BANCOS DE DADOS ACID E SISTEMAS OPERACIONAIS

ESTUDO DE CASO DO COMPORTAMENTO DO SQL SERVER COM RELAÇÃO AO SISTEMA OPERACIONAL E BENEFÍCIOS DESSA SINERGIA.

UFABC
CURSO DE SISTEMAS OPERACIONAIS / 2 QUADRIMESTRE DE 2016
PROFESSOR FERNANDO TEUBL FERREIRA

Lênin Cristi Fernandes
RA: 21030114

CONTEÚDO

BANCOS DE DADOS ACID E SISTEMAS OPERACIONAIS

Sistemas gerenciadores de bancos de dados têm algumas estruturas bastante semelhantes das vistas em sistemas operacionais, e implementam boa parte das técnicas de interação com o sistema aprendidas no curso, o objetivo do relatório é comparar suas estruturas de blocos, mostrar essas semelhanças e indicar algumas boas práticas de ambiente com reflexo na performance do banco de dados.

Dado os sistemas gerenciadores de bancos de dados (SGBD's) terem estruturas muito semelhantes entre si entre diferentes fornecedores, sejam o SQL Server da Microsoft, o DB2 ou o Informix ambos da IBM, o Oracle xg e o MySQL ambos da Oracle, apesar de usarmos aqui o banco multiplataforma SQL Server, as técnicas em especial no sentido de I/O podem ser consideradas em qualquer destes SGBD's que tenham aderência ao ACID, mas com implementação em nível de código distinta. A versão do banco utilizada foi a "SQL Server 2016 Developer Edition" que é gratuita para desenvolvimento e a última disponível na confecção do relatório.

O QUE É UM SGBD E O QUE É ACID?

SGBD

Um sistema gerenciador de banco de dados ou SGBD (do inglês DBMS - Data Base Management System) é um conjunto de aplicações responsável por gerenciar um armazenamento de dados, e permitir a outros aplicativos consultar e manipular dados.

No nosso caso, vamos nos ater a SGBD's relacionais ou seja, que são capazes de manter e forçar políticas de relação de chave entre tabelas de dados respeitando relacionamentos 1-para-muitos ou 1-para-1 por exemplo entre seus dados. Mas vou comparar rapidamente suas características com sistemas NoSQL (do inglês Not Only SQL).

ACID

ACID é o acrônimo de Atomicidade, Consistência, Isolamento e Durabilidade (do inglês ACID - Atomicity, Consistency, Isolation and Durability) que é um conceito geralmente ligado ao conceito de transação num SGBD. Transação num SGBD pode ser entendida como um conjunto de comandos de manipulação de dados ou alteração de esquema que devem ser executadas pelo SGBD.

- **Atomicidade** - Trata o conjunto de comandos da transação como indivisível, ou todos são executados ou os dados são mantidos como se a transação nunca tivesse ocorrido, os dados e esquema não devem ficar em estados intermediários (geralmente "commit" no caso de sucesso ou "rollback" para reverter);
- **Consistência** - A transação deve levar os dados de um estado consistente a um estado consistente com relação ao esquema, todas as políticas com relação a tipo ou relação por exemplo devem ser respeitadas;
- **Isolamento** - As transações paralelas num sistema multiusuário não interferem umas nas outras, transações diferentes não "veem" os estados intermediários dos dados de uma outra transação sendo executada nos seus dados;
- **Durabilidade** - Os efeitos de uma transação bem sucedida devem ser persistentes mesmo em casos de falha do próprio SGDB, do sistema ou hardware.

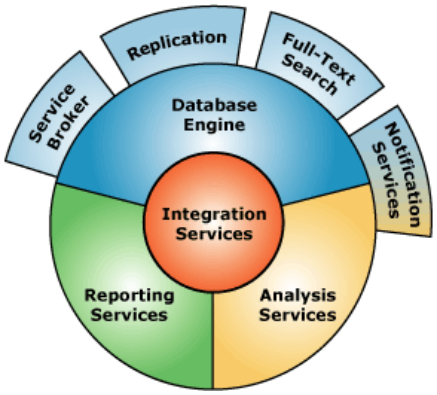
O QUE É O SQL SERVER?

