

## Entrada e Saída

- Tipos e Características de Dispositivos
- Arquitetura do Sistema de E/S
- Discos
  - \* Mecanismo, componentes de gravação e de posicionamento
  - \* Controlador
  - \* RAID
- Redes, barramentos, vazão e latência
- dispositivos, interfaces com CPU e com Sist Operacional
- Desempenho e projeto

## Características dos Dispositivos

- Comportamento
  - \* entrada (lê uma vez só)
  - \* saída (escreve uma vez só)
  - \* armazenagem (lê muitas vezes, também escreve)
- contra-parte
  - \* humano
  - \* computador
- taxa de transferência
  - \* taxa de pico
  - \* taxa sustentada (sustentável em condições “normais”)

## Classes de Periféricos

- Lentos e Preguiçosos:
  - \* teclado – 10 caracteres por segundo
  - \* mouse – 30 caracteres por segundo
- Rápidos e Gulosos:
  - \* disco rígido – 512 bytes em 0.2ms ( $\approx 2$  Mbytes/s)
  - \* interface de rede – 1024 bytes em 0.1ms ( $\approx 10$  Mbytes/s)
  - \* controlador de vídeo – 30 Kbytes em 1ms ( $\approx 30$  Mbytes/s)
- Tratamento diferente para as duas classes
  - \* periféricos lentos podem esperar
  - \* periféricos rápidos devem ser prontamente atendidos
  - \* tratamento de grandes volumes é mais complexo que o de caracteres individuais

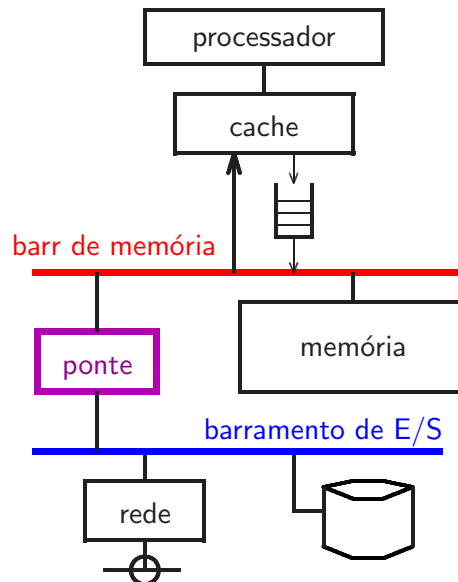
## Arquitetura do Sistema de E/S

### Hierarquia de vias:

largura de banda é menor  
a medida em que desce  
na hierarquia  
barramentos distintos  
em cada nível

### Processamento de E/S:

controlado por programa  
ADM  
processadores de E/S



## Vazão e Latência

**Vazão:** taxa de transferência [bytes/segundo]  
depende de:

- largura da via (largura do barramento: 8, 32 ou 256 bits)
- taxa de sinalização (velocidade do relógio)
- tipo de sinalização (síncrona ou assíncrona)

**Latência:** lapso entre comando e resposta [segundo]  
depende de:

- tipo de sinalização (síncrona ou assíncrona)
- tipo dos dispositivos (memória, disco, mouse)
- organização (mapeamento de endereços, segmentação do caminho)

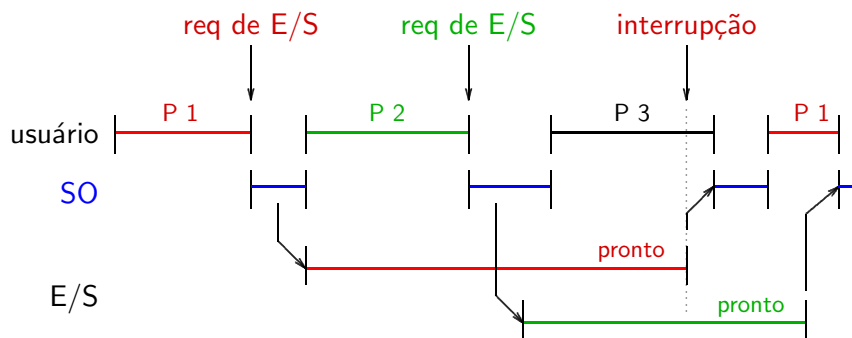
## Entrada e Saída

- Tipos e Características de Dispositivos
- Arquitetura do Sistema de E/S
- Discos
  - \* Mecanismo, componentes de gravação e de posicionamento
  - \* Controlador
  - \* RAID
- Redes, barramentos, vazão e latência
- dispositivos, interfaces com CPU e com Sist Operacional
- Desempenho e projeto

