UNIDADE 1 – ESTRUTURA DE ARQUIVOS E PROCESSAMENTO DE CONSULTAS
MÓDULO 1 – ARMAZENAMENTO DE DADOS EM DISCO.

01

1 - ARMAZENAMENTO DE DADOS EM DISCO

Olá, seja bem-vindo à disciplina Banco de dados II. Você já estudou as características básicas de um SGBD, como estruturar um banco de dados, como criar elementos (tabelas, atributos, relações, procedimentos armazenados, gatilhos), e como partir de um modelo conceitual até um modelo físico, passando pela normalização dos dados.

Nesta disciplina iremos focar em atividades de operação e manutenção de um servidor de banco de dados, bem como aprofundar em alguns dos elementos já estudados, principalmente aqueles voltados à manutenção e operação do SGBD.

Para tais estudos, é importante que você possa treinar os conceitos apresentados em um SGBD no seu computador. As rotinas e conceitos apresentados nesta disciplina são aplicáveis à maioria dos SGBDs. Entretanto, alguns SGBDs não possuirão determinado tipo de controle ou funcionalidade, bem como a forma pela qual determinada funcionalidade seja corretamente aplicada pode necessitar de algum ajuste do operador do SGBD.

Alguns SGBDs podem requerer que você instale *softwares* complementares, como o Apache, Java e/ou PHP. Para um maior aprendizado, você poder instalar e testar os comandos e exercícios apresentados desta matéria nos SGBDs sugeridos aqui. Dessa forma, além de conhecer as diferenças entre os diferentes SGBDs, você desenvolveria muito mais o seu aprendizado.

Os bancos de dados são armazenados fisicamente como arquivos de registros, que geralmente ficam em discos magnéticos (Disco Rígido). Trataremos aqui sobre a organização dos bancos de dados em locais de armazenamento e as técnicas para acessá-los de modo eficiente. Falaremos também sobre os índices e como essas estruturas realizam pesquisas eficientes para localização de registros.

aqui
Sugestões de SGBDs que possuem versões de avaliação ou são gratuitos que você poderia instalar no seu computador para executar as atividades e conhecimentos presentes nesta disciplina:

- Versão de avaliação:
  - Microsoft SQL Server Express. Disponível aqui: https://goo.gl/uuZnfS.
- Gratuitos:
  - MySQL. Disponível aqui: http://goo.gl/M6Z1nX.
  - PostgreSQL. Disponível aqui: http://goo.gl/2sRVnJ.

02

A coleção de dados que compõem um banco de dados computadorizado deve ser armazenada fisicamente em algum meio de armazenamento no computador. O software de SGBD pode então recuperar, atualizar e processar esses dados conforme a necessidade.

A mídia de armazenamento de computador forma uma hierarquia de armazenamento que inclui duas categorias principais:

a) **Armazenamento primário**

Essa categoria inclui a mídia de armazenamento que pode ser operada diretamente pela unidade central de processamento (CPU) do computador, como a memória principal do computador e memórias cache menores, porém mais rápidas. O armazenamento primário normalmente oferece acesso rápido aos dados, mas tem capacidade de armazenamento limitada. Embora as capacidades da memória principal estejam crescendo rapidamente nos últimos anos, elas ainda são mais caras e têm menos capacidade de armazenamento do que os dispositivos de armazenamento secundários e terciários.

b) **Armazenamento secundário e terciário**

Essa categoria inclui discos magnéticos, discos ópticos (CD-ROMs, DVDs e outros meios de armazenamento semelhantes) e fitas (para backup). As unidades de disco rígido são classificadas como armazenamento secundário, enquanto a mídia removível, como discos ópticos, pen drives, memórias flash e as fitas, é considerada armazenamento terciário. Esses dispositivos costumam ter uma capacidade maior e menor custo, entretanto, oferecem acesso mais lento aos dados do que os dispositivos de armazenamento primários.

Os dados no armazenamento secundário ou terciário não podem ser processados diretamente pela CPU; primeiro, eles precisam ser copiados para o armazenamento primário e, depois, processados pela CPU.

03

1.1 - Hierarquias de memória e dispositivos de armazenamento

Em um sistema de computador moderno, os dados residem e são transportados por uma hierarquia de meios de armazenamento. A memória de velocidade mais alta é a mais cara e, portanto, está disponível com a menor capacidade. A memória de velocidade mais lenta é o armazenamento em fita off-line, que basicamente está disponível em capacidade de armazenamento indefinida.

No nível de **armazenamento primário**, a hierarquia de memória inclui, no extremo mais caro, a memória cache, que é uma **RAM** (Random Access Memory) estática. A memória cache normalmente é usada pela CPU para agilizar a execução de instruções de programa usando técnicas como pré-busca e pipelining (técnicas que tentam adivinhar que informação a CPU precisa, deixando-as previamente disponíveis para a CPU). Esta memória localiza-se dentro da CPU.

O próximo nível de armazenamento primário é a **DRAM** (Dynamic RAM), que oferece a área de trabalho principal para a CPU, para manter instruções de programa e dados. Ela é popularmente chamada de memória principal. Veja aqui uma vantagem da DRAM.

No nível de **armazenamento secundário e terciário**, a hierarquia inclui **discos magnéticos**, bem como armazenamento em massa na forma de dispositivos de **CD-ROM**, **DVD** e **Blue Ray** e, por fim, fitas no extremo mais barato da hierarquia.

A capacidade de armazenamento é medida em kilobytes (Kbyte ou 1.024 bytes), megabytes (MB ou 1 milhão de bytes), gigabytes (GB ou 1 bilhão de bytes) e até mesmo terabytes (1.024 GB).

A palavra **petabyte** (1.024 terabytes ou 10^ 15 bytes) agora está se tornando relevante no contexto de repositórios muito grandes de dados na física, astronomia, ciências terrestres, aplicações científicas e também em sistemas de armazenamento enorme de dados (VLDB – Very Large Data Base).

Veja aqui
A vantagem da DRAM é seu baixo custo, que continua a diminuir; a desvantagem é sua volatilidade (os dados são perdidos ao desligar ou reiniciar o computador) e menor velocidade em comparação com a cache (RAM estática).

04

Os programas residem e são executados na DRAM. Em geral, grandes bancos de dados permanentes residem no armazenamento secundário (discos magnéticos) e partes do banco de dados são lidas e escritas de buffers na memória principal conforme a necessidade. Atualmente, os computadores pessoais e as estações de trabalho possuem grandes memórias principais na ordem de 4 a 128 gigabytes de RAM ou DRAM, de modo que está se tornando possível carregar uma grande parte do banco de dados na memória principal. 64 a 512 GB de memória principal em um único servidor está se tornando algo comum. Em alguns casos, bancos de dados inteiros podem ser mantidos na memória principal (com uma cópia de backup no disco magnético), levando a bancos de dados de memória principal. Estes são particularmente úteis em aplicações de tempo real que exigem tempos de resposta extremamente rápidos.

Entre a DRAM e o armazenamento em disco magnético, outra forma de memória, a memória flash, está se tornando comum, principalmente porque ela é não volátil. As memórias flash são de alta densidade, alto desempenho, e usam a tecnologia EEPROM (Electrically Erasable Programmable Read-Only Memory).

A **vantagem** da memória flash é a velocidade de acesso rápida.

A **desvantagem** é que um bloco inteiro precisa ser apagado e gravado simultaneamente.

As placas de memória flash estão aparecendo como o meio de armazenamento de dados em aparelhos domésticos com capacidades de quase um terabyte. Estão presentes em câmeras, MP3 players, telefones celulares, Smartphones, e assim por diante. Unidades flash USB (Universal Serial Bus) se tornaram o meio mais portátil para transportar dados entre computadores pessoais; elas têm um dispositivo de armazenamento de memória flash integrado a uma interface USB.

Exemplo
Um exemplo são as aplicações de comutação de telefonia, que armazenam bancos de dados que contêm informações de roteamento e linha na memória principal.

05

Discos de CD-ROM, DVD e BlueRay armazenam dados opticamente e são lidos por um laser. Esses discos contêm dados pré-gravados que não podem ser modificados. Os discos do tipo **WORM** (Write-Once-Read-Many) são uma forma de armazenamento óptico usado para arquivar dados; eles permitem que os dados sejam gravados uma vez e lidos qualquer número de vezes sem a possibilidade de apagamento. No caso dos Blue Rays, eles podem manter até 100 gigabytes de dados por disco e duram muito mais do que os discos magnéticos. A maioria das unidades de disco de computador pessoal modernas lê e grava discos de CD-ROM, DVD e Blue Ray.

Finalmente, as **fitas magnéticas** são usadas para arquivamento e armazenamento de backup dos dados. Os jukeboxes de fita — que contêm um banco de fitas que são catalogadas e podem ser carregadas automaticamente nas unidades de fita — estão se tornando populares como armazenamento terciário, para manter terabytes de dados. Por exemplo, o sistema satélite de observação da Terra (EOS — Earth Observation Satellitc) da NASA armazena bancos de dados arquivados dessa forma.

Muitas organizações grandes já estão achando normal ter bancos de dados com tamanhos na ordem dos terabytes. O termo banco de dados muito grande não pode mais ser definido com exatidão, pois as capacidades de armazenamento em disco estão aumentando e os custos, diminuindo. Logo, o termo pode ser reservado para bancos de dados acima de um petabyte.

06

1.2 - Armazenamento de bancos de dados

Os bancos de dados costumam armazenar grande quantidade de dados que precisam persistir por longos períodos de tempo e, portanto, eles costumam ser considerados dados persistentes. Partes

desses dados são acessadas e processadas repetidamente durante esse período. Isso é contrário à noção de **dados transientes**, que persistem apenas por um tempo limitado durante a execução do programa.

A maioria dos bancos de dados é armazenada de maneira permanente (ou **persistentemente**) no armazenamento secundário do disco magnético, pelos seguintes motivos: em geral, os bancos de dados são muito grandes para caber inteiramente na memória principal. As circunstâncias que causam perda permanente de dados armazenados surgem com menos frequência para o armazenamento de disco secundário do que para o armazenamento primário. Logo, referimo-nos ao disco — e a outros dispositivos de armazenamento secundário — como armazenamento não volátil, enquanto a memória principal normalmente é chamada de armazenamento volátil.

O custo de armazenamento por unidade de dados está na ordem de grandeza menor para o armazenamento de disco secundário do que para o armazenamento primário. Algumas das tecnologias mais novas — como discos ópticos, DVDs e jukeboxes de fita — provavelmente oferecem alternativas viáveis ao uso de discos magnéticos.

No futuro, portanto, os bancos de dados poderão residir em diferentes níveis de hierarquia de memória. Contudo, já se sabe que os discos magnéticos continuarão a ser o meio de escolha principal para bancos de dados grandes por muitos anos. Logo, é importante estudar e entender as propriedades e as características dos discos magnéticos e o modo como os arquivos podem ser organizados neles, a fim de projetar bancos de dados eficazes, com desempenho aceitável.

07

As **fitas magnéticas** são frequentemente usadas como meio de armazenamento para o *backup* de bancos de dados, pois o armazenamento em fita custa ainda menos que o armazenamento em disco. No entanto, o acesso aos dados na fita é muito lento. Os dados armazenados nas fitas são off-line, ou seja, alguma intervenção por um operador — ou um dispositivo de carga automática — para carregar uma fita é necessário antes que os dados se tornem disponíveis. Ao contrário, os discos são dispositivos on-line, que podem ser acessados diretamente a qualquer momento. As técnicas utilizadas para armazenar grande quantidade de dados estruturados em disco são importantes para os projetistas de banco de dados, para o DBA e para implementadores de um SGBD.

Os projetistas de banco de dados e o DBA precisam conhecer as vantagens e desvantagens de cada técnica de armazenamento quando projetam, implementam e operam um banco de dados em um SGBD específico. Em geral, o SGBD tem várias opções disponíveis para organizar os dados.

O processo de projeto físico de banco de dados envolve a escolha das técnicas de organização de dados em particular que mais se ajustam a determinados requisitos da aplicação dentre as opções disponíveis. Os implementadores de sistema de SGBD devem estudar as técnicas de organização de dados de modo que possam implementá-las de modo eficaz e, portanto, oferecer ao DBA e aos usuários do SGBD opções corretas de arquitetura.

As aplicações de banco de dados típicas só precisam de uma pequena parte do banco de dados de cada vez para processamento. Sempre que certa parte dos dados é necessária, ela precisa ser localizada no disco, copiada para a memória principal para processamento e, depois, reescrita para o disco se os dados forem alterados.

Os dados armazenados no disco são organizados como arquivos de registros. Cada registro é uma coleção de valores de dados que podem ser interpretados como fatos sobre entidades, seus atributos e relacionamentos. Os registros devem ser armazenados em disco de uma maneira que torne possível localizá-los de modo eficiente quando necessário.

08

2 - DISPOSITIVOS DE ARMAZENAMENTO SECUNDÁRIOS

Nesta seção, descrevemos algumas características dos dispositivos de armazenamento em disco magnético e fita magnética.

2.1 - Descrição de *hardware* dos dispositivos de disco

Os **discos magnéticos** são usados para armazenar grande quantidade de dados. A unidade de dados mais básica no disco é um único bit de informação. Ao magnetizar uma área do disco de certas maneiras, pode-se fazer que ele represente um valor de bit de O (zero) ou 1 (um).

Para codificar a informação, os bits são agrupados em bytes (ou caracteres). Os tamanhos de byte normalmente variam entre 4, 8, 16, 32, 64 ou 128 bits, dependendo do computador e do dispositivo. Consideramos que um caractere é armazenado em um único byte, e usamos os termos byte e caractere para indicar a mesma coisa. A capacidade de um disco é o número de bytes que ele pode armazenar, que em geral é muito grande. Pequenos disquetes usados no passado com microcomputadores costumavam manter de 400 KB a 1,5 MB; mas eles provavelmente já saíram de circulação em quase todos os locais do mundo. Foram substituídos por PenDrives, de 2 a 512 GBytes de espaço.

Os discos rígidos para computadores pessoais modernos mantêm de 250 GB até alguns TB; e grandes pacotes de disco usados com servidores e mainframes têm capacidades de centenas de TB. As capacidades de disco continuam a crescer à medida que a tecnologia se aperfeiçoa. Os discos rígidos são fabricados com padrões específicos de velocidade de rotação, quanto maior a velocidade de rotação, mais rápido eles conseguem ler e gravar informações. Há, basicamente, três padrões de velocidade: 5.400 RPM, 10.000 RPM e 15.000 RPM.

09

Há várias tecnologias de controle e acesso dos discos rígidos. As mais comuns são IDE, SATA e SCSI. Atualmente, a tecnologia de HDs SATA é indicada para estações de trabalho, enquanto que a tecnologia SCSI é indicada para computadores servidores.

Não vamos discutir aqui a fundo as características dessas tecnologias, visto não serem escopo na nossa disciplina. Entretanto, para saber mais sobre elas, consulte um mecanismo de busca ou mídias especializadas.

