesmo tempo, não poderá dar suporte a muitos caixas eletrônicos, ou então o custo do computador por caixa eletrônico será muito alto.

Resumindo, as três classes, desktops, servidores e computadores embutidos são sensíveis à confiabilidade e ao custo da E/S. Sistemas de desktop e sistemas embutidos se concentram mais no tempo de resposta e na diversidade dos dispositivos de E/S, enquanto sistemas servidores focalizam mais a vazão e a facilidade de expansão dos dispositivos de E/S.

## 6.2

## Confiança, confiabilidade e disponibilidade

Os usuários imploram por armazenamento confiável, mas como podemos definir isso? Na indústria de computação, a questão é mais difícil do que consultar o dicionário. Após um considerável debate, a definição considerada padrão é a seguinte (Laprie, 1985):

*Confiança de um sistema computacional é a qualidade do serviço entregue de modo que a confiança possa ser justificadamente depositada sobre esse serviço. O serviço entregue por um sistema é o seu comportamento real observado como percebido por outro(s) sistema(s) interagindo com os usuários desse sistema. Cada módulo possui um comportamento especificado ideal, no qual uma especificação de serviço é uma descrição combinada do comportamento esperado. Uma falha do sistema ocorre quando o comportamento real se desvia do comportamento especificado.*

Assim, você precisa que uma especificação de referência do comportamento esperado seja capaz de determinar a confiança. Os usuários podem, então, ver um sistema alternando entre dois estados de serviço fornecido com relação à especificação deste:

1. *Realização do serviço*, na qual o serviço é entregue conforme especificado.
2. *Interrupção do serviço*, na qual o serviço entregue é diferente do serviço especificado.

As transições do estado 1 para o estado 2 são causadas por falhas, e as transições do estado 2 para o estado 1 são causadas por *restaurações*. As falhas podem ser permanentes ou intermitentes. O último é o caso mais difícil de diagnosticar quando um sistema oscila entre os dois estados; as falhas permanentes são muito mais fáceis de diagnosticar. Essa definição ocasiona dois termos relacionados: confiabilidade e disponibilidade.

*Confiabilidade* é uma medida da realização contínua do serviço – ou, de forma equivalente, do tempo para a falha – de um ponto de referência. Logo, o *tempo médio para a falha* (MTTF) dos discos na Figura 6.5 é uma medida de confiabilidade. Um termo relacionado é a *taxa de falha anual* (AFR), que é simplesmente a porcentagem dos dispositivos que falhariam em um ano para determinado MTTF. A interrupção do serviço é medida como o *tempo médio para o reparo* (MTTR). O *tempo médio entre falhas* (MTBF) é simplesmente a soma MTTF + MTTR. Embora o MTBF seja muito utilizado, o MTTF normalmente é o termo mais apropriado.

*Disponibilidade* é uma medida da realização do serviço com relação à alternância entre os dois estados de realização e interrupção. A disponibilidade é quantificada estaticamente como

$$\text{Disponibilidade} = \frac{\text{MTTF}}{(\text{MTTF} + \text{MTTR})}$$

Observe que a confiabilidade e a disponibilidade são medidas quantificáveis, e não apenas sinônimos de confiança.

Qual é a causa das falhas? A Figura 6.3 resume muitos documentos que coletaram dados sobre motivos para falhas de sistemas computacionais e sistemas de telecomunicações. Logicamente, os operadores humanos são uma fonte de falhas significativa.

Operador	Software	Hardware	Sistema	Ano do dado coletado
42%	25%	18%	Centro de dados (Tandem)	1985
15%	55%	14%	Centro de dados (Tandem)	1989
18%	44%	39%	Centro de dados (DEC VAX)	1985
50%	20%	30%	Centro de dados (DEC VAX)	1993
50%	14%	19%	Rede telefônica pública dos EUA	1996
54%	7%	30%	Rede telefônica pública dos EUA	2000
60%	25%	15%	Serviços de internet	2002

**FIGURA 6.3 Resumo dos estudos dos motivos para falhas.** Embora seja difícil coletar dados para determinar se os operadores são a causa dos erros, como os operadores normalmente registram os motivos para as falhas, esses estudos capturaram esses dados. Constantemente havia outras categorias, como motivos ambientais para cortes de energia, mas eles em geral eram pequenos. As duas linhas iniciais vêm de um artigo clássico de Jim Gray [1990], que ainda é muito citado, quase 20 anos após a coleta dos dados. As duas linhas seguintes são de um artigo de Murphy e Gent, que estudaram casos de cortes em sistemas VAX com o tempo (“Measuring system and software reliability using an automated data collection process”, *Quality and Reliability Engineering International* 11:5, setembro–outubro de 1995, 341–53). As quinta e sexta linhas são estudos de dados de falhas do FCC sobre a rede telefônica pública dos Estados Unidos, por Kuhn (“Sources of failure in the public switched telephone network”, *IEEE Computer* 30:4, abril de 1997, 31–36) e por Patty Enriquez. O estudo mais recente de três servidores de internet vem de Oppenheimer, Ganapath e Patterson [2003].

Para aumentar o MTTF, você pode melhorar a qualidade dos componentes ou projetar sistemas para que continuem a operação na presença de componentes que falharam. Logo, a falha precisa ser definida em relação a um contexto. Uma falha em um componente pode não ocasionar uma falha do sistema. Para esclarecer essa distinção, o termo *falha* é usado indicando falha de um componente. Aqui estão três maneiras de melhorar o MTTF:

1. *Impedimento de falha*: evitar a ocorrência da falha pela construção.
2. *Tolerância a falhas*: uso de redundância para permitir que o serviço cumpra com a especificação de serviço apesar da ocorrência de falhas, o que se aplica principalmente a falhas do hardware. A Seção 6.9 descreve as técnicas de RAID para tornar o armazenamento confiável por meio da tolerância a falhas.
3. *Previsão de falha*: prever a presença e criação de falhas, o que se aplica a falhas do hardware e do software, permitindo que o componente seja substituído antes de falhar.

**Verifique  
você mesmo**

Diminuir o MTTR pode ajudar na disponibilidade tanto quanto aumentar o MTTF. Por exemplo, ferramentas para detecção, diagnóstico e reparo de falhas podem ajudar a reduzir o tempo e reparar falhas ocasionadas por pessoas, software e hardware.

Quais das seguintes afirmações são verdadeiras sobre confiança?

1. Se um sistema estiver ativo, então todos os seus componentes estão realizando seu serviço esperado.
2. A disponibilidade é uma medida quantitativa da porcentagem de tempo em que um sistema está realizando seu serviço esperado.
3. A confiabilidade é uma medida quantitativa da realização contínua do serviço por um sistema.
4. A principal fonte de interrupções hoje é o software.

### 6.3

## Armazenamento em disco

**não volátil** Dispositivo de armazenamento em que os dados retêm seu valor mesmo quando a alimentação é removida.

**trilha** Um dos milhares de círculos concêntricos que compõem a superfície de um disco magnético.

**setor** Um dos segmentos que compõem uma trilha em um disco magnético; um setor é a menor quantidade de informação lida ou escrita em um disco.

**seek** O processo de posicionar uma cabeça de leitura/gravação na trilha correta de um disco.

Como mencionamos no Capítulo 1, os discos magnéticos contam com um prato giratório coberto por uma superfície magnética e utiliza uma cabeça de leitura/escrita móvel para acessar o disco. O armazenamento em disco é **não volátil** – os dados permanecem mesmo quando a alimentação é removida. Um disco magnético consiste em uma coleção de pratos (1-4), cada qual com duas superfícies de disco graváveis. A pilha de pratos gira a uma velocidade entre 5.400 a 15.000RPM e tem um diâmetro entre 2,5cm e 9cm. Cada superfície do disco é dividida em círculos concêntricos, chamados **trilhas**. Normalmente, existem de 10.000 a 50.000 trilhas por superfície. Cada trilha, por sua vez, é dividida em **setores** que contêm as informações; cada trilha pode ter de 100 a 500 setores. Os setores normalmente possuem 512 bytes de tamanho, embora exista uma iniciativa para aumentar o tamanho do setor para 4.096 bytes. A sequência gravada em mídia magnética é um número de setor, um gap, a informação para esse setor incluindo o código de correção de erro (veja  Apêndice C, página C-66), um gap, o número de setor do próximo setor, e assim por diante.

Originalmente, todas as trilhas tinham o mesmo número de setores e, portanto, o mesmo número de bits, mas com a introdução da ZBR (Zone Bit Recording – registro de bits por zona) no início da década de 1990, as unidades de disco passaram para um número variável de setores (portanto, bits) por trilha, em vez de manter constante o espaçamento entre os bits. O ZBR aumenta o número de bits nas trilhas externas e, assim, aumenta a capacidade da unidade.

Como vimos no Capítulo 1, para ler e escrever informações, as cabeças de leitura/escrita precisam ser movidas de modo que estejam sobre o local correto. As cabeças de disco para cada superfície são conectadas e se movem em conjunto, de modo que cada cabeça esteja sobre a mesma trilha de cada superfície. O termo **cilindro** é usado para se referir a todas as trilhas sob as cabeças em determinado ponto para todas as superfícies.

Para acessar dados, o sistema operacional precisa direcionar o disco por um processo em três estágios. O primeiro passo é posicionar a cabeça sobre a trilha apropriada. Essa operação é chamada **seek**, e o tempo para mover a cabeça até a trilha apropriada é chamado *tempo de seek*.

Os fabricantes de disco informam o tempo de seek mínimo, o tempo de seek máximo e o tempo de seek médio em seus manuais. Os dois primeiros são fáceis de medir, mas a média está aberta a interpretações, pois ela depende da distância do seek. Os fabricantes decidiram calcular o tempo de seek médio como a soma do tempo para todos os seeks possíveis dividido pelo número de seeks possíveis. Os tempos de seek médios normalmente são anunciados como entre 3ms a 13ms, mas, dependendo da aplicação e do escalonamento das requisições de disco, o tempo de seek médio real pode ser de apenas 25% a 33% do número anunciado, devido à localidade das referências de disco. Essa localidade surge tanto por causa de acessos sucessivos ao mesmo arquivo quanto porque o sistema operacional tenta escalarizar esses acessos juntos.

Quando a cabeça tiver atingido a trilha correta, temos de esperar até o setor desejado girar sob a cabeça de leitura/escrita. Esse tempo é chamado de **latência rotacional** ou **atraso rotacional**. A latência média para a informação desejada está a meio caminho ao redor do disco. Como os discos giram entre 5.400RPM a 15.000RPM, a latência rotacional média é entre

$$\text{Latência rotacional média} = \frac{0,5 \text{ rotação}}{5400 \text{ RPM}} = \frac{0,5 \text{ rotação}}{5400 \text{ RPM} / \left( 60 \frac{\text{segundos}}{\text{minuto}} \right)} \\ = 0,0056 \text{ segundos} = 5,6 \text{ ms}$$

**latência rotacional** Também chamada de **atraso rotacional**. O tempo exigido para que o setor desejado de um disco gire sob a cabeça de leitura/escrita; normalmente considerado metade do tempo de rotação.

E

$$\text{Latência rotacional média} = \frac{0,5 \text{ rotação}}{15.000 \text{ RPM}} = \frac{0,5 \text{ rotação}}{15.000 \text{ RPM} / \left( 60 \frac{\text{segundos}}{\text{minuto}} \right)} \\ = 0,0020 \text{ segundos} = 2,0 \text{ ms}$$

O último componente de um acesso ao disco, o *tempo de transferência*, é o tempo para transferir um bloco de bits. O tempo de transferência é uma função do tamanho do setor, da velocidade de rotação e da densidade de gravação de uma trilha. As taxas de transferência em 2008 estavam entre 70 e 125MB/seg. A única complicação é que a maioria dos controladores de disco possui uma cache interna que armazena setores enquanto eles passam; as taxas de transferência da cache normalmente são maiores e poderiam chegar até 375MB/seg (3 Gbit/seg) em 2008. Hoje, a maioria das transferências de disco possui o tamanho de múltiplos setores.

Uma *controladora de discos* normalmente trata do controle detalhado do disco e da transferência entre o disco e a memória. A controladora acrescenta o componente final do tempo de acesso ao disco, o *tempo da controladora*, que é o overhead que a controladora impõe na realização do acesso de E/S. O tempo médio para realizar uma operação de E/S consistirá nesses quatro tempos mais qualquer espera que ocorra porque outros processos estão utilizando o disco.

### Tempo de leitura do disco

Qual é o tempo médio para ler ou escrever um setor de 512 bytes em um disco típico girando a 15.000RPM? O tempo de seek médio anunciado é de 4ms, a taxa de transferência é de 100MB/seg e o overhead da controladora é de 0,2ms. Suponha que o disco esteja ocioso, de modo que não existe um tempo de espera.

O tempo médio de acesso ao disco é igual ao Tempo médio de seek + Atraso rotacional médio + Tempo de transferência + Overhead da controladora. Usando o tempo de seek médio anunciado, a resposta é

$$4,0 \text{ ms} + \frac{0,5 \text{ rotação}}{15.000 \text{ RPM}} + \frac{0,5 \text{ KB}}{100 \text{ MB/seg}} + 0,2 \text{ ms} = 4,0 + 2,0 + 0,005 + 0,2 = 6,2 \text{ ms}$$

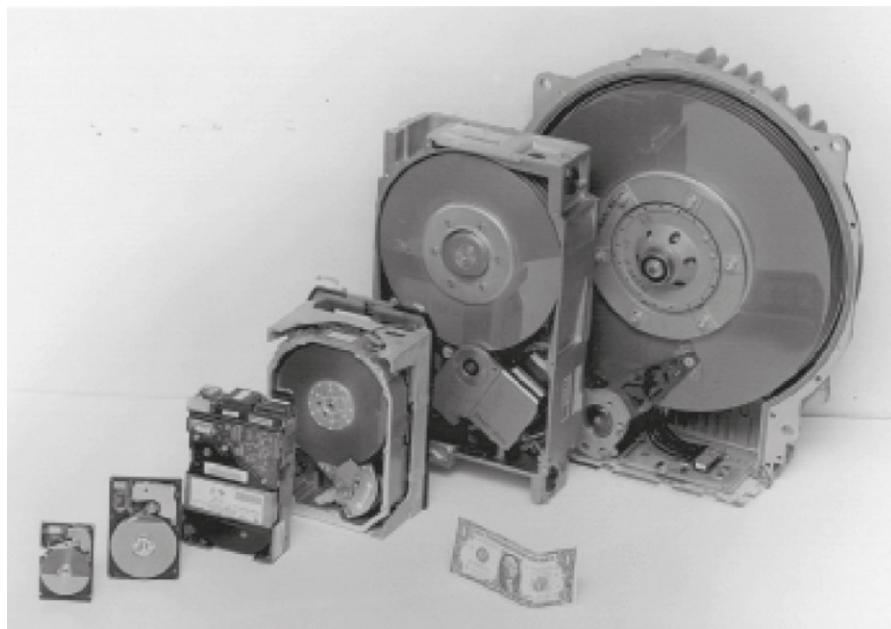
Se o tempo médio de seek medido for 25% do tempo médio anunciado, a resposta é

$$1,0 \text{ ms} + 2,0 \text{ ms} + 0,005 \text{ ms} + 0,2 \text{ ms} = 3,2 \text{ ms}$$

Observe que, quando consideramos o tempo médio de seek medido, ao contrário do tempo médio de seek anunciado, a latência rotacional pode ser o maior componente do tempo de acesso.

### EXEMPLO

### RESPOSTA



**FIGURA 6.4 Seis discos magnéticos, variando em diâmetro de 35cm até 4,5cm.** Os discos da figura foram introduzidos há mais de 15 anos e, portanto, não representam a melhor capacidade dos discos modernos desses mesmos diâmetros. Contudo, essa fotografia representa com precisão seus tamanhos físicos relativos. O maior dos discos é o DEC R81, contendo quatro pratos de 35,5cm de diâmetro e armazenando 456MB. Ele foi fabricado em 1985. O disco com diâmetro de 20cm vem da Fujitsu, e esse disco de 1984 armazena 130MB em seis pratos. O Micropolis RD53 possui cinco pratos de 13,3cm e armazena 85MB. O IBM 0361 também possui cinco pratos, mas possuem apenas 8,8cm de diâmetro. Esse disco de 1988 tem 320MB de capacidade. Em 2008, o disco de 8,8cm mais denso tinha dois pratos e tinha 1TB no mesmo espaço, ocasionando um aumento de densidade de aproximadamente 3000 vezes! O Conner CP 2045 possui dois pratos de 6,35cm, contendo 40MB, e foi fabricado em 1990. O menor disco desta fotografia é o Integral 1820. Esse disco de um único prato de 4,5cm contém 20MB e foi fabricado em 1992.

As densidades de disco têm continuado a aumentar há mais de 50 anos. O impacto dessa melhoria na densidade e na redução do tamanho físico de uma unidade de disco tem sido incríveis, como mostra a [Figura 6.4](#). Os objetivos de diferentes projetistas de discos têm levado a uma grande variedade de unidades disponíveis em determinado momento. A [Figura 6.5](#) mostra as características de quatro discos magnéticos. Em 2008, esses discos de um único fabricante custavam entre US\$0,30 e US\$5 por gigabyte. No mercado mais amplo, os preços geralmente variam entre US\$0,20 e US\$2 por gigabyte, dependendo do tamanho, da interface e do desempenho.

Embora os discos permaneçam viáveis por um futuro previsível, o mesmo não ocorre com a sabedoria convencional sobre onde os números de bloco são encontrados. As suposições do modelo de setor-trilha-cilindro são que os blocos próximos estão na mesma trilha, os blocos no mesmo cilindro levam menos tempo para acessar, pois não existe tempo de seek, e algumas trilhas são mais próximas que outras. O motivo para o desmembramento foi o aumento do nível das interfaces. As interfaces inteligentes de nível mais alto, como **ATA** e **SCSI**, exigiram um microprocessador dentro de um disco, o que leva a otimizações de desempenho.

Para aumentar a velocidade das transferências sequenciais, essas interfaces de nível mais alto organizam os discos mais como fitas do que como dispositivos de acesso aleatório. Os blocos lógicos são ordenados em formato de serpentina por uma única superfície, tentando capturar todos os setores que são gravados na mesma densidade de bits. Portanto, os blocos sequenciais podem estar em trilhas diferentes. Veremos um exemplo, na [Figura 6.19](#), da armadilha de considerar o modelo convencional de setor-trilha-cilindro.

**Advanced Technology Attachment (ATA)** Um conjunto de comandos utilizado como padrão para dispositivos de E/S, que é muito popular no PC.