O SQL Server é um sistema SGBD desenvolvido atualmente pela Microsoft, foi criado em 1988 pela Sybase para a plataforma OS/2, e em 1994 a Microsoft assumiu o desenvolvimento do produto. Ele é completamente aderente ao ACID e trabalha com o padrão ANSI SQL, mas o estende num dialeto específico chamado T-SQL. Sua versão 2014 foi a última a suportar plataforma 32bits e a versão atual 2016 será a primeira lançada também para sistemas operacionais Linux.

Logo do SQL Server

Uma importante virada no produto SQL Server ocorreu em novembro de 2005 quando foi lançado o SQL Server 2005 codenome “Yukon”, onde foram incluídos com uma visão mais completa de produtos “na caixa” do mecanismo de dados outros serviços, sendo os principais:

- **Database Engine** - É o mecanismo de núcleo para armazenamento, processamento e segurança dos dados (no sentido de autorização e recuperação). Ele é usado para criar armazenamentos de dados para OLTP ou OLAP;
- **Analysis Services** - Mecanismo para criação de “datawarehouses” e “datamarts” um tipo específico de armazenagem de dados otimizado para análise organizados em cubos OLAP, e ferramentas de mineração de dados “data minning”;
- **Integration Services** - É o mecanismo de ETL (Extract, Transform, Load) da plataforma, é possível integrar e transformar dados de diferentes plataformas e formatos com ele. Até a versão 2005 ele se chamava DTS (Data Transformation Services);
- **Reporting Services** - Uma plataforma de desenvolvimento, gerenciamento e distribuição de relatórios interativos, formato papel, tabular ou painéis (Dashboards) que podem ser consumidos na plataforma, entregues para agendamento ou consumidos por API’s multiplataforma.



Serviços do SQL Server a partir da versão 2005

Os componentes adicionais são essenciais a estruturas de tomada de decisão (DSS - Decision Support Systems) e de inteligência de negócios (BI - Business Intelligence).

A inclusão destes serviços na plataforma diminuiu drasticamente o custo de aquisição do produto, visto que os concorrentes disponibilizavam (e ainda disponibilizam) plataformas BI, OLAP e ETL licenciadas em separado e em alguns casos com custo semelhante ao do mecanismo de dados.

Quadrante mágico da Gartner para Outubro de 2015 em DBMS’s. Importante notar que eles consideraram bancos não ACID neste levantamento.

Dez anos após esse lançamento estamos na versão 2016 do produto e esse posicionamento agressivo foi um fator decisivo para colocá-lo em destaque no último levantamento da Gartner sobre SGDB’s conhecido popularmente como “quadrante mágico” frente aos concorrentes mais diretos.

ESTRUTURA DE UM OS X ESTRUTURA DO SQLOS

No passado a escalabilidade do SQL Server sempre foi questionada quando entravam em cena volumes transacionais muito grandes ou bancos de dados muito massivos. O mercado classificou o produto como um SGBD capaz de gerenciar volumes de dados de pequeno e médio porte devido limitações de seu mecanismo. Esse retorno de mercado levou a equipe de desenvolvimento a uma profunda alteração na engenharia do produto: A criação do SQLOS ou sistema operacional do SQL Server.

Resumidamente, o SQLOS é uma camada fina de software rodando em modo de usuário entre o SQL Server e o Windows. Ela é usada para operações de baixo nível como agendamento, entrada/saída, gerenciamento de memória e de recursos.

