ADMINISTRAÇÃO DE BANCO DE DADOS

Márcio Motta

SAGAH SOLUÇÕES EDUCACIONAIS INTEGRADAS

OBJETIVOS DE APRENDIZAGEM

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Identificar os problemas de performance mais comuns.
- Reconhecer os recursos para correção de problemas de performance.
- Conhecer as técnicas de otimização para bancos de dados.

INTRODUÇÃO

O desempenho no acesso a um Banco de Dados é ponto fundamental para a agilidade na busca de informação para a tomada de decisão nas organizações. Uma grande quantidade de dados e o tempo de busca das informações têm exigido preocupações com os possíveis problemas de performance apresentados nas bases de dados.

Este artigo apresenta como identificar esses problemas de performance e técnicas para a sua otimização, visando evitar congestionamentos, travamentos, desperdício de espaço e melhoras na velocidade das consultas.

APRESENTAÇÃO

A demanda por Sistemas de Gerenciamento de Bancos de Dados cresce continuamente. Juntamente com essa demanda, cresce também o volume de dados que estes sistemas devem gerenciar e a complexidade de suas aplicações. Neste cenário, realizar operações de forma eficiente precede de uma série de protocolos e boas práticas na implementação e manutenção das bases de dados.

Problemas de desempenho em banco de dados são toda e qualquer contenção que impeça a realização de uma tarefa específica em tempo hábil, devido a problemas de configuração, parâmetros com valores inadequados, dimensionamento errado das áreas de memórias e comandos SQL escritos de maneira ineficiente. Assim, o objetivo da otimização de desempenho em banco de dados é minimizar o tempo de resposta para cada consulta e maximizar o rendimento do servidor de banco de dados inteiro, reduzindo o tráfego de rede, disco e processamento da CPU. Para isso, pode ser preciso

mudar a forma como as aplicações são construídas, as estruturas de dados e parâmetros de um sistema de banco de dados, a configuração do sistema operacional, ou o hardware.

PRINCIPAIS PROBLEMAS DE PERFORMANCE

Muitas vezes o projeto de banco de dados não prevê um alto crescimento dos dados causando após a entrada em produção problemas de desempenho e até de operabilidade. Neste cenário, bases com desempenho abaixo do esperado fazem com que geralmente todos os sistemas dependentes desta tenham o tempo de resposta afetado negativamente.

Como principais problemas relacionados com a performance, podemos citar 3 pontos que devem ser considerados:

Problemas de Infraestrutura

Os recursos de Infraestrutura tem relação direta na performance do Banco de Dados, seja durante a sua fase de análise de requisitos para a as implementação, seja após o incremento de vários registros na base de dados.

A capacidade do servidor, assim como as conexões de rede são fator determinante na sua performance, em primeiro lugar deve-se analisar o volume de dados que serão trabalhados, levando em consideração o crescimento constante de registros e a demanda de consultas, assim a escolha do hardware necessário, capacidade da CPU, quantidade de memória, capacidade e desempenho dos discos deve ser calculada para atender as demandas com uma ampla margem de crescimento das bases de dados no futuro. Economizar nessa fase de implementação, causará além de um desempenho ruim, custos adicionais para remediar um servidor de desempenho insuficiente. As conexões de rede também devem ser consideradas para que não se crie gargalos, causando lentidão na velocidade da rede mesmo que o servidor seja suficiente.

A escolha das tecnologias também dependerá, além de custos de investimento, custos operacionais e administração da infraestrutura, também como a forma que as consultas serão realizadas, com mais ou menos usuários/clientes com acesso às bases de dados. A decisão da estrutura do servidor

deve ser calculada com inteligência, seja a decisão por um servidor físico local ou a utilização de um servidor na nuvem. Em muitos casos a escolha de um servidor na nuvem (*cloud*) diminuirá custos iniciais e manutenção, mas também é necessário o dimensionamento correto nesta contratação de serviços, no que diz respeito ao espaço e tecnologias e a velocidade de conexão à Internet.

Escolha do SGBD/Modelagem dos dados

O desempenho do banco de dados e suas consultas já se define no momento da escolha do Sistema Gerenciador de Banco de Dados (SGBD). O SGBD nada mais é do que a solução de software escolhida para fazer todo o gerenciamento e modelagem dos dados a serem inseridos. Há diversas soluções de SGBDs no mercado atual, tais como Oracle, SQL Server, PostgreSQL e MySQL. Existem soluções comerciais e Freeware/Open Source, mas devem ser levados em consideração fatores como: tamanho máximo total do banco de dados, número de bancos/tabelas permitidos e complexidade das consultas. Alguns SGBDs, dependendo do modelo e da versão utilizada, trabalham com o **padrão SQL-92**, o que não permitirá consultas avançadas que requerem o padrão **SQL:1999** (como OLAP, CUBE e Advanced), assim podendo trabalhar apenas com bancos de dados relacionais, não permitindo o uso em Data Warehouses e recursos de Data Mining.

Outro ponto determinante são as boas práticas no momento da modelagem dos dados e criação das tabelas e colunas no SGBD. A tipagem dos dados é decisiva na alocação das informações na memória e também no espaço alocado em disco para cada registro.

Quando uma tabela, índice ou coluna possui mais espaço alocado do que necessário, isso pode não aparecer como um problema, mas pode afetar negativamente o desempenho. Por exemplo, se as páginas de um objeto em SQL contidas em um Data Warehouse não estiverem 100% utilizadas, mais páginas do que o necessário serão examinadas durante leituras de tabelas. Isso, por sua vez, contribui com um maior tempo de resposta a consultas do usuário.

Uma tipagem equivocada também poderá causar "truncamento" dos dados, informações imprecisas, consumo de espaço desnecessário em disco e aumento do processamento da CPU.