**Small Computer Systems Interface (SCSI)** Um conjunto de comandos usado como um padrão para dispositivos de E/S.

**Detalhamento:** Essas interfaces de alto nível permitem que as controladoras de disco incluam caches. Essas caches permitem um acesso rápido aos dados lidos recentemente entre trans-

Características	Seagate ST33000655SS	Seagate ST31000340NS	Seagate ST973451SS	Seagate ST9160821AS
Diâmetro do disco (cm)	8,89	8,89	6,35	6,35
Capacidade do disco formatado (GB)	147	1000	73	160
Número de superfícies de disco (cabeças)	2	4	2	2
Velocidade de rotação (RPM)	15.000	7.200	15.000	5.400
Tamanho da cache de disco interna (MB)	16	32	16	8
Interface externa, largura de banda (MB/seg)	SAS, 375	SATA, 375	SAS, 375	SATA, 150
Taxa de transferência sustentada (MB/seg)	73–125	105	79–112	44
Tempo de seek mínimo (leitura/escrita) (ms)	0,2/0,4	0,8/1,0	0,2/0,4	1,5/2,0
Tempo médio de seek para leitura/escrita (ms)	3,5/4,0	8,5/9,5	2,9/3,3	12,5/13,0
Tempo médio para falha (MTTF) (horas)	1.400.000 a 25°C	1.200.000 a 25°C	1.600.000 a 25°C	—
Taxa de falha anual (AFR) (porcentagem)	0,62%	0,73%	0,55%	—
Ciclos de início-fim de contato	—	50.000	—	>600.000
Garantia (anos)	5	5	5	5
Erros de leitura não recuperáveis por bits lidos	< 1 setor por $10^{16}$	< 1 setor por $10^{15}$	< 1 setor por $10^{16}$	< 1 setor por $10^{14}$
Temperatura, limites de vibração (operando)	5°–55°C, 60 G	5°–55°C, 63 G	5°–55°C, 60 G	0°–60°C, 350 G
Tamanho: dimensões (cm), peso (gramas)	2,5 × 10,1 × 14,7, 861,8g	2,5 × 10,1 × 14,7, 635g	1,52 × 7,11 × 9,9, 226g	1,0 × 7,11 × 9,9, 90,6g
Potência: operando/ocioso/standby (watts)	15/11/—	11/8/1	8/5,8/—	1,9/0,6/0,2
GB/pol. cúb., GB/watt	6GB/pol. cúb., 10 GB/W	43 GB/pol. cúb., 91 GB/W	11 GB/pol.cúb., 6 GB/W	37 GB/pol. cúb., 84 GB/W
Preço em 2008, \$/GB	~US\$250, ~US\$1,70/GB	~US\$275, US\$0,30/GB	~US\$350, US\$5,00/GB	~US\$100, US\$0,60/GB

**FIGURA 6.5 Características de quatro discos magnéticos de um único fabricante em 2008.** As três unidades mais à esquerda são para servidores e desktops, enquanto a unidade mais à direita é para laptops. Observe que a terceira unidade tem apenas 6,35cm de diâmetro, mas é uma unidade de alto desempenho com a mais alta confiabilidade e tempo de seek mais rápido. Os discos mostrados aqui são versões seriadas da interface para SCSI (SAS), um barramento de E/S padrão para muitos sistemas, ou a versão serial da ATA (SATA), um barramento de E/S padrão para PCs. A taxa de transferência da cache é de 3-5 vezes mais rápida do que a taxa de transferência da superfície do disco. O custo muito mais baixo por gigabyte da unidade de 8,8cm SATA ocorre principalmente devido ao mercado hipercompetitivo dos PCs, embora existam diferenças em desempenho em E/Ss por segundo devido à rotação mais rápida e tempos de seek mais rápidos para SAS. A vida útil para esses discos é de cinco anos. Observe que o MTTF cotado considera potência e temperatura normais. Os tempos de vida do disco podem ser muito mais curtos se a temperatura e a vibração não forem controlados. Veja o link da Seagate em [www.seagate.com](http://www.seagate.com) a fim de obter mais informações sobre essas unidades.

ferências solicitadas pelo processador. Elas utilizam write-through e não atualizam quando há falha na escrita. Elas normalmente também incluem algoritmos de prefetch para tentar antecipar a demanda. As controladoras também utilizam uma fila de comandos que permite que o disco decida em que ordem irá realizar os comandos para maximizar o desempenho enquanto mantém o comportamento correto. Naturalmente, essas capacidades complicam a medida de desempenho do disco e aumentam a importância da escolha da carga de trabalho na comparação de discos.

Quais dos seguintes itens são verdadeiros sobre unidades de disco?

- Discos de 8,89cm realizam mais E/Ss por segundo que os discos de 6,35cm.
- Discos de 6,35cm oferecem os maiores índices de gigabytes por watt.
- São necessárias horas para ler o conteúdo de um disco de alta capacidade sequencialmente.
- São necessários meses para ler o conteúdo de um disco de alta capacidade usando setores aleatórios de 512 bytes.

**Verifique você mesmo**

## 6.4

## Armazenamento flash

Muitos tentaram inventar uma tecnologia para substituir os discos, e muitos falharam: memória CCD, memória de bolha e memória holográfica, todos ficaram a desejar. Quando uma nova tecnologia era entregue, os discos faziam avanços conforme já era previsto, os custos caíam proporcionalmente, e o produto desafiador ficava pouco atraente no mercado.

O primeiro desafiador convincente é a memória flash. Essa memória semicondutora é não volátil como os discos, mas a latência é 100-1000 vezes mais rápida que o disco, e ela é menor, gasta menos energia e é mais resistente ao choque. Igualmente importante, devido à popularidade da memória flash nos telefones celulares, câmeras digitais e players MP3, existe um grande mercado a pagar pelo investimento na melhoria da tecnologia de memória flash. Recentemente, o custo da memória flash por gigabyte tem caído 50% por ano. Em 2008, o preço por gigabyte da flash era de \$4 a \$10 por gigabyte, ou cerca de 2 a 40 vezes mais alto que o disco e 5 a 10 vezes mais baixo que a DRAM. A [Figura 6.6](#) compara três produtos baseados em flash.

Embora seu custo por gigabyte seja mais alto que os discos, a memória flash é popular nos dispositivos móveis em parte porque vem em capacidades menores. Como resultado, os discos rígidos de 1 polegada de diâmetro estão desaparecendo de alguns mercados de embutidos. Por exemplo, em 2008, o MP3 player iPod Shuffle da Apple era vendido por US\$50 e mantinha 1GB, enquanto o disco menor é de 4GB e é vendido por mais do que o MP3 player inteiro.

A memória flash é um tipo de memória somente de leitura programável e eletricamente apagável (*EEPROM*). A primeira memória flash, chamada *flash NOR* devido à semelhança da célula de armazenamento com uma porta NOR padrão, era um corrente direto com outras EEPROMs, sendo aleatoriamente endereçável, como qualquer memória. Há alguns anos, a memória *flash NAND* oferecia maior densidade de armazenamento, mas a memória só podia ser lida e escrita em blocos, pois a fiação necessária para os acessos aleatórios foi retirada. A *flash NAND* é muito menos dispendiosa por gigabyte e muito mais comum que a *flash NOR*; todos os produtos na [Figura 6.6](#) utilizam *flash NAND*. A [Figura 6.7](#) compara as principais características da memória *flash NOR* versus *NAND*.

Diferente dos discos e da DRAM, mas assim como as tecnologias EEPROM, os bits da memória flash se desgastam (ver [Figura 6.7](#)). A fim de lidar com esses limites, a maioria dos produtos de *flash NAND* inclui um controlador para espalhar as escritas, remapeando blocos que foram escritos muitas vezes para blocos menos utilizados. Essa técnica é chamada de *nivelamento de desgaste*. Com o nivelamento de desgaste, produtos de consumidor

Características	Kingston SecureDigital (SD) SD4/8 GB	Transcend Type I CompactFlash TS16GCF133	RiDATA Solid State Disk 2.5" SATA
Capacidade de dados formatados (GB)	8	16	32
Bytes por setor	512	512	512
Taxa de transferência de dados (leitura/escrita MB/seg)	4	20/18	68/50
Potência de operação/standby (W)	0,66/0,15	0,66/0,15	2,1—
Tamanho: altura × largura × profundidade (cm)	2,39 × 3,2 × 0,2	3,63 × 4,27 × 0,33	0,89 × 6,98 × 10,1
Peso em gramas (454 gramas/libra)	2,5	11,4	52
Tempo médio entre falhas (horas)	> 1.000.000	> 1.000.000	> 4.000.000
GB/pol. cúb., GB/watt	84 GB/pol.cúb., 12 GB/W	51 GB/pol.cúb., 24 GB/W	8 GB/pol.cúb., 16 GB/W
Melhor preço (2008)	~ US\$30	~ US\$70	~ US\$300

**FIGURA 6.6 Características de três produtos de armazenamento flash.** O pacote padrão CompactFlash foi proposto pela Sandisk Corporation em 1994 para as placas PCMCIA-ATA de PCs portáteis. Por seguir a interface ATA, ele simula uma interface de disco, incluindo comandos seek, trilhas lógicas e assim por diante. O produto RiDATA imita uma interface de disco SATA de 2,5 polegadas.

Características	Memória flash NOR	Memória flash NAND
Uso típico	Memória BIOS	Chave USB
Tamanho de acesso mínimo (bytes)	512 bytes	2048 bytes
Tempo de leitura (microsegundos)	0,08	25
Tempo de escrita (microsegundos)	10,00	1500 para apagar + 250
Largura de banda de leitura (MBytes/segundo)	10	40
Largura de banda de escrita (MBytes/segundo)	0,4	8
Desgaste (escritas por célula)	100.000	10.000 a 100.000
Melhor preço/GB (2008)	US\$65	US\$4

**FIGURA 6.7 Características da memória flash NOR versus NAND em 2008.** Estes dispositivos podem ler bytes e palavras de 16 bits apesar de seus tamanhos de acesso grandes.

como telefones celulares, câmeras digitais, MP3 players ou chaves de memória têm muito poucas chances de excederem os limites de escrita na flash. Esses controladores reduzem o desempenho em potencial da flash, mas são necessários, a não ser que o software de nível mais alto monitore o desgaste do bloco. Porém, os controladores também podem melhorar o rendimento, mapeando as células de memória que foram manufaturadas incorretamente.

Os limites de escrita são um motivo para a memória flash não ser comum nos computadores de desktop e servidor. Porém, em 2008, os primeiros laptops estão sendo vendidos com memória flash em vez de discos rígidos, a um custo considerável, para oferecer tempos de boot mais rápidos, tamanho menor e maior vida da bateria. Há também memórias flash disponíveis em tamanhos de disco padrão, como mostra a Figura 6.6. Combinando as duas ideias, os *discos rígidos híbridos* incluem, digamos, um gigabyte de memória flash, de modo que os laptops podem inicializar mais rapidamente e economizar energia, permitindo que os discos permaneçam ociosos com mais frequência.

Nos próximos anos, parece que a memória flash competirá com sucesso com os discos rígidos para muitos dispositivos operados por bateria. À medida que a capacidade aumenta e o custo por gigabyte continua a cair, será interessante ver se o desempenho mais alto e a eficiência de energia da memória flash gerarão oportunidades também nos mercados de desktop e servidor.

Quais dos seguintes itens são verdadeiros sobre a memória flash?

1. Assim como a DRAM, a memória flash é uma memória semicondutora.
2. Assim como os discos, a memória flash não perde informações se faltar energia.
3. O tempo de acesso de leitura da flash NOR é semelhante à DRAM.
4. A largura de banda de leitura da flash NAND é semelhante ao disco.

### Verifique você mesmo

## 6.5

### Conectando processadores, memória e dispositivos de E/S

Em um sistema computacional, os diversos subsistemas precisam ter interfaces entre si. Por exemplo, a memória e o processador precisam se comunicar, assim como o processador e os dispositivos de E/S. Durante muitos anos, isso tem sido feito com um *barramento*. Um barramento é um link de comunicação compartilhado, que utiliza um conjunto de fios para conectar diversos subsistemas. As duas vantagens principais da organização do barramento são versatilidade e baixo custo. Definindo um único esquema de conexão, novos dispositivos podem ser facilmente acrescentados, e os periféricos podem ainda ser

movidos entre os sistemas computacionais que utilizam o mesmo tipo de barramento. Além do mais, os barramentos são eficazes porque um único conjunto de fios é compartilhado de várias maneiras.

A principal desvantagem de um barramento é que ele cria um gargalo de comunicação, possivelmente limitando a vazão máxima de E/S. Quando a E/S tiver de passar por um único barramento, a largura de banda desse barramento limita a vazão máxima da E/S. O principal desafio é projetar um sistema de barramento capaz de atender às demandas do processador e também conectar grandes quantidades de dispositivos de E/S à máquina.

Os barramentos tradicionalmente são classificados como **barramentos processador-memória**, ou *barramentos de E/S*. Os barramentos processador-memória são curtos, geralmente de alta velocidade e correspondentes ao sistema de memória, de modo a maximizar a largura de banda memória-processador. Os barramentos de E/S, ao contrário, podem ser extensos, podem ter muitos tipos de dispositivos conectados a eles e normalmente possuem uma grande faixa de largura de banda de dados dos dispositivos conectados a eles. Os barramentos de E/S normalmente não realizam interface direta com a memória, mas utilizam um barramento processador-memória ou um **barramento backplane** para a conexão com a memória. Outros barramentos com características diferentes surgiram para funções especiais, como barramentos gráficos.

Um motivo para o projeto de barramento ser tão difícil é que sua velocidade máxima é limitada principalmente pelos fatores físicos: a extensão do barramento e o número de dispositivos. Esses limites físicos nos impedem de executar o barramento arbitrariamente rápido. Além disso, a necessidade de dar suporte a uma gama de dispositivos com latências e taxas de transferência de dados muito variáveis também torna o projeto do barramento desafiador.

Como é difícil trabalhar com muitos fios paralelos em alta velocidade devido a variações de clock e reflexão (veja  Apêndice C), o setor está em transição, passando de barramentos paralelos compartilhados para interconexões seriais ponto a ponto de alta velocidade com switches. Assim, essas redes estão gradualmente substituindo os barramentos em nossos sistemas.

Como resultado dessa transição, esta seção foi revisada nesta edição para enfatizar o problema geral de conectar dispositivos de E/S, processadores e memória, em vez de focalizar exclusivamente os barramentos.

## Fundamentos sobre conexão

**transação de E/S** Uma sequência de operações pela interconexão que inclui uma solicitação e pode incluir uma resposta, ambas podendo transportar dados. Uma transação é iniciada por uma única solicitação e pode exigir várias operações de barramento individuais.

Vamos considerar uma **transação de E/S** típica. Uma transação inclui duas partes: enviar o endereço e receber ou enviar os dados. As transações de barramento normalmente são definidas pelo que fazem com a memória. Uma transação de *leitura* transfere dados da memória (para o processador ou para um dispositivo de E/S), e uma transação de *escrita* escreve dados na memória. Logicamente, essa terminologia é confusa. Para evitar isso, vamos tentar usar os termos *entrada* e *saída*, que sempre são definidos do ponto de vista do processador: uma operação de entrada significa entrar dados do dispositivo para a memória, na qual o processador os poderá ler, e uma operação de saída significa sair com dados para um dispositivo a partir da memória, na qual o processador os escreve.

A interconexão de E/S serve como um modo de expandir a máquina e conectar novos periféricos. Para facilitar isso, o setor de computadores desenvolveu diversos padrões. Os padrões servem como uma especificação para o fabricante de computador e para o fabricante de periféricos. Um padrão garante ao projetista do computador que os periféricos estarão disponíveis para uma nova máquina, e garante ao montador do periférico que os usuários poderão se conectar ao seu novo equipamento. A [Figura 6.8](#) resume as principais características dos cinco padrões de E/S dominantes: Firewire, USB, PCI Express (PCIe), serial ATA (SATA) e Serial Attached SCSI (SAS). Eles conectam uma série de dispositivos aos computadores desktop, desde teclados a câmeras e discos.

Características	Firewire (1394)	USB 2.0	PCI Express	Serial ATA	Serial Attached SCSI
Uso intencionado	Externo	Externo	Interno	Interno	Externo
Dispositivos por canal	63	127	1	1	4
Largura básica do barramento de dados (sinais)	4	2	2 por pista	4	4
Largura de banda máxima teórica	50MB/seg (Firewire 400) ou 100MB/seg (Firewire 800)	0,2MB/seg (baixa velocidade), 1,5MB/seg (velocidade plena) ou 60MB/seg (velocidade alta)	250MB/seg por pista (1x); placas PCIe vêm como 1x, 2x, 4x, 8x, 16x ou 32x	300MB/seg	300MB/seg
Conectável mesmo ligado	sim	sim	Depende do <i>form factor</i>	sim	sim
Tamanho máximo do barramento (fio de cobre)	4,5 metros	5 metros	0,5 metro	1 metro	8 metros
Nome do padrão	IEEE 1394, 1394b	USB Implementors Forum	PCI-SIG	SATA-IO	Comitê T10

**FIGURA 6.8 Principais características dos cinco padrões de barramento de E/S dominantes.** A linha de uso intencionado indica se ele foi projetado para ser usado com cabos externos ao computador ou apenas dentro do computador, com cabos curtos ou fio nas placas de circuito impresso. PCIe pode admitir leituras e escritas simultâneas, de modo que muitas publicações dobram a largura de banda por pista, considerando uma divisão 50/50 de largura de banda de leitura *versus* escrita.

Os barramentos tradicionais são **síncronos**. Isso significa que o barramento inclui um clock nas linhas de controle e um protocolo fixo para comunicação que é relativo ao clock. Por exemplo, para realizar uma leitura da memória, poderíamos ter um protocolo que transmite o endereço e comando de leitura no primeiro ciclo de clock, usando as linhas de controle para indicar o tipo de solicitação. A memória poderia então precisar responder com a palavra de dados no quinto clock. Esse tipo de protocolo pode ser implementado com facilidade em uma máquina de estados finitos pequena. Como o protocolo é predeterminado e envolve pouca lógica, o barramento pode executar mais rapidamente, e a lógica da interface será pequena. Entretanto, os barramentos síncronos possuem duas grandes desvantagens. Primeiro, cada dispositivo no barramento precisa executar na mesma velocidade de clock. Segundo, devido a problemas de variação de clock, os barramentos síncronos não podem ser longos se forem rápidos (veja ☀ Apêndice C).

Esses problemas levaram a interconexões **assíncronas**, que não utilizam clock. Por não terem clock, as interconexões assíncronas podem acomodar uma grande variedade de dispositivos, e o barramento pode ser estendido sem preocupação com problemas de variação de clock ou sincronismo. Todos os exemplos da Figura 6.8 são assíncronos.

Para coordenar a transmissão de dados entre o emissor e o receptor, um barramento assíncrono utiliza um **protocolo de handshaking**. Um protocolo de handshaking consiste em uma série de etapas em que o emissor e o receptor prosseguem para a próxima etapa apenas quando as duas partes concordarem. O protocolo é implementado com um conjunto adicional de linhas de controle.

## As interconexões de E/S dos processadores x86

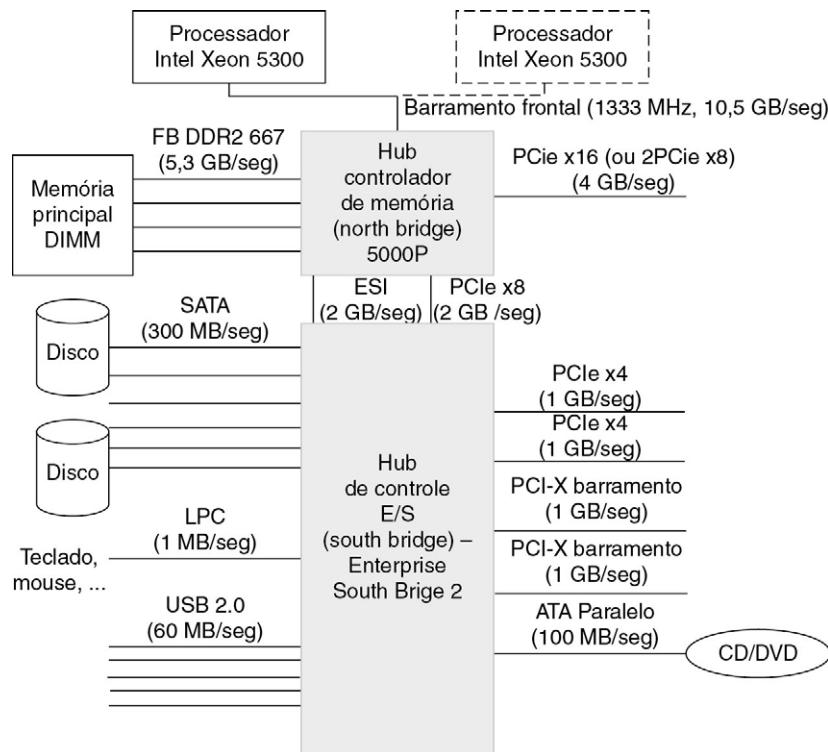
A Figura 6.9 mostra o sistema de E/S de um PC tradicional. O processador se conecta a periféricos por meio de dois chips principais. O chip próximo ao processador é o hub controlador da memória, normalmente chamado *bridge norte*, e aquele conectado a ele é o hub controlador de E/S, chamado de *bridge sul*.

A *bridge norte* é basicamente um controlador de DMA, conectando o processador à memória, possivelmente a uma placa gráfica e ao chip da *bridge sul*. A *bridge sul* conecta

**barramento síncrono** Um barramento que inclui um clock nas linhas de controle e um protocolo fixo para comunicação, relativo ao clock.

**interconexão assíncrona** Utiliza um protocolo de handshaking para coordenar o uso, em vez de um clock; pode acomodar uma grande variedade de dispositivos, de diferentes velocidades.

**protocolo de handshaking** Uma série de etapas usadas para coordenar as transferências em barramentos assíncronos em que o emissor e o receptor só prosseguem para a próxima etapa quando as duas partes concordarem que a etapa atual foi concluída.



**FIGURA 6.9 Organização do sistema de E/S em um servidor Intel usando o chip set Intel 5000P.**

Se você considerar que leituras e escritas são metade do tráfego cada, poderá dobrar a largura de banda por link para PCIe.

a bridge norte a diversos barramentos de E/S. A Intel, AMD, NVIDIA e outros fabricantes oferecem uma grande variedade de chip sets para conectar o processador ao mundo exterior.