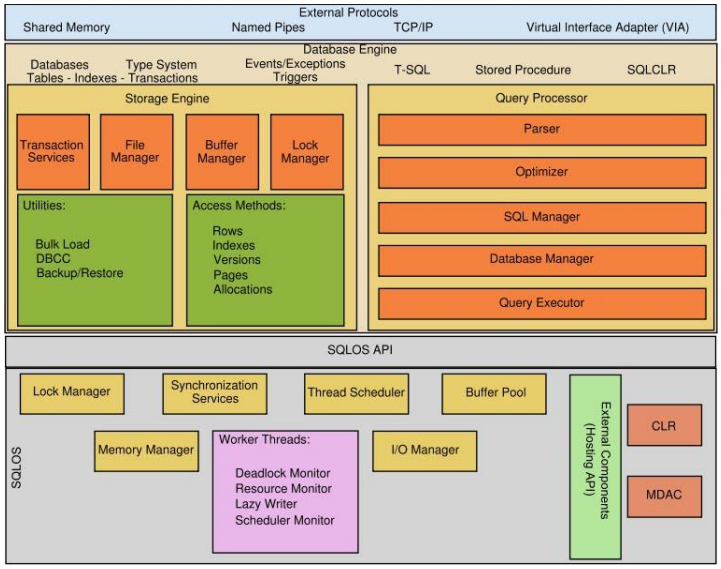


Diagrama de blocos do SQL Server e do SQLOS

CPU (AGENDADORES ESCALONAMENTO E THREADS)

O SQL Server pode rodar em modo thread ou fibra (thread mode, fiber mode) sendo o padrão o modo thread, o modo fibra é um modo de pool mais leve mas só recomendado em situações específicas.

Os elementos desse processo de trabalho são:

- **Agendadores (Schedulers)** - No inicio do serviço, o SQL Server reserva um agendador (scheduler) para cada processador lógico no sistema*. Ele controla a execução das requisições e não substitui o agendador do Windows, mas gerencia a execução sem retornar o controla para ele, o que diminui significativamente as trocas de contexto. Ele opera de modo não-preemptivo, uma vez que as requisições têm a obrigação de devolver o controle para o agendador quando termina seu tempo de CPU ou são suspensas;
    - Agendadores podem ser monitorados pela view dinâmica de gerenciamento (DMV) sys.dm_os_schedulers;
- **Trabalhadores (Workers)** - O trabalhador nessa estrutura é um elemento que define uma capacidade de trabalho, ele é ligado a uma thread do sistema e é dividido em grupos (pools) associados por sua vez aos agendadores. Eles recuperam tarefas de uma lista de trabalho e cada uma delas roda até completar. Eles podem ser trocados de agendador dependendo da afinidade de CPU configurada;
    - Trabalhadores são calculados segundo o critério onde “n” é o número de CPUs lógicas no sistema:
        - Se n é menor ou igual a 4 o máximo é de 512 trabalhadores;
        - Se n é maior que 4 o máximo de trabalhadores é dado por (512 + (n - 4)*16)
    - Trabalhadores são monitorados pela view sys.dm_os_workers e threads em uso são monitorados pela view sys.dm_os_threads;
- **Tarefas (Tasks)** - Tarefas são executadas quando retiradas de uma fila de trabalho (work queue) de modo não-preemptivo o que novamente denota o modelo cooperativo do SQL Server. Elas são atribuídas a um agendador

(para serem executadas por um trabalhador de seu grupo) baseada em NUMA, configurações de afinidade ou balanceamento de carga por pressão relativa nos agendadores**;

- Tarefas podem ser monitoradas pela view sys.dm_os_tasks, a view sys.dm_os_waiting_tasks que informa as tarefas esperando por um recurso qualquer;

Nota*: Na verdade, o SQL Server mostra dois agendadores por CPU, sendo que um deles é reservado para tarefas internas e o outro para execução de requisições, mais um na CPU zero para conexões administrativas diretas, usada somente em casos especiais para correções mais críticas. Para nosso escopo de compreender o agendamento, pode ser interessante simplificar em um por CPU.