Por ora, a única informação que precisamos saber é que um **disco rígido leva em torno de 1 a 100 milissegundos para ler e gravar um byte**. Embora isso pareça ser extremamente rápido, o disco rígido é o principal gargalo das aplicações de banco de dados, e usá-lo de modo eficiente é primordial para que nossos usuários consigam operar os sistemas de informação que utilizam banco de dados numa performance adequada ao trabalho. Para tal, precisamos minimizar o número de transferências de dados necessárias para localizar e transferir os dados entre os discos rígidos e a memória principal. Por exemplo, ao colocar informações relacionadas sequencialmente (em blocos contíguos), poderemos aumentar a performance em relação a ter esses mesmos dados de modo aleatório.

10

2.2 - Dispositivos de armazenamento de fita magnética

Discos são dispositivos de armazenamento secundário de acesso aleatório, pois um bloco de disco qualquer pode ser acessado aleatoriamente depois que especificamos seu endereço.

As fitas magnéticas são dispositivos de acesso sequencial; para acessar o enésimo bloco na fita, primeiro temos de varrer os n — 1 blocos anteriores.

Os dados são armazenados em bobinas de fita magnética de alta capacidade, semelhantes às fitas de áudio ou vídeo. Uma unidade de fita precisa ler os dados ou gravá-los em uma bobina de fita. Em geral, cada grupo de bits que forma um byte é armazenado na fita, e os próprios bytes são armazenados consecutivamente nela.

A principal característica de uma fita é seu requisito de que acessemos os blocos de dados em ordem sequencial.

Para chegar até um bloco no meio da bobina de fita, a fita é montada e depois varrida até que o bloco solicitado passe sob a cabeça de leitura/gravação. Por esse motivo, o acesso da fita pode ser lento e elas não são usadas para armazenar dados on-line, exceto para algumas aplicações especializadas. Porém, as fitas têm uma função muito importante — backup do banco de dados. Uma razão para haver backup é para manter cópias de arquivos de disco no caso de perda de dados por uma falha no disco, que pode

acontecer se a sua cabeça de leitura/gravação tocar na superfície por defeito mecânico. Por esse motivo, os arquivos de disco são copiados periodicamente para a fita.

11

Para muitas aplicações críticas on-line, como sistemas de reserva aérea, para evitar qualquer tempo de paralisação, são usados **sistemas espelhados** para manter três conjuntos de discos idênticos — dois em operação on-line e um como backup. Aqui, os discos off-line tornam-se um dispositivo de backup. Os três são usados de modo que possam ser trocados, caso haja uma falha em uma das unidades de disco ativas.

As fitas também podem ser utilizadas para armazenar arquivos de banco de dados excessivamente grandes. Os arquivos do banco de dados que raramente são usados ou estão desatualizados, mas são necessários para manutenção de registro histórico, podem ser arquivados em fita.

Saiba+ sobre o uso das fitas para o armazenamento de dados.

O backup de bancos de dados corporativos, de modo que nenhuma informação de transação seja perdida, é uma tarefa importante. Atualmente, as bibliotecas de fitas com slots para várias centenas de cartuchos são usadas com Digital e Superdigital Linear Tapes (DLTs e SDLTs), com capacidades na ordem de centenas de gigabytes, que registram dados em trilhas lineares. Braços robóticos são usados para gravar em vários cartuchos em paralelo, utilizando várias unidades de fita com *software* de rotulagem automático para identificar os cartuchos de backup.

Um exemplo de uma biblioteca gigante é o modelo da Sun Storage Technology, que pode armazenar até 70 petabytes (1 petabyte equivale a 1.000 TB) de dados usando até 448 unidades com uma taxa de vazão máxima de 193,2 TB/hora.

Saiba+
Originalmente, as unidades de fita com bobina de meia polegada eram usadas para o armazenamento de dados, empregando as chamadas fitas de nove trilhas. Mais tarde, fitas magnéticas menores, de 8mm (semelhantes às usadas em filmadoras portáteis), que podem armazenar até 50 GB, bem como os cartuchos de dados de varredura helicoidal de 4mm e CDs, DVDs e BlueRays graváveis, tornaram-se mídia popular para o backup de arquivos de dados dos PCs e estações de trabalho. Eles também são usados para armazenar imagens e bibliotecas de sistemas.

12

2.3 - Gravando registros de arquivo no disco

Nesta seção, definimos os conceitos de registros, tipos de registro e arquivos. Depois, discutimos sobre as técnicas para gravar registros de arquivo no disco.

2.3.1 - Registros e tipos de registro

Os dados costumam ser armazenados na forma de **registros**. Cada registro contém uma coleção de valores ou itens de dados relacionados, no qual cada valor é formado por um ou mais bytes e corresponde a um campo em particular do registro. Os registros normalmente descrevem entidades e seus atributos.

Por exemplo, um registro de FUNCIONARIO representa uma entidade de funcionário, e o valor de cada campo no registro especifica algum atributo desse funcionário, como ID, Nome, Data_nascimento, Salario ou Supervisor.

Uma coleção de nomes de campo e seus tipos de dados correspondentes constituem uma definição de tipo de registro ou formato de registro. Um tipo de dado, associado a cada campo, especifica os tipos de valores que um campo pode assumir. O tipo de dado de um campo normalmente é um dos tipos de dado padrão usados na programação. Estes incluem os tipos de dados:

- **numéricos** (inteiro, inteiro longo ou ponto flutuante),
- **cadeia de caracteres** (tamanho fixo ou variável),
- **booleanos** (tendo apenas valores O e 1, ou TRUE e FALSE),
- às vezes, tipos **data** e **tempo** especialmente codificados.

13

O número de bytes exigidos para cada tipo de dado é fixo para determinado sistema de computação. Um inteiro pode exigir 4 bytes, um inteiro longo, 8 bytes, um número real, 4 bytes, um booleano, 1 byte, e uma data, 10 bytes (considerando um formato DD-MM-AAAA), e uma string de tamanho fixo de x caracteres, x bytes. As strings de tamanho variável podem exibir tantos bytes quantos caracteres existirem no valor de cada campo.

Por exemplo, um tipo de registro FUNCIONARIO pode ser definido com a seguinte estrutura:

```
        FUNCIONARIO
                ID – inteiro – 4 bytes;
                Nome – 255 caracteres – 255 bytes;
                CPF – 11 caracteres – 11 bytes;
Data_nascimento – data – 10 bytes;
Salario – real – 4 bytes;
ID_Supervisor – inteiro – 4 bytes.
----------------------------------------
Total: 288 bytes.
```

Dessa forma, podemos perceber que cada registro armazenado contempla 288 bytes de informação.

Em algumas aplicações de banco de dados, pode haver necessidade de armazenar itens de dados que consistem em grandes objetos não estruturados, que representam imagens, vídeo digitalizado ou streams de áudio, ou então texto livre. Estes são conhecidos como **BLOB**s (objetos binários grandes).

Um item de dados BLOB costuma ser armazenado separadamente de seu registro, em um conjunto de blocos de disco, e um ponteiro para o BLOB é incluído no registro. Dessa forma, o registro, ao ser armazenado utiliza 4 bytes para registrar o ponteiro para o conteúdo BLOB além do espaço necessário para o conteúdo BLOB em si (que pode ser até na ordem de GigaBytes).

14

2.3.2 - Arquivos, registros de tamanho fixo

Um arquivo é uma sequência de registros.

Em muitos casos, todos os registros em um arquivo são do mesmo tipo de registro. Se cada registro no arquivo tem exatamente o mesmo tamanho (em bytes), o arquivo é considerado composto de registros de tamanho fixo. Se diferentes registros no arquivo possuem diversos tamanhos, o arquivo é considerado composto de registros de tamanho variável.

Um arquivo pode ter registros de tamanho variável por vários motivos:

- Os **registros do arquivo são do mesmo tipo de registro, mas um ou mais dos campos são de tamanho variável** (campos de tamanho variável). Por exemplo, o campo Nome de FUNCIONARIO pode ser um campo de tamanho variável (exemplo: varchar (255)).
- Os **registros do arquivo são do mesmo tipo de registro, mas um ou mais dos campos podem ter múltiplos valores para registros individuais**; esse campo é chamado de campo repetitivo (ou vetorial), e um grupo de valores para o campo normalmente é chamado de grupo repetitivo (ou vetor).
- Os **registros do arquivo são do mesmo tipo de registro, mas um ou mais dos campos são opcionais**, ou seja, eles podem ter valores para alguns, mas não para todos os registros do arquivo (campos opcionais).
- O **arquivo contém registros de tipos de registro diferentes** e, portanto, de tamanho variável (arquivo misto). Isso ocorreria se registros relacionados de diferentes tipos fossem agrupados (colocados juntos) em blocos de disco; por exemplo, os registros de HISTORICO_ESCOLAR de determinado aluno podem ser colocados após o registro desse ALUNO.

15

No exemplo a seguir, apresentaremos o registro ALUNO, com os seguintes campos: ID (em amarelo), Nome (em laranja), CPF (em verde) e Dt_Nascimento (em vermelho).

ID	NOME	CPF	DT_NASCIMENTO
0	Marcelo Souza Barros	54386155308	15051973
1	Rui Leal Brito	87686065775	01071967
2	Andreia Costa Luz	87686065775	03091987

Exemplo de espaço ocupado por cada um dos registros do banco de dados.

No exemplo acima, os registros de FUNCIONARIO de tamanho fixo têm um tamanho de registro de 52 bytes. Cada registro tem os mesmos campos, e os tamanhos dos campos são fixos, de modo que o sistema pode identificar a posição do byte inicial de cada campo em relação à posição inicial do registro. Isso facilita a localização de valores de campo pelos programas que acessam tais arquivos. Observe que essa estrutura desperdiça espaço em disco, pois há várias posições no registro (no campo nome) que não possuem informações.

Para **registros de tamanhos fixos**, a fórmula que dá a posição do início do registro é: S x T -1, sendo S o número da sequência do registro desejado e T o tamanho do registro. O -1 no final refere-se à que a posição inicial é o byte zero, por isso há necessidade de subtrair 1 do valor final.

Por exemplo, sabendo que cada registro ocupa 58 bytes, para localizar o registro de ID 3, devermos encontra-lo na posição 3x58-1, que seria o byte de posição 173. A partir desse byte, deveríamos ler 52 bytes para ler o registro completo.

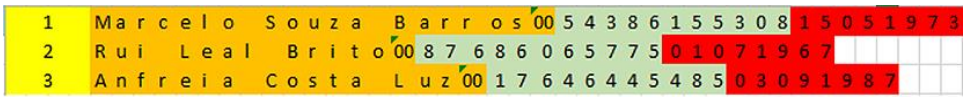
Uma das grandes vantagens dos registros de tamanho fixo é a sua performance. Por usar espaços pré-determinados, não há fragmentação. Lembre-se: só haverá desperdício de espaço se houver campos texto com espaço em branco. Uma tabela que não possua campos texto ou que os campos texto ocupem todas as posições não apresenta desperdício.

16

2.3.3 - Registros de tamanho variável

É possível representar um arquivo que logicamente deveria ter registros de tamanho variável como um arquivo de registros de tamanho fixo. Por exemplo, no caso dos campos opcionais, poderíamos ter cada campo incluído em cada registro de arquivo, mas armazenar um valor especial NULL se não houver valor para esse campo. Para um campo repetitivo, poderíamos alocar tantos espaços em cada registro quanto o número máximo possível de ocorrências do campo. De qualquer forma, o espaço é desperdiçado quando certos registros não têm valores para todos os espaços físicos fornecidos em cada registro.

Agora, vamos considerar outras opções para formatar registros de um arquivo de registros de tamanho variável. Para campos de tamanho variável, cada registro tem um valor para cada campo, mas não sabemos o tamanho exato de alguns valores de campo. Para determinar os bytes em um registro em particular que representa cada campo, podemos usar caracteres separadores especiais (como o caractere ASC 00) para terminar os campos de tamanho variável. Veja o exemplo a seguir onde o byte 00 foi usado como terminador do campo nome. Veja como ficaria a mesma tabela ALUNO do exemplo anterior, com os seguintes campos: ID (em amarelo), Nome (em laranja), CPF (em verde) e Dt_Nascimento (em vermelho):

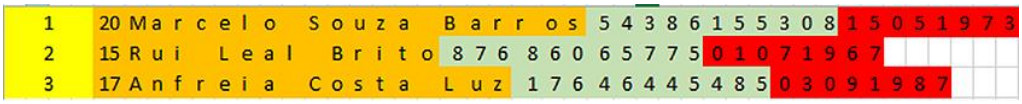


Registros de tamanho variável com byte identificador de término de informação

No exemplo acima, podemos ver o cadastro de três pessoas de nomes e comprimento de caracteres diferentes. O byte 00 foi utilizado como terminador de tamanho do campo nome de cada um dos registros. Caso o primeiro caractere de nome fosse 00, entenderíamos que esse campo possui valor NULL.

17

Outra forma é armazenar o tamanho do campo em bytes no próprio registro, antes do valor do campo, conforme exemplo a seguir:



Registros de tamanho variável com identificação do tamanho do campo em bytes antes da informação.

No exemplo acima, os valores 20, 15 e 17 antes do campo nome representa os tamanhos em caracteres (bytes) de cada nome. Caso o tamanho do campo fosse 0, entenderíamos que esse campo possui valor NULL.

Os controles utilizados pelos SGBDs para delimitar os registros de tamanho variável são transparentes para os usuários. Portanto, não se preocupe qual técnica o seu SGBD utiliza, basta saber que ele emprega uma técnica que minimiza o desperdício de espaço. Entretanto, registros de tamanho variável podem gerar fragmentação de dados e possuem menor performance que os registros de tamanho fixo.

18

2.3.4 - Armazenando dados em sequência ou espalhados

Dentro de um disco rígido, os dados dos registros podem ser armazenados de duas formas:

- em sequência ou
- espalhados.

Quando **em sequência**, conseguimos garantir o máximo de performance de consulta, pois podemos prever claramente a posição de um determinado registro na sequência. Entretanto, quando há acréscimo de informações, podemos criar um problema grande de necessidade de reorganização dos registros afim de que o espaço seja ajustado de forma a garantir que espaços com outras informações sejam desocupadas (movidas) para garantir a sequência ininterrupta dos registros.	De outra forma, uma **organização espalhada** é aquela que ao final de um registro existe um ponteiro para o endereço do próximo registro da sequência. Dessa forma, conseguimos a compactação máxima, pois conseguir utilizar todos os espaços disponíveis. O problema aqui está na recuperação, quem, sem um índice coerente, será necessário pesquisar registro a registro até encontrar o registro desejado.

2.4 - Conclusão

A principal conclusão que devemos ter é quando escolher registros de tamanho fixo ou de tamanho variável. Lembrando que essa questão se aplica tão somente a arquivos do tipo texto. Para registros texto de tamanho fixo usamos o tipificador CHAR e para tamanho variável VARCHAR.