*An introduction to Disk Drive Modeling*, Ruemmler & Wilkes, IEEE Computer  
27(3):17-28, Mar 1994

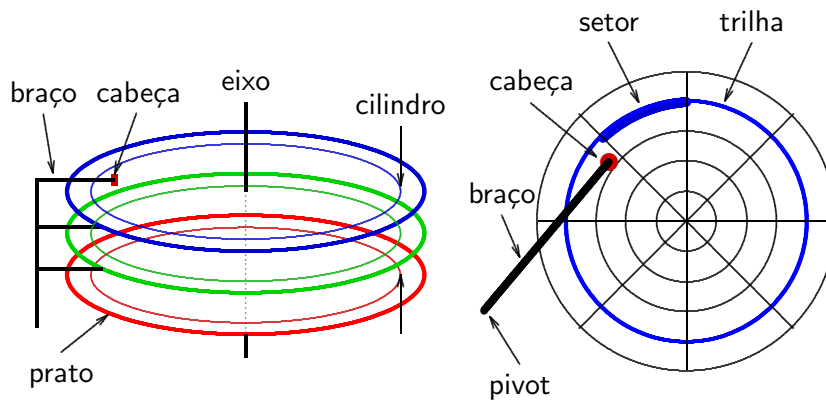
## E/S e Computação

E/S concorre com computação de maneiras complexas



$$\mathcal{T}_{\text{tarefa}} = \mathcal{T}_{\text{cpu}} + \mathcal{T}_{\text{E/S}} - \mathcal{T}_{\text{concorr}}$$

## Discos Magnéticos



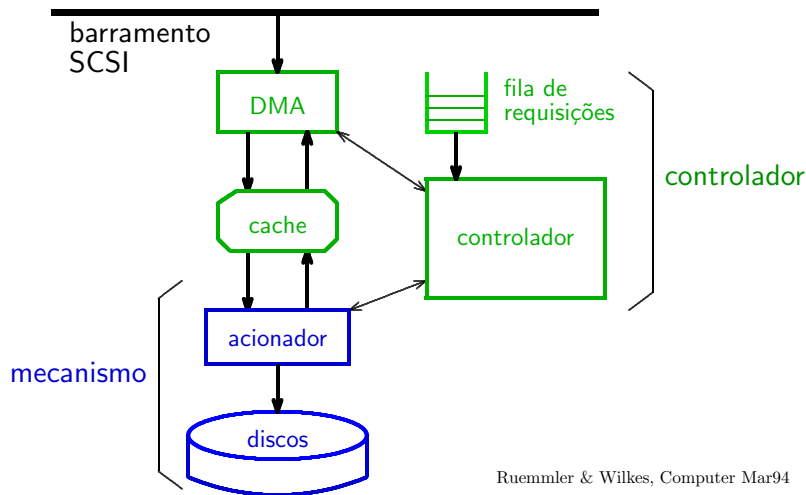
Ruemmler & Wilkes, Computer Mar94

## Discos Magnéticos – Parâmetros Típicos

| Característica            | mín   | máx    |
|---------------------------|-------|--------|
| diâmetro [polegadas]      | 1,0   | * 3,5  |
| capacidade formatada [GB] | 4     | >200   |
| discos/pratos             | 1     | 20     |
| trilhas por superfície    | 6.000 | 25.000 |
| setores por trilha        | 100   | 600    |
| bytes por setor           | 512   | 4.096  |
| velocidade [rpm]          | 5.400 | 15.000 |
| cache [MB]                | 0,5   | ≥8     |
| taxa transferência [MB/s] | 2,5-5 | 27-40  |