Nota**: Antes do SQL 2005, o agendador do SQL Server trabalhava no conceito de round-robin, e a tarefa não tinha garantias de ser agendada num agendador sub-utilizado. Essa característica afetava o mecanismo a ponto de tornar desafiadores trabalhos volumosos OLTP ou OLAP.

Simplificadamente o plano de execução é:

1. O usuário conecta ao mecanismo;
2. O usuário executa comandos;
3. Os comandos são divididos em uma ou mais tarefas (no caso de paralelismo ser necessário);
4. Cada tarefa é atribuída a um trabalhador;
5. O agendador "agenda" o trabalhador para execução;
6. O trabalhador entra nos estados RUNNING, RUNNABLE ou SUSPENDED;
7. A tarefa é concluída;

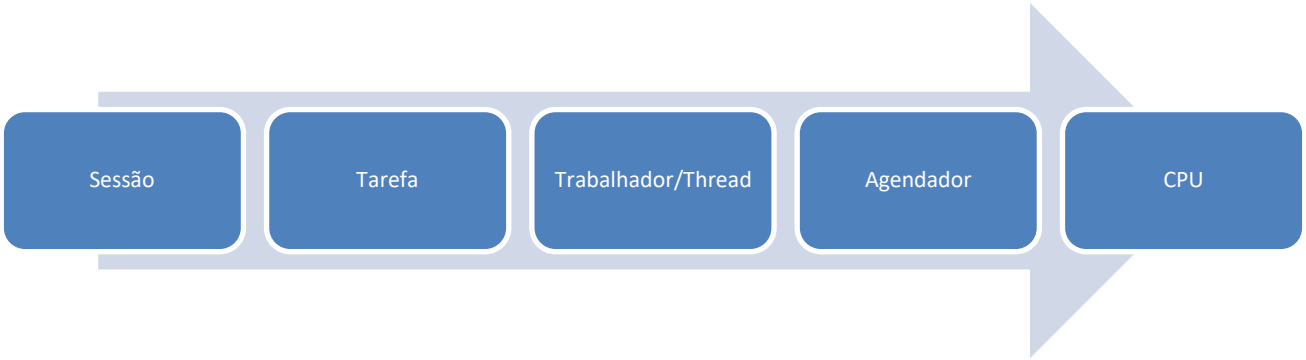


Diagrama de blocos do plano de execução

CONECTIVIDADE (SHARED MEMORY, NAMED PIPES E SOCKETS)

Para conectar ao SQL Server, um protocolo de conectividade deve estar ativo:

- Memória compartilhada (Shared Memory) - É o protocolo de conexão mais simples, e como só funciona no escopo local geralmente não é muito utilizado, mas é muito útil em casos de conexão para resolução de problemas, por exemplo;
- Pipes nomeados (Named pipes) - É um protocolo pensado para redes locais, uma parte da memória é usada para passar informação para outros processos, num esquema de "tubos" de entrada/saída, a outra ponta pode estar na máquina local ou num computador da rede;
- TCP/IP (Sockets) - É o protocolo de rede mais usado na internet hoje. Ele é interoperável em várias arquiteturas e sistemas operacionais, tem padrões de roteamento e oferece características avançada sde segurança.

Nota: Em redes locais rápidas, não há diferenças significativas de velocidade entre TCP/IP e pipes nomeados, mas esta diferença se torna mais clara em redes mais lentas como as de área mais ampla (WAN's) ou de conectividade discada.

MEMÓRIA (IN-MEMORY OLTP)

Nota: Não vou abordar aqui como no caso da CPU o gerenciamento de memória do SQL Server como um todo, mas uma tecnologia específica que ajuda a lidar com um problema discutido durante o curso: A diferença de latência e possível troca de contexto de um thread que deixa de acessar dados em memória e passa a acessar dados em disco, e como esse problema foi endereçado nessa plataforma. É importante notar que grandes fornecedores de SGBD também tem soluções a grosso modo semelhantes para esse mesmo problema.

