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")

Tipos de Dispositivos

dispositivo	comportamento	parceiro	taxa [b/s]
teclado	entrada	humano	10
mouse	entrada	humano	20
scanner	entrada	humano	400 K
monitor gráfico	saída	humano	60 M
rede local	E/S	computador	0.5-10 M
fita	armazenagem	computador	2 M
disco	armazenagem	computador	2-10 M

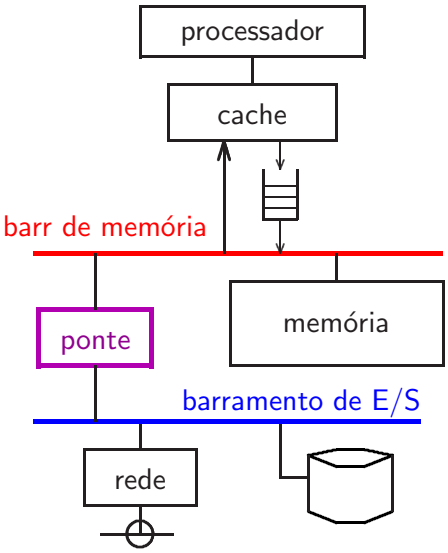
Classes de Periféricos

- Lentos e Preguiçosos:
  - teclado – 10 caracteres por segundo
  - mouse – 30 caracteres por segundo
- Rápidos e Gulosos:
  - disco rígido – 512 bytes em 0.2ms (≈2 Mbytes/s)
  - interface de rede – 1024 bytes em 0.1ms (≈10 Mbytes/s)
  - controlador de vídeo – 30 Kbytes em 1ms (≈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)