A Figura 6.10 mostra três exemplos dos chip sets. Observe que a AMD engoliu o chip da bridge norte no Opteron e outros produtos, reduzindo assim a quantidade de chips e a latência até a memória e placas gráficas, pulando uma travessia de chip.

Visto que a Lei de Moore continua a vigorar, um número cada vez maior de controladoras de E/S, que antes estavam disponíveis como placas opcionais conectadas aos barramentos de E/S, têm sido incorporadas por esses chip sets. Por exemplo, o AMD Opteron X4 e o Intel Nehalem incluem a bridge norte dentro do microprocessador, e o chip da bridge sul do Intel 975 inclui uma controladora RAID (ver Seção 6.9).

Essas interconexões de E/S oferecem conectividade elétrica entre os dispositivos de E/S, processadores e memória, e também definem o protocolo de mais baixo nível para a comunicação. Acima desse nível básico, temos de definir os protocolos de hardware e software a fim de controlar as transferências de dados entre os dispositivos de E/S e a memória, e de modo que o processador especifique comandos aos dispositivos de E/S. Esses assuntos serão abordados na próxima seção.

**Verifique você mesmo** Redes e barramentos conectam componentes. Quais das seguintes afirmações são verdadeiras:

1. As redes e os barramentos de E/S são quase sempre padronizados.
2. As redes e os barramentos de E/S são quase sempre síncronos.

	<b>Chip set Intel 5000P</b>	<b>Chip set Intel 975X</b>	<b>Chip set AMD 580X CrossFire†</b>
Segmento de destino	Servidor	PC com desempenho	Servidor/PC com desempenho
Barramento do sistema (64 bits)	1066/1333 MHz	800/1066MHz	—
<b>Hub controlador de memória (“bridge norte”)</b>			
Nome do produto	Blackbird 5000P MCH	975X MCH	
Pinos	1432	1202	
Tipo e velocidade da memória	DDR2 FBDIMM 667/533	DDR2 800/667/533	
Barramentos de memória, larguras	4 × 72	1 × 72	
Número de DIMMs, DRAM/DIMM	16, 1GB/2GB/4GB	4, 1GB/2GB	
Capacidade máxima da memória	64GB	8GB	
Correção de erro da memória disponível?	sim	não	
PCIe/ Interface Gráfica Externa	1 PCIe x16 ou 2 PCIe x	1 PCIe x16 ou 2 PCIe x8	
Interface da bridge sul	PCIe x8, ESI	PCIe x8	
<b>Hub controlador de E/S (“bridge sul”)</b>			
Nome do produto	6321 ESB	ICH7	580X CrossFire
Tamanho do pacote, pinos	1284	652	549
Barramento PCI: largura, velocidade	Dois 64 bits, 133MHz	32 bits, 33MHz, 6 masters	—
Portas PCI Express	Três PCIe x4		Duas PCIe x16, Quatro PCI x1
Controlador MAC Ethernet, interface	—	1000/100/10Mbps	—
Portas USB 2.0, controladoras	6	8	10
Portas ATA, velocidade	Uma 100	Duas 100	Uma 133
Portas Serial ATA (SATA)	6	2	4
Controlador de áudio AC-97, interface	—	sim	sim
Gerenciamento de E/S	SMbus 2.0, GPIO	SMbus 2.0, GPIO	ASF 2.0, GPIO

**FIGURA 6.10 Dois chip sets de E/S da Intel e um da AMD.** Observe que as funções da bridge norte estão incluídas no microprocessador AMD, pois estão no Intel Nehalem mais recente.

## 6.6

### Interface dos dispositivos de E/S com processador, memória e sistema operacional

Um protocolo de barramento ou de rede define como uma palavra ou bloco de dados devem ser comunicados em um conjunto de fios. Isso ainda deixa várias outras tarefas que precisam ser realizadas para realmente fazer com que os dados sejam transferidos de um dispositivo para o espaço de endereçamento da memória de algum programa de usuário. Esta seção focaliza essas tarefas e responde a perguntas como estas:

- Como uma solicitação de E/S de um usuário é transformada em um comando de dispositivo e comunicada ao dispositivo?
- Como os dados são realmente transferidos de ou para um local da memória?
- Qual é o papel do sistema operacional?

Como veremos na resposta a essas perguntas, o sistema operacional desempenha um papel importante no tratamento da E/S, atuando como interface entre o hardware e o programa que solicita a E/S.

As responsabilidades do sistema operacional surgem de três características dos sistemas de E/S:

1. Diversos programas usando o processador compartilham o sistema de E/S.
2. Os sistemas de E/S normalmente usam interrupções (exceções geradas externamente) para comunicar informações sobre operações de E/S. Como as interrupções causam

uma transferência ao modo kernel ou supervisor, elas precisam ser tratadas pelo sistema operacional (SO).

3. O controle de baixo nível de um dispositivo de E/S é complexo, pois exige o gerenciamento de um conjunto de eventos simultâneos e porque os requisitos para o controle correto do dispositivo normalmente são muito detalhados

## Interface hardware/software

As três características dos sistemas de E/S anteriores levam a diversas funções diferentes que o sistema operacional precisa oferecer:

- O sistema operacional garante que o programa de um usuário acessa apenas as partes de um dispositivo de E/S para as quais o usuário possui direitos. Por exemplo, o sistema operacional não pode permitir que um programa leia ou escreva num arquivo no disco se o proprietário do arquivo não tiver acesso a esse programa. Em um sistema com dispositivos de E/S compartilhados, a proteção não poderia ser fornecida se os programas de usuário pudessem realizar E/S diretamente.
- O sistema operacional oferece abstrações para acessar dispositivos fornecendo rotinas que tratam as operações de baixo nível dos dispositivos.
- O sistema operacional trata as interrupções geradas pelos dispositivos de E/S, assim como trata as exceções geradas por um programa.
- O sistema operacional tenta oferecer acesso equilibrado aos recursos de E/S, além de escalarizar acessos a fim de melhorar a vazão do sistema.

Para realizar essas funções em favor dos programas de usuário, o sistema operacional precisa ser capaz de se comunicar com os dispositivos de E/S e impedir que o programa do usuário se comunique com os dispositivos de E/S diretamente. Três tipos de comunicação são necessários:

1. O sistema operacional precisa ser capaz de dar comandos aos dispositivos de E/S. Esses comandos incluem não apenas operações como ler e escrever, mas também outras operações a serem feitas no dispositivo, como uma busca em um disco.
2. O dispositivo precisa ser capaz de notificar o sistema operacional quando o dispositivo de E/S tiver completado uma operação ou tiver encontrado um erro. Por exemplo, quando um disco completar uma busca, ele notificará o sistema operacional.
3. Os dados precisam ser transferidos entre a memória e um dispositivo de E/S. Por exemplo, o bloco sendo lido em uma leitura de disco precisa ser movido do disco para a memória.

Nas próximas seções, veremos como essas comunicações são realizadas.

## Dando comandos a dispositivos de E/S

Para dar um comando a um dispositivo de E/S, o processador precisa ser capaz de endereçar o dispositivo e fornecer uma ou mais palavras de comando. Dois métodos são usados para endereçar o dispositivo: E/S mapeada em memória e instruções de E/S especiais. Na **E/S mapeada em memória**, partes do espaço de endereçamento são atribuídas a dispositivos de E/S. Leituras e escritas para esses endereços são interpretadas como comandos aos dispositivos de E/S.

Por exemplo, uma operação de escrita pode ser usada para enviar dados a um dispositivo de E/S, em que os dados serão interpretados como um comando. Quando o processador coloca o endereço e os dados no barramento da memória, o sistema de memória ignora a operação, porque o endereço indica uma parte do espaço de memória usado para E/S. O controlador de dispositivos, porém, vê a operação, registra os dados e os transmite

**E/S mapeada em memória** Um esquema de E/S em que partes do espaço de endereçamento são atribuídas a dispositivos de E/S e leituras e escritas para esses endereços são interpretadas como comandos aos dispositivos de E/S.

ao dispositivo como um comando. Os programas de usuário são impedidos de realizar operações de E/S diretamente, pois o sistema operacional não oferece acesso ao espaço de endereçamento atribuído aos dispositivos de E/S e, assim, os endereços são protegidos pela tradução de endereços. A E/S mapeada em memória também pode ser usada para transmitir dados, escrevendo ou lendo para selecionar endereços. O dispositivo utiliza o endereço para determinar o tipo de comando, e os dados podem ser fornecidos por uma escrita ou obtidos por uma leitura. Em qualquer evento, o endereço codifica a identidade do dispositivo e o tipo de transmissão entre o processador e o dispositivo.

Na realidade, fazer uma leitura ou escrita de dados para cumprir uma solicitação do programa normalmente exige várias operações de E/S separadas. Além do mais, o processador pode ter de interrogar o status do dispositivo entre comandos individuais para determinar se o comando foi concluído com sucesso. Por exemplo, uma simples impressora possui dois registradores de dispositivo de E/S – um para informações de status e um para dados a serem impressos. O registrador de status contém um *bit de pronto*, ligado pela impressora quando ela tiver impresso um caractere, e um *bit de erro*, indicando que a impressora está com papel preso ou sem papel. Cada byte de dados a ser impresso é colocado no registrador de dados. O processador precisa, então, esperar até que a impressora ligue o bit pronto antes que possa colocar outro caractere no buffer. O processador também precisa verificar o bit de erro para determinar se houve um problema. Cada uma dessas operações exige um acesso separado ao dispositivo de E/S.

**Detalhamento:** A alternativa à E/S mapeada em memória é usar **instruções de E/S** dedicadas no processador. Essas instruções de E/S podem especificar o número do dispositivo e a palavra de comando (ou o local da palavra de comando na memória). O processador comunica o endereço do dispositivo por meio de um conjunto de fios normalmente incluídos como parte do barramento de E/S. O comando real pode ser transmitido pelas linhas de dados do barramento. Exemplos de computadores com instruções de E/S são os computadores Intel x86 e o IBM 370. Tornando as instruções de E/S ilegais para serem executadas quando fora do modo kernel ou supervisor, os programas de usuário são impedidos de acessar os dispositivos diretamente.

**instrução de E/S** Uma instrução dedicada, usada para dar um comando a um dispositivo de E/S e que especifica o número do dispositivo e a palavra de comando (ou o local da palavra de comando na memória).

## Comunicação com o processador

O processo de verificar periodicamente os bits de status para ver se é hora da próxima operação de E/S, como no exemplo anterior, é chamado de **polling**. O polling é a forma mais simples para um dispositivo de E/S se comunicar com o processador. O dispositivo de E/S simplesmente coloca a informação no registrador de status, e o processador deve vir e apanhar a informação. O processador está totalmente no controle e realiza todo o trabalho.

O polling pode ser usado de várias maneiras diferentes. As aplicações embutidas de tempo real sondam os dispositivos de E/S porque as taxas de E/S são predeterminadas e isso torna o overhead da E/S mais previsível, o que é útil para tempo real. Como veremos, isso permite que o polling seja usado mesmo quando a taxa de E/S é um pouco maior.

A desvantagem do polling é que ele pode desperdiçar muito tempo de processador, pois os processadores são muito mais rápidos do que os dispositivos de E/S. O processador pode ler o registrador de status muitas vezes, para descobrir que o dispositivo não completou uma operação de E/S comparativamente lenta, ou que o mouse não saiu do lugar desde a última vez em que foi sondado. Quando o dispositivo completar uma operação, ainda teremos de ler o status para determinar se ele teve sucesso.

O overhead em uma interface de polling foi reconhecido há muito tempo, levando à invenção de interrupções para notificar o processador quando um dispositivo de E/S exigir atenção do processador. A **E/S controlada por interrupção**, usada por quase todos os sistemas pelo menos para alguns dispositivos, emprega interrupções de E/S para indicar ao processador que um dispositivo de E/S precisa de atenção. Quando um dispositivo deseja notificar o processador de que completou alguma operação ou que precisa de atenção, isso faz com que o processador seja interrompido.

**polling** O processo de verificar periodicamente o status de um dispositivo de E/S para determinar a necessidade de atender ao dispositivo.

Uma interrupção de E/S é exatamente como as exceções vistas nos Capítulos 4 e 5, com duas distinções importantes:

**E/S controlada por interrupção** Um esquema de E/S que emprega interrupções para indicar ao processador que um dispositivo de E/S precisa de atenção.

1. Uma interrupção de E/S é assíncrona com relação à execução da instrução. Ou seja, a interrupção não é associada a qualquer instrução e não impede o término da instrução. Isso é muito diferente de quaisquer exceções de falta de página ou exceções como overflow aritmético. Nossa unidade de controle só precisa verificar uma interrupção de E/S pendente no momento em que iniciar uma nova instrução.
  2. Além do fato de que uma interrupção de E/S ocorreu, gostaríamos de transmitir informações adicionais, como a identidade do dispositivo gerando a interrupção. Além do mais, as interrupções representam dispositivos que podem ter diferentes prioridades e cujas solicitações de interrupção possuem diferentes urgências associadas a elas.

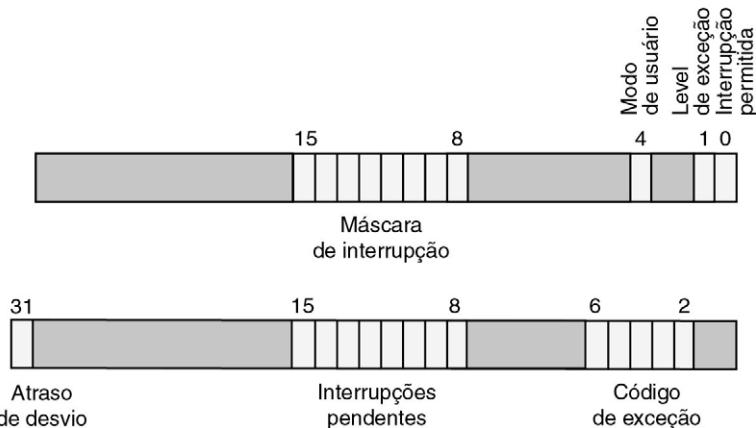
Para comunicar informações ao processador, como a identidade do dispositivo que gera a interrupção, um sistema pode usar interrupções vetorizadas ou um registrador de causa da exceção. Quando o processador reconhece a interrupção, o dispositivo pode enviar o endereço do vetor ou um campo de status para colocar no registrador de causa. Como resultado, quando o sistema operacional adquire o controle, ele sabe a identidade do dispositivo que causou a interrupção e pode interrogar imediatamente o dispositivo. Um mecanismo de interrupção elimina a necessidade de o processador sondar o dispositivo e, em vez disso, permite que o processador seja focalizado nos programas em execução.

## **Níveis de prioridade de interrupção**

Para lidar com as diferentes prioridades dos dispositivos de E/S, a maioria dos mecanismos de interrupção possui vários níveis de prioridade; sistemas operacionais UNIX utilizam de quatro a seis níveis. Essas prioridades indicam a ordem em que o processador deverá processar interrupções. Exceções geradas internamente e interrupções de E/S externas possuem prioridades; em geral, as interrupções de E/S possuem prioridade menor do que as exceções internas. Pode haver várias prioridades de interrupção de E/S, com dispositivos de alta velocidade associados às prioridades mais altas.

Para dar suporte a níveis de prioridade para interrupções, o MIPS oferece as primitivas que deixam o sistema operacional implementar a política, de modo semelhante ao modo como o MIPS trata de falhas de TLB. A Figura 6.11 mostra os principais registradores, e a Seção B.7 no Apêndice B oferece mais detalhes.

O registrador Status determina quem pode interromper o computador. Se o bit Interrupções habilitadas for 0, então ninguém poderá interromper. Um bloqueio de interrupções mais refinado está disponível no campo de máscara de interrupções. Existe um bit na máscara correspondente a cada bit no campo interrupções pendentes do registrador Cause. Para habilitar a interrupção correspondente, é preciso haver um 1 no campo de máscara no bit dessa posição. Quando ocorre uma interrupção, o sistema operacional pode encontrar o



**FIGURA 6.11 Os registradores Cause e Status.** Essa versão do registrador Cause corresponde à arquitetura MIPS-32. A arquitetura MIPS I mais antiga tinha três conjuntos aninhados de bits kernel/usuário e de bits de habilitação de interrupções para dar suporte a interrupções aninhadas. A Seção B.7 no Apêndice B contém mais detalhes sobre esses registradores.

motivo no campo de código de exceção do registrador Status: 0 significa que uma interrupção ocorreu, com outros valores para as exceções mencionadas no Capítulo 5.

Aqui estão as etapas que precisam ocorrer no tratamento de uma exceção:

1. Realize um AND lógico entre o campo interrupções pendentes e o campo máscara de interrupções para ver quais interrupções ativas poderiam ser as culpadas. São feitas cópias desses dois registradores usando a instrução `mfc0`.
2. Selecione a prioridade mais alta dessas interrupções. A convenção do software é que a mais à esquerda seja a prioridade mais alta.
3. Salve o campo de máscara de interrupções do registrador Status.
4. Mude o campo de máscara de interrupções para desativar todas as interrupções de prioridade igual ou inferior.
5. Salve o estado do processador necessário para lidar com a interrupção.
6. A fim de permitir interrupções de prioridade mais alta, coloque o bit interrupções habilitadas do registrador Cause em 1.
7. Chame a rotina de interrupção apropriada.
8. Antes de restaurar o estado, coloque o bit interrupções habilitadas do registrador Cause em 0. Isso permite restaurar o campo de máscara de interrupções.

O Apêndice B mostra um handler de exceções para uma tarefa de E/S simples.

Como os *níveis de prioridade de interrupção* (IPL – Interrupt Priority Levels) correspondem a esses mecanismos? O IPL é uma invenção do sistema operacional. Ele é armazenado na memória do processo, e cada processo recebe um IPL. No IPL mais baixo, todas as interrupções são permitidas. Ao contrário, no IPL mais alto, todas as interrupções são bloqueadas. Levantar e reduzir o IPL envolve mudanças no campo de máscara de interrupção do registrador Status.

**Detalhamento:** Os dois bits menos significativos dos campos interrupções pendentes e máscara de interrupções são para interrupções de software, que são de prioridade inferior. Eles normalmente são usados por interrupções de prioridade mais alta para deixar trabalho para interrupções de menor prioridade realizarem depois que o motivo imediato da interrupção for tratado. Quando a interrupção de maior prioridade terminar, as tarefas de prioridade inferior serão observadas e tratadas.

## Transferindo os dados entre um dispositivo e a memória

Vimos dois métodos diferentes que permitem que um dispositivo se comunique com o processador. Essas duas técnicas – polling e interrupções de E/S – formam a base para dois métodos de implementação da transferência de dados entre o dispositivo de E/S e a memória. Essas duas técnicas funcionam melhor com dispositivos de menor largura de banda, nos quais estamos mais interessados em reduzir o custo do controlador de dispositivo e interface do que oferecer uma transferência com largura de banda alta. Tanto o polling quanto as transferências controladas por interrupção colocam o trabalho de mover dados e gerenciar a transferência sob a responsabilidade do processador. Depois de examinar esses dois esquemas, veremos um outro mais adequado para dispositivos de maior desempenho ou coleções de dispositivos.

Podemos usar o processador para transferir dados entre um dispositivo e a memória com base no polling. Em aplicações de tempo real, o processador carrega dados dos registradores do dispositivo de E/S e os armazena na memória.

Um outro mecanismo é fazer a transferência de dados controlada por interrupção. Nesse caso, o sistema operacional ainda transferiria dados em pequenos números de bytes de ou para o dispositivo. Entretanto, como a operação de E/S é controlada por interrupção, o sistema operacional simplesmente atua sobre outras tarefas enquanto os dados estão sendo lidos ou escritos no dispositivo. Quando o sistema operacional reconhece uma interrupção a partir do dispositivo, ele lê o status para verificar a ocorrência de erros. Se não houver,

o sistema operacional poderá fornecer a próxima parte dos dados, por exemplo, por uma sequência de escritas mapeadas em memória. Quando o último byte de uma solicitação de E/S tiver sido transmitido e a operação de E/S for concluída, o sistema operacional poderá informar ao programa. O processador e o sistema operacional realizam todo o trabalho nesse processo, acessando o dispositivo e a memória para cada item de dados transferido.