\* tamanho típico (+popular) em 2004

## Discos – organização



## Mecanismo – componentes de gravação

- Diâmetro: 1.0, 1.3, 2.5, 3.5 ,8 polegadas
- densidade linear de gravação: 100 Kbpi [bits/inch]
- densidade de trilhas: 20 Ktpi [tracks/inch]
- efeito combinado: densidade por área cresce 60% aa
- velocidade 3.600, 7.200, 10.000, 15.000 cresce 12% aa
- uma cabeça ativa por vez, taxa de leitura  $\geq 100$  Mbps
- conteúdo de um setor
  - ★ número do setor;
  - ★ espaço;
  - ★ informação do setor com código de detec&corr de erros
  - ★ espaço;
  - ★ ...

## Mecanismo – componentes de posicionamento

- Densidade é tão alta que noção de cilindro é quase irrelevante
- busca consiste de
  - ★ aceleração até atingir  $1/2 V_{\text{máx}}$
  - ★  $V_{\text{máx}}$  em distâncias longas
  - ★ desaceleração até trilha desejada
  - ★ estabilização da cabeça sobre a trilha (1-3 ms)
  - ★ seeks longos:  $\mathcal{T} \propto \text{distância} \text{ (@} V_{\text{máx}})$
  - ★ seeks médios:  $\mathcal{T} \propto \sqrt{\text{distância}}$
- recalibrar posição a cada 15-30 min, durante 500-800 ms
- acompanhamento de trilhas
  - ★ troca de cabeças  $\rightarrow$  reposicionar braço ( $\approx 0.5-1.5$  ms)
  - ★ troca de trilha  $\rightarrow$  estabilizar braço ( $\approx 1-3$  ms)
  - ★ controlador tenta leitura otimista assim que chega na trilha  
pode fazer escrita otimista? ( $\approx 0.75$  ms até estabilizar para escrever)

## Mecanismo – leiaute dos dados

- Disco visto pelo SO como vetor linear de blocos (256-1024 bytes)
- controlador mapeia vetor nos setores físicos  $1D \rightsquigarrow 3D$   
 $\text{blocos}[i] \rightsquigarrow \text{disco}[\text{superfície, trilha, setor}]$
- #bits cresce  $\approx$  linearmente com comprimento da trilha  
**zoneamento: número de setores depende do raio**  
 $\exists$  3-50 zonas com mesmo número de setores /zona
- deslocamento de setores nas trilhas:  
**setor0** de cada trilha deslocado para esconder  
tempo de reposicionamento *track skewing*
- trilhas/setores sobressalentes: referências a setores danificados  
são re-mapeadas para setores/trilhas de reserva
  - ★ na formatação – pula endereço da trilha com defeito *slip sparing*
  - ★ em uso – re-mapeia endereço do setor/trilha para sobressalentes

## Controlador

- Funções do controlador SCSI
  - ★ mediar acessos ao mecanismo
  - ★ executar sistema de acompanhamento de trilhas
  - ★ transferir dados entre disco e cliente
  - ★ gerenciar buffers/cache
- operação do controlador custa 0.3-1 ms (caindo lentamente)  
eletrônica segue Lei de Moore mas funcionalidade cada vez mais complexa
- interface com barramento
  - ★ transferências em modo síncrono, na veloc máxima do barramento
  - ★ pode operar com *split transactions* (latências enormes)
  - ★  $\exists$  buffer entre mecanismo e barramento por causa das diferenças de velocidade
- buffer usado como cache

## Controlador – cache (leitura/escrita)

- Políticas da cache: *read-ahead*  $\approx$  busca antecipada
  - \* *on-arrival read-ahead*: assim que chegar na trilha, lê trilha toda
  - \* *read-ahead* agressivo: atravessa trilhas e/ou cilindros
  - \* *read-ahead 'zen'*: pára no final de trilha/cilindro
  - \* cache associativa:  
particionar cache para  $\neq$ s seqüências entrelaçadas
- Cache pode corromper sistema de arquivos se faltar energia
  - \* controlador avisa que completou operação  
após escrever **na cache**
  - \* se cache tem bateria, problema desaparece (?)
- Cache com fila de comandos: controlador pode otimizar operações  
porque conhece geometria do disco