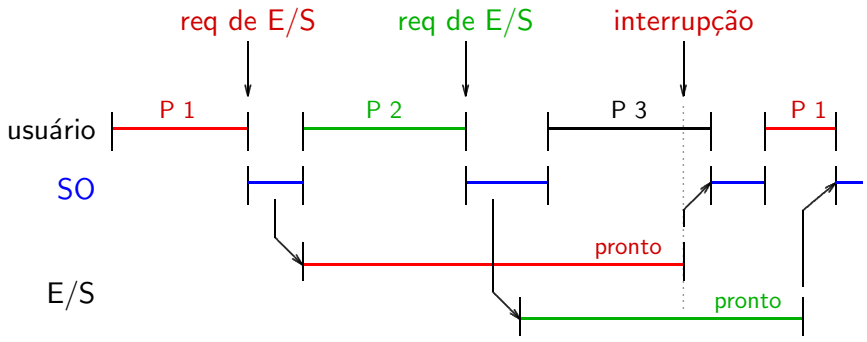
Entrada e Saída

- 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

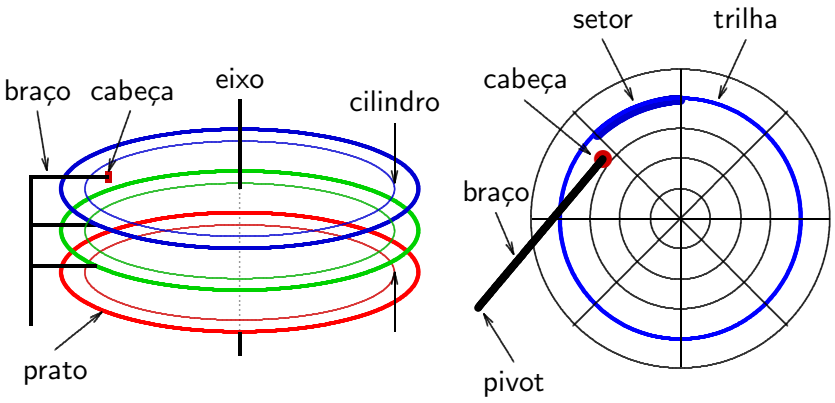
E/S e Computação

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



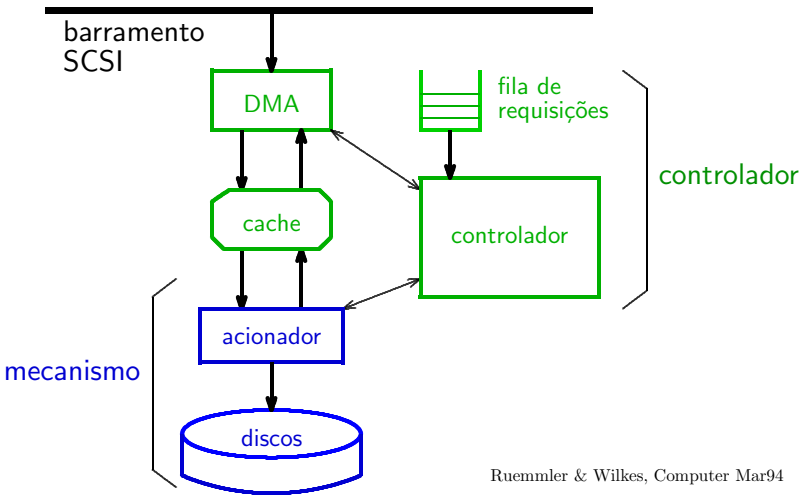
Ruemmler & Wilkes, Computer Mar94

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: $T \propto \text{distância}\ \ (@V_{\text{máx}})$
  - ⋆ seeks médios: $T \propto \sqrt{\text{distância}}$
- recalibrar posição a cada 15-30 min, durante 500-800 ms
- acompanhamento de trilhas
  - ⋆ troca de cabeças ⟶ reposicionar braço (≈ 0.5-1.5 ms)
  - ⋆ troca de trilha ⟶ estabilizar braço (≈ 1-3 ms)
  - ⋆ controlador tenta leitura otimista assim que chega na trilha
    pode fazer escrita otimista? (≈ 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 ⇝ 3D
  blocos[i] ⇝ disco[superfície, trilha, setor]
- #bits cresce ≈ linearmente com comprimento da trilha
  zoneamento: número de setores depende do raio
  ∃ 3-50 zonas com mesmo número de setores /zona
- deslocamento de setores nas trilhas:
  setor**0** 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)
  - ⋆ ∃ buffer entre mecanismo e barramento por causa das diferenças de velocidade
- buffer usado como cache

Controlador – cache (leitura/escrita)

- Políticas da cache: *read-ahead* ≈ 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 ≠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 (≈1/2 volta)
  - na média, 0.5/rpm → 0,5 / (7200rpm / 60spm) = 4,2ms
- 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)

  10ms + 0,5/(7200rpm/60spm) + 4KB/10MB/s + 0,5ms

  10ms + 4,2ms + 0,4ms + 0,5ms = 14,65ms

Desempenho (cont)

Localidade:
discos exibem localidade
⟶ em "acessos locais" seek cai em 1/3

Cache:
buffer em memória (*Unix buffer cache*)
e na unidade de disco
⟶ latência cai para hit+transferência

distância # trilhas	fração
0	24%
15	23%
30	8%
45	4%
60	3%
75	3%
90	1%
105	3%
120	3%
135	2%
150-195	11%

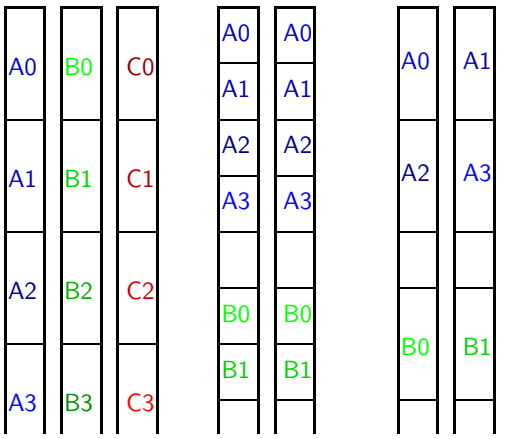
unix time-sharing

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



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 ≈ 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

Redundant Arrays of Inexpensive Disks – RAIDs

- Conjuntos de discos pequenos e baratos resultam em alto desempenho e alta confiabilidade
  D = número de discos de dados no conjunto
  V = número de discos de verificação no conjunto
- Nível 0: só listras, sem redundância *(striping)*
- Nível 1: discos espelhados (D=1, V=1)
  - desperdício é elevado
- Nível 2: código de detecção de erros (D=10, V=4)
  - mesmo tipo de código de detecção de erros usado com DRAMs
  - todos os bits do conjunto são lidos
  - agrega bits atualizados com bits que permanecem; recomputa a paridade
  - re-escreve todo o conjunto, incluindo a verificação