A E/S controlada por interrupção libera o processador de ter de esperar por cada evento de E/S, embora, se usássemos esse método para transferir dados de ou para um disco rígido, o overhead ainda poderia ser intolerável, pois isso poderia consumir uma grande fração do processador quando o disco estivesse transferindo. Para dispositivos com alta largura de banda, como discos rígidos, as transferências consistem principalmente em blocos de dados relativamente grandes (centenas a milhares de bytes). Assim, os projetistas de computadores inventaram um mecanismo para desafogar o processador e fazer com que o controlador de dispositivo transfira dados diretamente de ou para a memória sem envolver o processador. Esse mecanismo é chamado de **acesso direto à memória** (DMA – Direct Memory Access). O mecanismo de interrupção ainda é usado pelo dispositivo para a comunicação com o processador, mas somente no término da transferência de E/S ou quando ocorre um erro.

O DMA é implementado com um controlador especializado, que transfere dados entre um dispositivo de E/S e a memória, independente do processador. O controlador de DMA torna-se o **master** e direciona as leituras e escritas entre si mesmo e a memória. Existem três etapas em uma transferência de DMA:

1. O processador configura o DMA fornecendo a identidade do dispositivo, a operação a realizar no dispositivo, o endereço de memória que é a origem ou o destino dos dados a serem transferidos e o número de bytes a transferir.
2. O DMA inicia a operação no dispositivo e arbitra o acesso à interconexão. Quando os dados estão disponíveis (do dispositivo ou da memória), ele transfere os dados. O dispositivo de DMA fornece o endereço de memória para a leitura ou a escrita. Se a solicitação exigir mais de uma transferência, a unidade de DMA gera o próximo endereço de memória e inicia a próxima transferência. Usando esse mecanismo, a unidade de DMA pode completar uma transferência inteira, que pode ter milhares de bytes de tamanho, sem incomodar o processador. Muitos controladores de DMA contêm alguma memória para permitir que eles tratem de modo flexível atrasos na transferência ou aqueles ocorridos na espera para se tornar o master.
3. Quando a transferência de DMA termina, o controlador interrompe o processador, que pode então determinar, interrogando o dispositivo de DMA ou examinando a memória, se a operação inteira foi concluída com sucesso.

Pode haver vários dispositivos de DMA em um sistema de computador. Por exemplo, em um sistema com um único barramento processador-memória e vários barramentos de E/S, cada controlador de barramento de E/S normalmente terá um processador de DMA que trata de quaisquer transferências entre um dispositivo no barramento de E/S e a memória.

Ao contrário do polling ou da E/S controlada por interrupção, o DMA pode ser usado para realizar interface de um disco rígido sem consumir todos os ciclos de processador para uma única E/S. Naturalmente, se o processador também estiver brigando pela memória, ele será atrasado quando a memória estiver ocupada realizando uma transferência de DMA. Usando caches, o processador pode evitar ter de acessar a memória na maior parte do tempo, deixando assim a maior parte da largura de banda da memória livre para uso por dispositivos de E/S.

**Detalhamento:** Para reduzir ainda mais a necessidade de interromper o processador e ocupá-lo no tratamento de uma solicitação de E/S que possa envolver a realização de várias operações reais, o controlador de E/S pode se tornar mais inteligente. Controladores inteligentes normalmente são chamados de *processadores de E/S* (bem como *controladores de E/S* ou *controladores de canal*). Esses processadores especializados executam uma série de operações de E/S, chamadas de *programa de E/S*. O programa pode estar armazenado no processador de E/S, ou pode estar armazenado na memória e ser buscado pelo processador de E/S. Ao usar um processador de E/S, o sistema operacional normalmente configura um programa de E/S que

indica as operações de E/S a serem realizadas, além do tamanho e do endereço de transferência para quaisquer leituras ou escritas. O processador de E/S, então, busca as operações do programa de E/S e interrompe o processador apenas quando o programa inteiro estiver completo. Os processadores de DMA são processadores de uso especial (normalmente, de único chip e não programáveis), enquanto os processadores de E/S normalmente são implementados com microprocessadores de uso geral, que executam um programa de E/S especializado.

### Acesso direto à memória e o sistema de memória

Quando o DMA é incorporado a um sistema de E/S, o relacionamento entre o sistema de memória e o processador muda. Sem DMA, todos os acessos ao sistema de memória vêm do processador e, assim, prosseguem pela tradução de endereços e acesso à cache como se o processador gerasse as referências. Com DMA, existe outro caminho para o sistema de memória – que não passa pelo mecanismo de tradução de endereços ou pela hierarquia de cache. Essa diferença gera alguns problemas nos sistemas de memória virtual e em sistemas com caches. Esses problemas normalmente são solucionados com uma combinação de técnicas de hardware e suporte do software.

As dificuldades de ter DMA em um sistema de memória virtual surgem porque as páginas possuem um endereço físico e um endereço virtual. O DMA também cria problemas para sistemas com caches, pois pode haver duas cópias de um item de dados: uma na cache e uma na memória. Como o processador de DMA realiza solicitações de memória diretamente à memória, e não pela cache do processador, o valor de um local de memória visto pela unidade de DMA e pelo processador pode ser diferente. Considere uma leitura do disco que a unidade de DMA coloque diretamente na memória. Se alguns dos locais em que o DMA escreve estiverem na cache, o processador receberá o valor antigo quando fizer uma leitura. De modo semelhante, se a cache for write-back, o DMA poderá ler um valor diretamente da memória quando um valor mais novo estiver na cache, e o valor não foi escrito de volta. Isso é chamado de *problema de dados antigos*, ou *problema de coerência* (veja Capítulo 5).

Vimos três métodos diferentes para transferir dados entre um dispositivo de E/S e a memória. Ao passar do polling para uma E/S controlada por interrupção e para uma interface de DMA, mudamos o peso do gerenciamento de uma operação de E/S do processador para um controlador de E/S progressivamente mais inteligente. Esses métodos têm a vantagem de liberar os ciclos do processador. Sua desvantagem é que eles aumentam o custo do sistema de E/S. Por causa disso, determinado sistema computacional pode escolher qual ponto nesse espectro é apropriado para os dispositivos de E/S se conectarem a ele.

Antes de discutirmos o projeto dos sistemas de E/S, vejamos rapidamente as medidas de desempenho deles na próxima seção.

Na avaliação das três maneiras de realizar E/S, quais afirmações são verdadeiras?

1. Se quisermos a menor latência para uma operação de E/S a um único dispositivo de E/S, a ordem é polling, DMA e E/S controlada por interrupção.
2. Em termos de menor impacto na utilização do processador a partir de um único dispositivo de E/S, a ordem é DMA, E/S controlada por interrupção e polling.

### Verifique você mesmo

Em um sistema com memória virtual, o DMA deverá funcionar com endereços virtuais ou com endereços físicos? O problema óbvio com os endereços virtuais é que a unidade de DMA precisará traduzir os endereços virtuais em endereços físicos. O problema principal com o uso de um endereço físico em uma transferência de DMA é que a transferência não pode cruzar com facilidade um limite de página. Se uma solicitação de E/S cruzasse um limite de página, então os locais de memória para os quais ela estava sendo transferida não necessariamente seriam contíguos na memória virtual. Consequentemente, se usarmos endereços físicos, teremos de restringir todas as transferências de DMA para permanecerem dentro de uma página.

Um método para permitir que o sistema inicie transferências de DMA que cruzam limites de página é fazer com que o DMA funcione em endereços virtuais. Nesse sistema, a unidade

### Interface hardware/software

de DMA possui um pequeno número de entradas de mapa que oferecem mapeamento virtual para físico para uma transferência. O sistema operacional provê o mapeamento quando a E/S for iniciada. Usando esse mapeamento, a unidade de DMA não precisa se preocupar com o local das páginas virtuais envolvidas na transferência.

Outra técnica é que o sistema operacional divide a transferência de DMA em uma série de transferências, cada uma confinada dentro de uma única página física. As transferências, então, são *encadeadas* e entregues a um processador de E/S ou unidade de DMA inteligente, que executa a sequência inteira de transferências; como alternativa, o sistema operacional pode solicitar as transferências individualmente.

Qualquer que seja o método utilizado, o sistema operacional ainda precisa cooperar não remapeando as páginas enquanto uma transferência de DMA que envolve essa página estiver em andamento.

## Interface hardware/software

O problema de coerência para dados de E/S é evitado pelo uso de uma de três técnicas importantes. Uma técnica é rotear a atividade de E/S por meio da cache. Isso garante que as leituras vejam o valor mais recente enquanto as escritas atualizam quaisquer dados na cache. O roteamento de toda a E/S pela cache é dispendioso e possui um grande impacto potencial negativo no desempenho do processador, pois os dados de E/S raramente são usados de imediato e podem deslocar dados úteis de que um programa em execução precisa. Uma segunda opção é ter o sistema operacional invalidando a cache seletivamente para uma leitura de E/S ou forçar a ocorrência de write-backs para uma escrita de E/S (normalmente chamado de *flush* de cache). Essa técnica exige uma pequena quantidade de suporte do hardware e provavelmente é mais eficiente se o software puder realizar a função de forma fácil e eficiente. Como esse flush de grandes partes da cache só precisa acontecer nos acessos em bloco ao DMA, ele será relativamente pouco frequente. A terceira técnica é oferecer um mecanismo de hardware para fazer o flush (ou invalidar) seletivamente às entradas de cache. A invalidação do hardware para garantir coerência da cache é comum em sistemas multiprocessador, e a mesma técnica pode ser usada para E/S; discutimos esse assunto com detalhes no Capítulo 5.

## 6.7

### Medidas de desempenho de E/S: exemplos de sistemas de disco e de arquivos

Como devemos comparar sistemas de E/S? Essa é uma pergunta complexa, porque o desempenho da E/S depende de muitos aspectos do sistema e diferentes aplicações enfatizam diferentes aspectos do sistema de E/S. Além do mais, um projeto pode fazer escolhas complexas entre tempo de resposta e vazão, tornando impossível medir apenas um aspecto isoladamente. Por exemplo, tratar um pedido o mais cedo possível em geral minimiza o tempo de resposta, embora uma vazão maior possa ser alcançada se tentarmos lidar com solicitações relacionadas juntas. De acordo com isso, podemos aumentar a vazão em um disco agrupando solicitações que acessam locais próximos. Essa política aumentará o tempo de resposta para algumas solicitações, provavelmente levando a uma variação maior no tempo de resposta. Embora a vazão seja maior, alguns benchmarks restringem o tempo de resposta máximo a qualquer solicitação, tornando tais otimizações potencialmente problemáticas.

Nesta seção, damos alguns exemplos de medidas propostas para determinar o desempenho dos sistemas de disco. Esses benchmarks são afetados por uma variedade de recursos do sistema, incluindo tecnologia de disco, como os discos são conectados, o sistema de memória, o processador e o sistema de arquivos fornecido pelo sistema operacional.

Antes de discutirmos esses benchmarks, precisamos explicar um ponto confuso sobre terminologia e unidades. O desempenho dos sistemas de E/S depende da velocidade em

que o sistema transfere dados. A velocidade de transferência depende da velocidade do clock, que normalmente é dada em GHz =  $10^9$  ciclos por segundo. A taxa de transferência normalmente é cotada em GB/seg. Nos sistemas de E/S, GBs são medidos usando a base 10 (ou seja,  $1\text{GB} = 10^9 = 1.000.000.000$  bytes), diferente da memória principal, em que a base 2 é utilizada (ou seja,  $1\text{GB} = 2^{30} = 1.073.741.824$ ). Além de aumentar a confusão, essa diferença gera a necessidade de conversão entre a base 10 ( $1\text{K} = 1000$ ) e a base 2 ( $1\text{K} = 1024$ ), porque muitos acessos à E/S são para blocos de dados que possuem um tamanho que é uma potência de dois. Em vez de complicar todos os nossos exemplos, convertendo com precisão uma das duas medidas, ressaltamos aqui essa distinção e o fato de que tratar as duas medidas como se as unidades fossem idênticas produz um pequeno erro. Ilustramos esse erro na Seção 6.12.

## Benchmarks de E/S de processamento de transações

Aplicações de **processamento de transações** (TP – Transaction Processing) envolvem um requisito de tempo de resposta e uma medida de desempenho baseada na vazão. Além do mais, a maioria dos acessos de E/S é pequena. Por causa disso, as aplicações de TP tratam principalmente da **taxa de E/S**, medida como o número de acessos ao disco por segundo, ao contrário da **taxa de dados**, medida como bytes de dados por segundo. As aplicações de TP geralmente envolvem mudanças em um banco de dados grande, com o sistema atendendo a alguns requisitos de tempo de resposta e tratando de forma controlada certos tipos de falhas. Essas aplicações são muito críticas e sensíveis ao custo. Por exemplo, os bancos normalmente utilizam sistemas de TP porque se preocupam com uma série de características, entre elas: garantir que as transações não são perdidas, tratar das transações rapidamente e minimizar o custo do processamento de cada transação. Embora a confiabilidade em face da falha seja um requisito absoluto em tais sistemas, o tempo de resposta e a vazão são fundamentais para criar sistemas econômicos.

Diversos benchmarks de processamento de transações foram desenvolvidos. O conjunto mais conhecido de benchmarks é uma série desenvolvida pelo Transaction Processing Council (TPC).

O TPC-C, inicialmente criado em 1992, simula um ambiente de consulta complexo. O TPC-H modela o apoio à decisão ocasional – as consultas não são relacionadas, e o conhecimento de consultas passadas não pode ser usado para otimizar futuras consultas; o resultado é que os tempos de execução da consulta podem ser muito longos. O TPC-W é um benchmark de aplicações baseadas na Web, que simula as atividades de um servidor Web transacional orientado a negócios. Ele exercita o sistema de banco de dados e também o software básico do servidor Web. O TPC-App é um benchmark de servidor de aplicações e Web services. O mais recente é o TPC-E, que simula a carga de trabalho de processamento de transações de uma firma de corretagem. Os benchmarks TPC são descritos em [www\(tpc.org](http://www(tpc.org)

Todos os benchmarks de TCP medem o desempenho em transações por segundo. Além disso, eles incluem um requisito de tempo de resposta, de modo que o desempenho da vazão é medido apenas quando o limite do tempo de resposta é atendido. Para modelar sistemas do mundo real, as velocidades de transação mais altas também estão associadas a sistemas maiores, tanto em termos de usuários quanto o tamanho do banco de dados ao qual as transações são aplicadas. Logo, a capacidade de armazenamento precisa se expandir com o desempenho. Finalmente, o custo do sistema para um sistema de benchmark também precisa ser incluído, permitindo comparações precisas de custo-desempenho.

## Benchmarks de E/S para sistema de arquivos e para Web

Além de benchmarks de processador, o SPEC oferece um benchmark de servidor de arquivos (SPECFS) e um benchmark de servidor Web (SPECWeb). O SPECFS é um benchmark destinado a medir o desempenho do NFS (Network File System) usando um script de solicitações para servidores de arquivos; ele testa o desempenho do sistema de E/S, incluindo disco e rede, além do processador. SPECFS é um benchmark orientado a vazão, mas com requisitos importantes de tempo de resposta. SPECWeb é um benchmark de servidor Web que simula vários clientes solicitando páginas estáticas e dinâmicas de um servidor, além de clientes postando dados ao servidor (veja Capítulo 1).

### processamento de transações

Um tipo de aplicação que envolve o tratamento de pequenas operações curtas (chamadas transações) que normalmente exigem tanto E/S quanto cálculo. As aplicações de processamento de transações normalmente possuem requisitos de tempo de resposta e uma medida de desempenho baseada na vazão das transações.

**taxa de E/S** A medida de desempenho das E/Ss por unidade de tempo, como leituras por segundo.

**taxa de dados** Medida de desempenho de bytes por unidade de tempo, como GB/segundo

O esforço SPEC mais recente é para medir a potência. O SPECPower mede as características de potência e desempenho de pequenos servidores.

A Sun recentemente anunciou o *filebench*, um framework de benchmark do sistema de arquivos. Em vez de uma carga de trabalho padrão, ele oferece uma linguagem que lhe permite descrever a carga de trabalho que você gostaria de executar nos seus sistemas de arquivos. Porém, existem exemplos de cinco cargas de trabalho que têm como finalidade simular aplicações comuns de sistemas de arquivos.

**Verifique  
você mesmo**

As seguintes afirmativas são verdadeiras ou falsas? Ao contrário dos benchmarks de processador, os benchmarks de E/S:

1. concentram-se na vazão, em vez da latência.
2. podem exigir que os dados definam a escala em tamanho ou número de usuários para conseguir os marcos de desempenho.
3. normalmente relatam o desempenho em termos de custo.

**6.8****Projetando um sistema de E/S**

Existem dois tipos principais de especificação que os projetistas encontram nos sistemas de E/S: restrições de latência e restrições de largura de banda. Nos dois casos, o conhecimento do padrão de tráfego afeta o projeto e a análise.

As restrições de latência envolvem garantir que a latência para completar uma operação de E/S esteja limitada por uma certa quantidade. No caso simples, o sistema pode ser descarregado, e o projetista também precisa garantir que algum limite de latência seja realizado, pois isso é fundamental para a aplicação ou porque o dispositivo precisa receber certo serviço garantido que impeça erros. Da mesma forma, determinar a latência de um sistema não carregado é relativamente fácil, pois envolve rastrear o caminho da operação de E/S e somar as latências individuais.

Encontrar a latência média (ou a distribuição da latência) sob uma carga é um problema muito mais complexo. Esses problemas são resolvidos ou por teoria de filas (quando o comportamento das solicitações da carga de trabalho e os tempos de atendimento de E/S podem ser aproximados por distribuições simples) ou por simulação (quando o comportamento dos eventos de E/S é complexo). Os dois tópicos estão além do escopo deste texto.

Projetar um sistema de E/S para atender a um conjunto de restrições de largura de banda dado uma carga de trabalho é o outro problema comum que os projetistas enfrentam. Como alternativa, o projetista pode receber um sistema de E/S parcialmente configurado e ser solicitado a balancear o sistema para manter a largura de banda máxima alcançável conforme ditado pela parte pré-configurada do sistema. Esse último problema de projeto é uma versão simplificada do primeiro.

A técnica geral para projetar tal sistema é a seguinte:

1. Encontrar o elo mais fraco no sistema de E/S, que é o componente no caminho da E/S que restringirá o projeto. Dependendo da carga de trabalho, esse componente pode estar em qualquer lugar, incluindo nos processadores, nos controladores de E/S ou nos dispositivos. Os limites da carga de trabalho e de configuração podem ditar onde está localizado o elo mais fraco.
2. Configurar esse componente para sustentar a largura de banda exigida.
3. Determinar os requisitos para o restante do sistema e configurá-los para dar suporte a essa largura de banda.
4. O modo mais fácil de entender essa metodologia é com um exemplo. Faremos uma análise simples do sistema de E/S do servidor Sun Fire x4150 na Seção 6.10, para mostrar como essa metodologia funciona.

## 6.9

# Paralelismo e E/S: Redundant Arrays of Inexpensive Disks (RAID)

A lei de Amdahl no Capítulo 1 nos lembra que é precipitado negligenciar a E/S nessa revolução paralela. Um exemplo simples demonstra isso.

### Impacto da E/S sobre o desempenho do sistema

Suponha que tenhamos um benchmark executado em 100 segundos de tempo decorrido, dos quais 90 segundos é tempo de CPU e o restante é tempo de E/S. Suponha que o número de processadores dobra a cada dois anos, mas os processadores permanecem na mesma velocidade, e o tempo de E/S não melhora. O quanto mais rápido nosso programa será executado ao final de seis anos?

Sabemos que

$$\begin{aligned}\text{Tempo decorrido} &= \text{tempo de CPU} + \text{tempo de E/S} \\ 100 &= 90 + \text{tempo de E/S} \\ \text{Tempo de E/S} &= 10 \text{ segundos}\end{aligned}$$

### EXEMPLO

### RESPOSTA

Os novos tempos de CPU e os tempos decorridos resultantes são calculados na tabela a seguir.

Após $n$ anos	Tempo de CPU	Tempo de E/S	Tempo decorrido	% de tempo de E/S
0 ano	90 segundos	10 segundos	100 segundos	10%
2 anos	$\frac{90}{2} = 45$ segundos	10 segundos	55 segundos	18%
4 anos	$\frac{45}{2} = 23$ segundos	10 segundos	33 segundos	31%
6 anos	$\frac{23}{2} = 11$ segundos	10 segundos	21 segundos	47%

A melhoria no desempenho da CPU após seis anos é

$$\frac{90}{11} = 8$$

Porém, a melhoria no tempo decorrido é de apenas

$$\frac{100}{21} = 4,7$$

e o tempo de E/S aumentou de 10% para 47% do tempo decorrido.

Logo, a revolução paralela precisa chegar à E/S e também ao cálculo, ou o esforço gasto paralelizando poderia ser gasto sempre que programas realizam E/S, o que todos eles precisam fazer.