## Operação de Discos

- Comandos
  - \* Latência/atraso no controlador + tempo na fila (OS)
  - \* 0,5ms se não encontra na cache, 0,1ms se encontra na cache
- Seek (movimentação do braço entre trilhas/cilindros)
  - \* move a cabeça até a trilha desejada
  - \* tempo depende da posição inicial da cabeça
  - \* valores típicos médios entre 5-12ms
- Latência rotacional
  - \* espera até que setor desejado passe sob a cabeça ( $\approx 1/2$  volta)
  - \* na média,  $0,5/\text{rpm} \rightarrow 0,5 / (7200\text{rpm} / 60\text{spm}) = 4,2\text{ms}$
- Transferência de dados
  - \* taxa de transferência entre 2 e 40 MByte/s

## Desempenho

- Tempo médio de acesso
  - = tempo médio de movimentação do braço (seek)
  - + latência rotacional média
  - + tempo de transferência
  - + tempo do controlador
- Exemplo: 7200 rpm, 10MByte/s
  - tempo médio de movimentação do braço: 10ms
  - tempo do controlador: 0,5ms
  - tempo para ler bloco de 4Kbytes (uma página)
$$10\text{ms} + 0,5 / (7200\text{rpm} / 60\text{spm}) + 4\text{KB} / 10\text{MB/s} + 0,5\text{ms}$$

$$10\text{ms} + 4,2\text{ms} + 0,4\text{ms} + 0,5\text{ms} = 14,65\text{ms}$$

## Desempenho (cont)

|  | distância |        |
|--|-----------|--------|
|  | # trilhas | fração |
|  | 0         | 24%    |
|  | 15        | 23%    |
| <b>Localidade:</b>                             | 30        | 8%     |
| discos exibem localidade                       | 45        | 4%     |
| → em “acessos locais” seek cai em 1/3          | 60        | 3%     |
|  | 75        | 3%     |
| <b>Cache:</b>                                  | 90        | 1%     |
| buffer em memória ( <i>Unix buffer cache</i> ) | 105       | 3%     |
| e na unidade de disco                          | 120       | 3%     |
| → latência cai para hit+transferência          | 135       | 2%     |
|  | 150-195   | 11%    |