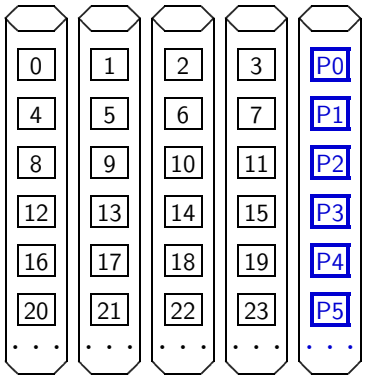
Redundant Arrays of Inexpensive Disks – RAIDs

- Nível 3: paridade com bits entrelaçados (D=4, V=1)
  - disco com falha é identificado facilmente pelo controlador
  - não é necessário código especial para descobrir disco em falha
- Nível 4: paridade com blocos entrelaçados
  - usado com blocos grandes
  - similar ao RAID 3, mas pode efetuar mais de um acesso com poucos dados a cada vez
  - escrita deve atualizar disco com dados e disco com paridade

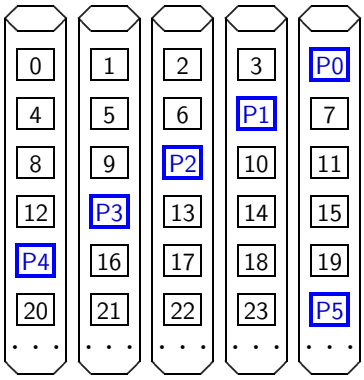
Redundant Arrays of Inexpensive Disks – RAIDs

- Nível 5: paridade distribuída por blocos entrelaçados
  - ∗ paridade é distribuída pelos discos no conjunto
  - ∗ atualizações distintas de paridade vão para discos distintos
- Nível 6: matriz bi-dimensional
  - ∗ matriz de dados é bi-dimensional, com paridade nas linhas e nas colunas
  - ∗ permite recuperação de duas falhas

RAID 4/5 – Blocos de Paridade



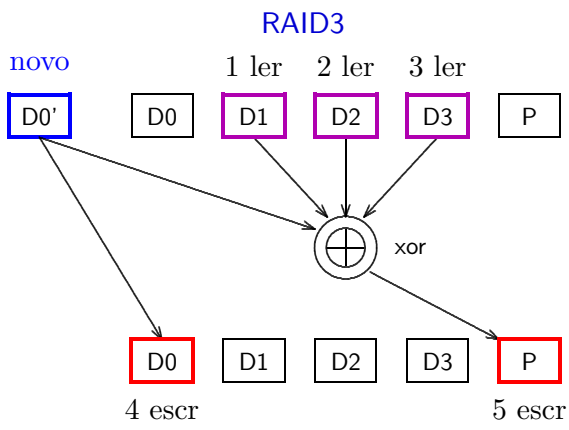
RAID 4



RAID 5

RAID – Atualização "Pequena"

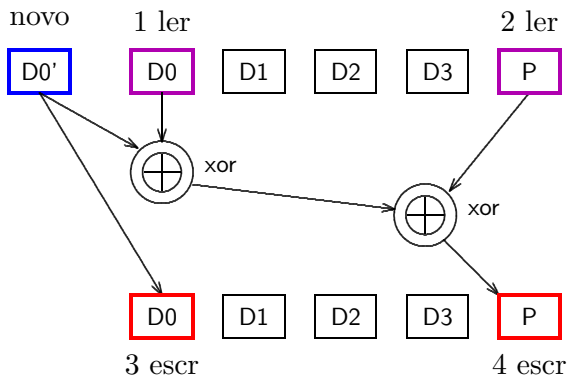
Qual é o número de operações de leitura/escrita nos discos individuais para efetuar escrita de poucos dados (= atualização pequena)?



RAID – Atualização "Pequena"

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

RAID4 / RAID 5



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