Já observamos as vantagens e desvantagens de cada um; agora, a grande pergunta é: **quando devo usar um e quando devo usar o outro?** Aqui vão algumas dicas:

- Quando todos os possíveis valores dos campos forem de tamanho padronizado fixo, use o tipo CHAR. Exemplos: CEP – char (8), sigla da unidade da federação – char (2), CPF – char (11), CNPJ – char (14), e, em alguns casos, campos de códigos internos da organização, como número de protocolo (exemplo: 483.000.233/2015-00), número de matrícula (exemplo: 4432/9) e outros.
- Quando os tamanhos dos valores forem variáveis, mas o maior valor possível for até 10 caracteres, utilize o tamanho fixo com o tamanho máximo possível. Exemplos: lista de cores – char (10), exemplo: amarelo, rosa, branco, roxo... Lista de estados civis – char(10): solteiro, casado, desquitado, viúvo...
- Quando os tamanhos dos valores forem variáveis, e há valores com mais de 10 caracteres e menos de 4000, utilize varchar e um número grande adequado. Exemplo: Nome varchar(255), Endereco varchar (1000), Descricao varchar (4000).
- Para informações com mais de 4000 caracteres, recomenda-se o uso de campos especiais, como TEXTO ou BLOB.

RESUMO

Neste módulo, aprendemos que:

a) Os bancos de dados são armazenados como arquivos físicos dentro de discos rígidos. Alguns bancos de dados especiais são mantidos em memória RAM ou mesmo em fitas magnéticas.

b) Manter um banco de dados em memória RAM aumenta, em muito, a performance do banco de dados, mas, caso ocorra perda de energia (ou travamento da máquina), os dados não salvos em arquivos físicos serão perdidos. Já os sistemas que usam bancos de dados em fitas magnéticas são casos muito específicos, que lidam com enorme quantidade de informação ou tecnologia que não permite outra solução viável.

c) O armazenamento primário representa o hardware diretamente conectado ao processador central do computador. Possui altíssima performance, baixa capacidade de armazenamento e alto custo por byte armazenado (quando comparados com outros tipos de armazenamento). São usados para manter os dados mais comumente utilizados pela CPU bem próximo dela, fazendo assim consultas e operações em velocidades altíssimas.

d) O armazenamento secundário representa todos os dispositivos de armazenamento, cuja existência não é obrigatória, e que está diretamente ligado à CPU. Neste quesito, os discos rígidos podem não existir em alguns tipos de computador, e por isso são considerados secundário.

e) O armazenamento terciário corresponde aos dispositivos externos (e portanto, móveis e opcionais) à placa mãe e ao computador. Nesta classificação enquadram-se pen drives, discos óticos, fitas magnéticas e outros.

f) Um computador sempre obedece à hierarquia de armazenamento. Um dado não passa diretamente do disco rígido para a CPU sem passar antes pela memória DRAM.

g) Quanto à disponibilidade de acesso, os dispositivos de armazenamento são classificados em on-line (quando os dispositivos estão sempre conectados e disponíveis, como os discos rígidos) e off-line (quando os dispositivos precisam ser conectados, como discos óticos e fitas magnéticas).

h) Os arquivos de banco de dados referem-se à estrutura de dados e aos registros mantidos. Cada registro corresponde em bytes ao tamanho definido na estrutura, esteja ele preenchido com dados ou não. Uma exceção se dá aos campos de texto variável (como o VARCHAR), cujo espaço ocupado é proporcional ao conteúdo armazenado.

i) Os motivos pelos quais um registro pode possuir tamanho variável são: existência de campos variáveis, valores múltiplos, campos opcionais, registros de tipos diferentes aglutinados.

j) Registros de tamanho fixo não geral fragmentação e são mais eficientes, entretanto, podem desperdiçar espaço em disco se houver espaços em branco em campos do tipo textual.

k) Registros de tamanho variável possuem duas formas de identificação do tamanho do registro: ou se utiliza um terminador de campo (como o caractere ASC 00), ou se inicia o registro apresentando o tamanho do mesmo.

UNIDADE 1 – ESTRUTURA DE ARQUIVOS E PROCESSAMENTO DE CONSULTAS
MÓDULO 2 – TECNOLOGIA RAID (PARTE 1)

01

1 - TECNOLOGIA RAID

Com o crescimento exponencial no desempenho e na capacidade dos dispositivos semicondutores e memórias, microprocessadores mais rápidos, com memórias primárias cada vez maiores, estão continuamente se tornando disponíveis. Para corresponder a esse crescimento, é natural esperar que a tecnologia de armazenamento secundário também deva acompanhar a tecnologia do processador em desempenho e confiabilidade. Um avanço importante na tecnologia de armazenamento secundário é representado pelo desenvolvimento do **RAID**, que originalmente significava *Redundant Array of Inexpensive Disks*. Mais recentemente, o I em RAID passou a significar *Independent*.

RAID (Redundant Array of Independent Disks ou Conjunto Redundante de Discos Independentes) é uma tecnologia utilizada principalmente em servidores que consiste em um conjunto de dois ou mais discos rígidos.

A ideia do RAID recebeu um endosso muito positivo da indústria, e desenvolveu-se em um elaborado conjunto de arquiteturas RAID alternativas (RAID níveis 0 a 6, RAID 01, RAID 10, RAID 50 e RAID 100).

Há dois **objetivos** em se utilizar a tecnologia RAID:

a) O objetivo principal do RAID é nivelar as diferentes taxas de melhoria de desempenho dos discos contra aquelas na memória e nos microprocessadores.
b) Oferecer medidas de proteção a falhas de discos (evitando assim perda de dados).

02

Enquanto a capacidade da RAM tem se quadruplicado a cada dois ou três anos, os tempos de acesso do disco estão melhorando em menos de 10% ao ano, e as taxas de transferência do disco estão melhorando em aproximadamente 20% ao ano. As capacidades do disco estão realmente melhorando em mais de 50% ao ano, mas as melhorias em velocidade e tempo de acesso são de uma grandeza muito menor.

Existe uma segunda disparidade qualitativa entre a capacidade dos microprocessadores especiais de atender a novas aplicações envolvendo vídeo, áudio, imagem e processamento de dados espaciais, com a correspondente falta de acesso rápido a conjuntos de dados grandes e compartilhados.

A solução natural é um grande conjunto (conjunto empilhado) de pequenos discos independentes, que atuam como um único disco lógico de maior desempenho.

A criação de um sistema RAID depende de dois ou mais discos, de capacidade idêntica, organizados em determinado modelo RAID e controlados por um gerenciador do RAID. Atualmente, muitas placas mães são capazes de gerenciar discos em combinação RAID. Há também placas específicas capazes de gerenciar o RAID.

03

1.1 - Striping

O Striping (distribuição) caracteriza-se pela associação de discos em paralelo, que promove dois **benefícios**:

a) **Aumenta-se a capacidade de armazenamento**, pois o conjunto de discos é visto como uma única unidade pelo sistema operacional, representada pelo somatório de espaço de todos os discos.
b) O **armazenamento de arquivos é dividido proporcionalmente** em todos os discos, assim, como vários discos trabalham ao mesmo tempo para ler e gravar informações, o desempenho do conjunto aumenta.

Para uma configuração striping, é necessário no mínimo dois discos rígidos. Mas você pode criar um striping com uma quantidade de discos muito grande (grandes datacenters possuem conjuntos na ordem de 1.000 discos ou mais).

Exemplo de configuração: suponha que você tenha 5 discos de 1 TB, ao colocá-los em um sistema de striping, o computador enxergará esses 5 discos como um único disco de 5 TB. Ao armazenar informações nesse conjunto, o sistema gerenciador dos discos irá dividir a informação em 5 pedaços, colocando cada pedaço em um dos discos. Ao ler a informação, o sistema irá coletar cada pedaço de cada disco, juntá-los e disponibilizar o arquivo para o computador (geralmente na memória RAM).

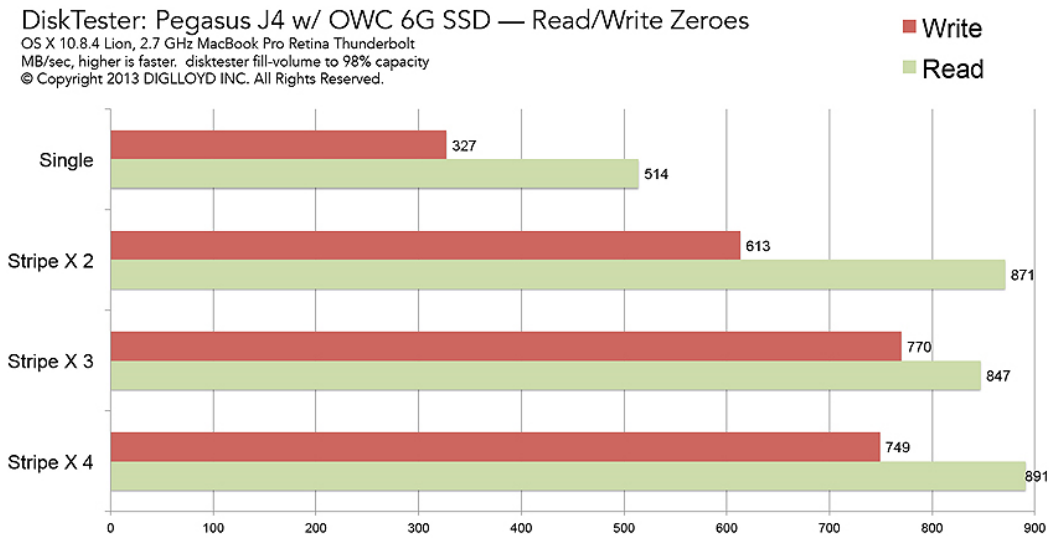
A imagem acima ilustra 3 arquivos divididos em 5 partes e cada parte estando armazenada em um disco diferente.

04

O striping melhora o desempenho geral de acesso a disco, permitindo que cada solicitação de acesso aos dados seja fragmentada e atendida em paralelo, oferecendo assim altas taxas de transferência. Consequentemente, o striping de dados **balanceia a carga entre os discos**.

Vamos ver um exemplo teórico de como funcionaria esse sistema. Suponha que a taxa de transferência de um disco rígido seja de 10 megabytes por segundo. Logo, um arquivo de 600 megabytes leva 60 segundos para ser gravado. Ao utilizar um conjunto de 5 discos, 1/5 do tamanho do arquivo é armazenado em cada disco, dessa forma, para o arquivo de 600 MB, cada disco seria responsável por armazenar 120 MB. O tempo necessário para armazenar 120MB é de 12 segundos, dessa forma, esse sistema de 5 discos aumentaria a performance em 5x, diminuindo de 60 segundos para 12 segundos o tempo de transferência de um arquivo de 600 MB. Entretanto, como há a necessidade de montagem e desmontagem do arquivo e o tempo utilizado entre o barramento da placa mãe, memória e processador, a melhora real obtida não é exatamente essa performance que calculamos.

No mundo real, há uma perda de 10 a 30% de eficiência. A imagem abaixo apresenta um exemplo real de striping de 2, 3 e 4 discos, comparando-os com taxas de leitura e gravação de um disco apenas.



Comparativo de taxas de leitura e gravação em discos montados em strip.
(Fonte http://macperformanceguide.com/PromisePegasusJ4-performance.html)

Observe que as taxas de melhoria real foram de 1,9x (ao invés de 2x para dois discos), 2,4x (ao invés de 3x para três discos) e 2,3x (ao invés de 4x para quatro discos, neste caso houve até mesmo uma piora em relação ao sistema de três discos).

05

A técnica de striping possui uma desvantagem. Um conjunto de discos possui probabilidade maior de falha, e, ao falhar um disco, como o conteúdo de cada arquivo é distribuído entre todos os discos, todos os arquivos serão corrompidos, dessa forma, perde-se todo o conteúdo do conjunto. Por exemplo, um conjunto de 1.000 discos em striping tem 1.000 vezes maior chance de falha, se comparado a um disco único.

Veja num exemplo prático de como isso pode ser um grave problema: se a vida média de um HD é de 20 anos (equivalente a 7.300 dias), um conjunto de 1.000 discos possui uma chance de falha de 7,3 dias! Estatisticamente falando, quando o conjunto é novo, há baixíssima probabilidade de falha, mas na medida em que os discos vão envelhecendo, a taxa de falhas pode ser bem alta.

Uma vez que o striping oferece um risco alto de falhas, é importantíssimo manter atualizado o *backup* dos dados, garantindo assim eventual reconstrução do conjunto. Para recuperar um conjunto em caso de falha, o *backup* deve ser restaurado por completo em todos os discos (e não só no disco defeituoso, visto que é impossível determinar qual parte do arquivo foi perdido do disco defeituoso). Essa reconstrução pode ser uma tarefa muito demorada.

A tecnologia de striping é muito indicada para sistemas que exigem muitas consultas (e pouca ou nenhuma atualização de dados), visto a alta performance de consultas e pouca necessidade de realização frequente de *backup*s. Exemplos de uso dessa tecnologia são os grandes repositórios de multimídia.

É possível montar striping com discos de tamanhos diferentes, o espaço útil será sempre a soma individual dos discos. Exemplo: um striping com um disco de 800GB e um disco de 1 TB será uma unidade de 1,8 TB.

Há várias técnicas RAID que utilizam o conceito do striping, como veremos mais adiante.

06

1.2 - Espelhamento

Manter uma única cópia de dados em um conjunto de discos organizados em striping causará uma perda de confiabilidade significativa. Uma solução é empregar a **redundância de dados** de modo que as falhas de disco possam ser toleradas.

O espelhamento ou redundância refere-se a ter todos os dados duplicados em dois locais diferentes, dessa forma, a perda de dados de um disco poderia ser recuperada on-line do outro disco.

Entretanto, essa técnica possui várias **desvantagens**:

a) As operações de entrada e saída são duplicadas durante a gravação, o que pode sobrecarregar o barramento de troca de dados. Com isso, o tempo exigido para leitura e gravação é um pouco superior se comparado a taxa de transferência de um disco isolado.
b) Há necessidade de computação extra para manter a redundância e realizar a recuperação de erros, o que pode consumir recursos de processamento.
c) Há necessidade de dobrar a capacidade de disco para armazenar informações redundantes. Ou o espaço útil será metade do espaço bruto total disponível.
d) Enquanto a reconstrução do disco danificado não é finalizada, o sistema inteiro fica vulnerável.

Uma técnica para introduzir redundância é chamada de espelhamento ou sombreamento. Os dados são gravados de maneira redundante em dois discos físicos idênticos, que são tratados como um disco lógico. Quando os dados são lidos, eles podem ser apanhados do disco menos sobrecarregado no momento. Se um disco falhar, o outro é usado até que o primeiro seja reparado. As informações serão perdidas se esse segundo disco também falhar antes da reconstrução do conjunto (substituição do HD defeituoso e sincronização dos arquivos), o que cria uma necessidade muito grande de se efetuar trocas rápidas e manter dados em *backup* off-line.

07

1.3 - Bit de paridade e Hamming

Outra solução para o problema de confiabilidade é armazenar informações extras que não são necessárias normalmente, mas que podem ser usadas para reconstruir a informação pedida no caso de falha no disco. A incorporação de redundância precisa considerar dois **problemas**:

a) Selecionar uma técnica para calcular a informação redundante e;
b) Selecionar um método de distribuição da informação redundante pelo conjunto de disco.

