

Dispositivos de Armazenamento

Paulo Sérgio Almeida

Grupo de Sistemas Distribuídos
Departamento de Informática
Universidade do Minho



Discos rígidos

- Discos rígidos são o dispositivo mais comum
 - rodam entre 60 a 200 vezes por segundo
 - são constituídos por vários **platters**, cada um com
 - várias **pistas**, cada uma com **sectores**
 - o **braço** contém **cabeças**, que se deslocam juntas
 - o conjunto de pistas numa posição do braço é um **cilindro**
- São relevantes num disco:
 - **transfer rate**, mede a velocidade de transferência entre o disco e o computador
 - **random-access time** é o tempo que demora a posicionar a cabeça para ler um sector; é constituído por
 - **seek time**, para mover o braço e
 - **rotational latency** para o sector ficar por baixo da cabeça



Discos rígidos

- Disco são ligados ao computador por um **I/O bus**
 - vários tipos: EIDE, ATA, SATA, USB, Fibre Channel, SCSI
 - **host controller** no computador usa o bus para falar com o
 - **disk controller** no dispositivo, que contém uma cache
- Existem discos amovíveis; e.g. **floppy disks**, em desuso
- Discos podem sofrer acidentes; e.g. **head crash** em que a cabeça entra em contacto com a superfície



Fita magnética

- Usada inicialmente para armazenamento secundário
- Guarda grande quantidade de dados
- Acesso é lento: necessário bobinar até os dados estarem por baixo da cabeça
- Transferência de dados rápida, depois de bobinada
- Acesso aleatório muito lento
- Actualmente usada para backup



Estrutura dos discos

- Discos acedidos como array unidimensional de **blocos lógicos**
- Bloco é a unidade básica de transferência; e.g. 512 bytes
- Array unidimensional é mapeado em sectores sequencialmente:
 - o sector 0 é o primeiro da primeira pista do cilindro exterior
 - o mapeamento prossegue ao longo da pista
 - depois nas outras pistas do cilindro
 - e depois ao longo dos cilindros em direcção ao centro



Ligação dos discos

- **Host-attached storage** é acessada via portos de I/O que falam com um I/O bus; e.g. SATA, SCSI, FC
- SCSI:
 - SCSI é um bus que serve até 16 dispositivos num cabo
 - o **SCSI initiator** faz um pedido a um **SCSI target**
 - cada target pode ter até 8 unidades lógicas; e.g. componentes de um array RAID
- FC (Fibre Channel):
 - arquitectura série de alta velocidade
- Vários tipos de dispositivos podem ser usados: discos rígidos, arrays RAID, CD, DVD, fita



Network-attached storage

- **Network-attached storage (NAS)** é um dispositivo acessado via rede em vez de um bus
- Permite conjunto de máquinas numa LAN partilhar disco
- São usados protocolos como NFS ou CIFS implementados via RPC
- Dispositivo implementa interface RPC
- Tipicamente menor performance do que armazenamento local
- Usam rede IP normal; e.g LAN
- iSCSI disponibiliza protocolo SCSI sobre IP



Storage Area Network

- Permite vários computadores ligarem-se a vários dispositivos
- Está a tornar-se comum em ambientes de grande dimensão
- Usa rede específica, separada da rede de dados normal
- Usa protocolos específicos de armazenamento
- Usa normalmente Fibre Channel



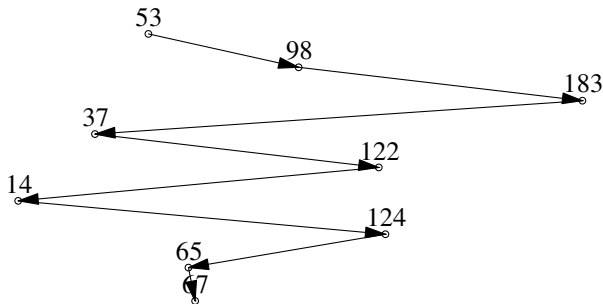
Escalonamento de disco

- Com multiprogramação é normal haver fila de pedidos de transferência de disco
- Gestão dos pedidos de transferência tenta minimizar tempo de acesso e aumentar largura de banda
- Tempo de acesso composto por:
 - seek time
 - rotational latency
- Tempo de acesso dominado por movimento da cabeça
- Diferentes algoritmos de escalonamento de pedidos de transferência, com diferentes resultados