Uma característica importante ao considerar bancos de dados é o acesso a disco, e numa aplicação usual, esse acesso geralmente causa um tempo de espera pelo recurso disco (que é mais lento) e potencial troca de contexto da thread do ponto de vista do sistema operacional, uma solução para esse problema é a tecnologia chamada in-Memory OLTP.

Quando uma tabela é muito usada em memória, não é incomum os mecanismos mais avançados de banco de dados carregarem porções dela em memória para otimizar a leitura, na verdade isso evita acessos mais frequentes a disco e em bancos que não abstraiam os agendadores do sistema, causar trocas de contexto com outras aplicações rodando e degradar mais seriamente a performance já na leitura de dados.

Mas quando abordamos o cenário OLTP, ou transacional com parcelas relevantes de escrita e leitura, manter uma tabela na memória e manter o banco estritamente aderente ao ACID pode se tornar um desafio, pois uma tabela que receba um "commit" de alterações na memória, para todos os efeitos não passa no quesito "durabilidade" do ACID, visto que a memória é sujeita a falhas de energia por exemplo que fariam com que fosse esvaziada e as transações confirmadas perdidas.

O recurso in-Memory OLTP vem para resolver esse problema, de um lado impedindo acessos constantes ao disco para confirmação de transações de escrita e dando durabilidade a transações mesmo em memória, ela também é otimizada para trabalhar com dados residentes na memória, ao invés de dados baseados em disco.

Essas características são conseguidas do seguinte modo:

- A informação escrita em disco é resumida a log streams e checkpoint streams
    - Log streams são as alterações feitas por transações
    - Checkpoint streams são divididos em
        - Data streams que contém versões de linha num intervalo de tempo
        - Delta streams que são um registro de deleção de linha relacionado aos data streams

Um par de checkpoints (CFP – Check point file pairs), data e deltas representam um segmento do log de transação de aproximadamente 100MB de novas versões de linha. Esses conjuntos se acumulam num checkpoint completo e esse checkpoint completo mais o log conseguem reconstruir a tabela na memória caso ocorra a falha.

Para tabelas que usam o in-Memory OLTP, o log de registros só é gerado no momento da confirmação da transação, diferente das tabelas em disco, que usam a escrita de disco "antecipada" (WAL - Write Ahead Logging). Na WAL o banco de dados escreve no log antes de qualquer alteração em disco. Para tabelas in-Memory OLTP, dados sujos (dirty data, ou alterações não confirmadas) nunca são escritas em disco e essa tecnologia as agrupa em blocos maiores de até 24kb o que resulta em menos atividade de escrita de log em disco e atividades de I/O maiores.

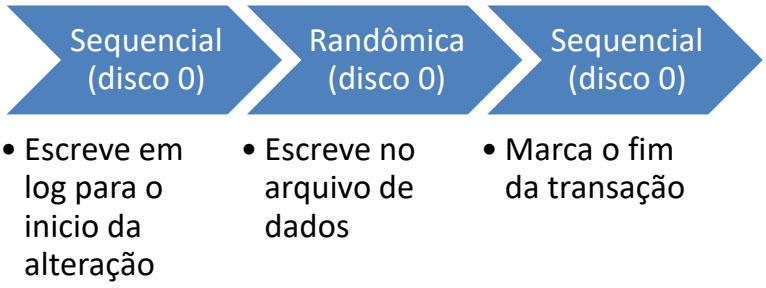
I/O (DESIGN DE STORAGE, DISTRIBUIÇÃO DE CARGA)

Nota: Durante o curso foram levantadas dúvidas sobre a utilidade de um design de disco que isole pedaços do banco de dados fisicamente no servidor e seu ganho de performance, sendo esse design atrelado fortemente ao hardware e sistema operacional, acho interessante dar uma visão desse caso particular de solução de design. Importante ressaltar aqui que esse design e os conceitos apresentados também são semelhantes nos maiores fornecedores de SGBD atuais.