O **primeiro problema** é resolvido usando códigos de correção de erro que envolvem bits de paridade, ou códigos especializados como os códigos de Hamming.

Sob o esquema de paridade, um disco redundante pode ser considerado como tendo a soma de todos os dados nos outros discos. Quando um disco falha, a informação que falta pode ser construída por um processo semelhante à subtração.

Para o **segundo problema**, as duas técnicas principais são armazenar a informação redundante em um pequeno número de discos ou distribuí-las uniformemente por todos os discos. A última resulta em melhor balanceamento de carga. Os diferentes níveis de RAID escolhem uma combinação dessas opções para implementar a redundância e melhorar a confiabilidade.

A seguir veremos um **exemplo de como o bit de paridade funciona**.

08

Há apenas duas opções para o bit de paridade, 0 (zero) para paridade PAR ou 1 (um) para paridade ímpar. Suponha agora que determinada informação foi dividida em 8 discos, logo, a informação foi dividida em um bloco de 8 bits. Agora, suponha que essa informação seja 01101101. Ao contar o número de "1" desse bloco, obtemos o número 5, que é ímpar, logo, o bit de paridade para esse bloco é 1, o disco que armazenar o bit de paridade irá armazenar o número 1. Agora vamos supor que o quarto disco falhe e o bit desse disco seja perdido. Dessa forma, o bloco da nossa informação será 011?1101. Sendo que o "?" representa o bit perdido. Ao consultar o bit de paridade, identificamos que o bloco deve ter paridade ímpar. Vamos agora contar os bits que conhecemos do bloco 011?1101: o resultado é 5 bits. Como esse bloco deve possuir paridade ímpar, podemos inferir que o bit desconhecido só pode ser 0.

Vejamos outros exemplos:

- 0?1 – bit de paridade 0: como só temos um número 1, o bit desconhecido é 1 (para que a soma seja par).
- 0?11 – bit de paridade 1: como temos dois números 1, o bit desconhecido é 1 (para que a soma seja ímpar).
- 001011?00111 – bit de paridade 0: como temos seis números 1, o bit desconhecido é 0 (para que a soma seja par).
- 011001101? – bit de paridade 1: como temos cinco números 1, o bit desconhecido é 0 (para que a soma seja ímpar).

Note que, ao perder um disco que possua um bit significativo, o sistema pode reconstruir o disco perdido por meio do cálculo do bit de paridade. Entretanto, se dois ou mais discos forem perdidos, e consequentemente dois ou mais bits forem perdidos, então não há como reconstruir o bloco.

Exemplo: 01?00?101 – bit de paridade 0. O resultado é par e a soma atual é 3, então ou o primeiro bit é 0 e o segundo é 1 ou o primeiro bit é 1 e o segundo é 0. Dessa forma, não há como calcular os bits perdidos.

09

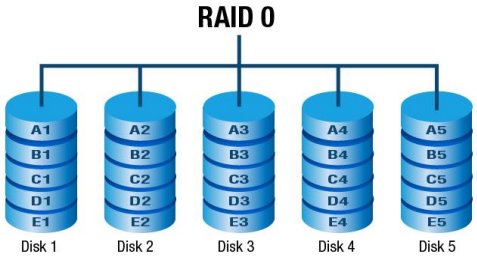
2 - Organizações e níveis de RAID

Diferentes organizações de RAID foram definidas com base em diversas combinações dos dois fatores de detalhamento da intercalação (striping) e do padrão de dados usados para calcular informações redundantes. Na proposta inicial, os níveis de 1 a 5 de RAID foram propostos, e dois níveis adicionais — O e 6 — foram acrescentados depois. Posteriormente surgiram outros modelos: RAID 01 (ou 0+1), RAID

1+0, RAID 50 e RAID 100. Veremos os modelos RAID 0 e RAID 1 neste módulo, e os demais modelos RAID mais adiante.

2.1 - RAID 0

O RAID O (zero) usa apenas a técnica de striping de dados. Dessa forma, todas as características de striping se aplica ao modelo RAID 0. Muitas organizações utilizam o RAID 0 quando o objetivo é prover um grande espaço usado principalmente para leitura, com poucas modificações. Para esse sistema é fundamental possuir um *backup* das informações, visto que a perda de um disco paralisa todo o sistema.



RAID 0 – striping de dados

Observe na imagem acima como os arquivos A, B, C, D e E são fragmentados em cinco partes e cada parte é colocada em um disco. Para ler o arquivo, as cinco partes devem ser juntadas.

10

Veja a seguir quais são as vantagens e desvantagens do Raid 0.

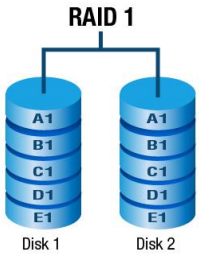
Vantagens	Desvantagens
•Acesso rápido as informações (até três vezes mais rápido); •Custo baixo para expansão de memória.	•Não possui tolerância a falhas de disco. Caso algum dos setores de algum dos HDs venha a apresentar perda de informações, o mesmo arquivo, que está dividido entre os mesmos setores dos demais HDs, não terão mais sentido existir, pois uma parte do arquivo foi corrompida, ou seja, caso algum disco falhe, não tem como recuperar.

11

2.2 - RAID 1

O RAID 1 usa apenas a técnica de espelhamento de dados. Dessa forma, todas as características de espelhamento se aplica ao modelo RAID 1. Apenas dois discos podem ser utilizados nessa configuração. Para aumento de número de discos (e consequentemente aumento de espaço em disco), outra técnica RAID deve ser configurada (como RAID 10 ou RAID 01).

Muitas organizações de pequeno porte utilizam o RAID 1 quando o objetivo é prover um espaço seguro, que aceite muitas leituras e modificações. Esse sistema oferece tolerância a falhas, visto que a perda de um disco não paralisa o sistema. Portanto, é a escolha ideal (e mais cara) para sistemas críticos.



RAID 1 – espelhamento

Observe na imagem acima como os arquivos A1, B1, C1, D1 e E1 são duplicados em cada um dos discos. Para ler o arquivo, o sistema pode escolher o disco mais ocioso para realizar a consulta. Nesse modelo, não é possível pegar parte do arquivo de um disco e a outra parte do arquivo do outro disco (o que poderia gerar algum tipo de melhoria de performance de leitura).

12

Observe a seguir quais são as vantagens e desvantagens do Raid 1.

Vantagens	Desvantagens
•Tolerância a falhas. Caso algum setor de um dos discos venha a falhar, basta recuperar o setor defeituoso copiando os arquivos contidos do segundo disco; •Segurança nos dados (com relação a possíveis defeitos que possam ocorrer no HD); •Reconstrução rápida dos dados.	•Custo alto, pois o espaço útil é apenas metade do espaço bruto total; •Baixa performance para escrita de dados; •Limitado a dois discos.

13

RESUMO

Vimos neste módulo as duas técnicas que os sistemas de disco utilizam para aumento de performance e tolerância a falhas.

Neste módulo, aprendemos que:

a) RAID é uma tecnologia de configuração de dois ou mais discos rígidos a fim de se obter alguma vantagem em relação à utilização individual de cada um.
b) Os padrões de RAID mais comuns são: RAID 0 a 6, RAID 01, RAID 10, RAID 50 e RAID 100.

c) Os modelos RAID focam em dois benefícios: oferecer um ambiente tolerante a falhas e/ou aumentar a performance do conjunto.

d) Os sistemas RAID são controlados ou diretamente pela placa mãe ou por placas específicas.

e) Striping refere-se à capacidade de distribuir um conjunto de dados em vários discos diferentes. O striping oferece aumento da capacidade de armazenamento (ao somar as unidades de disco como um disco único maior) e aumento da velocidade de acesso (pelo fato de que cada disco grava e lê apenas um pedaço da informação).

f) O espelhamento refere-se à técnica de copiar (e manter sincronizados) os arquivos de um disco em outro disco. Na eventual falha de um dos discos, o outro poderá ser utilizado.

g) O bit de paridade é uma técnica utilizada para identificação de um bit de um determinado conjunto de bits. Essa técnica é comumente utilizada em sistemas que realizam striping para ser possível reconstruir um disco defeituoso. Outra técnica semelhando é denominada Hamming.

h) O RAID 0 (zero) é o modelo RAID que utiliza a técnica de striping (distribuição). Nele, todas as unidades de disco são juntadas como uma única unidade para o sistema operacional. Os dados são distribuídos entre vários discos, com isso há um ganho de performance. O maior problema do RAID 0 é que ele não é tolerante a falhas, dessa forma, a perda de um disco paralisa todo o sistema.

i) O RAID 1 é o modelo RAID que utiliza a técnica de espelhamento. Nele, dois discos são utilizados para criar duas áreas separadas, mas sincronizadas entre si. As informações são duplicadas nos dois discos, dessa forma, caso um disco falhe de um conjunto, o outro disco continua funcional.

UNIDADE 1 – ESTRUTURA DE ARQUIVOS E PROCESSAMENTO DE CONSULTAS
MÓDULO 3 – TECNOLOGIA RAID (PARTE 2)
01

1 - TECNOLOGIA RAID

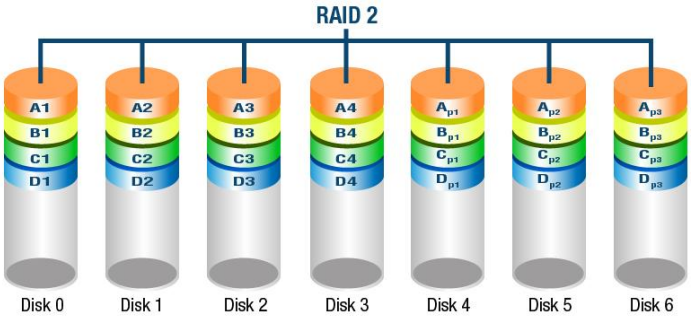
No módulo passado, vimos como as técnicas de striping, espelhamento e bit de paridade podem ser utilizadas para criar sistemas de discos seguros, eficientes e de alta capacidade. Vimos também como os modelos de RAID 0 (zero) utiliza a técnica de striping e o modelo RAID 1 utiliza a técnica de espelhamento. Vimos também que o RAID 0 pode ser construído com qualquer quantidade de discos acima de dois, já o RAID 1 só suporta dois discos. Neste modelo vamos ver outras técnicas de arranjo RAID, especialmente aquelas que permitem as vantagens da segurança do RAID 1 mas que permitam mais discos no sistema.

1.1 - RAID 2

O RAID 2 utiliza os benefícios da redundância de dados ao utilizar a técnica de Hamming. A redundância de dados trata da técnica de se obter, de alguma forma, uma cópia de cada bit armazenado ou uma forma de calcular esse bit.

O RAID 2 surgiu no final dos anos 80, onde os HDs não possuíam checagem de erros. Assim, pode-se dizer que o RAID 2 é similar ao RAID 0, mas possuindo algoritmos de Hamming ECC (Error Correcting Code), que é a informação de controle de erros, no lugar da paridade. Utiliza, no mínimo, dois discos, um de dados e outro de ECC, porém o sistema só se torna eficiente com 4 discos ou mais. Além disso, podem-se ter várias configurações, como 10 discos normais + 4 discos somente para ECC. Este fato possibilita uma proteção adicional, porém o RAID 2 ficou obsoleto pelas novas tecnologias de disco já possuírem este tipo de correção internamente.

O RAID 2 origina uma maior consistência dos dados se houver queda de energia durante a escrita. Baterias de segurança e um encerramento correto podem oferecer os mesmos benefícios.



RAID 2 – Hamming ECC

Observe na imagem acima que os discos 4, 5 e 6 servem de repositório para as informações de Hamming ECC.

02

Veja a seguir quais são as vantagens e desvantagens do Raid 2.

Vantagens	Desvantagens
•Usa Hamming ECC, diminuindo a quase zero as taxas de erro, mesmo com falhas de energia.	•Tecnologia obsoleta. Hoje em dia, há tecnologias melhores para o mesmo fim. •Dependendo da configuração, é necessário a mesma quantidade de discos ECC para discos normais, desperdiçando espaço que poderia ser usado para dados (como veremos logo adiante com técnicas mais modernas de RAID).

03

1.2 - RAID 3

O RAID 3 utiliza a técnica de striping nos discos de dados e um único disco de paridade contando com o controlador de disco para descobrir qual disco falhou. Dessa forma, exige no mínimo três discos.

O RAID 3 é uma versão simplificada do RAID nível 2. Nesse arranjo, um único bit de paridade é computado para cada palavra de dados e escrito em um drive de paridade. À primeira vista, pode parecer que um único bit de paridade dá somente detecção de erro, e não **correção** de erro. Para o caso de erros aleatórios não detectados, essa observação é verdadeira.

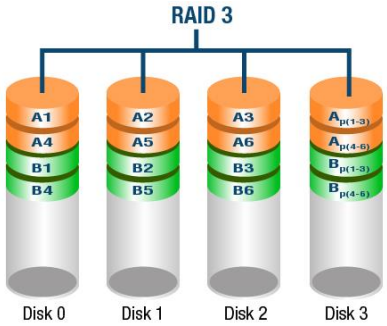
Contudo, para o caso de uma falha de drive, ela provê correção total de erros de um bit, uma vez que a posição do bit defeituoso é conhecida. Se um drive falhar, o controlador apenas finge que todos os seus bits são "zeros". Se uma palavra apresentar erro de paridade, o bit que vem do drive extinto deve ter sido um "um", portanto, é corrigido. Veja um exemplo.

A fim de evitar o atraso em razão da latência rotacional, o RAID 3 exige que todos os eixos das unidades de disco estejam sincronizados. A maioria das unidades de disco mais recentes não possuem a opção de sincronização do eixo, ou se são capazes disto, faltam os conectores necessários, cabos e documentação do fabricante.

Veja um exemplo
Suponha um sistema de 5 discos de dados e um de paridade. Suponha que o quarto disco falhe e a informação consultada no momento seja 011?0, bit de paridade 1. O sistema RAID 2 irá colocar “0” no bit ausente, ficando assim: 011**0**0. Feito isso, é calculado a paridade, que no caso é par, bit 0. Como o bit de paridade é 1, o sistema reconhece que foi um erro colocar o bit 0, e então ajusta o quarto bit para 1, ficando assim: 01110. Dessa forma, consegue-se corrigir o bit ausente.

04

Observe na imagem que o disco 3 serve de repositório para as informações de paridade. O RAID 3 utiliza striping em nível de byte (os sistemas RAID mais modernos utilizam striping em nível de bloco, o que garante maior performance).



RAID 3 – Disco dedicado à Paridade dos bytes.

Na imagem, A1, A2 e A3 são bytes e Ap é a paridade desses bytes. Os discos precisam rodar de maneira sincronizada para que os arquivos possam ser lidos e montados adequadamente (o mesmo vale para a gravação).

05

Veja a seguir quais são as vantagens e desvantagens do Raid 3.