Acelerar o desempenho de E/S foi a motivação original dos arrays de disco (veja [Seção 6.14](#) no site). No final dos anos 1980, o armazenamento em alto desempenho preferido eram discos grandes e dispendiosos, como os maiores na [Figura 6.4](#). O argumento foi que, substituindo alguns discos grandes por muitos discos pequenos, o desempenho melhoraria porque haveria mais cabeças de leitura. Essa passagem é uma boa escolha para processadores múltiplos

também, pois muitas cabeças de leitura/escrita significam que o sistema de armazenamento poderia dar suporte a muito mais acessos independentes, e também transferências grandes se espalhariam por muitos discos. Ou seja, você poderia conseguir altas taxas de E/S por segundo e altas taxas de transferência de dados. Além do desempenho mais alto, poderia haver vantagens no custo, na potência e no espaço, pois discos menores geralmente são mais eficientes por gigabyte do que discos maiores.

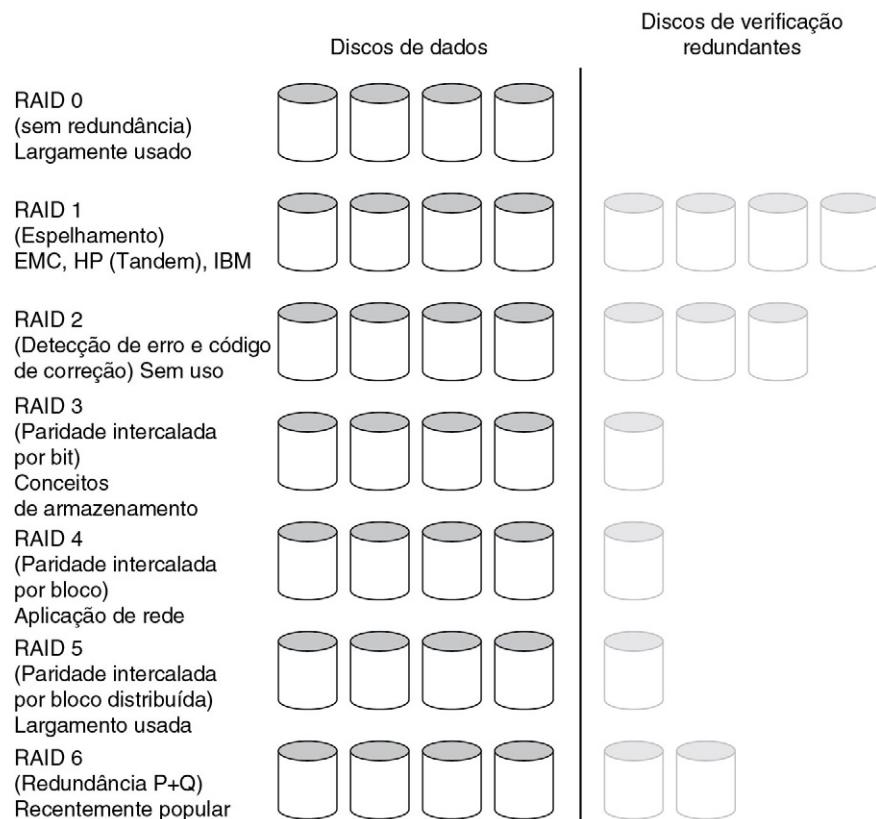
A falha no argumento foi que os arrays de disco poderiam tornar a confiabilidade muito pior. Essas unidades menores e menos dispendiosas tinham menores valores de MTTF que as unidades grandes, porém, mais importante que isso, substituindo uma única unidade por, digamos, 50 unidades pequenas, a taxa de falha subiria por um fator de pelo menos 50!

A solução foi acrescentar redundância de modo que o sistema pudesse lidar com as falhas de disco sem perder informações. Tendo muitos discos pequenos, o custo da redundância extra para melhorar a confiabilidade é pequeno em relação a solução com alguns discos grandes. Assim, a confiabilidade era mais econômica se você construísse um array redundante de discos mais baratos. Essa observação levou ao seu nome: **array redundante de discos pouco dispendiosos**, abreviado como **RAID**.

Em retrospecto, embora sua invenção fosse motivada pelo desempenho, a confiabilidade foi o principal motivo para a popularidade geral do RAID. A revolução paralela destacou a questão do desempenho original do RAID. O restante desta seção analisa as opções para confiabilidade e seus impactos sobre custo e desempenho.

De quanta redundância você precisa? Você precisa de informações extras para encontrar as falhas? Importa como você organiza os dados e as informações de verificação extra nesses discos? O artigo que criou o termo deu uma resposta evolutiva a essas questões, começando com a solução mais simples, porém mais dispendiosa. A [Figura 6.12](#) mostra a evolução e um exemplo de custo no número de discos de verificação extras. Para acompanhar a evolução, os autores numeraram os estágios do RAID, e eles ainda são usados hoje.

**RAID (Redundant Arrays of Inexpensive Disks)** Uma organização de discos que usa um array de discos pequenos e baratos para aumentar o desempenho e a confiabilidade.



**FIGURA 6.12 RAID para um exemplo de quatro discos de dados, mostrando discos de verificação extras por nível de RAID e empresas que utilizam cada nível.** As [Figuras 6.13 e 6.14](#) explicam a diferença entre RAID 3, RAID 4 e RAID 5.



## Nenhuma redundância (RAID 0)

O simples espalhamento dos dados por vários discos, chamado **striping**, força automaticamente os acessos a vários discos. O striping por um conjunto de discos faz com que a coleção apareça ao software como um único disco grande, que simplifica o gerenciamento do armazenamento. Isso também melhora o desempenho para acessos grandes, pois muitos discos podem operar ao mesmo tempo. Os sistemas de edição de vídeo, por exemplo, normalmente repartem seus dados e podem não se preocupar com a confiabilidade tanto quanto, digamos, os bancos de dados.

RAID 0 é um nome errado, pois não existe redundância. Entretanto, os níveis de RAID normalmente são deixados para o operador definir ao criar um sistema de armazenamento, e RAID 0 normalmente está listado como uma das opções. Logo, o termo RAID 0 tornou-se muito utilizado.

**striping** Alocação de blocos logicamente sequenciais por discos separados para permitir maior desempenho do que um único disco pode oferecer.

## Espelhamento (RAID 1)

Esse esquema tradicional para tolerar falhas de disco, chamado **espelhamento** ou *shadowing*, utiliza o dobro da quantidade de discos do RAID 0. Sempre que os dados são gravados em um disco, esses dados também são gravados em um disco redundante, de modo que sempre existem duas cópias da informação. Se um disco falhar, o sistema simplesmente vai ao “espelho” e lê seu conteúdo para obter a informação desejada. O espelhamento é a solução de RAID mais dispendiosa, pois exige mais discos.

**espelhamento** Escrever dados idênticos em vários discos, para aumentar a disponibilidade dos dados.

## Código de detecção e correção de erros (RAID 2)

RAID 2 utiliza um esquema de detecção e correção de erros que é mais utilizado para memórias (veja Apêndice C). Como RAID 2 caiu em desuso, não iremos descrevê-lo aqui.

## Paridade intercalada por bit (RAID 3)

O custo da disponibilidade mais alta pode ser reduzido para  $1/n$ , onde  $n$  é o número de discos em um **grupo de proteção**. Em vez de ter uma cópia completa dos dados originais para cada disco, só precisamos acrescentar informações redundantes suficientes para restaurar a informação perdida em uma falha. Leituras ou escritas vão para todos os discos no grupo, com um disco extra para manter as informações de verificação caso haja uma falha. RAID 3 é comum em aplicações com grandes conjuntos de dados, como multimídia e alguns códigos científicos.

**grupo de proteção** O grupo de discos de dados ou blocos que compartilham um disco ou bloco de verificação comum.

*Paridade* é um esquema desse tipo. Os leitores não acostumados com a paridade podem pensar no disco redundante como aquele com a soma de todos os dados dos outros discos. Quando um disco falha, então você subtrai todos os dados nos discos bons do disco de paridade; a informação restante deverá ser a informação que falta. A paridade é simplesmente a soma módulo dois.

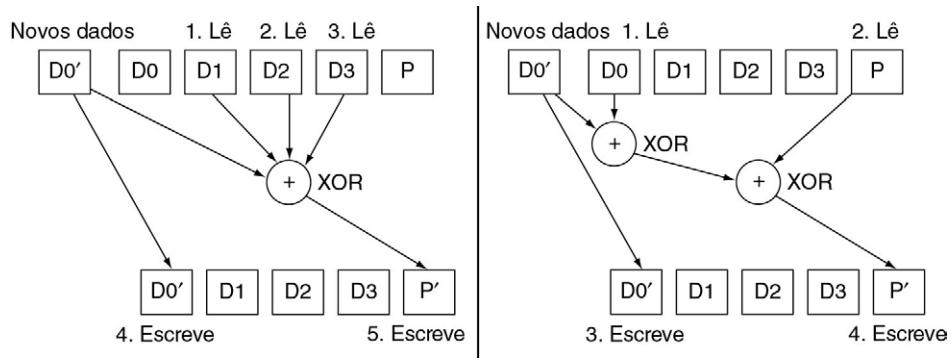
Diferente de RAID 1, muitos discos precisam ser lidos para determinar os dados que faltam. A suposição por trás dessa técnica é a de que levar mais tempo para recuperar-se de uma falha, mas gastar menos com armazenamento redundante, é uma boa escolha.

## Paridade intercalada por bloco (RAID 4)

RAID 4 usa a mesma razão de discos de dados e discos de verificação do RAID 3, mas eles acessam dados de formas diferentes. A paridade é armazenada como blocos e associada a um conjunto de blocos de dados.

Em RAID 3, cada acesso ia para todos os discos. Contudo, algumas aplicações preferem acessos menores, permitindo que acessos independentes ocorram em paralelo. Essa é a finalidade do RAID níveis 4 a 6. Como a informação de detecção de erro em cada setor é verificada nas leituras para ver se os dados estão corretos, essas “leituras pequenas” a cada disco podem ocorrer de forma independente, desde que o acesso mínimo seja de um setor. No contexto do RAID, um acesso pequeno vai para apenas um disco em um grupo de proteção, enquanto um acesso grande vai para todos os discos em um grupo de proteção.

As escritas são outro problema. Pode parecer que cada escrita pequena exigiria que todos os outros discos fossem acessados para ler o restante das informações necessárias no recálculo da nova paridade, como na [Figura 6.13](#). Uma “escrita pequena” exigiria a



**FIGURA 6.13 Pequena atualização de escrita em RAID 4.** Essa otimização para pequenas escritas reduz a quantidade de acessos ao disco, bem como a quantidade de discos ocupados. Essa figura considera que temos quatro blocos de dados e um bloco de paridade. O ingênuo cálculo de paridade do RAID 4 à esquerda da figura lê os blocos D1, D2 e D3 antes de acrescentar o bloco D0' para calcular a nova paridade P'. (Caso você esteja questionando, os novos dados D0' vêm diretamente da CPU, de modo que os discos não estão envolvidos na sua leitura.) O atalho RAID 4 à direita lê o valor antigo D0 e o compara com o novo valor D0' para ver quais bits mudaram. Em seguida, você lê a paridade antiga P e depois muda os bits correspondentes para formar P'. A função lógica OR exclusivo faz exatamente o que queremos. Esse exemplo substitui três leituras de disco (D1, D2, D3) e duas escritas (D0', P') envolvendo todos os discos para duas leituras de disco (D0, P) e duas escritas de disco (D0', P'), que envolvem apenas dois discos. Aumentar o tamanho do grupo de paridade aumenta as economias do atalho. RAID 5 utiliza o mesmo atalho.

leitura dos dados antigos e da paridade antiga, adicionando as novas informações e depois escrevendo a nova paridade no disco de paridade e os novos dados no disco de dados.

A ideia principal para reduzir esse overhead é que a paridade é simplesmente uma soma de informações; observando quais bits mudam quando escrevemos as novas informações, só precisamos mudar os bits correspondentes no disco de paridade. O lado direito da Figura 6.13 mostra o atalho. Temos de ler os dados antigos do disco sendo escrito, comparar os dados antigos com os novos para ver quais bits mudam, ler a paridade antiga, alterar os bits correspondentes, depois escrever os novos dados e a nova paridade. Assim, a pequena escrita envolve quatro acessos de disco a dois discos, em vez de acessar todos os discos. Essa organização é RAID 4.

### Paridade distribuída intercalada por bloco (RAID 5)

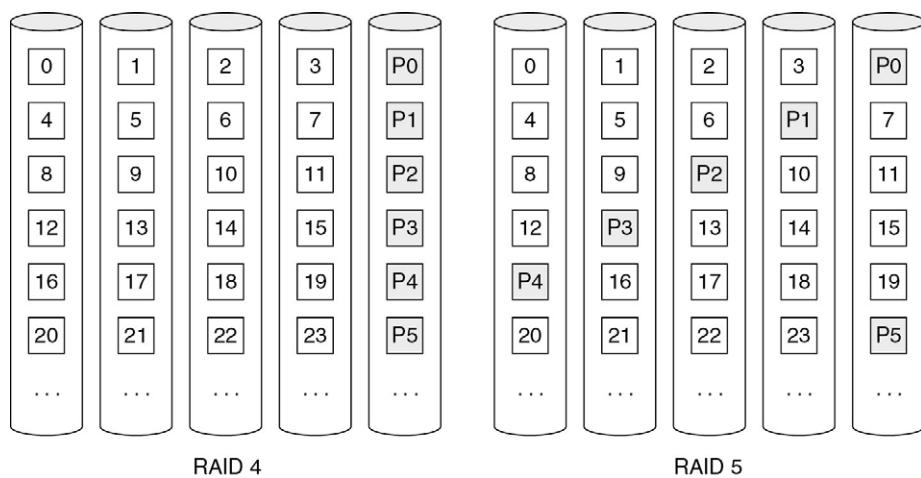
RAID 4 aceita de forma eficiente uma mistura de leituras grandes, escritas grandes e leituras pequenas, e também permite escritas pequenas. Uma desvantagem para o sistema é que o disco de paridade precisa ser atualizado em cada escrita, de modo que o disco de paridade é o gargalo para escritas back-to-back.

Para resolver o gargalo da escrita de paridade, a informação de paridade pode ser espalhada por todos os discos, de modo que não haja um único gargalo para escritas. A organização da paridade distribuída é RAID 5.

A Figura 6.14 mostra como os dados são distribuídos no RAID 4 versus RAID 5. Como vemos na organização da direita, em RAID 5, a paridade associada a cada linha de blocos de dados não é mais restrita a um único disco. Essa organização permite que várias escritas ocorram simultaneamente, desde que os blocos de paridade não estejam localizados no mesmo disco. Por exemplo, uma escrita no bloco 8 à direita também precisa acessar seu bloco de paridade P2, ocupando assim o primeiro e terceiro discos. Uma segunda escrita no bloco 5, à direita, implicando uma atualização no seu bloco de paridade P1, acessa o segundo e quarto discos e, assim, poderia ocorrer simultaneamente com a escrita no bloco 8. Essas mesmas escritas na organização à esquerda resultam em mudanças nos blocos P1 e P2, ambas no quinto disco, que é um gargalo.

### Redundância P + Q (RAID 6)

Os esquemas baseados em paridade protegem contra uma única falha autoidentificável. Quando uma correção de única falha não é suficiente, a paridade pode ser generalizada



**FIGURA 6.14 Paridade intercalada por bloco (RAID 4) versus paridade distribuída intercalada por bloco (RAID 5).** Distribuindo os blocos de paridade a todos os discos, algumas escritas pequenas podem ser realizadas em paralelo.

para ter um segundo cálculo sobre os dados e outro disco de verificação de informações. Esse segundo bloco de verificação permite a recuperação de uma segunda falha. Assim, o overhead do armazenamento é o dobro daquele do RAID 5. O atalho de escrita pequena da Figura 6.13 também funciona, exceto que agora existem seis acessos a disco, em vez de quatro para atualizar as informações de P e Q.

### Resumo de RAID

RAID 1 e RAID 5 são bastante utilizados em servidores; uma estimativa é de que 80% de discos nos servidores se encontrem em algum sistema RAID.

Um ponto fraco dos sistemas RAID é o reparo. Primeiro, para evitar tornar os dados indisponíveis durante o reparo, o array precisa ser designado de modo a permitir que os discos que falharam sejam substituídos sem ter de desligar o sistema. RAIDs possuem redundância suficiente para permitir a operação contínua, mas o **hot swapping** de discos impõe demandas sobre o projeto físico e elétrico do array e as interfaces de disco. Segundo, outra falha poderia ocorrer durante o reparo, de modo que o tempo de reparo afeta as chances de perder dados: quanto maior for o tempo de reparo, maiores as chances de outra falha que causará perda de dados. Em vez de ter de esperar que o operador traga um disco bom, alguns sistemas incluem **reservas em standby**, de modo que os dados podem ser reconstruídos imediatamente na descoberta da falha. O operador pode, então, substituir os discos que falharam sem tanta pressa. Perceba que um operador humano que, em última análise, determina quais discos devem ser removidos. Como mostra a Figura 6.3, os operadores são apenas humanos, de modo que ocasionalmente poderão remover um disco bom no lugar do disco com defeito, ocasionando uma falha de disco irrecuperável.

Além de projetar o sistema RAID para reparo, existem questões sobre como a tecnologia de disco muda com o tempo. Embora os fabricantes de disco citem um MTTF muito alto para seus produtos, esses números estão sob condições nominais. Se um array de disco em particular tiver sido sujeito a ciclos de temperatura devido a, digamos, a falha do sistema de ar-condicionado, ou a sacudidas devido a um projeto, uma construção ou uma instalação de rack ineficaz, as taxas de falha podem ser de três a seis vezes maior (veja a falácia posteriormente neste capítulo). O cálculo de confiabilidade RAID considera independência entre falhas de disco, mas as falhas poderiam estar correlacionadas, pois tal dano devido ao ambiente provavelmente aconteceria em todos os discos no array. Outro problema é que, a largura de banda do disco está crescendo mais lentamente que a capacidade do disco, o tempo para reparo de um disco em um sistema RAID está aumentando, o que, por sua vez, aumenta as chances de uma segunda falha. Por exemplo, um disco SATA de 1000GB

**hot swapping** Substituição de um componente de hardware enquanto o sistema está em execução.

**reservas em standby** Recursos de hardware de reserva que podem imediatamente tomar o lugar de um componente defeituoso.

poderia levar quase três horas para ser lido sequencialmente, sem considerar interferência. Dado que o RAID danificado provavelmente continuará a atender com dados, a reconstrução poderia ser bastante esticada. Além de aumentar esse tempo, outro problema é que a leitura de muito mais dados durante a reconstrução significa aumentar a chance de uma falha irrecuperável de leitura de mídia, que resultaria em perda de dados. Outros argumentos para preocupação com múltiplas falhas simultâneas são o aumento do número de discos nos arrays e o uso de discos SATA, que são mais lentos e têm maior capacidade que os discos tradicionais para empresa.

Logo, essas tendências levaram a um interesse cada vez maior na proteção contra mais de uma falha, e, portanto, o RAID 6 está cada vez mais sendo oferecido como uma opção e sendo usado no setor.

### Verifique você mesmo

Quais das seguintes afirmações são verdadeiras sobre os níveis RAID 1, 3, 4, 5 e 6?

1. Os sistemas RAID contam com a redundância para conseguir a alta disponibilidade.
2. RAID 1 (espelhamento) possui o mais alto overhead de disco de verificação.
3. Para pequenas escritas, RAID 3 (paridade intercalada por bit) possui a pior vazão.
4. Para grandes escritas, RAID 3, 4 e 5 possuem a mesma vazão.

**Detalhamento:** Uma questão é como o espelhamento interage com o striping. Suponha que você tivesse, digamos, quatro discos de dados para armazenar e oito discos físicos para usar. Você criaria quatro pares de discos – cada um organizado como RAID 1 – e depois faria striping dos dados nos quatro pares RAID 1? Como alternativa, você criaria dois conjuntos de quatro discos – cada um organizado como RAID 0 – e depois espelharia as escritas nos dois conjuntos RAID 0? A terminologia RAID evoluiu para chamar o primeiro de RAID 1 + 0, ou RAID 10 (“espelhos com striping”) e o segundo de RAID 0 + 1 ou RAID 01 (“striping espelhado”).