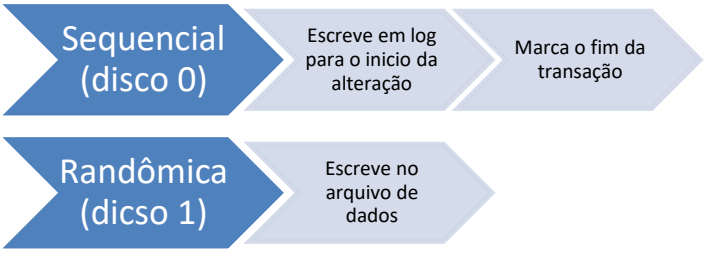
Para entender design de disco e sua utilidade é importante entender que um banco de dados (vamos partir de um exemplo simples, sem divisões inicialmente em grupos de arquivos) é composto de duas partes: O arquivo de dados e o arquivo de log, e essas duas partes têm características de acesso muito distintas:

- Arquivos de dados primários - Têm extensão MDF e é sujeito a leitura constante, escrita de eventual para constante e de característica **randômica**;
- Arquivos de log - Têm extensão LDF e mantém toda informação de transação necessária a fazer a recuperação dos dados se necessário, é sujeito a escrita **sequencial.**

Seu comportamento de acesso ao disco pode ser entendido como no seguinte diagrama:



Geralmente a escrita e leitura sequencial em disco é muito superior em performance a escrita e leitura randômica, então separar estes comportamentos em arquivos distintos, com subsistemas de disco separados* tira proveito não só da característica de agendamento de tarefas em CPUs exposta no primeiro tópico que permitem essas operações de escrita randômica e sequencial serem paralelizadas mas também do hardware de disco de permanecer em escrita sequencial durante todo o processo, como neste segundo diagrama:



Nota*: Para tirar proveito desse design, os discos precisam ser fisicamente separados, simplesmente dividir o disco em partições lógicas não oferece esse ganho de performance, pois fisicamente o mesmo disco (ou a mesma agulha) ainda estaria sendo usada para as duas escritas. É importante notar também que ainda que os comportamentos de escrita sejam semelhantes em difernetes grupos de arquivos, dividir os subsistemas ainda dá ganhos de performance pela arquitetura de execução massiva multi-thread e de paralelismo otimizado inerente a SGBD's modernos.

Com estes conceitos iniciais em mente, vou passar para as boas práticas da EMC, uma conhecida fabricante de armazenamento (Storages) para o design de disco do SQL Server, que recomenda no geral 5 grupos de armazenamento:

- Binários do SO e do SQL Server
  - Numa configuração típica do SQL Server, o servidor é dedicado e os binários do SQL Sever estão na mesma unidade do sistema operacional;
  - O tamanho recomendado dessa unidade fica entre 60GB a 120GB e tipicamente discos de alta capacidade e baixa performance em RAID 5 atingem os requisitos para esta necessidade;
- Bancos de dados do sistema

    - Na maioria dos ambientes, bancos de sistema não são frequentemente alterados e podem ficar na mesma unidade do sistema operacional;
- Arquivos de log dos bancos de usuário
    - Logs de bancos de usuário geralmente necessitam de baixa capacidade de operações de IO/s (na maioria a escrita é sequencial) ainda que estejam em cenários de replicação destes arquivos de log, no entento não é incomum que sejam armazenados em unidades ligadas por canais de fibra.
- Banco temporario TempDB
    - Num cenário OLTP, o banco temporário geralmente não é tão requisitado, e pode seguir o mesmo principio de design adotado para os arquivos de log. Neste caso específico, eles podem estar na mesma unidade dos arquivos de log.
    - Quando existem trabalhos agendados ou geração on-line de relatórios, o banco temporário por sofrer uso muito alto. É necessário traçar o perfil de uso do sistema para determinar o uso do banco temporário e se necessário, seu isolamento em unidade específica.
    - O banco temporário num datawarehouse para trabalhos OLAP geralmente está sob pressão intensiva, e exige atenção especial nestes ambientes e geralmente vai necessitar de ter unidade separada.
- Arquivos de dados dos bancos de usuário
    - Estes arquivos gerlamente são o foco principal do design do armazenamento. Os tipos vão depender da capacidade e carga de trabalho.