Vantagens	Desvantagens
•Leitura rápida; •Escrita rápida; •Possui controle de erros.	•Montagem difícil via software; •Incompatibilidade com discos diferentes; •Enquanto a reconstrução do disco danificado não é finalizada, o sistema inteiro fica vulnerável. •Atualmente, é considerado obsoleto.

06

1.3 - RAID 4

O RAID 4 funciona com três ou mais discos iguais. Um dos discos guarda a **paridade** (uma forma de soma de segurança) da informação contida nos discos. Se algum dos discos avariar, a paridade pode ser imediatamente utilizada para reconstituir o seu conteúdo. Os discos restantes, usados para

armazenar dados, são configurados para usarem segmentos suficientemente grandes (tamanho medido em blocos) para acomodar um registro inteiro.

Isto permite leituras independentes da informação armazenada, fazendo do RAID 4 um conjunto perfeitamente ajustado para ambientes transacionais que requerem muitas leituras pequenas e simultâneas.

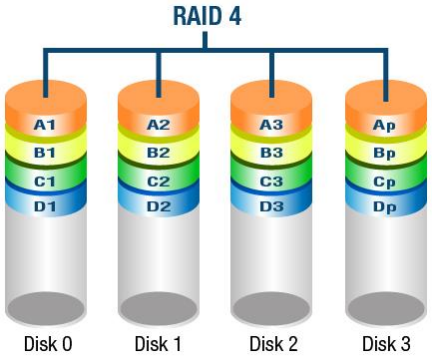
O RAID 4, assim como outros RAIDs, cuja característica é utilizarem paridade, usam um processo de recuperação de dados mais envolvente que conjuntos espelhados, como RAID 1. Este nível também é útil para criar discos virtuais de grande dimensão, pois consegue somar o espaço total oferecido por todos os discos (usando striping), exceto o disco de paridade. O desempenho oferecido é razoável nas operações de leitura, pois podem ser utilizados todos os discos em simultâneo.

Sempre que os dados são escritos no conjunto de disco, as informações são lidas do disco de paridade e um novo dado sobre paridade deve ser escrito para o respectivo disco antes da próxima requisição de escrita ser realizada.

Por causa dessas duas operações de I/O, o disco de paridade é o fator limitante do desempenho total do conjunto. Devido ao fato do disco requerer somente um disco adicional para proteção de dados, este RAID é mais acessível em termos monetários que a implementação do RAID 1.

07

O espaço total do sistema é calculado pelo número total de discos menos 1 (que é o disco de paridade). Exemplo: 11 discos de 1 TB organizados em RAID 4 possuem 10 TB de dados (e 1 TB de paridade). Se o disco de paridade falhar, basta trocar esse disco e esperar a nova carga de bits de paridade. Se um disco de dados falhar, o processo de sincronia é bem mais demorado, pois envolve a consulta a todos os discos e cálculo do bit do disco danificado.



RAID 4 – Disco dedicado à Paridade dos blocos.

Observe na imagem acima que o discos 3 serve de repositório para as informações de paridade. O RAID 4 utiliza striping em nível de bloco (muito mais eficiente que o RAID 3 que usa striping de byte). Na imagem, A1, A2 e A3 são blocos de dados e Ap é a paridade desses blocos. Os discos não precisam rodar de maneira sincronizada, pois os blocos são rearranjados em memória RAM da forma correta.

08

Observe a seguir quais são as vantagens e desvantagens do Raid 4.

Vantagens	Desvantagens
•Taxa de leitura rápida; •Só utiliza um disco para o cálculo de paridade (desperdiça apenas o espaço de um único disco). •Possibilidade do aumento de área de discos físicos (basta acrescentar mais discos ao sistema).	•Taxa de gravação lenta; •Em comparação com o RAID 1, em caso de falha do disco, a reconstrução é difícil, pois o RAID 1 já tem o dado pronto no disco espelhado; •Enquanto a reconstrução do disco danificado não é finalizada, o sistema inteiro fica vulnerável. •Tecnologia obsoleta, não sendo mais utilizada.

09

1.4 - RAID 5

O RAID 5 é, sem dúvida, o modelo mais amplamente utilizado atualmente (não necessariamente o melhor).

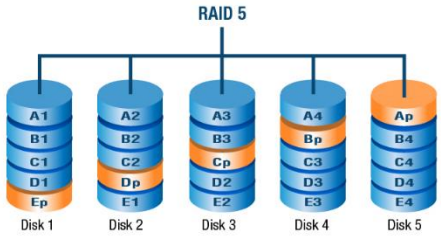
Ele funciona de modo similar ao RAID 4, mas supera alguns dos problemas mais comuns sofridos por esse tipo. As informações sobre paridade para os dados do conjunto de discos são distribuídas ao longo de todos os discos, ao invés de serem armazenadas num disco dedicado, oferecendo assim mais desempenho que o RAID 4, e, simultaneamente, tolerância a falhas. O RAID 5 utiliza, no mínimo, 3 discos.

A quantidade de espaço disponível do sistema é o total de discos menos 1. Deve-se usar discos do mesmo tamanho, do contrário, o sistema se auto ajustará para que cada disco seja utilizado apenas o tamanho do menor disco. Exemplo, se você utilizar dois discos de 1 TB e um disco de 500MB, o conjunto utilizará apenas 500 MB de cada disco, gerando um espaço útil de 1TB e mais 500MB de paridade.

Para aumentar o desempenho de leitura de um RAID 5, o tamanho de cada segmento em que os dados são divididos pode ser optimizado para o número de discos a ser utilizado. O desempenho geral do RAID 5 é equivalente ao do RAID 4, exceto no caso de leituras sequenciais, que reduzem a eficiência dos algoritmos de leitura por causa da distribuição das informações sobre paridade. A informação sobre paridade é distribuída por todos os discos; perdendo-se um, reduz-se a disponibilidade, até à recuperação do disco que falhou. Isto causa degradação do desempenho de leitura e de escrita.

10

Os níveis 4 e 5 usam o striping de dados em nível de bloco, com o nível 5 distribuindo informações de dados e paridade por todos os discos. Se um disco falha, os dados que faltam são calculados com base na paridade disponível dos discos restantes.



RAID 5 – Paridade dos blocos espalhada por todos os discos.

Observe na imagem acima que todos os discos servem de repositório de dados (em azul) e de repositório para as informações de paridade (em laranja). Assim, todos os discos funcionam de maneira similar. O RAID 5 utiliza striping em nível de bloco. Na imagem, A1, A2 e A3 são blocos de dados e Ap é a paridade desses blocos. Os discos não precisam rodar de maneira sincronizada, pois os blocos são rearranjados em memória RAM da forma correta.

11

Veja a seguir quais são as vantagens e desvantagens do Raid 5.

Vantagens	Desvantagens
•Maior rapidez com tratamento de ECC; •Leitura rápida (porém escrita não tão rápida). •Fácil expansão do sistema, basta acrescentar mais discos.	•Sistema complexo de controle dos HDs. •Fica sujeito a falhas enquanto um disco defeituoso não é reconstruído pelo sistema.

12

1.5 - RAID 6

O RAID 6 se aplica ao chamado **esquema de redundância de paridade** para proteger contra até duas falhas de discos usando o espaço de apenas dois discos do conjunto. Esse modelo é similar ao RAID 5, só que agora guarda-se a paridade em dois discos distintos. Para tal, o RAID 6 utiliza um mínimo de 4 discos.

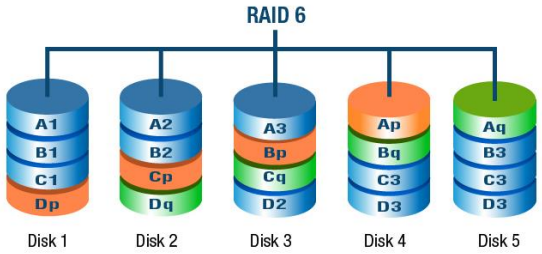
Essa técnica foi criada na ideia de não deixar o sistema de disco vulnerável em nenhum momento. Observe que no sistema RAID 5, no caso de falha de um disco, o sistema ficará vulnerável até que o disco defeituoso seja substituído e reconstruído (o que pode levar horas), se outro disco falhar enquanto a reconstrução não tiver sido completada, todo o conjunto será perdido.

Já no sistema RAID 6, há uma espécie de ***backup* da paridade**, dessa forma, o sistema seria capaz de funcionar com até dois discos defeituosos ao mesmo tempo. Como uma falha de dois discos no mesmo momento é raríssimo de acontecer, enquanto um disco defeituoso estiver sendo substituído e reconstruído, o sistema não estará vulnerável.

A quantidade de espaço disponível do sistema é o total de discos menos 2 (espaço ocupado pelos bits de paridade). Deve-se usar discos do mesmo tamanho, da mesma forma que o RAID 5, a fim de evitar desperdício de espaço.

13

Observe na imagem que todos os discos servem de repositório de dados (em azul) e de repositório para as informações de paridade (em laranja e verde).



RAID 6 – Paridade dos blocos espalhada por todos os discos.

Observe também os dois conjuntos de paridade, um em verde e outro em laranja. Assim, todos os discos funcionam de maneira similar. O RAID 6 utiliza striping em nível de bloco. Na imagem, A1, A2 e A3 são blocos de dados e Ap e Aq representam a paridade desses blocos. Os discos não precisam rodar de maneira sincronizada, pois os blocos são rearranjados em memória RAM da forma correta.

14

Observe a seguir quais são as vantagens e desvantagens do Raid 6.

Vantagens	Desvantagens
•Possibilidade falhar 2 discos ao mesmo tempo sem perdas. •Não fica vulnerável no tempo de reconstrução de um disco.	•Precisa de dois discos a mais no conjunto por causa dos dois conjuntos de paridade; •Escrita lenta; •Sistema complexo de controle dos HDs.

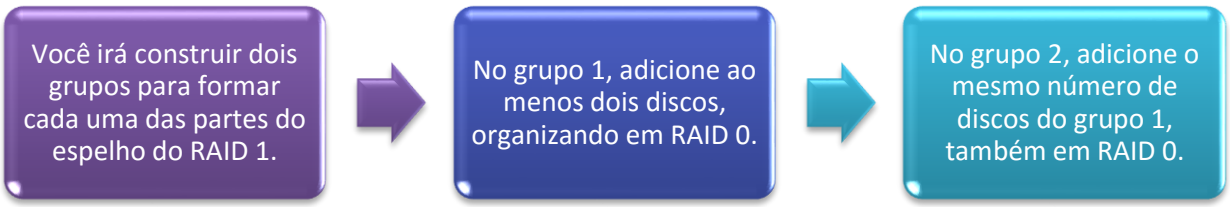
15

1.6 - RAID 01 ou RAID 0+1

O RAID 0 + 1 é uma combinação dos níveis 0 (Striping) e 1 (Espelhamento), onde os dados são divididos entre os discos para melhorar o rendimento, mas também utilizam outros discos para duplicar as informações. Assim, é possível utilizar o bom rendimento do nível 0 com a redundância do nível 1.

No entanto, são necessários pelo menos 4 discos para montar um RAID desse tipo. O aumento desses discos nesse sistema deve ser feito sempre de 2 em 2 discos. Tais características fazem do RAID 0 + 1 o mais rápido e seguro, porém o mais caro de ser implantado. No RAID 0+1, se um dos discos vier a falhar, o sistema vira um RAID 0. O espaço em disco útil corresponde a 50% do somatório dos discos, visto que há o espelhamento dos dados.

Para montar um RAID 01, devem ser seguidos os passos abaixo:



Pronto, você já tem um sistema em RAID 01. Para expandir o sistema, adicione um disco em cada um dos grupos, portanto, é possível montar um RAID 01 com 6, 8, 10, 12 discos etc.

16

Na imagem note que os discos espelhos estão afastados um do outro, o que pode ser gerenciado por controladoras diferentes. Se um disco falhar, o sistema permanece funcionando (mas sem tolerância a outra falha e sem o ganho de velocidade).

Entretanto, se dois discos que possuam os mesmos dados falharem ao mesmo tempo (como por exemplo, o primeiro e o terceiro disco da imagem), haverá perda de dados.

RAID 01 – Striping e Espelhamento.

17

Perceba a seguir as vantagens e desvantagens do Raid 01.

Vantagens	Desvantagens
•Segurança contra perda de dados; •Pode falhar 1 dos HD's, ou os dois HD's do mesmo todos os discos do mesmo grupo.	•Alto custo de implantação (o espaço útil é de apenas 50% do espaço total do sistema). •Alto custo de expansão de hardware (sempre de dois em dois discos); •Os discos devem ficar em sincronismo de velocidade para obter a máxima performance.

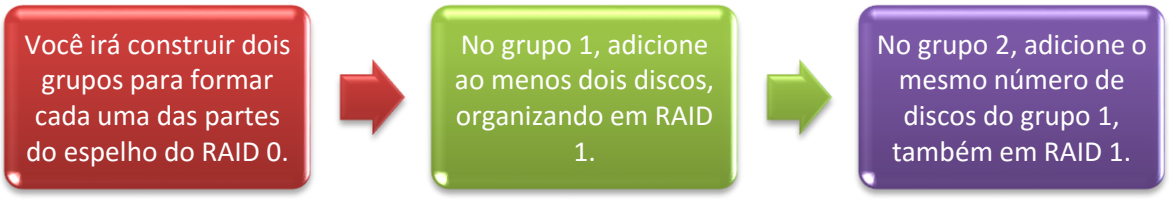
18

1.7 - RAID 10 ou 1+0

O RAID 1+0, ou 10, é algo parecido com o RAID 01. Também exige ao menos 4 discos rígidos, e depois múltiplos de 4 para crescimento do conjunto. Cada par será espelhado, garantindo redundância, e os pares serão distribuídos, melhorando desempenho.

Até metade dos discos pode falhar simultaneamente, sem colocar o conjunto a perder, desde que não falhem os dois discos de lados opostos do espelhamento. É o nível recomendado para bases de dados, por ser o mais seguro e dos mais velozes, assim como qualquer outro uso onde a necessidade de economia não se sobreponha à segurança e desempenho.

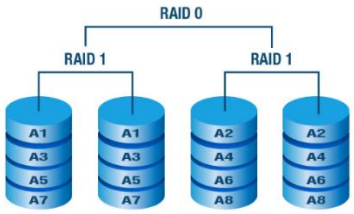
Para montar um RAID 10, devem ser adotados os seguintes passos:



Pronto, você já tem um sistema em RAID 10. Para expandir o sistema, adicione dois discos em cada um dos grupos, portanto, é possível montar um RAID 10 com 4, 8, 12, 16 discos etc.

19

Na imagem, note que os discos espelhos estão juntos um do outro, normalmente gerenciados pela mesma controladora. Se um disco falhar, o sistema permanece funcionando (mas sem tolerância a outra falha e sem o ganho de velocidade), entretanto, se dois discos que possuam os mesmos dados falharem ao mesmo tempo (como por exemplo, o primeiro e o terceiro disco da imagem), haverá perda de dados.



RAID 01 – Espelhamento e Striping.