*unix time-sharing*

Hennessy&Patterson QA Fig-7.52

## Matrizes de Discos

### Conjunto de discos individuais:

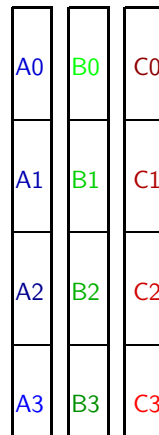
cada disco com seu braço/cabeça

### Distribuição dos dados:

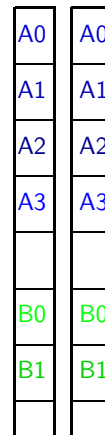
endereçamento independente

listras de blocos pequenos

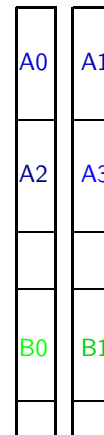
listras de blocos grandes



independente



granularidade fina



granularidade grossa

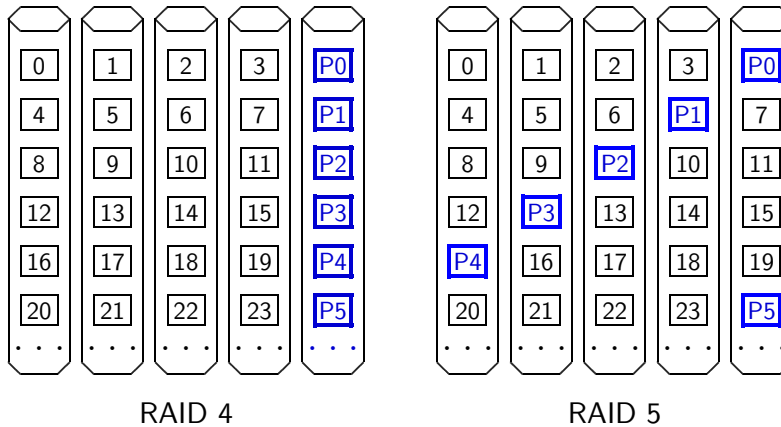
## Matrizes de Discos

- Endereçamento independente
  - ★ software/usuário distribui os dados
  - ★ balanceamento de carga entre discos pode ser problemático
- Listras de blocos pequenos (*fine-grain striping*)
  - ★ um bit, um byte, ou um setor
  - ★  $\#discos * |bloco|$  define menor quantidade de dados acessível
  - ★ balanceamento de carga perfeito; só uma requisição atendida por vez
  - ★ taxa efetiva de transferência  $\approx N$  vezes melhor que um disco só
  - ★ tempo de acesso pode aumentar, a não ser que discos sejam sincronizados
- Listras de blocos grandes (*coarse-grain striping*)
  - ★ paralelismo na transferência de grandes volumes de dados
  - ★ concorrência para transferências pequenas
  - ★ balanceamento de carga pela aleatoriedade
- Granularidade escolhida em função da aplicação e tipo de carga

## Mecanismos de Redundância

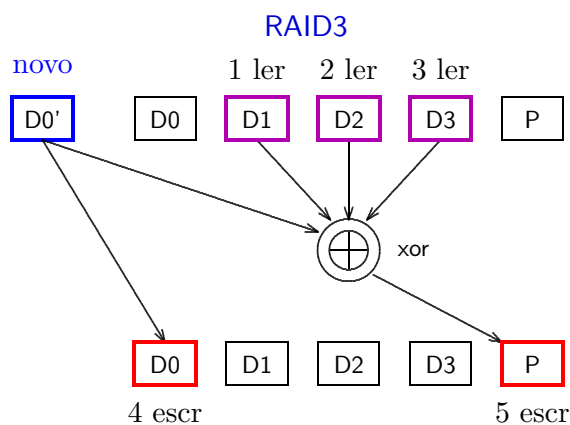
- Falhas em discos são parcela grande de falhas de hardware
  - \* striping aumenta o número de arquivos perdidos por falha
- Replicação dos dados
  - espelhamento dos discos
    - permite leituras em paralelo
    - escritas devem ser sincronizadas
- Proteção com paridade
  - \* usar disco para manter a paridade

## RAID 4/5 – Blocos de Paridade



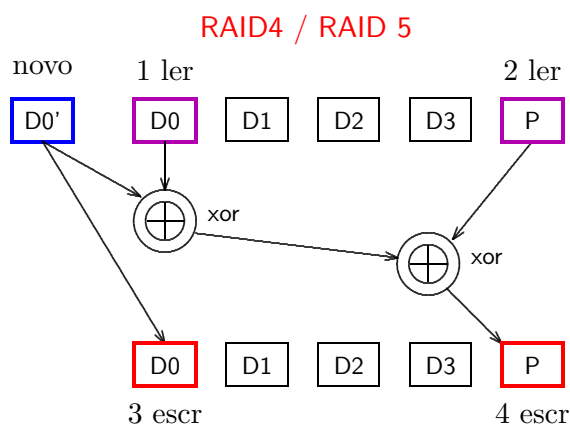
## RAID – Atualização “Pequena”

Qual é o número de operações de leitura/escrita nos discos individuais para efetuar escrita de poucos dados (= atualização pequena)?



## RAID – Atualização “Pequena”

Qual é o número de operações de leitura/escrita nos discos individuais para efetuar escrita de poucos dados (= atualização pequena)?





## resumo – Discos

- Tempo médio de acesso
  - = tempo médio de movimentação do braço (*seek*)
  - + latência rotacional média
  - + tempo de transferência
  - + tempo do controlador
- Cache no controlador para tirar proveito de localidade
  - ★ falta de energia durante escrita de metadados corrompe sist de arquivos
  - ★ mesmos problemas que fila de escrita (riscos RAW a WAW)
- RAID – usar discos baratos para
  - aumentar desempenho – acessos em paralelo (*striping*)
  - e
  - melhorar confiabilidade – paridade