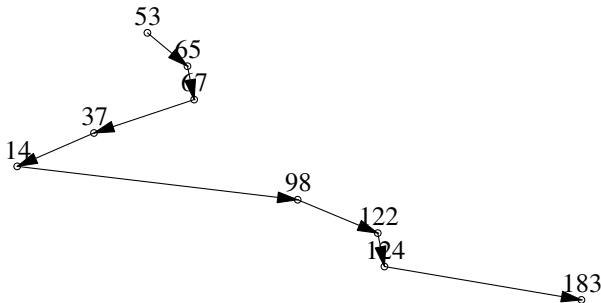
FCFS – first-come first-served

- Serve pedidos por ordem
- Justo mas pouco eficiente
- Suponhamos cilindros entre 0 e 199; cabeça no cilindro 53 e
- fila de pedidos a cilindros: 98, 183, 37, 122, 14, 124, 65, 67
- Movimento total da cabeça: 640 cilindros



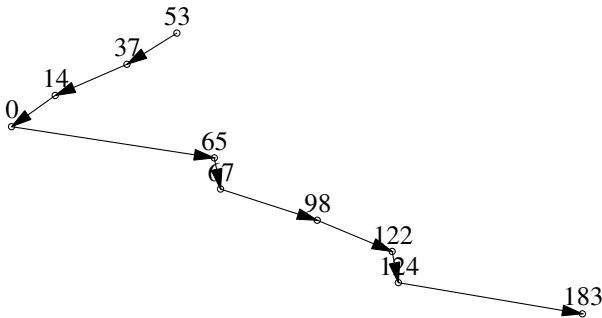
SSTF – Shortest-seek-time-first

- Escolhe pedido com o menor seek-time
- É escolhido o pedido mais perto da posição actual da cabeça
- Performance consideravelmente melhor; mas não óptimo
- Análogo a shortest-job-first; pode causar **starvation**
- Suponhamos mesma fila: 98, 183, 37, 122, 14, 124, 65, 67
- Movimento total da cabeça: 236 cilindros



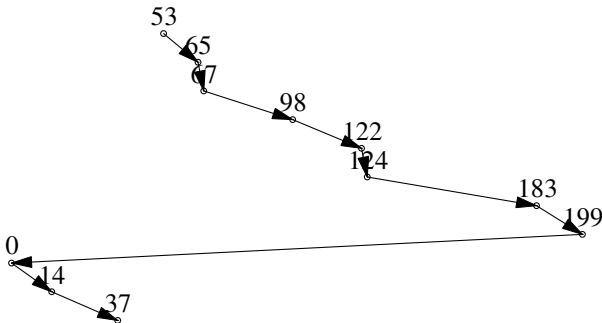
SCAN – elevator algorithm

- Cabeça desloca-se até ao fim, servindo pedidos pelo caminho
- Ao chegar ao fim, muda de direcção
- Análogo ao que faz um elevador
- Suponhamos mesma fila: 98, 183, 37, 122, 14, 124, 65, 67
- Movimento total da cabeça: 236 cilindros



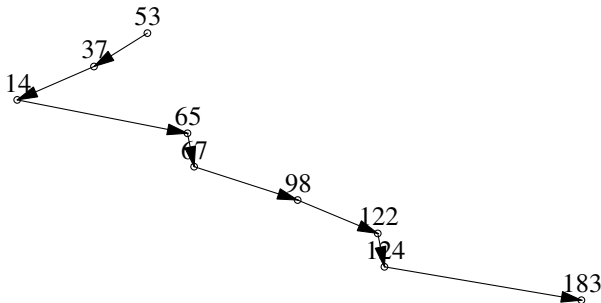
C-SCAN – Circular SCAN

- Variante de SCAN com tempo de espera mais uniforme
- Ao chegar a uma ponta, volta imediatamente ao início sem servir pedidos no caminho de volta
- Equivalente a considerar uma lista circular de cilindros
- Suponhamos mesma fila: 98, 183, 37, 122, 14, 124, 65, 67
- Movimento total da cabeça: 382 cilindros