20

Perceba a seguir as vantagens e desvantagens do Raid 10.

Vantagens	Desvantagens
•Segurança contra perda de dados; •Pode falhar um ou vários dos HDs ao mesmo tempo, dependendo de qual avaria.	•Alto custo de expansão de *hardware* (de quatro em quatro unidades); •Os discos devem ficar em sincronismo de velocidade para obter a máxima performance.

A diferença básica entre o RAID 01 e o RAID 10 está no fato de que na perda de um segundo disco, o RAID 01 ficaria em desvantagem em relação ao RAID 10; você poderia perder até metade dos discos que o sistema continuaria funcionando no RAID 10. Outra diferença seria a velocidade de recuperação, no caso do RAID 10 será necessário apenas sincronizar com o disco espelho, já no RAID 01 será necessário espelhar todo o conjunto segmentado.

Atualmente, para sistemas de bancos de dados as duas **recomendações** são:

- Se o dinheiro não é problema, utilize RAID 10, é o mais rápido de todos;
- Se o dinheiro é limitado, utilize o RAID 5 ou RAID 6, é razoavelmente rápido e bem seguro.

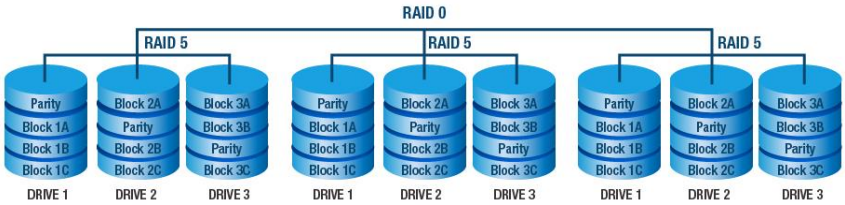
21

1.8 - RAID 50

O RAID 50, assim como os próximos modelos que veremos, são pouco comuns. Entretanto, apresentam boas ideias de segurança e performance.

O RAID 50 é uma combinação híbrida que usa as técnicas de RAID com paridade em conjunção com a segmentação de dados. Um arranjo RAID 50 é essencialmente um arranjo com as informações segmentadas através de dois ou mais arranjos.

A ideia principal é utilizar a segurança do modelo RAID 5 aliado com a performance de leitura do RAID 0.



RAID 50 – Vários RAID 5 em striping para aumento de performance.

22

Note a seguir as vantagens e desvantagens do Raid 50.

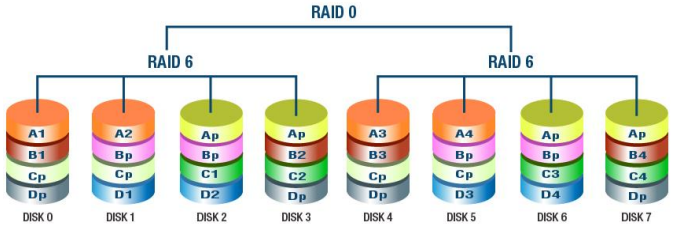
Vantagens	Desvantagens
•Alta taxa de transferência; •Ótimo para uso em servidores.	•Alto custo de implementação e expansão de memória.

23

1.9 - RAID 60

Assim como o RAID 6 propõe a utilização de dois discos de paridade a fim de se evitar o momento de risco enquanto um disco defeituoso é reconstruído, o RAID 60 é a combinação de várias estruturas RAID 6 espelhadas para aumento de performance.

Possui as mesmas vantagens e desvantagens do RAID 50, sendo pouquíssimo usado em ambientes reais.



RAID 60 – Vários RAID 6 espelhados para aumento de performance.

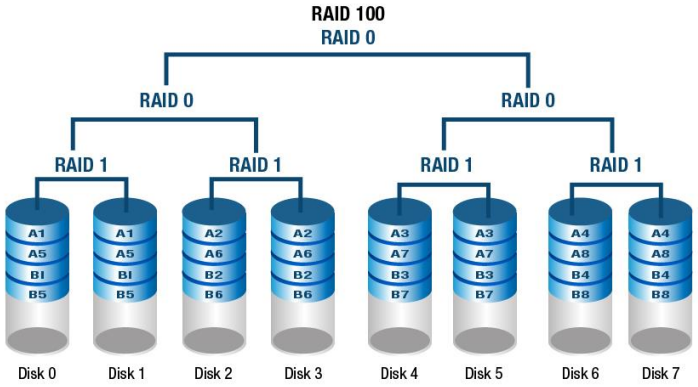
24

1.10 - RAID 100

O RAID 100 basicamente é composto do RAID 10+0. Normalmente ele é implementado utilizando uma combinação de software e *hardware*, ou seja, implementa-se o RAID 0 via software sobre o RAID 10 via *hardware*.

O RAID 10 é muito usado atualmente para sistemas de banco de dados. O objetivo dos RAID 50, 60 e 100 é o aumento da performance de leitura. No caso específico do RAID 100, mantém-se os objetivos e benefícios do RAID 10 aliados ao ganho de performance do espelhamento (RAID 0).

Assim como os modelos RAID 50 e 100, o RAID 100 é muito raro de ser encontrado no mundo real. Devido à necessidade de sincronização, velocidade, capacidade e compatibilidade entre os discos, esses modelos são muito caros e difíceis de serem implementados. Enquanto que na teoria podemos observar vantagens nesses modelos complexos, o mundo real tende a repudiar soluções extremamente caras e complexas.



RAID 100 – um espelhamento de vários RAID 10.

25

2 - COMO IMPLEMENTAR UM RAID EM UMA EMPRESA

A implementação real de um RAID é uma decisão estratégica da organização, não é um profissional ou outro quem vai decidir pela compra de discos e a implantação do RAID. Normalmente, há um grupo de profissionais (entre elas o DBA e o administrador de redes) que propõe uma ou outra arquitetura RAID como a mais indicada.

Depois disso, o responsável por aquisições poderá aprovar a compra e assim a empresa terá acesso à tecnologia.

A instalação e configuração do RAID não é trivial. O computador que gerenciará o RAID precisa ser compatível com tal tecnologia. Além disso, é necessário decidir por um bom modelo de implementação, pois há placas mães mais simples que, apesar de suportarem modelos RAID, em caso de falha da placa mãe, pode ser que você não consiga recuperar os seus discos em outra placa mãe. Isso, por exemplo, é mais comum em RAID 0, onde a tecnologia usada para espelhamento pode não ser compatível entre placas mães diferentes.

Há ainda equipamentos específicos para implementar soluções de RAID, como servidores NAS e SAS (pesquise sobre elas na Web).

Por fim, ao montar um ambiente RAID, é necessário formatar e preparar os discos, dessa forma, podem ser necessários vários dias até que a solução esteja configurada completamente.

26

3 - TESTANDO RAIDS TOLERANTE A FALHAS

Outro aspecto fundamental é realizar **testes periódicos** de sistemas RAIDs tolerantes a falha. Esses testes confirmarão se o seu ambiente realmente é seguro.

Para fazer tais testes, escolha janelas de tempo onde os sistemas poderão eventualmente ficar fora do ar, como finais de semana ou durante a madrugada. Um teste inicial logo após a configuração pode ser facilmente realizado, mas testes em ambiente de produção real pode ser bem mais complicado. O ideal seria que a empresa realizasse ao menos dois testes por ano. Isso daria um bom controle do ambiente tolerante a falhas.

Para realizar o teste é bem simples, acha o computador que gerencia os RAID e desconecte o cabo de energia de um dos discos. Verifique se o sistema continua acessando normalmente os arquivos. Se a solução for, por exemplo, um RAID 6, desconecte dois discos e abra o gerenciador do RAID para ver os informes. Atenção: esse teste é para apenas RAIDs tolerantes a falhas, RAIDs como RAID 0 que provê apenas mais espaço não é tolerante a falhas.

27

RESUMO

Neste módulo, aprendemos que:

a) O RAID 2 utiliza a técnica de hamming para criar um ambiente tolerante a falhas, entretanto esse modelo encontra-se tecnologicamente ultrapassado.
b) O RAID 3 utiliza as técnicas de striping e paridade. Um disco é colocado especificamente para armazenar os bits de paridade.

c) O RAID 4 utiliza três ou mais discos iguais, um disco guarda a paridade, e os demais dados. Em relação ao RAID 3, a principal diferença é que o RAID 4 efetua paridade de blocos e não de bytes.

d) O RAID 5 é um dos modelos mais utilizados pelas grandes empresas. Pois oferece tolerância a falhas e possui boa performance. Nele, o bit de paridade é distribuído por todos os discos. Ou seja, todos os discos armazenam dados e uma fração de paridade.

e) O RAID 6 é um modelo semelhante ao modelo RAID 5, utiliza 2 bits de paridade, o que garante um sistema tolerante a falhar mesmo com um disco defeituoso no sistema.

f) O RAID 01 (ou 0+1) combina as técnicas de RAID 0 e RAID 1. Enquanto o modelo RAID 0 só permite 2 discos, o modelo RAID 01, permite 4 ou mais discos (sempre múltiplos de 2). Esse

sistema oferece tolerância a falhas e eficiência, porém é o mais caro de todos. O espaço em disco útil corresponde a 50% do total de discos.

g) O RAID 10 (ou 1+0) combina as técnicas de RAID 0 e RAID 1 de uma forma mais interessante que o RAID 01. Enquanto o modelo RAID 01 permite crescimento de 2 discos em 2 discos, o modelo RAID 10, só pode crescer de 4 em 4 discos. O espaço em disco útil corresponde a 50% do total de discos.

h) O RAID 50 combina um RAID 5 com o striping do RAID 0, permitindo um sistema tolerante a falhas e de alta performance.

i) O RAID 60 equivale a um RAID 6 com o striping do RAID 0.

j) O RAID 100 equivale a um RAID 10 com um striping do RAID 0.

UNIDADE 1 – ESTRUTURA DE ARQUIVOS E PROCESSAMENTO DE CONSULTAS
MÓDULO 4 – OTIMIZANDO AS OPERAÇÕES DE DADOS.

01

1 - ESTRUTURAS FÍSICAS DE ARQUIVOS

Vimos anteriormente como a tecnologia RAID pode nos ajudar na construção de espaços de memória em disco que sejam **eficientes** (rápidas) e **confiáveis** (tolerante a falhas). O objetivo de se escolher uma arquitetura RAID que proveja esses dois benefícios é orientado de forma ao SGBD prover a melhor performance possível para os usuários nas operações de consultas e atualização de dados.

Já estudamos como os modelos RAID podem ser configurados para pequenos, médios e grandes repositórios de dados, especialmente os modelos RAID 0, 1, 5, 6, 10, 50, 60 e 100 (que são os modelos mais utilizados atualmente). Partindo do pressuposto que no nível físico da tecnologia já podemos configurar os discos rígidos de forma a se obter o melhor desempenho e confiabilidade possível, agora iremos apresentar algumas técnicas e tecnologias de como as informações devem ser armazenadas tanto em disco quanto em memória RAM a fim de se maximizar todas as questões que possam beneficiar as operações de consulta e atualização de dados.

Os SGBDs utilizam vários tipos de arquivos para operações distintas, vamos ver nesta etapa do nosso estudo como organizar os arquivos em áreas separadas para obter maior performance do SGBD. A ideia principal é **diminuir a concorrência de acesso ao disco**, como veremos a seguir.

02

1.1 - Separação física da área de dados

O disco rígido executa várias operações de leitura e gravação ao mesmo tempo. Essas operações são necessidades concorrentes que podem estar acontecendo ao mesmo tempo. Para poder resolver todas

as solicitações de acesso a disco, o sistema operacional enfileira essas solicitações, tradando-as uma a uma sequencialmente.

São exemplos de tipos dessas operações de acesso a disco:

- arquivos que o sistema operacional está consultando,
- arquivos que o SGBD pode estar solicitando,
- dados do banco de dados que estão sendo lidos e armazenados.

Para eliminar essa concorrência de acesso ao disco, o ideal seria que o disco (ou o conjunto de discos) especificado para o repositório dos bancos de dados seja **dedicado**, isto é, seja **exclusivamente utilizado para armazenar dados**.

Essa solução implica, por exemplo, em montar um ambiente físico para dados de banco de dados e outra estrutura de discos para os arquivos do sistema operacional e aplicativos.

03

Muitas organizações de grande porte usam até mesmo tecnologias de RAID diferentes para essas necessidades, por exemplo, dois discos em RAID 1 para o sistema operacional e aplicativos e vários discos em RAID 5 para os bancos de dados.

Separação dos bancos de dados em discos distintos. Cada círculo da direita representa um disco distinto

04

1.2 - Separação física da área de dados da área de log (transacional)

Os SGBDs modernos são **transacionais**, ou seja, eles armazenam todas as operações realizadas nos bancos de dados (transações) onde é possível, entre outras funcionalidades, desfazer operações feitas ou voltar o banco de dados a um exato momento no tempo (técnica chamada de *rollback*).

Para tal funcionalidade, os SGBDs utilizam ao menos dois arquivos:

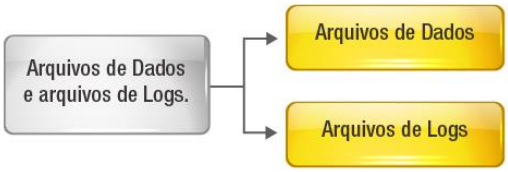
- um **arquivo de dados**, onde ficam armazenadas as tabelas, registros, e demais elementos do banco de dados;

- um **arquivo de log**, onde as transações ficam armazenadas.

O arquivo de dados sofre operações de consulta e alterações, enquanto que o arquivo de log sofre, geralmente, operações de inserção de dados. Ainda, algumas rotinas de *backup* podem "limpar" o arquivo de transações.

Dessa forma, para obter máxima performance do banco de dados, também é interessante que esses arquivos (dados e transações) fiquem em **áreas físicas distintas** (e provavelmente em sistemas RAIDs diferentes).

Ao separar os arquivos, as controladoras e os discos passam a ficar mais dedicados a um objetivo, aumentando a performance geral do sistema.



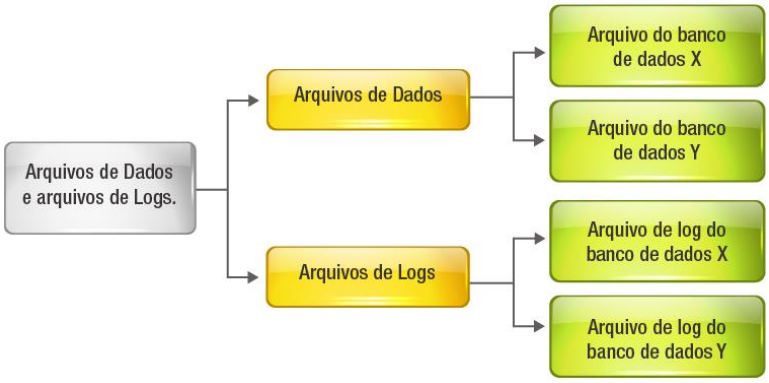
Separação de dados e logs em discos distintos. Cada círculo da direita representa um disco distinto.