## 6.10

### Vida real: servidor Sun Fire x4150

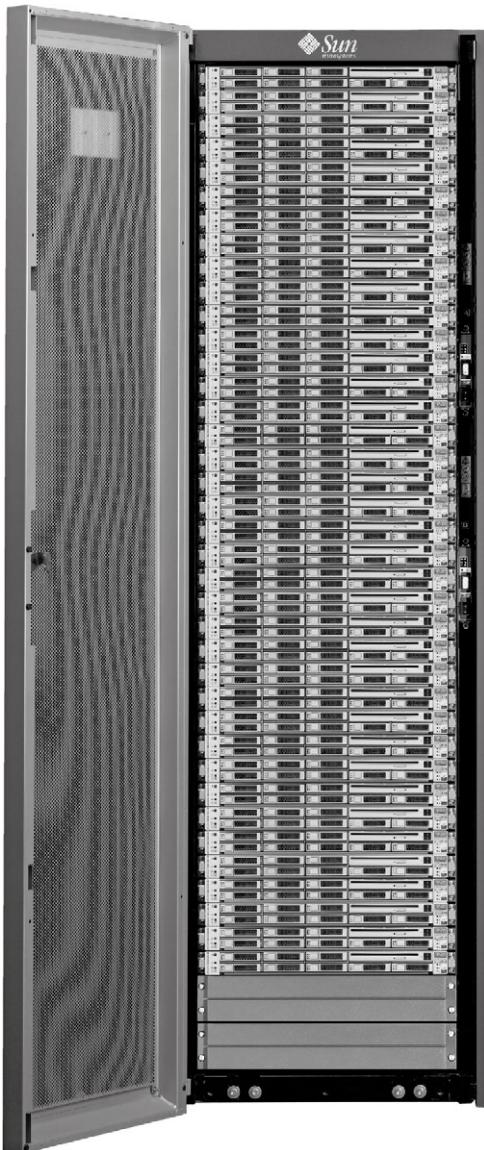
Além da revolução no modo como os microprocessadores são construídos, estamos vendo uma revolução no modo como o software é entregue. Em vez do modelo tradicional do software vendido em um CD ou entregue pela Internet para ser instalado no seu computador, a alternativa é o *software como um serviço*. Ou seja, você vai à Internet para realizar seu trabalho em um computador que roda o software que você deseja usar para fornecer o serviço desejado. O exemplo mais comum provavelmente seja a pesquisa na Web, mas existem serviços para edição e armazenamento de foto, processamento de documentos, armazenamento de banco de dados, mundos virtuais e outros. Se você procurar, provavelmente poderá encontrar uma versão de serviço de quase todo programa que utiliza no seu computador desktop.

Essa mudança levou à construção de grandes centros de dados para manter computadores e discos executando os serviços utilizados por milhões de usuários externos. Como deverão ser os computadores se eles forem projetados para serem colocados nesses grandes centros de dados? Certamente não é preciso que todos tenham monitores e teclados. Claramente, a eficiência no espaço e a eficiência na energia serão importantes se você tiver 10.000 deles em um centro de dados, além dos aspectos tradicionais do custo e desempenho.

A questão relacionada é: como deverá ser o armazenamento em um centro de dados? Embora existam muitas opções, uma versão comum é incluir discos com o processador e a memória, e tornar essa unidade inteira o bloco de montagem. Para contornar questões sobre confiabilidade, a própria aplicação faz cópias redundantes e é responsável por mantê-las coerentes e recuperar-se de falhas.

A indústria de TI em grande parte concorda com alguns padrões no projeto físico dos computadores para o centro de dados, especificamente o rack utilizado para manter os computadores no centro de dados. O mais comum é o rack de 19 polegadas (48cm) de largura. Os computadores projetados para o rack são chamados, naturalmente, de *montagem de rack*, mas também são chamados de *subrack* ou simplesmente *prateleira*. Como o espaçamento tradicional entre os furos para conectar as prateleiras é de 1,75 polegadas (4,45cm), essa distância normalmente é chamada de *unidade de rack*, ou simplesmente *unidade (U)*. O rack mais comum de 48cm tem 42 U de altura, que é 42 x 4,45, ou aproximadamente 187cm de altura. A profundidade da prateleira varia.

Logo, o menor computador para montagem em rack é de 48cm de largura e 4,45cm de altura, normalmente chamados de computadores 1U ou servidores 1U. Devido às suas dimensões, eles ganharam o apelido de *caixas de pizza*. A Figura 6.15 mostra um exemplo de um rack padrão, preenchido com 42 servidores 1U.



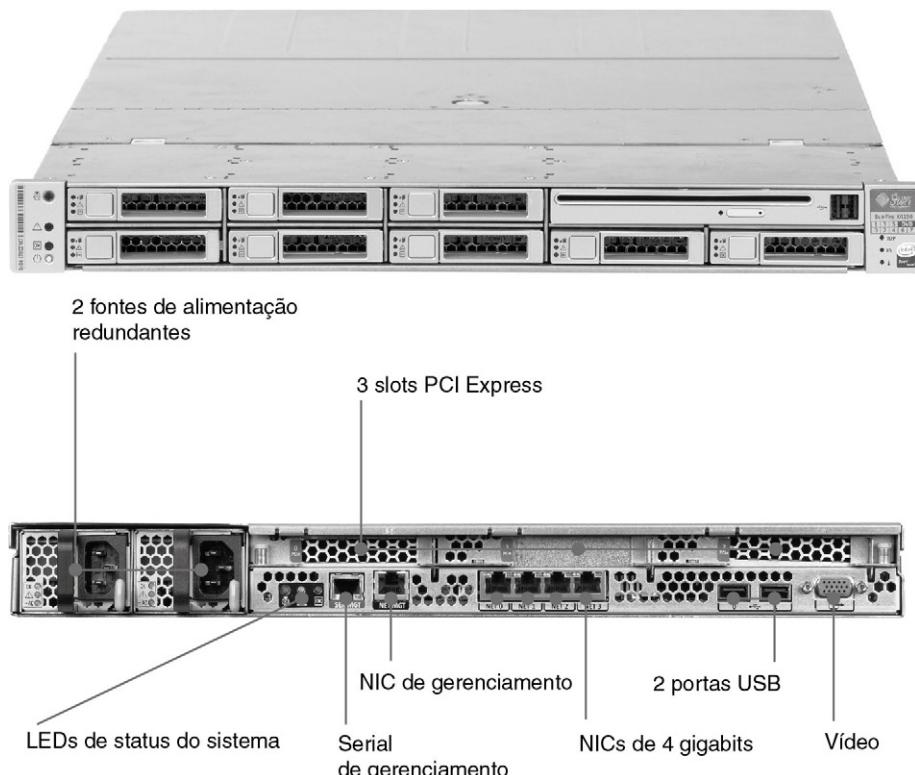
**FIGURA 6.15 Um rack padrão de 48cm preenchido com 42 servidores 1U.** Este rack tem 42 servidores “caixa de pizza” 1U. Fonte: [http://gchelpdesk.ualberta.ca/news/07mar06/cbhd\\_news\\_07mar06.php](http://gchelpdesk.ualberta.ca/news/07mar06/cbhd_news_07mar06.php).

A [Figura 6.16](#) mostra o Sun Fire x4150, um exemplo de um servidor 1U. Configurado de forma máxima, essa caixa 1U contém:

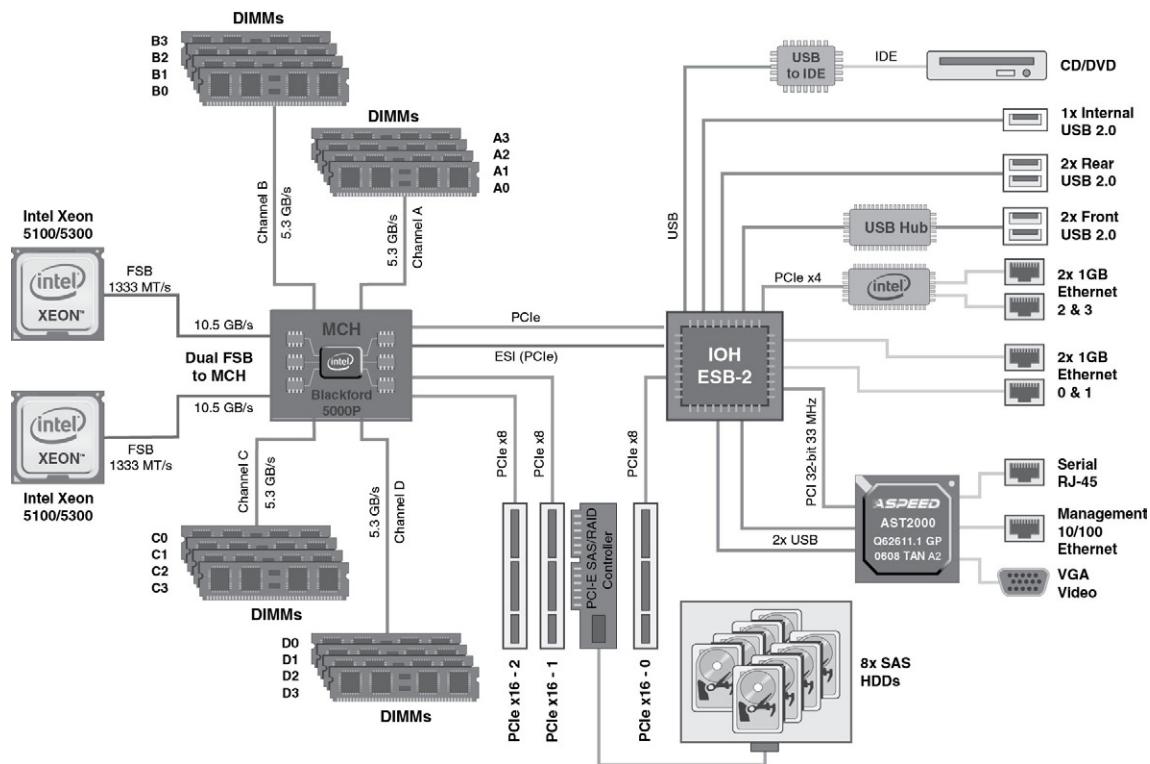
- 8 processadores de 2,66GHz, espalhados por dois soquetes (2 Intel Xeon 5345).
- 64GB de DRAM DDR2-667, espalhadas por 16 FBDIMMs de 4GB.
- 8 unidades de disco SAS de 73GB e 6,35cm a 15.000 RPM.
- 1 controlador RAID (admitindo RAID 0, RAID 1, RAID 5 e RAID 6).
- 4 portas Ethernet 10/100/1000.
- 3 portas PCI Express x8.
- 4 portas USB 2.0 externas e 1 interna.

A [Figura 6.17](#) mostra a conectividade e larguras de banda dos chips da placa mãe. As [Figuras 6.9 e 6.10](#) descrevem o chip set de E/S para o Intel 5345, e a [Figura 6.5](#) descreve os discos SAS no Sun Fire x4150.

Para esclarecer o aviso sobre o projeto de um sistema de E/S na Seção 6.8, vamos realizar uma avaliação de desempenho simples para ver onde poderiam estar os gargalos de uma aplicação hipotética.



**FIGURA 6.16 Frente e fundos do servidor 1U Sun Fire x4150.** As dimensões são 4,45cm de altura por 48cm de largura. As oito unidades de disco de 6,35cm podem ser substituídas da frente. No canto superior direito está um DVD e duas portas USB. A figura de baixo rotula os itens na parte traseira do servidor. Ela tem fontes de alimentação e ventiladores redundantes, para permitir que o servidor continue operando apesar de falhas de um desses componentes.



**FIGURA 6.17 Conexões lógicas e larguras de banda dos componentes no Sun Fire x4150.** Os três conectores PCIe permitem que placas x16 sejam conectadas, mas somente oferece oito pistas de largura de banda ao MCH. Fonte: Figura 5 do “SUN FIRE™ X4150 AND X4450 SERVER ARCHITECTURE” (veja [www.sun.com/servers/x64/x4150/](http://www.sun.com/servers/x64/x4150/)).

### Projeto do sistema de E/S

Considere o seguinte sobre o Sun Fire x4150:

### EXEMPLO

- O programa do usuário utiliza 200.000 instruções por operação de E/S.
- O sistema operacional utiliza em média 100.000 instruções por operação de E/S.
- A carga de trabalho consiste em leituras de 64KB.
- Cada processador sustenta 1 bilhão de instruções por segundo.

Ache a taxa máxima de E/S sustentável para um Sun Fire x4150 totalmente carregado para leituras aleatórias e leituras sequenciais. Considere que as leituras sempre podem ser feitas em um disco ocioso, se houver um (ou seja, ignore conflitos de disco) e que o controlador RAID não é o gargalo.

Vamos primeiro achar a taxa de E/S de um único processador. Cada E/S utiliza 200.000 instruções do usuário e 100.000 instruções do SO, de modo que  
Taxa de E/S máxima de 1 processador =

### RESPOSTA

$$\frac{\text{Taxa de execução da instrução}}{\text{Instruções por E/S}} = \frac{1 \times 10^9}{(200 + 100) \times 10^3} = 3,333 \frac{\text{E/Ss}}{\text{seg}}$$

Como um único soquete Intel 5345 tem quatro processadores, ele pode realizar 13,333 IOPS. Dois soquetes com oito processadores podem realizar 26.667 IOPS.

Vamos determinar os IOPS por disco para leituras aleatórias e sequenciais para o disco SAS de 6,35cm descrito na [Figura 6.5](#). Em vez de usar o tempo médio de busca do fabricante de disco, vamos supor que ele seja apenas um quarto desse tempo, como normalmente acontece (veja Seção 6.3). O tempo por leitura aleatória de um único disco:

$$\text{Tempo por E / S} = \text{Busca} + \text{tempo rotacional} + \text{Tempo de transferência}$$

$$= \frac{2,9}{4 \text{ ms}} + 2,0 \text{ ms} + \frac{64 \text{ KB}}{112 \text{ MB/seg}} = 3,3 \text{ ms}$$

Assim, cada disco pode completar 1000ms/3,3ms ou 303 E/Ss por segundo, e oito discos realizam 2424 leituras aleatórias por segundo.

Para leituras sequenciais, isso é apenas o tempo de transferência dividido pela largura de banda do disco:

$$\frac{112 \text{ MB/seg}}{64 \text{ KB}} = 1750 \text{ IOPS}$$

Oito discos podem realizar 14.000 leituras sequenciais de 64KB.

Precisamos ver se os caminhos dos discos para a memória e os processadores são um gargalo. Vamos começar com a interconexão PCI Express da placa RAID para chip da bridge norte. Cada pista de uma PCIe é de 250MB/segundo, de modo que oito pistas podem realizar 2GB/segundo.

$$\text{Taxa de E / S máxima da PCIe x8} = \frac{\text{Largura de banda PCI}}{\text{Bytes por E / S}} = \frac{2 \times 10^9}{64 \times 10^3} = 31,250 \frac{\text{E / Ss}}{\text{segundo}}$$

Até mesmo oito discos transferindo sequencialmente utilizam menos de metade do link PCIe x8.

Quando os dados chegam à MCB, eles precisam ser escritos na DRAM. A largura de banda de uma FBDIMM DDR2 de 667MHz é de 5336MB/segundo. Uma única DIMM pode executar

$$\frac{5336 \text{ MB / seg}}{64 \text{ KB}} = 83,375 \text{ IOPS}$$

A memória não é um gargalo mesmo com uma DIMM, e temos 16 em um Sun Fire x4150 totalmente configurado.

O link final na cadeia é o Front Side Bus que conecta o hub da bridge norte ao soquete Intel 5345. Sua largura de banda de pico é de 10,6GB/seg, mas a Seção 7.10 lhe sugere não obter mais de metade do pico. Cada E/S transfere 64KB, de modo que

$$\text{Taxa máx. E/S do FSB} = \frac{\text{Largura de banda do barramento}}{\text{Bytes por E/S}} = \frac{5,3 \times 10^9}{64 \times 10^3} = 81.540 \frac{\text{E/Ss}}{\text{segundo}}$$

Existe um Front Side Bus por soquete, de modo que o pico FSB dual é mais de 150.000 IOPS, e mais uma vez, o FSB não é um gargalo.

Logo, um Sun Fire x4150 totalmente configurado pode sustentar a largura de banda de pico dos oito discos, que é 2424 leituras aleatórias por segundo ou 14.000 leituras sequenciais por segundo.

Observe o número significativo de suposições de simplificação que são necessárias para realizar este exemplo. Na prática, muitas dessas simplificações poderiam não ser mantidas para aplicações críticas com uso intenso de E/S. Por esse motivo, executar uma carga de trabalho realista ou benchmark relevante normalmente é a única forma plausível de avaliar o desempenho da E/S.

Conforme mencionamos no início desta seção, esses novos centros de dados se preocupam com a potência e o espaço, além do custo e do desempenho. A [Figura 6.18](#) mostra a potência ociosa e de pico exigida por um Sun Fire x4150 totalmente configurado, com um desmembramento por cada componente. Vejamos as configurações alternativas do Sun Fire x4150 para economizar energia.

### Avaliação de potência do sistema de E/S

Reconfigure um Sun Fire x4150 para minimizar a potência, supondo que a carga de trabalho no exemplo anterior seja a única atividade nesse servidor 1U.

### EXEMPLO

Para conseguir as 2424 leituras aleatórias de 64KB por segundo do exemplo anterior, precisamos de todos os oito discos e da controladora PCI RAID. Pelos cálculos anteriores, uma única memória DIMM pode admitir mais de 80.000 IOPS, de modo que podemos economizar potência na memória. A memória mínima do Sun Fire x4150 é de duas DIMMs, de modo que podemos economizar a potência (e custo) de 14 DIMMs de 4GB. Um único soquete pode admitir 13.333 IOPS, de modo que também podemos reduzir o número de soquetes Intel E5345 por um. Usando os números na [Figura 6.18](#), a potência total do sistema agora é:

### RESPOSTA

$$\begin{aligned} \text{Potência Ociosa}_{\text{leituras aleatórias}} &= 154 + 2 \times 10 + 8 \times 8 + 15 = 253 \text{ watts} \\ \text{Potência Pico}_{\text{leituras aleatórias}} &= 215 + 2 \times 11 + 8 \times 8 + 15 = 316 \text{ watts} \end{aligned}$$

ou uma redução na potência por um fator de 1,6 a 1,7.

O sistema original pode desempenhar 14.000 leituras sequenciais de 64KB por segundo. Ainda precisamos de todos os discos e da controladora de disco, e o mesmo número de DIMMs pode tratar dessa carga mais alta. Essa carga de trabalho excede uma potência de processamento do único soquete Intel E5345, de modo que precisamos acrescentar um segundo.

$$\begin{aligned} \text{Potência Ociosa}_{\text{leituras sequenciais}} &= 154 + 22 + 2 \times 10 + 8 \times 8 + 15 = 275 \text{ watts} \\ \text{Potência Pico}_{\text{leituras sequenciais}} &= 215 + 79 + 2 \times 11 + 8 \times 8 + 15 = 395 \text{ watts} \end{aligned}$$

ou uma redução na potência por um fator de 1,4 a 1,5.

Item	Componentes			Sistema			
	Idle	Pico	Número	Idle		Pico	
Socket único Intel 2,66 GHz E5345, chipset MCB/IOH Intel 5000, controles Ethernet, fontes de força, ventiladores	154 W	215 W	1	154 W	37%	215 W	39%
Socket adicional Intel 2,66 GHz E5345	22 W	79 W	1	22 W	5%	79 W	14%
DDR2-667 5300 FBDIMM 4GB	10 W	11 W	16	160 W	39%	176 W	32%
Drive de disco 15 K, 73 GB SAS	8 W	8 W	8	64 W	15%	64 W	12%
PCIe x8 RAID Controladora de Disco	15 W	15 W	1	15 W	4%	15 W	3%
Total	—	—	—	415 W	100%	549 W	100%

**FIGURA 6.18 Potência de pico e idle do Sun Fire x4150 totalmente configurado.** Esses experimentos vieram enquanto executando SPECJBB com 29 configurações diferentes, então o pico de potência poderia ser diferente enquanto executando aplicações diferentes. Fonte: [www.sun.com/servers/x64/x4150/calc](http://www.sun.com/servers/x64/x4150/calc).

## 6.11

### Tópicos avançados: Redes

As redes estão ganhando mais popularidade com o passar do tempo e, diferente de outros dispositivos de E/S, existem muitos livros e cursos sobre elas. Para os leitores que não fizeram nenhum curso nem leram livros sobre redes, a  Seção 6.11, no site, oferece uma visão geral dos tópicos e da terminologia, incluindo interligação de redes, o modelo OSI, famílias de protocolos, como TCP/IP, redes de longa distância, como ATM, redes locais, como Ethernet, e redes sem fio, como IEEE 802.11.