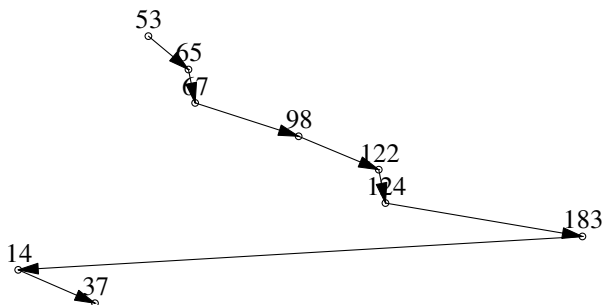
LOOK e C-LOOK

- Variantes de SCAN e C-SCAN
- Em vez de ir até ao fim, a cabeça só continua se ainda houver pedidos nessa direcção
- LOOK; mesma fila: 98, 183, 37, 122, 14, 124, 65, 67
- Movimento total da cabeça: 208 cilindros



C-LOOK

- C-LOOK, mesma fila: 98, 183, 37, 122, 14, 124, 65, 67
- Movimento total da cabeça: 322 cilindros



Escolha de um algoritmo

- SSTF escolha comum
- SCAN e C-SCAN melhores para grande carga; evitam starvation
- SSTF ou LOOK são escolhas razoáveis por omissão
- Performance depende do número e tipo de pedidos
- Pedidos influenciados pela alocação de ficheiros em disco
- Discos modernos fazem escalonamento interno
- Ordem de pedidos importante, para além de performance
 - prioridade à paginação sobre I/O de ficheiros
 - prioridade a escritas se cache tem poucos blocos
 - ordem de escrita de meta-dados do sistema de ficheiros
- SO pode escalonar conjuntos de pedidos; disco gere cada conjunto



Gestão de discos

- **Formatação física:** divisão do disco em sectores a que o controlador acede
- **Particionamento:** divisão do disco em partes independentes com diferentes sistemas de ficheiros
- **Formatação lógica:** criação de sistema de ficheiros, com estruturas de dados em disco
- **Raw disks:**
 - partições que não contêm sistemas de ficheiros
 - vistas como array de blocos lógicos
 - acedidas directamente por aplicações como bases de dados
- Bloco de **boot**:
 - é carregado pelo programa inicial de **bootstrap** em ROM
 - contém programa de bootstrap que carrega o SO para memória



Bad Sectors

- **Bad sectors:**
 - alguns vêm marcados de fábrica
 - formatação física deixa sectores extra de lado, invisíveis ao SO
 - quando se danificam outros, são substituídos por sectores extra
 - ECC **error-correcting code** usado para cada sector
 - substituição antecipada se ocorrerem **soft errors**, recuperáveis, antes que aconteçam **hard errors**
- Substituição de bad sectors – **sector sparing ou forwarding**
 - sector danificado é mapeado em sector extra
 - feita pelo controlador; transparente para o SO
 - pode invalidar optimizações do algoritmo de escalonamento
 - solução: são mantidos sectores extra em cada cilindro
- Substituição de bad sectors – **sector slipping**
 - em alternativa a sparing
 - move um conjunto de sectores na direcção de sector extra livre
 - re-mapeia o conjunto para a nova gama



Área de swap

- Área de swap é usada pelo sistema de memória virtual
- Dimensionamento por excesso pode ser conveniente:
 - discos baratos
 - evita terminar processos
- Duas alternativas de gestão:
 - ficheiro grande, guardado dentro de sistema de ficheiros
 - é mantida numa partição raw, à parte
- Partição própria de swap mais eficiente:
 - evita aceder a estruturas do sistema de ficheiros
 - não corre o perigo de o ficheiro estar disperso pelo disco
 - pode usar algoritmos especializados
 - desvantagem: dificuldade em aumentar área de swap
- Pode estar dividida por vários discos: aumenta largura de banda



Gestão da área de swap

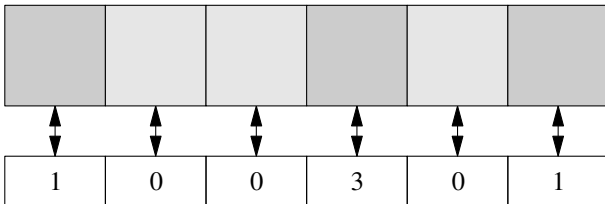
Evolução:

- Inicialmente era feito swap de processos inteiros para região contígua (daí nome)
- Com hardware de paginação, swapping e paging coexistiram
- 4.3BSD: swap alocado quando processo arranca; para código e dados
- Solaris 1 (SunOS): páginas de código são trazidas do sistema de ficheiros e descartadas quando necessário; não vão para swap
- Solaris 2: swap alocado apenas quando página é rejeitada; não quando criada



Swap map em Linux

- Linux pode ter várias áreas de swap, em ficheiro ou partição raw
- Cada área é um vector de blocos de 4KB para guardar páginas
- A cada área é associada um **swap map**:
 - vector de inteiros, 1 por bloco
 - = 0: bloco livre
 - > 0: contador de número de processos que partilham página



RAID

- RAID – redundant array of independent disks
- Ligação de vários discos trabalhando cooperativamente
- Podem melhorar performance trabalhando em paralelo
- Podem melhorar fiabilidade à custa de redundância
- Disco **hot spare** em stand-by permite reconstituir o RAID se houver falha de disco
- Implementação por software, ou hardware transparente ao SO
- Diferentes variantes de RAID



Melhoria de fiabilidade

- Introdução de redundância no armazenamento
- **Mirroring**: duplicação de cada disco
 - cada disco lógico é constituído por 2 discos físicos
 - cada escrita é feita nos 2 discos
 - se um disco falha, dados podem ser lidos do outro
- Quando um disco falha é reparado/substituído e dados copiados
- Probabilidade de perda de dados muito menor
- Minimizar correlação de falhas: e.g. usar discos de diferentes fabricantes ou com diferentes datas de fabrico



Melhoria de desempenho

- Com mirroring:
 - as escritas têm que ir para os 2 discos: mesmo desempenho
 - 2 leituras podem ser feitas em paralelo: dobro do desempenho
- **Striping** – dividir os dados ao longo de n discos
 - **bit-level**: 1 bit para cada disco: bit i para disco $(i \bmod n)$
 - outras possibilidades: **byte-level**, **sector-level**, **block-level**
 - mais comum: block-level – espalha blocos de ficheiro
- Striping permite, através de paralelismo:
 - melhorar throughput de acessos pequenos, via **load-balancing**
 - reduzir tempo de resposta de acessos grandes



Níveis de RAID

- Várias combinações são possíveis em termos de mirroring, striping, ou uso de bits de paridade
- **RAID Level 0:** striping a nível do bloco
- **RAID Level 1:** uso de mirroring
- **RAID Level 2: memory-style error-correcting code organization**
 - uso de bit-level striping com bits ECC em discos extra
 - 3 discos extra por cada 4 discos de dados
 - melhora fiabilidade com menos discos do que RAID 1
- **RAID Level 3: bit-interleaved parity organization**
 - melhora RAID 2, usando apenas 1 disco extra com bit de paridade
 - aproveita o facto de discos já conterem ECCs e detectarem erros
 - sabendo qual o disco com erro 1 bit de paridade é suficiente
 - melhora fiabilidade; usado em vez de RAID 2



Níveis de RAID

- **RAID Level 4: block-interleaved parity organization**
 - como RAID 3 mas com block-level striping
 - leituras a blocos diferentes podem seguir em paralelo
 - melhora throughput para acessos pequenos
 - escritas pequenas exigem ler e escrever dados e paridade
- **RAID Level 5: block-interleaved distributed parity**
 - parecido com RAID 4; mas com paridade espalhada pelos discos
 - com d discos, paridade do bloco b no disco $(b \bmod d)$
 - evita sobrecarregar um único disco de paridade como em RAID 4
 - é o nível de RAID com paridade mais comum
- **RAID Level 6: P + Q redundancy scheme**
 - parecido com RAID 5 mas com ECC em vez de paridade
 - com 2 bits redundantes por 4 bits de dados tolera 2 falhas



Níveis de RAID

- **RAID Level 0+1**: combinação de RAID 0 com RAID 1
 - RAID 0 para desempenho e RAID 1 para fiabilidade
 - discos são striped e o conjunto mirrored
 - dispendioso como RAID 1 (dobro dos discos)
 - mais performance do que RAID 5
- **RAID Level 1+0**: variante de RAID 0+1
 - discos são mirrored aos pares
 - os pares são striped
 - vantagem sobre 0+1: se um disco falha, o par continua disponível (em 0+1 se disco falha, a stripe inteira fica inacessível)