05

1.3 - Separação física de bancos de dados

Muitas vezes um SGBD de uma organização pode gerenciar vários bancos de dados. É possível que, em um determinado contexto, um ou mais bancos de dados deveriam ter maior prioridade de processamento.

Veja um exemplo.



Separação dos arquivos de dados e logs em discos distintos. Cada círculo da extremidade da direita representa um disco distinto.

Exemplo

Suponha que uma organização possua os seguintes bancos de dados: PRODUTOS, RH, FOLHA_DE_PAGAMENTO, PATRIMONIO. Desses bancos, suponha que ela queira priorizar o banco de dados de PRODUTOS, para que o sistema de e-Commerce funcione em máxima performance. Isso poderia ser feito por várias ações estruturais da arquitetura de sistemas de informação, entre elas, a separação deste banco de dados dos demais. A organização poderia, por exemplo, criar uma área de discos exclusiva para o banco de dados de PRODUTOS e outra área de disco para os demais bancos de dados. Separando assim, não haverá concorrência de acesso a disco entre o banco de dados de PRODUTOS e os demais bancos.

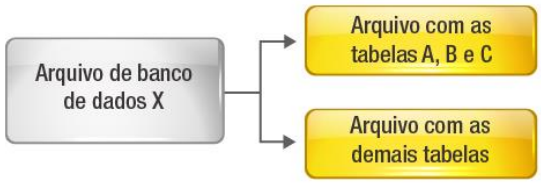
06

1.4 - Separação física de tabelas

Em um nível mais preciso de otimização, poderíamos ter um modelo de dados em que uma ou mais tabelas precisariam ter uma prioridade de processamento ou que essas tabelas, em relação às demais, teriam uma concorrência alta com muitos acessos simultâneos. Da mesma forma que separamos os arquivos do SGBD e depois os bancos de dados em áreas físicas distintas, também é possível separar tabelas de um banco de dados em áreas físicas separadas.

Conforme falamos, um banco de dados normalmente possui dois arquivos, um para dados e outro para transações. Porém essa configuração padrão normalmente pode ser alterada para vários arquivos de dados e/ou vários arquivos de transações. A fim de se obter uma maior performance para um grupo de tabelas específicas, um banco de dados pode conter, por exemplo, dois arquivos de dados, e cada arquivo de dados ser armazenado em uma área física distinta.

Veja um exemplo prático.



Separação das tabelas em arquivos de dados localizados em discos distintos. Cada círculo da extremidade da direita representa um disco distinto.

exemplo prático
Suponha que uma organização utiliza um sistema e-Commerce que possui várias tabelas, entre elas, tabelas de produtos, de clientes e de vendas. Suponha que a funcionalidade de consulta de produtos esteja um pouco lenta (pois ela é executada por centenas de potenciais clientes ao mesmo tempo que acessam o sistema pela Internet). Uma das formas de se aumentar a performance desse sistema seria separar a tabela de produtos em um arquivo (dataset) separado, que seria armazenado, por exemplo, em um sistema RAID à parte. Dessa forma, o banco de dados do sistema e-Commerce poderia ter duas áreas de dados, uma para as tabelas de produtos e outra para as demais tabelas. Isolando as tabelas de dados em um conjunto de discos distinto das demais tabelas auxiliaria na

otimização da performance do sistema, já que não haveria concorrência entre as operações de consulta de produtos e as operações sobre as demais tabelas.

07

1.5 - Conclusão

Como vimos, há diversas formas de se otimizar uma arquitetura de discos que atende a um SGBD. Cabe ao DBA identificar a estrutura física adequada, em consonância com o orçamento de TI da empresa.

A forma mais correta de se aprimorar uma arquitetura de discos é a forma “evolucional”, na qual os gargalos vão sendo identificados à medida que a utilização do banco de dados aumenta.

Dessa forma, apesar de reativa, o DBA possuirá informações estatísticas de acesso e de gargalos, provendo uma solução mais adequada à necessidade.

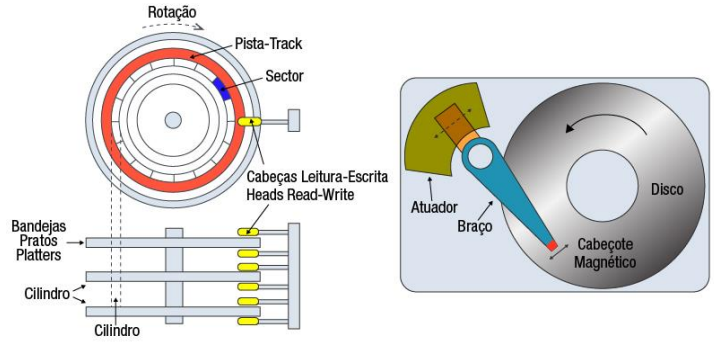
08

2 - Formas de acesso ao disco

Outra ideia de otimização está relacionada com o tempo de acesso ao disco. Veremos aqui algumas técnicas de gravação de dados que otimizam o tempo de uso do disco rígido.

As operações de consulta e gravação de dados são, na maioria das vezes, compostas de pequenos registros de dados. Já aprendemos que o principal gargalo para a leitura e gravação de dados são os discos rígidos, pois, comparados com a velocidade da memória RAM, do barramento interno e do processador, a velocidade de transferência de dados dos discos rígidos é mais de 1.000 vezes mais lenta do que a velocidade dos demais componentes citados.

O disco rígido é uma peça mecânica, ele utiliza dois motores internos que permitem a leitura e gravação dos dados. Um dos motores faz os discos internos girarem enquanto o outro motor faz a cabeça magnética mover-se para a frente e para trás, posicionando o braço da cabeça magnética nos exatos setores onde as informações são lidas e gravadas.



Diagramas exemplificando o funcionamento do disco rígido

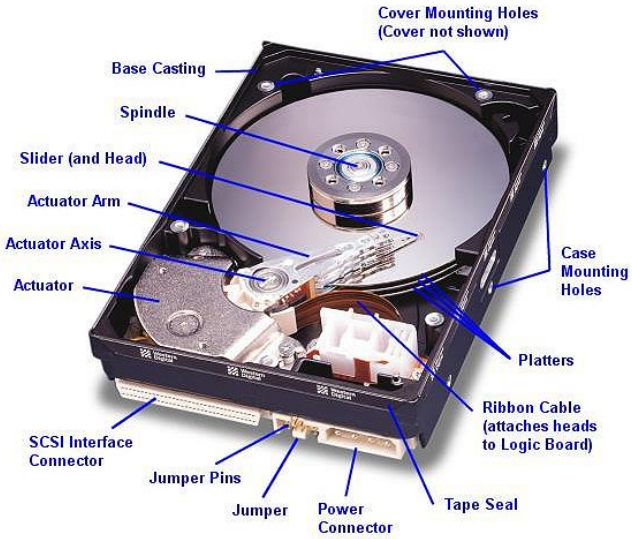


Imagem real de um disco rígido

09

O tempo que o disco rígido gasta para posicionar a cabeça de leitura é um tempo inútil para fins de transferência de dados, pois nenhum trabalho efetivo de transferência de dados está sendo feito nesse momento.

Se pudéssemos eliminar esse tempo de posicionamento da cabeça magnética, conseguiríamos um aumento de performance muito grande, pois garantiríamos que o disco rígido estaria sempre pronto/posicionado para ler ou gravar algo.

Infelizmente, ainda não é possível eliminar o tempo de posicionamento da cabeça magnética, mas é possível minimizar esse tempo. As técnicas descritas aqui pretendem minimizar esse tempo de posicionamento da cabeça magnética.

Atualmente há uma nova tecnologia de armazenamento de dados denominada SSD (discos de estado sólido), que são equipamentos que utilizam chips de memória semelhantes a cartões de memória que utilizamos em *smartphones* e pendrives.

O contexto apresentado neste módulo está focado na tecnologia de discos magnéticos, que ainda é largamente utilizada.

10

2.1 - Gravação em bloco

A ideia por trás da gravação em bloco é bem simples: ao invés de mandar vários registros (poucos bytes) para serem gravados no disco rígido (o que exigiria vários reposicionamentos da cabeça magnética do disco), devem ser juntados em um grande bloco e enviados como uma única operação (diminuindo a quantidade de operações de reposicionamento da cabeça, e, consequentemente, reduzindo o tempo de concorrência entre outras operações).

Vamos a um exemplo para ver como isso se daria. Suponha que você tenha os seguintes dados de performance de um disco rígido:

- tempo médio de posicionamento da cabeça (também denominado de tempo médio de acesso) como sendo de 5ms (milissegundos),
- taxa de gravação igual a 10 MB/s (mega bytes por segundo).

Vamos supor agora que temos que armazenar no disco 1.000 blocos de 10Kbytes de informação. Qual seria a duração desse processo?

Primeiro vamos computar o **tempo de duração de um bloco de 10Kb**.

Depois vamos calcular **quanto tempo leva para armazenar 1.000 blocos**.

tempo de duração de um bloco de 10Kb
Para computar o tempo de duração de um bloco de 10Kb usaremos a regra de três: se 10MB (10.000Kb) leva um segundo, então 10Kb leva X segundos: 10.000/1 = 10/x ➔ x = 1/1000 = 0,001 segundo ou 1 ms (milissegundo). Então já sabemos que o tempo que cada bloco leva para ser gravado é de 1 ms.

quanto tempo leva para armazenar 1.000 blocos
Agora vamos computar quanto tempo leva para armazenar 1.000 blocos. Esse tempo (T) é composto pelo tempo de acesso (Ta) acrescido pelo tempo de gravação (Tg) e o resultado deve ser multiplicado pelo número de blocos (N): T = (Ta+Tg) x N ➔ T = (5+1) x 1.000 = 6.000 ms = 6 s.

Resposta: o tempo que o disco leva para gravar os 1.000 blocos de 10Kb é de 6 segundos.

	1 segundo	2 segundos	3 segundos	4 segundos	5 segundos	6 segundos
Nossa sequência de 1.000 blocos	167 blocos	167 blocos	167 blocos	167 blocos	167 blocos	167 blocos

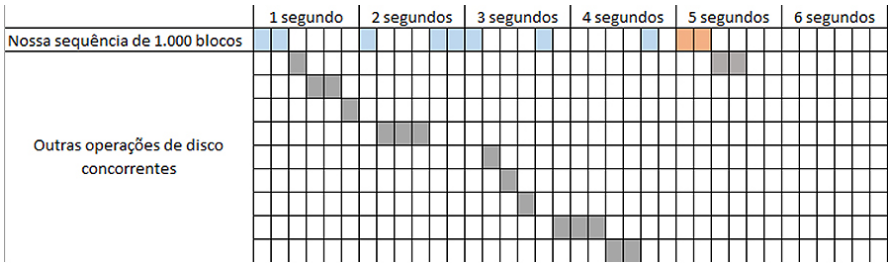
Tempo teórico para gravar os 1.000 blocos, em cada segundo grava-se aproximadamente 167 blocos.

11

O tempo obtido nos cálculos (6 segundos) é o tempo que qualquer operação de gravação leva para ser realizada, dessa forma, se o disco rígido não funciona de maneira dedicada, ele tem outros dados para serem lidos e gravados também.

Consequentemente, o tempo de gravação dos nossos dados concorre com o tempo de gravação dos demais dados, e, quando cronometrada em um relógio, o tempo real gasto desde a gravação do primeiro bloco até a gravação do último bloco dos nossos dados será bem mais do que seis segundos.

Dependendo da concorrência com outras operações, poderia levar 10x, 100x, 1.000x mais tempo!



Tempo real dada a concorrência com outras operações.

12

Uma das formas de se aumentar a performance de gravação, conforme já vimos, é **dedicar os discos** às operações de banco de dados.

Outra forma de otimização é forçar a gravação em um único bloco de dados. Ao invés de gravar 1.000 blocos de 10 Kb (onde cada um é localizado e gravado de cada vez), a ideia aqui seria gravar, por exemplo, todo o conjunto de dados de uma só vez, com isso, nossas informações seriam tratadas como um único bloco e, portanto, armazenadas sem a fragmentação do tempo de concorrência no disco rígido.

Veja como seria uma representação disso:

Observe que nossa sequência de dados foi armazenada de uma única vez.

13

2.2 - Leitura e gravação sequencial

Uma outra forma de diminuir o tempo de leitura e gravação é garantir que os dados sejam gravados, e posteriormente lidos, de maneira sequencial. Dessa forma, o tempo de localização do setor pela cabeça magnética diminui muito, tendendo a ser quase zero segundo.

Vamos explicar isso.

Normalmente, as operações de leitura e gravação são aleatórias, ora o dado está posicionado na parte central do disco, ora está posicionado no extremo externo e ora está localizado no extremo interno. Essa movimentação da cabeça magnética, entre os extremos, é muito distante e por isso consome muito tempo. Se diminuirmos as distâncias da movimentação da cabeça, diminuiremos, consequentemente o tempo de acesso.

Vamos a um exemplo prático. A largura de um disco rígido é de aproximadamente 4 cm. A cabeça magnética percorre esses 4 cm em aproximadamente 4 ms. A distância entre um setor e outro de um disco de 1 TB é de aproximadamente 1 nm (nanômetro), ou seja, 10.000 vezes menor que um centímetro.

Em uma regra de três simples, o tempo que a cabeça magnética leva para ser reposicionada para o setor ao lado da posição atual é de 4 ms / 10.000 = 0,004 ms, ou seja, quase zero (ou mil vezes mais rápido do que o movimento de um extremo ao outro do disco).

Vimos no exemplo anterior que 1.000 blocos levavam 6 segundos para serem armazenados. Que tal agora recalcular a mesma fórmula para dados sequencialmente gravados, onde o tempo de reposicionamento é de apenas 0,004 ms?

Veja aqui a resposta.

Veja aqui a resposta
A nova equação seria: T = (Ta+Tg) x N ➔ T = (0,004+1) x 1.000 = 1.004 ms, praticamente 1 segundo. Ou seja, sendo os dados sequencialmente gravados teríamos uma performance 6x mais rápida. Um ganho muito grande em termos de uso de disco.

14

Para implementar a **gravação sequencial** você precisa de:

- Uma grande massa de dados a ser armazenada, o que geralmente seria um conjunto de muitos registros ou um registro com dados muito grandes (BLOB), como imagens ou arquivos multimídia.
- No caso de vários registros sendo armazenados, a estrutura da tabela precisará de um índice sequencial, para que os registros sejam salvos na ordem desejada para o momento de leitura. Posteriormente, a leitura só poderá ser sequencial se obedecer a mesma ordem em que os registros forma salvos.
- No caso de dados BLOB, precisamos de uma área de disco desfragmentada e pré-alocada, o que pode ser feito no momento da criação do espaço em disco a ser utilizado pelo banco de dados. Podemos, por exemplo, ao criar um banco de dados, pré-alocar um espaço em disco de 500 MB para o banco, garantindo assim que o espaço não seja fragmentado.