## 6.12

### Falácia e armadilhas

*Falácia: o tempo médio para falha indicado para discos é 1.200.000 horas ou quase 140 anos, de modo que os discos praticamente nunca falham.*

As práticas de marketing atuais dos fabricantes de disco podem enganar os usuários. Como esse MTTF é calculado? No início do processo, os fabricantes colocam milhares de discos em uma sala, os colocam para trabalhar por alguns meses, e contam a quantidade que falha. Eles calculam o MTTF como o número total de horas que os discos estiveram acumuladamente ativos dividido pelo número que falhou.

Um problema é que esse número é muito superior ao tempo de vida de um disco, que normalmente é cinco anos ou 43.800 horas. Para esse grande MTTF fazer algum sentido, esses fabricantes argumentam que o cálculo corresponde a um usuário que compra um disco, e depois continua substituindo o disco a cada cinco anos – o tempo de vida planejado do disco. A reivindicação é que, se muitos clientes (e seus bisnetos) fizessem isso para o próximo século, na média eles substituiriam um disco 27 vezes antes de uma falha, ou cerca de 140 anos.

Uma medida mais útil seria a porcentagem de discos que falham, chamada taxa anual de falha (AFR). Considere 1.000 discos com um MTTF de 1.200.000 horas e que os discos sejam usados 24 horas por dia. Se você substituisse os discos que falharam por um novo com as mesmas características de confiabilidade, o número que falharia por ano (8.760 horas) é

$$\text{Discos falhos} = \frac{1.000 \text{ unidades} \times 8.760 \text{ horas/unidade}}{1.200.000 \text{ horas/falha}} = 7,3$$

Explicando de uma forma alternativa, a AFR é 0,73%. Os fabricantes de disco estão começando a citar a AFR além do MTTF para dar aos usuários uma melhor intuição sobre o que esperar a respeito de seus produtos.

*Falácia: as taxas de falha de disco em campo combinam com suas especificações.*

Dois estudos recentes avaliaram grandes coleções de discos para verificar o relacionamento entre os resultados em campo comparados com as especificações. Um estudo foi de quase 100.000 discos ATA e SCSI que tinham uma cotação de MTTF de 1.000.000 a 1.500.000 horas, ou AFR de 0,6% a 0,8%. Eles descobriram que AFRs de 2% a 4% são comuns, normalmente três a cinco vezes as taxas especificadas [Schroeder e Gibson, 2007]. Um segundo estudo de mais de 100.000 discos ATA, que tinham um valor AFR de aproximadamente 1,5%, viu taxas de falha de 1,7% das unidades em seu primeiro ano subirem para 8,6% das unidades em seu terceiro ano, ou cerca de cinco a seis vezes a taxa especificada [Pinheiro, Weber e Barroso, 2007].

*Falácia: uma interconexão de 1GB/seg pode transferir 1GB de dados em 1 segundo.*

Primeiro, você em geral não pode usar 100% de qualquer recurso do computador. Para um barramento, você ficaria satisfeita em conseguir 70% a 80% da largura de banda de pico. O tempo para enviar o endereço, o tempo para confirmar os sinais e os atrasos enquanto se espera para usar um barramento ocupado estão entre os motivos para você não poder usar 100% de um barramento.

Segundo, a definição de um gigabyte de armazenamento e um gigabyte por segundo de largura de banda não correspondem. Conforme discutimos na Seção 6.7, as medidas de largura de banda de E/S normalmente são cotadas em base 10 (ou seja, 1GB/seg =  $10^9$  bytes/seg), enquanto 1GB de dados normalmente é uma medida na base 2 (ou seja, 1GB =  $2^{30}$  bytes). Qual é o significado dessa distinção? Se pudéssemos usar 100% do barramento para a transferência de dados, o tempo para transferir 1GB de dados em uma interconexão de 1GB/seg seria, na realidade,

$$\frac{2^{30}}{10^9} = \frac{1.073.741.824}{1.000.000.000} = 1,073741824 \approx 1,07 \text{ segundo}$$

*Armadilha: tentar oferecer recursos apenas dentro da rede versus fim a fim.*

O problema é fornecer em um nível inferior recursos que só podem ser cumpridos no nível mais alto, satisfazendo assim apenas parcialmente à demanda da comunicação. Saltzer, Reed e Clark [1984] explicam o *argumento de fim a fim* como

*A função em questão só pode ser especificada completa e corretamente com o conhecimento e a ajuda da aplicação que fica nas extremidades do sistema de comunicação. Portanto, não é possível oferecer essa função questionada como um recurso do próprio sistema de comunicação.*

Seu exemplo da armadilha foi uma rede no MIT que usava vários gateways, cada qual acrescentando uma soma de verificação de um gateway para o seguinte. Os programadores da aplicação assumiram a precisão garantida pela soma de verificação, acreditando incorretamente que a mensagem estava protegida enquanto armazenada na memória de cada gateway. Um gateway tinha uma falha intermitente que trocava um par de bytes para cada milhão de bytes transferidos. Com o tempo, o código-fonte de um sistema operacional era repetidamente passado pelo gateway, adulterando, dessa forma, o código. A única solução foi corrigir os arquivos-fonte infectados, comparando as listagens em papel e reparando o código manualmente! Se as somas de verificação tivessem sido calculadas e verificadas pela aplicação rodando nos sistemas na ponta, a segurança teria sido garantida.

No entanto, existe uma função útil para verificações intermediárias, desde que a verificação fim a fim esteja disponível. Ela pode mostrar que *algo* está errado entre dois nós, mas não aponta onde se encontra o problema. As verificações intermediárias podem descobrir *qual componente* está errado. Você precisa de ambos para reparar.

*Armadilha: mover funções da CPU para o processador de E/S, esperando melhorar o desempenho sem uma análise cuidadosa.*

Existem muitos exemplos dessa armadilha pegando as pessoas, embora os processadores de E/S, quando usados de forma correta, certamente podem melhorar o desempenho. Um caso frequente dessa falácia é o uso de interfaces de E/S inteligentes que, devido ao maior overhead para configurar uma requisição de E/S, pode ter uma latência pior do que uma atividade de E/S controlada pelo processador (embora, se o processador for liberado suficientemente, a vazão do sistema ainda possa aumentar). Constantemente, o desempenho cai quando o processador de E/S tem um desempenho muito inferior ao do processador principal. Como consequência, uma quantidade pequena do tempo de processador principal é substituída por uma quantidade maior de tempo do processador de E/S. Os projetistas de estações de trabalho têm visto esses dois fenômenos repetidamente.

Myer e Sutherland [1968] escreveram um artigo clássico sobre a escolha entre complexidade e desempenho nos controladores de E/S. Apanhando emprestado o conceito religioso da “roda da reencarnação”, eles, por fim, observaram que eram apanhados em

um loop de aumentar continuamente a potência de um processador de E/S até que ele precisasse do seu próprio coprocessador mais simples:

*Enfrentamos a tarefa começando com um esquema simples e depois acrescentando comandos e recursos que achamos que melhorariam o poder da máquina. Gradualmente, o processador [de vídeo] tornava-se mais complexo... Finalmente, o processador de vídeo ficou semelhante a um computador completo, com alguns recursos gráficos especiais. E depois aconteceu uma coisa estranha. Sentimo-nos compelidos a acrescentar ao processador um segundo processador subsidiário que, por si só, começou a aumentar em complexidade. Foi então que descobrimos a verdade perturbadora. Projetar um processador de vídeo pode se tornar um processo cíclico sem fim. Na verdade, descobrimos que o processo era tão frustrante que passamos a chamá-lo de “roda da reencarnação”.*

*Armadilha: usar fitas magnéticas para o backup de discos.*

Mais uma vez, isso é uma falácia e uma armadilha. As fitas magnéticas têm feito parte dos sistemas de computador tanto quanto os discos, pois utilizam tecnologia semelhante aos discos e, por isso, historicamente têm seguido as mesmas melhorias na densidade. A diferença de custo-desempenho histórica entre discos e fitas é baseada em um disco selado, rotativo, com menor tempo de acesso do que o acesso sequencial à fita, mas os spools removíveis de fita magnética significam que muitas fitas podem ser usadas por leitora e que elas podem ser muito longas, de modo que possuem alta capacidade. De modo que, no passado, uma única fita magnética poderia manter o conteúdo de muitos discos, e por ser de 10 a 100 vezes mais barata por gigabyte do que os discos, esse era um meio de backup útil.

A alegação foi de que as fitas magnéticas precisam acompanhar os discos, pois as inovações nos discos precisam ajudar as fitas. Essa alegação foi importante porque as fitas eram um pequeno mercado e não poderiam dispor de um grande esforço de pesquisa e desenvolvimento separado. Um motivo para o mercado ser pequeno é que os proprietários de desktop geralmente não fazem backup de discos em fita, e assim, enquanto os desktops são um grande mercado para discos, eles são um pequeno mercado para fitas.

Infelizmente, o maior mercado levou os discos a melhorarem muito mais rapidamente do que as fitas. Entre 2000 a 2002, o disco muito mais popular era maior do que a maior fita. Nesse mesmo espaço de tempo, o preço por gigabyte de discos ATA caiu para menos do que o das fitas. Os defensores da fita agora alegam que elas possuem requisitos de compatibilidade que não são impostos sobre os discos; as leitoras de fita precisam ler ou escrever a geração atual e anterior de fitas e precisam ler as quatro últimas gerações de fitas. Como os discos são sistemas fechados, as cabeças de disco só precisam ler os pratos embutidos, e essa vantagem explica por que os discos estão melhorando muito mais rapidamente.

Hoje, algumas organizações retiraram as fitas, usando redes e discos remotos para replicar os dados geograficamente. Na verdade, muitas empresas oferecendo software como serviço utilizam componentes baratos, mas replicam os dados em nível de aplicação por diferentes locais. Os locais são selecionados de modo que os desastres não prejudiquem os dois locais, permitindo um tempo de recuperação instantâneo. (Um tempo de recuperação longo é outra desvantagem séria da natureza serial das fitas magnéticas.) Essa solução depende dos avanços na capacidade do disco e na largura de banda da rede, para fazer sentido economicamente, mas esses dois estão recebendo um investimento muito maior e, portanto, possuem registros de realização recentes melhores do que a fita.

*Falácia: os sistemas operacionais são o melhor local para programar acessos ao disco.*

Como dissemos na Seção 6.3, interfaces de nível mais alto, como ATA e SCSI, oferecem endereços de bloco lógicos para o sistema operacional hospedeiro. Dada essa abstração de alto nível, o melhor que um SO pode fazer para tentar ajudar no desempenho é classificar os endereços lógicos de bloco em ordem crescente. Porém, como o disco conhece o mapeamento real dos endereços lógicos na geometria física de setores, trilhas e superfícies, ele pode reduzir as latências de rotação e de busca pelo reescalonamento.

Por exemplo, suponha que a carga de trabalho seja quatro leituras [Anderson, 2003]:

Operação	LBA inicial	Tamanho
Leitura	724	8
Leitura	100	16
Leitura	9987	1
Leitura	26	128

O hospedeiro poderia reordenar as quatro leituras por ordem de bloco lógico:

Operação	LBA inicial	Tamanho
Leitura	26	128
Leitura	100	16
Leitura	724	8
Leitura	9987	1

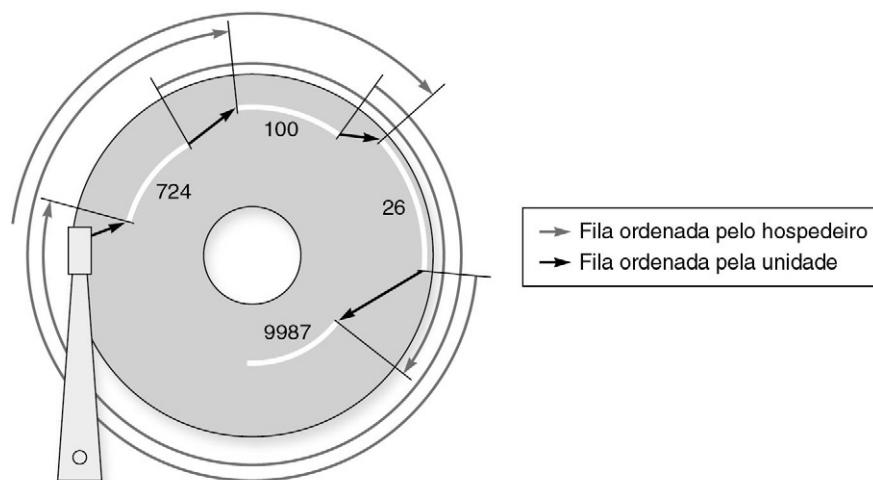
Dependendo do local relativo dos dados no disco, a reordenação poderia tornar isso pior, como mostra a [Figura 6.19](#). As leituras programadas pelo disco terminam em três quartos de uma rotação do disco, mas as leituras programadas pelo SO exigem três rotações.

*Armadilha: usar uma taxa de transferência de pico de uma parte do sistema de E/S para fazer projeções de desempenho ou comparações de desempenho.*

Muitos dos componentes de um sistema de E/S, desde os dispositivos até os controladores e barramentos, são especificados por meio de suas larguras de banda de pico. Na prática, essas medidas de largura de banda de pico em geral são baseadas em suposições irrealistas sobre o sistema ou não são alcançáveis, devido a outras limitações do sistema. Por exemplo, cotando o desempenho do barramento, a velocidade de transferência de pico às vezes é especificada usando um sistema de memória impossível de criar. Para sistemas em rede, o overhead do software para iniciar a comunicação é ignorado.

O barramento PCI de 32 bits, 33MHz, possui uma largura de banda de pico de cerca de 133MB/seg. Na prática, até mesmo para transferências longas, é difícil sustentar mais do que cerca de 80MB/seg para sistemas de memória reais.

A Lei de Amdahl também nos lembra que a vazão de um sistema de E/S será limitada pelo componente de menor desempenho no caminho de E/S.



**FIGURA 6.19 Exemplo mostrando acessos programados pelo SO versus disco, rotulados com “fila ordenada pelo hospedeiro” e “fila ordenada pela unidade”.** O primeiro leva três rotações para completar as quatro leituras, enquanto o segundo as completa em apenas três quartos de uma rotação (de Anderson [2003]).

## 6.13

### Comentários finais

Os sistemas de E/S são avaliados em diversas características diferentes: confiança; a variedade de dispositivos de E/S aceitos; o número máximo de dispositivos de E/S; custo; e desempenho, medidos tanto em latência quanto em vazão. Esses objetivos levam a esquemas bastante variados para interface de dispositivos de E/S. Nos sistemas inferiores e intermediários, o DMA com buffer provavelmente será o mecanismo de transferência dominante. Nos sistemas de alto nível, a latência e a largura de banda podem ser ambos importantes, e o custo pode ser secundário. Vários caminhos para dispositivos de E/S com buffer limitado normalmente caracterizam sistemas de E/S de alto nível. Em geral, ser capaz de acessar os dados em um dispositivo de E/S a qualquer tempo (alta disponibilidade) torna-se mais importante quando os sistemas crescem. Como resultado, a redundância e os mecanismos de correção de erros tornam-se mais e mais prevalentes enquanto ampliamos o sistema.

As demandas de armazenamento e rede estão crescendo em velocidades sem precedentes, em parte devido às demandas crescentes para que toda a informação esteja na ponta dos seus dedos. Uma estimativa é que a quantidade de informação criada em 2002 foi de 5 exabytes – equivalente a 500.000 cópias do texto da Biblioteca do Congresso dos Estados Unidos –, e essa quantidade total de informações no mundo dobrou nos últimos três anos [Lyman e Varian, 2003].

As direções futuras da E/S incluem expandir o alcance das redes com e sem fio, com quase todo dispositivo potencialmente tendo um endereço IP, e a expansão do papel da memória flash nos sistemas de armazenamento.

## Entendendo o desempenho dos programas

O desempenho de um sistema de E/S, seja ele medido por largura de banda ou latência, depende de todos os elementos no caminho entre o dispositivo e a memória, incluindo o sistema operacional que gera os comandos de E/S. A largura de banda da interconexão, da memória e do dispositivo determinam a velocidade de transferência máxima do dispositivo ou para o dispositivo. De modo semelhante, a latência depende da latência do dispositivo, junto com qualquer latência imposta pelo sistema de memória ou barramentos. A largura de banda efetiva e a latência de resposta também dependem de outras requisições de E/S que podem causar disputa por algum recurso no caminho. Finalmente, o sistema operacional é um gargalo. Em alguns casos, o sistema operacional leva muito tempo para entregar uma solicitação de E/S de um programa de usuário a um dispositivo de E/S, levando a uma alta latência. Em outros casos, o sistema operacional efetivamente limita a largura de banda de E/S, devido às limitações no número de operações de E/S simultâneas que ele pode admitir.

Lembre-se de que, embora o desempenho possa ajudar a vender um sistema de E/S, os usuários, em sua maioria, exigem confiabilidade e capacidade dos seus sistemas de E/S.

## 6.14

### Perspectiva histórica e leitura adicional

A história dos sistemas de E/S é fascinante. A  Seção 6.14 oferece um breve histórico dos discos magnéticos, RAID, memória flash, bancos de dados, a internet, a world wide web e como a ethernet continua a triunfar sobre seus desafiantes.

## 6.15 Exercícios<sup>1</sup>

### Exercício 6.1

A [Figura 6.2](#) descreve diversos dispositivos de E/S em termos de seu comportamento, parceria e taxa de dados. Porém, essas classificações normalmente não oferecem uma imagem completa do fluxo de dados dentro de um sistema. Explore as classificações de dispositivo para os seguintes dispositivos:

a.	Piloto automático
b.	Termostato automatizado

**6.1.1** [5] <6.1> Em relação aos dispositivos listados na tabela, identifique as interfaces de E/S e classifique-as em termos de seu comportamento e parceria.

**6.1.2** [5] <6.1> Para as interfaces identificadas no problema anterior, estime sua taxa de dados.

**6.1.3** [5] <6.1> Para as interfaces identificadas no problema anterior, determine se a taxa de dados ou a taxa de operação é a melhor medida do desempenho.

### Exercício 6.2

Mean Time Between Failures (MTBF), Mean Time To Replacement (MTTR) e Mean Time To Failure (MTTF) são medidas úteis para avaliar a confiabilidade e a disponibilidade de um recurso de armazenamento. Explore esses conceitos respondendo às perguntas sobre dispositivos com as métricas a seguir.

	MTTF	MTTR
a.	3 anos	1 dia
b.	7 anos	3 dias

**6.2.1** [5] <6.1, 6.2> Calcule o MTBF para cada um dos dispositivos na tabela.

**6.2.2** [5] <6.1, 6.2> Calcule a disponibilidade para cada um dos dispositivos na tabela.

**6.2.3** [5] <6.1, 6.2> O que acontece à disponibilidade quando o MTTR se aproxima de 0. Essa situação é real?

**6.2.4** [5] <6.1, 6.2> O que acontece com a disponibilidade quando o MTTR se torna muito alto, ou seja, um dispositivo é difícil de reparar? Isso significa que o dispositivo tem baixa disponibilidade?

### Exercício 6.3

Os tempos médio e mínimo para ler e escrever nos dispositivos de armazenamento são medições comuns usadas para comparar dispositivos. Usando as técnicas do Capítulo 6, calcule os valores relacionados ao tempo de leitura e escrita para discos com as características a seguir.

<sup>1</sup> Contribuição de Perry Alexander, da Universidade do Kansas.

	Tempo de busca médio	RPM	Taxa de transferência de disco	Taxa de transferência da controladora
a.	10 ms	7.500	90 MBytes/s	100 MBits/s
b.	7 ms	10.000	40 MBytes/s	200 MBits/s

**6.3.1** [10] <6.2, 6.3> Calcule o tempo médio para ler ou escrever um setor de 1024 bytes de cada disco listado na tabela.

**6.3.2** [10] <6.2, 6.3> Calcule o tempo mínimo para ler ou escrever um setor de 2048 bytes de cada disco listado na tabela.

**6.3.3** [10] <6.2, 6.3> Para cada disco na tabela, determine o fator dominante ao desempenho. Especificamente, se você pudesse fazer uma melhoria em qualquer aspecto do disco, o que escolheria? Se não houver um fator dominante, explique por quê.