É importante notar que as características de uso dos diferentes grupos de arquivos do banco de dados também são afetadas pelo layout de RAID dos discos, pois alguns layouts de RAID têm características de leitura e escrita bem distintas, como na tabela:

Table 8. RAID level performance characteristics

RAID level	Random Read	Random Write	Sequential Read	Sequential Write	RAID write overhead value	Storage utilization
RAID 1/0	Excellent	Excellent	Excellent	Excellent	2	Low
RAID 5	Excellent	Moderate	Good	Moderate	4	High
RAID 6	Good	Poor	Good	Moderate	6	Medium

Tabela comparativa de diferentes designs de RAID

BOAS PRÁTICAS BÁSICAS DO SQL SERVER

Acredito que foi possível passar um panorama geral de uso de recursos do SQL Server no sistema operacional em alguns dos principais recursos do sistema. Para finalizar, vou rapidamente detalhar as boas práticas mais gerais de uma implementação SQL Server, várias delas ligadas a assuntos tratados ou características do sistema hospedeiro (estas opções são mais específicas a ambientes de produção, e não de desenvolvimento):

- Utilize a política “lock pages in memory” para a conta utilizada pelo SQL Server para prevenir “swapping” de páginas da memória.
    - Isso é feito via políticas de grupo e é detalhado num rápido artigo nas referências;
- Pré aloque espaço suficiente dentro dos arquivos de dados para prevenir que eles cresçam durante o uso de pico;
    - Se o arquivo estiver próximo de cheio, o tempo de “crescimento” dos arquivops de dados pode ocasionar bloqueios, ter uma boa estratégia de crescimento para os arquivos de dados é essencial para evitar crescimentos frequentes ou durante o pico de uso;
- Desative a opção de “autoshrink” nos arquivos de dados e log;
    - Esta opção desativa o recurso de “devolver” ao sistema operacional o espaço não utilizado por arquivos de dados ou log , eventualmente o banco devolve esse espaço ao sistema seja no arquivo de log por dados já confirmados e backupeados ou no banco temporário que é volátil por natureza;
- Crie arquivos de dados de tamanhos semelhantes no meso banco de dados;
    - O SQL Server usa um algoritmo de “preenchimento proporcional” que favorece alocação em arquivos com mais espaço livre;
- Faça manutenções periódicas nos indices e estatísticas
    - Atualizar estatísticas e reconstruir ou reorganizar os indices em períodos de baixa atividade aumenta significativamente a eficácia dos planos de execução do mecanismo de dados e sua performance na utilização dos índices existentes;

BIBLIOGRAFIA

Nota: Organizei a bibliografia pesquisada e utilizada em seções para facilitar a consulta por assuntos específicos, os artigos marcados com "*" foram essenciais para a confecção do relatório e apresentação e em alguns casos tiveram trechos isolados traduzidos.