Para implementar a **leitura sequencial** você precisa de:

- Ler uma grande massa de dados armazenada (e não apenas poucos bytes de informação), o que pode ser necessário em:

a) relatórios que consomem muitos dados;

b) leitura de dados BLOB.

- Para o primeiro caso, relatório que consomem muito dados, é importante que as consultas SELECT sejam otimizadas a lerem os dados na sequência em que foram armazenadas. Daí a importância de que o índice principal da tabela, seja aquele que é utilizado por essas consultas. Muitas vezes, esses índices referem ao ID da tabela, ou a um determinado campo, como data de cadastro, valor, quantidade, classificação alfabética de um campo de nome etc.

Hoje em dia os SGBDs são capazes de ler dados sequenciais e depois organizá-los em memória na ordem de classificação desejada. Dessa forma, independentemente da ordem de consulta desejada pelo usuário, sempre será possível ler os dados de maneira sequencial. Veja um exemplo.

Veja um exemplo
Os dados de das vendas realizadas por uma empresa podem estar gravados sequencialmente pelo ID da venda (o que acaba coincidindo com a ordem cronológica), independentemente se queremos consultar essas vendas, classificando-as por valor da venda, data da venda, vendedor, cliente etc., sempre o SGBD poderá fazer a leitura sequencial dos dados e organizá-los em memória RAM conforme a necessidade dos usuários.

15

3 - BUFFER

3.1 - Cache de leitura (com otimização por estatísticas)

Uma das funcionalidades presentes em vários SGBDs refere-se à memória cache com utilização de estatísticas de acesso a dados.

Aparentemente, as operações de consulta e atualização de dados sobre um banco de dados é aleatória: qualquer dado tem a mesma probabilidade de ser escolhido para uma operação de consulta ou gravação. Entretanto, é notado que há tabelas e/ou registros que são mais comumente acessados pelos sistemas de informação. O gerenciamento de **estatísticas de acesso** registra quais tabelas e quais registros são mais frequentemente consultados.

A ideia por traz dessa funcionalidade é tentar manter em memória RAM (que é mais de 1.000 vezes mais rápida que os discos rígidos) essas informações frequentemente acessadas. Dessa forma, quando o sistema de informação solicita tal conjunto de registros, ao invés de eles serem consultados do disco rígido onde estão armazenados, eles são consultados da memória RAM, gerando uma resposta muito mais rápida. O SGBD utiliza cálculos de probabilidade para aumentar as chances de que as informações desejadas estejam previamente mantidas em memória RAM.

O gerenciador de estatísticas diz qual informação é mais utilizada, mas é o **controlador de cache de leitura** o responsável por manter as informações nessa parte da memória RAM. Quanto maior a área de cache, mais informação pode ser mantida em memória, porém, menos espaço haverá para as outras operações do banco de dados e do próprio sistema operacional.

Normalmente há duas formas de se gerenciar esse espaço de cache:

- **automática**, baseada em estatísticas de uso que atende à 99% dos casos,
- **manual**, onde o DBA pode configurar o tamanho em bytes da área de cache.

16

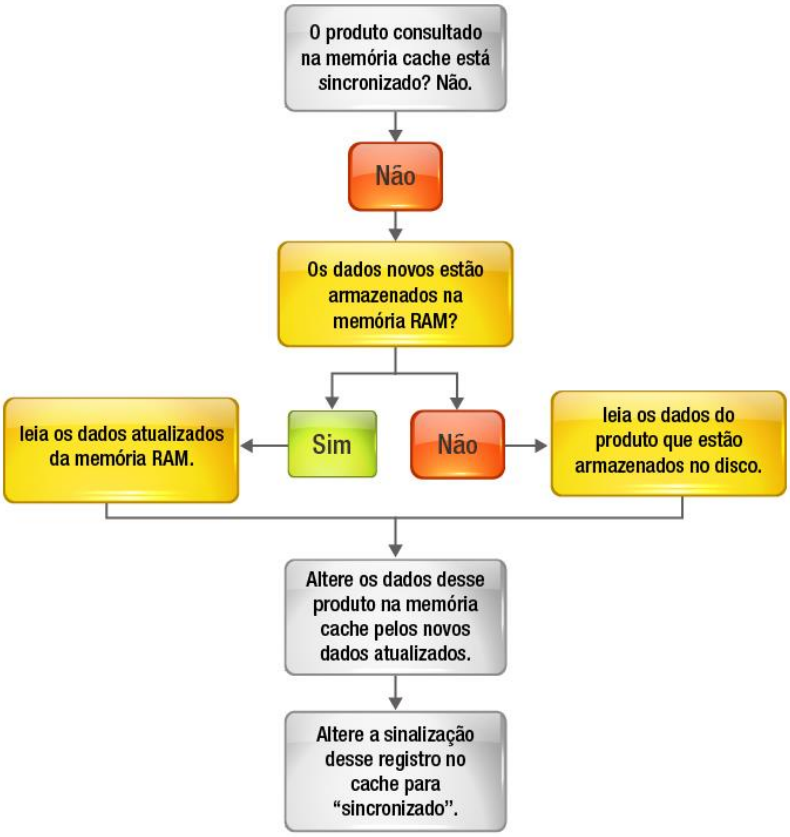
O gerenciador do cache também precisa realizar uma atividade importantíssima: **assegurar que as informações presentes no cache são exatamente as mesmas presentes no disco**. Normalmente, o processador do SGBD tem acesso à quais dados estão armazenados no cache, e sempre que um dado é atualizado em disco, ele sinaliza à área de cache que determinado dado foi alterado. Se essa informação alterada é uma informação que estava na área de cache, então o cache atualiza com a nova informação, garantindo a sincronia entre disco e área de cache. Exemplo

Vamos ver, na prática, como isso funciona.
Suponha que sua organização possua um sistema e-Commerce e que várias pessoas normalmente consultam vários produtos a todo momento. O gerenciador de cache poderia, por exemplo, armazenar toda a tabela de produtos na memória cache, dessa forma, cada vez que um novo potencial cliente consulta um produto, ao invés de buscar os dados desse produto no disco rígido, o SGBD informaria os dados que já estão no cache.
Suponha agora que um operador do sistema na organização altere o preço de um determinado produto. Ao realizar a operação de UPDATE, o processador do SGBD iria informar ao gerenciador de cache que os dados daquele produto foram mudados, e com isso aquele registro ficaria sinalizado como não sincronizado. Note que todos os demais produtos estão sincronizados na memória cache, apenas um produto não está. O gerenciador da cache continua a trabalhar normalmente, oferecendo as informações dos produtos que estão sincronizados no cache para os usuários.

Exemplo
Suponha que no disco exista uma informação A e que o cache também possua essa informação A. Ao alterar a informação do disco de A para B, o gerenciador de cache detecta a mudança e faz nova leitura para atualizar o valor da cache de A para B também, sincronizando ambas informações. Entretanto, se o dado B estiver em memória RAM, o controlador do cache não lê o dado do disco, mas sim da própria memória RAM, evitando assim um acesso a disco desnecessário.

17

Entretanto, se um usuário do sistema consultar exatamente aquele produto que teve o preço alterado, o gerenciamento do cache trabalhará assim:



18

3.2 - Buffer de gravação

O buffer de gravação tem por objetivo diminuir o número de operações pequenas de acesso ao disco para atualização de dados.

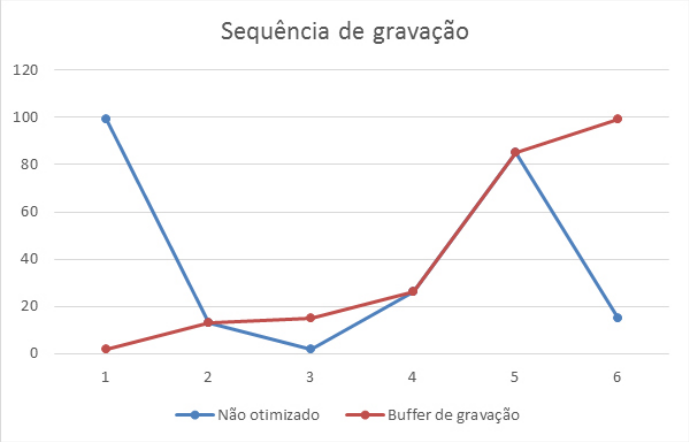
Estatisticamente, é comprovado que uma grande operação com atualização de muitos dados é mais eficiente do que várias operações de pequeno volume de dados. Isso se deve ao fato de que o SGBD pode organizar os dados a serem atualizados em memória RAM de forma a otimizar o movimento da cabeça de gravação.

Vamos ver um exemplo prático:
Suponha que existam 6 conjuntos de dados a serem armazenados, e que esses dados estejam nas seguintes posições do disco rígido: 99, 13, 2, 26, 85, 15. Se a gravação obedecer o enfileiramento proposto, a cabeça magnética primeiro se posicionará no setor 99, gravar os dados, depois voltar para a posição 13, gravar os dados, mover para a posição 2, gravar dados e assim por diante. Observe que a cabeça magnética vai para frente e para trás várias vezes a fim de completar sua tarefa. Somando os movimentos da cabeça (99-13+13-2+26-2+85-26+85-15), teríamos que a cabeça percorreu 349 posições.

De outra forma, o SGBD poderia organizar os dados o mais sequencialmente possível, na seguinte ordem: 2, 13, 15, 26, 85 e 99. Dessa nova forma, a cabeça só andaria para a frente, sem ter que voltar. A distância que ela percorreria seria de 99-2, ou seja, apenas 97 posições. A distância percorrida desta maneira organizada é de 349/97 = 3,6 vezes menor. Assim, o tempo de movimentação da cabeça também seria 3,6 vezes menor.

19

Observe no gráfico formas de gravação **não otimizada** e com o **buffer de gravação**. Note que a distância percorrida pela linha laranja é menor do que a da linha azul, especialmente da primeira para a segunda operação e da quinta para a sexta operação.



Para fazer essa funcionalidade operar de maneira correta, o SGBD precisa interagir com o sistema operacional, e este com o gerenciador dos discos rígidos. Somente assim é possível saber onde gravar o dado e, portanto, organizar a ordem de gravação.

Resumindo, ao organizar os dados de maneira sequencial, o buffer de gravação consegue gravar os dados de maneira mais rápida.

20

4 - Quando implementar técnicas de otimização?

A maioria dos sistemas de informação começam pequenos e vão crescendo ao longo do tempo. É muito difícil estimar o uso de sistemas de informação novos. Por isso, a grande maioria dos trabalhos de otimização não podem ser feitos no início do projeto, mas sim, somente após um período de uso.

Com isso, também qualquer mudança realizada se torna mais sensível, você pode estar considerando uma melhora no ambiente, mas sua alternativa pode acabar piorando a performance. Teoria e prática às vezes não caminham juntos na TI.

Alguns fatos importantes a serem considerados são:

a) Você sempre deve ser capaz de desfazer qualquer ação feita.
b) Teste primeiro em ambiente de testes, depois na produção.
c) Faça as modificações em horários ou dias em que o sistema possa eventualmente ficar parado como durante a madrugada ou finais de semana.
d) Adicione um tempo extra na sua manutenção, caso seja necessário desfazer algo feito.
e) Alerte seus usuários de que o sistema ficará indisponível durante um período de tempo
f) Tenha a disposição todos os profissionais de TI que atuam para a solução ficar disponível (como DBA, analista de rede, programadores etc.).
g) Antes de mudar, realize backups de tudo o que for afetado.

21

5 - COMO ANALISAR E PROPOR UMA MELHORIA DE PERFORMANCE?

Todos os locais por onde a informação passar ou for armazenada é alvo de análise de performance. Portanto, quase tudo relacionado ao computador que gerencia o SGBD, à rede de dados e aos discos de armazenamento podem gerar informações importantes para análise.

Veja alguns exemplos:

- O uso do processamento (taxa de uso do processador) pode indicar que muitas consultas estão demorando porque o processador não está suportando. Trocar por um processador mais atual ou propor uma técnica de processamento paralelo pode ser uma solução.
- O uso de memória RAM pode indicar gargalo no processamento das consultas. Aumentar a capacidade da memória RAM pode ser uma solução.
- O uso do servidor para outros serviços pode indicar gargalos no servidor, separar aplicações em servidores distintos pode ser uma boa solução.
- Muitos índices em uma tabela podem gerar gargalos em operações de inserção ou alteração de dados.
- Um SGBD antigo pode ser tecnologicamente ultrapassado e gerar baixa performance.
- Pouco espaço livre em disco pode gerar problemas de gravação de dados.
- Uma tecnologia de RAID errada pode gerar baixa performance do sistema.
- Tráfego alto de dados em placas de redes podem indicar gargalos de rede.
- Taxas muito alta de troca de dados podem indicar gargalos no gerenciador dos discos.
- Realização de backups nos horários de alto uso do sistema podem gerar baixa performance do SGBD.
- Sistemas de informação com tecnologia inadequada podem gerar problemas de performance.

Dessa forma, analise sempre o seu sistema como um todo. Às vezes, um problema de performance pode ser simplesmente um antivírus ruim que foi instalado na rede.

22

RESUMO

Neste módulo, aprendemos que:

a) Uma das formas de se aumentar a performance dos SGBDs é separar os discos utilizados pelo SGBD dos discos utilizados pelos demais programas do computador.

b) Um SGBD utiliza, geralmente, dois ou mais arquivos relacionados aos bancos de dados. Normalmente, utiliza-se um arquivo para manter os dados e outro para manter as transações realizadas (banco de dados transacional). Uma forma de se obter um ganho de performance nesse tipo de arquitetura é separar os discos de dados dos discos de transações.

c) Algumas soluções de SGBD gerenciam mais de um banco de dados ao mesmo tempo. Para evitar a concorrência de discos entre dois ou mais bancos de dados, uma boa alternativa é separar os arquivos de cada banco de dados em discos distintos, assim como os respectivos arquivos de log transacional.

d) Por fim, alguns SGBDs são tão grandes e concorrentes que há um grupo ou mais de tabelas que são altamente impactadas pelo acesso contínuo e concorrente. Para esse ambiente, uma solução é separar essas tabelas em discos distintos.

e) A gravação em bloco é um modelo de se obter maior performance nas operações de gravação de dados visto que ela diminui a quantidade de operações de movimentação da cabeça magnética.

f) A leitura e gravação sequencial diz respeito aos dados serem gravados sequencialmente nos discos (lado a lado). Dessa forma, o tempo gasto para reposicionamento da cabeça é diminuído a quase zero.

g) Caches de leitura mantém em memória RAM os dados estatisticamente mais acessados pelos sistemas de informação, dessa forma, há uma grande probabilidade de o SGBD não precisar consultar os dados nos discos.

h) O gerenciador de estatísticas é o mecanismo do SGBD responsável por identificar as tabelas e dados mais acessados. E o gerenciador de cache o responsável por manter os dados em memória RAM.

i) O buffer de gravação organiza as informações a serem gravadas, de modo a minimizar a movimentação da cabeça de gravação.

j) Há várias formas de se otimizar um sistema de informação. Para a análise da causa raiz, é necessário conhecer todos os indicadores de performance disponíveis e todas as características do ambiente de TI.