### Exercício 6.4

No fim, o projeto do sistema de armazenamento requer consideração de cenários de uso e também de parâmetros de disco. Diferentes situações exigem diferentes métricas. Vamos tentar avaliar sistematicamente os sistemas de disco. Explore diferenças no modo como os sistemas de armazenamento devem ser avaliados respondendo as perguntas sobre as aplicações a seguir.

a.	Sistema de controle de aeronaves
b.	Central telefônica

**6.4.1** [5] <6.2, 6.3> Para cada aplicação, diminuir o tamanho do setor durante leituras e escritas melhoraria o desempenho? Explique sua resposta.

**6.4.2** [5] <6.2, 6.3> Para cada aplicação, aumentar a velocidade de rotação de disco melhora o desempenho? Explique sua resposta.

**6.4.3** [5] <6.2, 6.3> Para cada aplicação, aumentar a velocidade de rotação do disco melhora o desempenho do sistema dado que o MTTF diminui? Explique sua resposta.

### Exercício 6.5

A memória FLASH é um dos primeiros competidores verdadeiros para as unidades de disco tradicionais. Explore as implicações da memória FLASH respondendo as perguntas sobre as aplicações a seguir.

a.	Sistema de controle de aeronaves
b.	Central telefônica

**6.5.1** [5] <6.2, 6.3, 6.4> Ao passarmos para unidades de estado sólido construídas de memória FLASH, o que mudará sobre os tempos de leitura de disco considerando que a taxa de transferência de dados permanece constante?

**6.5.2** [10] <6.2, 6.3, 6.4> Cada aplicação se beneficiaria de uma unidade FLASH em estado sólido, dado que o custo é um fator de projeto?

**6.5.3** [10] <6.2, 6.3, 6.4> Cada aplicação seria imprópria para uma unidade FLASH no estado sólido, dado que o custo NÃO é um fator de projeto?

## Exercício 6.6

Explore a natureza da memória FLASH respondendo as perguntas relacionadas a desempenho para memórias FLASH com as características a seguir.

	Taxa de transferência de dados	Taxa de transferência da controladora
a.	120 MB/s	100 MB/s
b.	100 MB/s	90 MB/s

**6.6.1** [10] <6.2, 6.3, 6.4> Calcule o tempo médio para leitura ou escrita de um setor de 1024 bytes para cada memória FLASH listada na tabela.

**6.6.2** [10] <6.2, 6.3, 6.4> Calcule o tempo mínimo para leitura ou escrita de um setor de 512 bytes para cada memória FLASH listada na tabela.

**6.6.3** [5] <6.2, 6.3, 6.4> A [Figura 6.6](#) mostra que os tempos de acesso de leitura e escrita da memória FLASH aumentam à medida que a memória FLASH se torna maior. Isso é inesperado? Que fatores causam isso?

## Exercício 6.7

A E/S pode ser realizada sincrônica ou assincronicamente. Explore as diferenças respondendo as perguntas de desempenho sobre os periféricos a seguir.

a.	Impressora
b.	Scanner

**6.7.1** [5] <6.5> Qual seria o tipo de barramento mais apropriado (síncrono ou assíncrono) para tratar das comunicações entre uma CPU e os periféricos listados na tabela?

**6.7.2** [5] <6.5> Que problemas os barramentos longos e síncronos causariam para as conexões entre uma CPU e os periféricos listados na tabela?

**6.7.3** [5] <6.5> Que problemas os barramentos assíncronos causariam para as conexões entre uma CPU e os periféricos listados na tabela?

## Exercício 6.8

Entre os tipos de barramento mais comuns utilizados na prática atualmente estão Fire-Wire (IEEE 1394), USB, PCI e SATA. Embora todos os quatro sejam assíncronos, eles são implementados de diferentes maneiras, dando-lhes diferentes características. Explore as diferentes estruturas de barramento respondendo as perguntas sobre os barramentos e os periféricos a seguir.

a.	Mouse
b.	Coprocessador Gráfico

**6.8.1** [5] <6.5> Selecione um barramento apropriado (FireWire, USB, PCI ou SATA) para os periféricos listados na tabela. Explique por que o barramento selecionado é apropriado. (Veja na [Figura 6.8](#) as principais características de cada barramento.)

**6.8.2** [20] <6.5> Use os recursos *on-line* ou de biblioteca e resuma a estrutura de comunicação para cada tipo de barramento. Identifique o que o controlador de barramento faz e onde o controle se encontra fisicamente.

**6.8.3** [15] <6.5> Explique as limitações de cada um dos tipos de barramento. Explique por que essas limitações precisam ser levadas em consideração quando se usa o barramento.

### Exercício 6.9

A comunicação com dispositivos de E/S é alcançada por meio de combinações de polling, tratamento de interrupção, mapeamento de memória e comandos especiais de E/S. Responda as perguntas sobre a comunicação com subsistemas de E/S para as aplicações a seguir usando combinações dessas técnicas.

a.	Piloto automático
b.	Termostato automatizado

**6.9.1** [5] <6.6> Descreva o polling do dispositivo. Cada aplicação na tabela seria apropriada para a comunicação usando as técnicas de polling? Explique.

**6.9.2** [5] <6.6> Descreva a comunicação controlada por interrupção. Para cada aplicação na tabela, se o polling for impróprio, explique as técnicas controladas por interrupção que poderiam ser usadas.

**6.9.3** [10] <6.6> Para as aplicações listadas na tabela, esboce um projeto de comunicação mapeada na memória. Identifique os locais de memória reservados e esboce seu conteúdo.

**6.9.4** [10] <6.6> Para as aplicações listadas na tabela, esboce um projeto para os comandos implementando a comunicação controlada por comando. Identifique os comandos e sua interação com o dispositivo.

**6.9.5** [5] <6.6> Faz sentido definir os subsistemas de E/S que usam uma combinação de mapeamento de memória e comunicação controlada por comando? Explique sua resposta.

### Exercício 6.10

A Seção 6.6 define um processo de oito etapas para tratar das interrupções. Os registradores Cause e Status juntos oferecem informações sobre a causa da interrupção e o status do sistema de tratamento da interrupção. Explore o tratamento da interrupção respondendo as perguntas sobre as seguintes combinações de interrupções.

a.	Controlador de dados Ethernet	Controlador do Mouse	Reiniciar
b.	Controlador do Mouse	Desligamento	Superaquecimento

**6.10.1** [5] <6.6> Quando uma interrupção é detectada, o registrador Status é salvo e tudo além da interrupção de mais alta prioridade é desabilitado. Por que as interrupções de baixa prioridade são desabilitadas? Por que o registrador Status é salvo antes de desabilitar as interrupções?

**6.10.2** [10] <6.6> Priorize as interrupções a partir dos dispositivos listados em cada linha da tabela.

**6.10.3** [10] <6.6> Esboce como uma interrupção de cada um dos dispositivos listados na tabela seria tratada.

**6.10.4** [5] <6.6> O que acontece se o bit “interrupt enable” do registrador Cause não for definido no tratamento de uma interrupção? Que valor é assumido pela máscara de interrupção para realizar a mesma coisa?

**6.10.5** [5] <6.6> A maioria dos sistemas de tratamento de interrupção é implementada no sistema operacional. Que suporte do hardware poderia ser acrescentado de modo a tornar o tratamento de interrupção mais eficiente? Compare sua solução com o suporte de hardware em potencial para as chamadas de função.

**6.10.6** [5] <6.6> Em algumas implementações de tratamento de interrupção, uma interrupção causa um salto imediato para um vetor de interrupção. Em vez de um registrador Cause, em que cada interrupção define um bit, cada interrupção tem seu próprio vetor de interrupção. O mesmo sistema de interrupção de prioridade pode ser implementado usando essa técnica? Existe alguma vantagem nessa técnica?

### Exercício 6.11

Direct Memory Access (DMA) permite que os dispositivos acessem a memória diretamente em vez de utilizar a CPU. Isso pode agilizar bastante o desempenho dos periféricos, mas aumenta a complexidade das implementações do sistema de memória. Explore as implicações do DMA respondendo as perguntas sobre os periféricos a seguir.

a.	Controlador do mouse
b.	Controlador da ethernet

**6.11.1** [5] <6.6> A CPU abre mão do controle da memória quando o DMA está ativo? Por exemplo, um periférico pode simplesmente se comunicar com a memória diretamente, evitando a CPU por completo?

**6.11.2** [10] <6.6> Dos periféricos listados na tabela, qual se beneficiaria com o DMA? Que critérios determinam se o DMA é apropriado?

**6.11.3** [10] <6.6> Dos periféricos listados na tabela, qual poderia causar problemas de coerência com o conteúdo da cache? Que critérios determinam se as questões de coerência devem ser enfocadas?

**6.11.4** [5] <6.6> Descreva os problemas que poderiam ocorrer quando se mistura DMA e memória virtual. Qual dos periféricos na tabela poderia gerar esses problemas? Como eles podem ser evitados?

### Exercício 6.12

A métrica para desempenho de E/S pode variar bastante de uma aplicação para outra. Enquanto o número de transações processadas domina o desempenho em algumas situações, a vazão de dados domina em outras. Explore a avaliação do desempenho de E/S respondendo as perguntas para as aplicações a seguir.

a.	Computações matemáticas
b.	Chat on-line

**6.12.1** [10] <6.7> Para cada aplicação na tabela, o desempenho da E/S domina o desempenho do sistema?

**6.12.2** [10] <6.7> Para cada aplicação na tabela, o desempenho da E/S é medido melhor usando a vazão de dados brutos?

**6.12.3** [5] <6.7> Para cada aplicação na tabela, o desempenho da E/S é medido melhor usando o número de transações processadas?

**6.12.4** [5] <6.7> Existe algum relacionamento entre as medidas de desempenho dos dois problemas anteriores e escolher entre o uso da comunicação por polling ou controlada por interrupção? E a escolha entre usar E/S mapeada pela memória ou controlada por comando?

### Exercício 6.13

Os benchmarks desempenham um papel importante na avaliação e seleção de dispositivos periféricos. Para que os benchmarks sejam úteis, eles devem exibir propriedades semelhante àquelas experimentadas por um dispositivo em uso normal. Explore os benchmarks e a seleção de dispositivo respondendo as perguntas sobre as aplicações a seguir.

a.	Computações matemáticas
b.	Chat <i>on-line</i>

**6.13.1** [5] <6.7> Para cada aplicação na tabela, defina as características que um conjunto de benchmarks deve exibir quando se avaliar um subsistema de E/S?

**6.13.2** [15] <6.7> Usando recursos *on-line* ou de biblioteca, identifique um conjunto de benchmarks padrão para aplicações na tabela. Por que os benchmarks padrão ajudam?

**6.13.3** [5] <6.7> Faz sentido avaliar um subsistema de E/S fora do sistema maior do qual ele faz parte?

### Exercício 6.14

RAID está entre as técnicas mais comuns de paralelismo e redundância nos sistemas de armazenamento. O nome Redundant Arrays of Inexpensive Disks implica em várias coisas sobre arrays RAID que exploraremos no contexto das atividades a seguir.

a.	Computações matemáticas de alta performance
b.	Serviços de vídeo <i>on-line</i>

**6.14.1** [10] <6.9> RAID 0 utiliza o striping para forçar o acesso paralelo entre muitos discos. Por que o striping melhora o desempenho do disco? Para cada uma das atividades listadas na tabela, o striping ajudará a alcançar melhor seus objetivos?

**6.14.2** [5] <6.9> RAID 1 espelha dados entre vários discos. Supondo que discos pouco dispendiosos possuem MTBF mais baixo que os discos dispendiosos, como a redundância usando discos pouco dispendiosos pode resultar em um sistema com MTBF inferior? Use a definição matemática do MTBF para explicar sua resposta. Para cada uma das atividades listadas na tabela, RAID 1 ajudará a conseguir melhor seus objetivos?

**6.14.3** [5] <6.9> Assim como RAID 1, RAID 3 oferece disponibilidade de dados mais alta. Explique a escolha entre RAID 1 e RAID 3. Cada uma das aplicações listadas na tabela se beneficiaria de RAID 3 em vez de RAID 1?

### Exercício 6.15

RAID 3, RAID 4 e RAID 5 utilizam o sistema de paridade para proteger blocos de dados. Especificamente, um bloco de paridade está associado a uma coleção de blocos de dados. Cada linha na tabela a seguir mostra os valores dos blocos de dados e paridade, conforme descritos na [Figura 6.13](#).

	Novo D0	D0	D1	D2	D3	P
a.	7453	AB9C	AABB	0098	549C	2FFF
b.	F245	7453	DD25	AABB	FEFE	FEFF

**6.15.1** [10] <6.9> Calcule a nova paridade P' para RAID 3 para as linhas a e b da tabela.

**6.15.2** [10] <6.9> Calcule a nova paridade P' para RAID 4 para as linhas a e b da tabela.

**6.15.3** [5] <6.9> RAID 3 ou RAID 4 é mais eficiente? Existem motivos para RAID 3 ser preferível a RAID 4?

**6.15.4** [5] <6.9> RAID 4 e RAID 5 utilizam aproximadamente o mesmo mecanismo para calcular e armazenar a paridade para blocos de dados. Como RAID 5 difere de RAID 4 e para que aplicações RAID 5 seria mais eficiente?

**6.15.5** [5] <6.9> As melhorias de velocidade do RAID 4 e RAID 5 crescem com relação a RAID 3 à medida que o tamanho do bloco protegido aumenta. Por que isso acontece? Existe alguma situação em que RAID 4 e RAID 5 não seria mais eficiente do que RAID 3?

### Exercício 6.16

O aparecimento de servidores Web para e-commerce, armazenamento *on-line* e comunicação tornou os servidores de disco aplicações fundamentais. A disponibilidade e a velocidade são medidas bem conhecidas para servidores de disco, mas o consumo de energia está se tornando cada vez mais importante. Responda as perguntas sobre configuração e avaliação de servidores de disco com os parâmetros a seguir.

	Instruções de programa/ Operações de E/S	Instruções do SO/Operação de E/S	Carga de Trabalho (KB lidos)	Velocidade do processador (Instruções/Segundo)
a.	100.000	150.000	64	2 bilhões
b.	200.000	200.000	128	3 bilhões

**6.16.1** [10] <6.8, 6.10> Ache a taxa de E/S sustentada máxima para leituras e escritas aleatórias. Ignore os conflitos de disco e suponha que a controladora RAID não seja o gargalo. Siga a mesma técnica esboçada na Seção 6.10, fazendo suposições semelhantes onde for necessário.

**6.16.2** [10] <6.8, 6.10> Suponha que estejamos configurando um servidor Sun Fire x4150 conforme descrito na Seção 6.10. Determine se uma configuração de oito discos apresenta um gargalo de E/S. Repita para as configurações de 16, 4 e 2 discos.

**6.16.3** [10] <6.8, 6.10> Determine se o barramento PCI, DIMM ou o Front Side Bus apresenta um gargalo de E/S. Use os mesmos parâmetros e suposições usados na Seção 6.10.

**6.16.4** [5] <6.8, 6.10> Explique por que os sistemas reais utilizam benchmarks ou aplicações reais para avaliar o desempenho real.

### Exercício 6.17

Determinar o desempenho de um único servidor com dados relativamente completos é uma tarefa fácil. Porém, ao comparar servidores de diferentes vendedores oferecendo dados diferentes, escolher entre as alternativas pode ser difícil. Explore o processo de encontrar e avaliar servidores respondendo as perguntas sobre a aplicação a seguir.

Servidor de banco de dados

**6.17.1** [15] <6.8, 6.10> Para a aplicação listada, identifique as características de runtime para o sistema operacional. Escolha características que darão suporte à avaliação semelhante à que foi realizada para o Exercício 6.16.

**6.17.2** [15] <6.8, 6.10> Com relação à aplicação listada anteriormente, encontre um servidor disponível no mercado que você acredita que seria apropriado para executar a aplicação. Antes de avaliar o servidor, identifique motivos pelos quais ele foi selecionado.

**6.17.3** [20] <6.8, 6.10> Usando métricas semelhantes às que foram usadas no Capítulo 6 e no Exercício 6.16, avalie o servidor que você identificou no Exercício 6.17.2 em comparação com o servidor Sun Fire x4150 avaliado no Exercício 6.16. Qual você escolheria? Os resultados da sua análise o surpreenderam? Especificamente, você escolheria de outra forma diferente?

**6.17.4** [15] <6.8, 6.10> Identifique um conjunto de benchmark padrão que seria útil para comparar o servidor que você identificou no Exercício 6.17.2 com o Sun Fire x4150.

### Exercício 6.18

As medições e as estatísticas fornecidas pelos vendedores de armazenamento devem ser cuidadosamente interpretadas para se obter previsões significativas sobre o comportamento do sistema. A tabela a seguir oferece dados para diversas unidades de disco.

	Número de unidades	Horas/Unidade	Horas/Falha
a.	1.000	10.512	1.200.000
b.	1.250	8.760	1.200.000

**6.18.1** [10] <6.12> Calcule a taxa de falha anual (AFR) para os discos na tabela.

**6.18.2** [10] <6.12> Suponha que a taxa de falha anual varie pelo tempo de vida dos discos na tabela anterior. Especificamente, suponha que a AFR seja três vezes esse valor no primeiro mês de operação e o dobro a cada ano começando no quinto ano. Quantos discos seriam substituídos após sete anos de operação? E depois de dez anos?

**6.18.3** [10] <6.12> Suponha que os discos com taxas de falha inferiores sejam mais dispendiosos. Especificamente, os discos estão disponíveis a um custo mais alto, que começará a dobrar sua taxa de falha no ano 8, ao invés do ano 5. Quanto mais você pagaria pelos discos se a sua intenção for mantê-los por 7 anos? E por 10 anos?

### Exercício 6.19

Para os discos na tabela do Exercício 6.18, considere que o seu vendedor ofereça uma configuração RAID 0 que aumentará a vazão do sistema de armazenamento em 70% e uma configuração RAID 1 que reduzirá a AFR dos pares de discos por 2. Suponha que o custo de cada solução é 1,6 vezes o custo da solução original.

**6.19.1** [5] <6.9, 6.12> Dados apenas os parâmetros do problema original, você recomendaria fazer o upgrade para RAID 0 ou RAID 1, supondo que os parâmetros individuais do disco permaneçam iguais aos da tabela anterior?

**6.19.2** [5] <6.9, 6.12> Dado que sua empresa opera um mecanismo de busca global com uma grande farm de disco, o upgrading para RAID 0 ou RAID 1 faz sentido econômico, visto que seu modelo de receita é baseado no número de anúncios atendidos?

**6.19.3** [5] <6.9, 6.12> Repita o Exercício 6.19.2 para uma grande farm de discos operada por uma empresa de backup *on-line*. O upgrading para RAID 0 ou RAID 1 faz sentido econômico, visto que seu modelo de receita é baseado na disponibilidade do seu servidor?

## Exercício 6.20

A avaliação e a manutenção diárias dos sistemas operando no computador envolvem muitos dos conceitos discutidos no Capítulo 6. Explore os detalhes da avaliação dos sistemas explorando as perguntas a seguir.

**6.20.1** [20] <6.10, 6.12> Configure o Sun Fire x4150 para fornecer 10 terabytes de armazenamento para um array de processadores de 1000 processadores, rodando simulações de bioinformática. Sua configuração deverá minimizar o consumo de potência enquanto enfatiza questões de vazão e disponibilidade para o array de discos. Certifique-se de considerar as propriedades de grandes simulações ao realizar sua configuração.

**6.20.2** [20] <6.10, 6.12> Recomende um sistema de backup e arquivamento de dados para o array de discos do Exercício 6.20.1. Compare as capacidades de disco, fita e backup *on-line*. Use a internet e recursos de biblioteca para identificar servidores em potencial. Avalie o custo e a adequação para a aplicação usando parâmetros descritos no Capítulo 6. Selecione parâmetros de comparação usando propriedades da aplicação e também os requisitos especificados.

**6.20.3** [15] <6.10, 6.12> Vendedores concorrentes para os sistemas que você identificou no Exercício 6.20.2 se ofereceram para permitir que você avalie seus sistemas no local. Identifique os benchmarks que você usará para determinar qual sistema é melhor à sua aplicação. Determine quanto tempo será necessário para colher dados suficientes e tomar a sua decisão.

§6.2: 2 e 3 são verdadeiros.

§6.3: 3 e 4 são verdadeiros.

§6.4: Todos são verdadeiros (considerando que 40MB/s seja compatível com 100MB/s).

§6.5: 1 é verdadeiro.

§6.6: 1 e 2.

§6.7: 1 e 2. 3 é falso, pois a maioria dos benchmarks TPC inclui custo.

§6.9: Todos são verdadeiros.

## Respostas das Seções “Verifique você mesmo”