- Arquitetura básica
  - https://pt.wikipedia.org/wiki/Sistema_de_gerenciamento_de_banco_de_dados SGBD
  - https://pt.wikipedia.org/wiki/ACID * ACID
  - https://pt.wikipedia.org/wiki/Microsoft_SQL_Server * SQL Server
  - http://searchsqlserver.techtarget.com/definition/SQL-Server SQL Server
  - https://technet.microsoft.com/en-us/library/ms166352(v=sql.90).aspx SQL Server 2005 e seus serviços
  - https://msdn.microsoft.com/en-us/library/ms143506(v=sql.130).aspx Requisitos para o SQL Server 2016
- SQLOS & CPU
  - https://blogs.msdn.microsoft.com/sqlmeditation/2012/12/13/tasks-workers-threads-scheduler-sessions-connections-requests-what-does-it-all-mean/ * Introdução aos componentes
  - http://www.rdc.co.za/blog/pdf/BLOG-0010_Introduction_to_Microsoft_SQL_Server_SQLOS_2012.pdf *Introdução ao SQLOS
  - https://blogs.msdn.microsoft.com/slavao/2005/07/20/platform-layer-for-sql-server/ Funcionamento da camada SQLOS - O blog está sem uso mas o conteúdo é bem rico
  - http://mscerts.programming4.us/sql_server/sql%20server%202012%20%20%20sql%20server%20architecture%20-%20sql%20server%E2%80%99s%20execution%20model%20and%20the%20sqlos.aspx Execução
  - https://sqlconsultant.wordpress.com/2011/07/09/sql-server-operating-system/ *SQLOS
  - http://sqlmagu.blogspot.com.br/2013/05/arquitetura-sql-server-parte-2-o-sqlos.html SQLOS em português
  - http://www.sqlpanda.com/2013/07/threadpool-and-sql-server-threads.html Demonstração das threads do SQL Server (usado na apresentação)
  - https://www.simple-talk.com/sql/learn-sql-server/understanding-and-using-parallelism-in-sql-server/ Sobre paralelismo
  - http://www.sqlservercentral.com/blogs/livingforsqlserver/2012/12/03/sqlos-basics-query-and-cpu/ Demonstração do SQLOS (usado na apresentação)
- Conectividade
  - https://technet.microsoft.com/en-us/library/ms187892(v=sql.105).aspx * Diferentes protocolos
- Memória
  - http://download.microsoft.com/download/5/f/8/5f8d223f-e08b-41cc-8ce5-95b79908a872/sql_server_2014_in-memory_oltp_tdm_white_paper.pdf * Artigo técnico abrangente sobre a tecnologia in-Memory OLTP
  - https://msdn.microsoft.com/en-us/library/dn720242.aspx Por dentro do in-Memory OLTP
  - https://msdn.microsoft.com/en-us/library/dn133186.aspx artigo rápido sobre in-Memory OLTP
  - https://msdn.microsoft.com/en-us/library/dn530757.aspx Demonstração de in-Memory OLTP (usada na apresentação)
  - https://msdn.microsoft.com/en-us/library/ms178067.aspx Configurações de servidor
- I/O
  - https://www.emc.com/collateral/white-papers/h12341-sqlserver-bp-wp.pdf * Artigo abrangente da EMC sobre armazenamento
  - https://msdn.microsoft.com/en-us/library/ee410782(v=SQL.100).aspx Artigo de análise sobre características de I/O no SQL Server
  - https://support.microsoft.com/en-us/kb/2154845 Artigo sobre alocação de espaço no banco temporário
  - http://cc.davelozinski.com/sql/increase-sql-server-tempdb-performance Aumento de performance no banco temporário
  - https://www.brentozar.com/archive/2009/02/when-should-you-put-data-and-logs-on-the-same-drive/ Artigo sobre quando compensa colocar os arquivos de dados e log no mesmo disco
  - https://www.brentozar.com/archive/2008/09/finding-your-san-bottlenecks-with-sqlio/ Uso do SQLIO (usado na apresentação)
- Boas práticas
  - https://www.emc.com/collateral/white-papers/h12341-sqlserver-bp-wp.pdf * Artigo abrangente da EMC sobre armazenamento
  - https://technet.microsoft.com/pt-br/library/cc966534.aspx 10 boas práticas a se adotar no armazenamento de bancos de dados
  - https://msdn.microsoft.com/en-us/library/ms190730.aspx Como configurar a política de prevenir swapping de páginas de memória no Windows