Tabela nº 1 – Tipagem de Textos

TIPO	CAPACIDADE
TinyText	Até 255 caracteres
Text	Até 65 mil caracteres
MediumText	Até 16 milhões de caracteres
BigText	Até 4 bilhões de caracteres
VarChar	Definido pelo programador (valores pequenos)

Fonte: Autor

A Tabela 1 demonstra as tipagens possíveis para os registros de texto mais comuns na linguagem DDL (definição de dados), onde cada tipo deve ser utilizado de forma coerente com a quantidade de texto previsto para cada registro. Não faz sentido (e prejudica a performance) a utilização de um tipo que prevê uma grande quantidade de caracteres para que sejam armazenadas pequenas informações de texto. O mesmo vale quando se deseja armazenar números e se opta, de forma equivocada, por um tipo de texto para os registros.

Quanto ao armazenamento de números, também há uma grande quantidade de tipos que podem ser utilizados, apresentando problemas semelhantes aos dos registros de texto, quando mal utilizados. A Tabela 2 apresenta os tipos relacionados a números e a sua capacidade de armazenamento.

Tabela nº 2 – Tipagem de Números

TIPO	CAPACIDADE
Bit ou Bool	Número inteiro 0 ou 1
TinyInt	Número inteiro de 0 até 255
Integer (Int)	Número inteiro de 0 até 429.496.295
SmallInt	Número inteiro de 0 até 65.535
MediumInt	Número inteiro de 0 até 16.777.215
BigInt	Número inteiro de 0 até 18.446.744.073.709.551.615

TIPO	CAPACIDADE
Decimal	Definido pelo usuário as casas antes e depois da vírgula
Float	Número com vírgula de 1,75494351E-38 até 3,402823466E+38 (positivos ou negativos)
Double	Número com vírgula de dupla precisão de 2,2250738585072014E-308 até 1,7976931348623157E+308 (positivos ou negativos)

Fonte: Autor

Outro ponto a ser considerado na criação das tabelas é a questão da Indexação. Índices além de facilitar consultas e buscas, têm relação até mesmo com o desempenho do disco na busca dos dados em um servidor. Quando índices possuem uma ordem lógica que não corresponde à ordem física real dos dados armazenados, o desempenho do acesso aos índices pode ser afetado. Se as ordens lógica e física estiverem sincronizadas, a cabeça de leitura do disco poderá ler em uma só direção, ao invés de mover-se para frente e para trás para obter as informações necessárias. De outra forma, a cabeça de leitura terá de efetuar saltos através do disco com muita frequência. Em servidores com a estrutura bem dimensionada, mas com acesso lento ao disco quase sempre indica uma situação que requer uma reorganização da ordem dos dados e ajuste dos índices.

Otimização das consultas

O desempenho do SGBD está relacionado com a simplicidade de suas consultas, em um grande número de casos, a reorganização de consultas mal formuladas já garantirá um ganho de performance, evitando a lentidão ao obter os resultados e garantindo informações mais diretas e precisas.

De uma forma geral, existem alguns critérios principais que podem ser adotados para identificar as consultas que devem ser modificadas, são eles:

- Monitorar as sessões ativas que estão sendo executadas no banco de dados;
- Separar as consultas que estão com execuções demoradas;

- Dividir as consultas em grupos, como: prioridade, frequência de execução e fraco desempenho;
- Implementar os ajustes reescrevendo as consultas que estão com fraco desempenho.

Cada banco de dados fornece suas ferramentas específicas para capturar as consultas citadas nos itens acima, porém a maneira mais eficiente e explícita é o próprio uso feito pelo usuário final. A otimização das consultas SQL apresenta vários critérios para a melhora do desempenho.

Campos a Selecionar

Selecionar exclusivamente aqueles campos que são necessários, o SGDB deve ler primeiro a estrutura da tabela antes de executar a sentença, não buscando campos desnecessários para a consulta.

Exemplo:

SELECT * FROM TABELA

Neste caso todos os campos estão sendo selecionados, independente de sua necessidade real, causando demora na obtenção dos resultados. Ao invés disso deve-se evitar o (*) que significa "selecionar tudo" e sim definir cada coluna a ser consultada, como no exemplo abaixo:

SELECT id,nome,produto,valor FROM TABELA

Trabalhando dessa forma serão selecionadas as colunas id(índice), nome, produto e valor, evitando uma sobrecarga de processamento. Mesmo que exista um número muito grande de colunas, a seleção será específica.

Campos de Filtro

A filtragem da consulta através da cláusula WHERE garante o resultado apenas nos campos que fazem parte da chave do pelo qual está sendo consultada. Quando há uma grande quantidade de registros, a filtragem evita a apresentação de dados desnecessários que dificultam a obtenção de resultados precisos e causa lentidão.

Filtros sempre devem ser específicos com informações exatas, como por exemplo:

SELECT vendas FROM TABELA WHERE DATA = 2016-10-20

Isso fará com que somente as vendas realizadas na data específica sejam apresentadas na consulta. Os filtros não devem ter informações incompletas e a cláusula LIKE deve ser evitada pois necessita de uma busca em todos os registros fazendo que obtenção dos resultados seja lenta.

CONCLUSÃO

Para conseguir os melhores resultados no acesso a um Banco de Dados, evitando problemas de performance, devem ser considerados todos os critérios analisados (Infraestrutura, modelagem e consultas) nas fases iniciais da implementação do servidor e do SGBD. Com isso também se evita que seja necessário um grande esforço com upgrades, remodelagem ou recodificação para se atingir um nível de desempenho satisfatório. Além disso nem sempre é possível conseguir os mesmos ganhos de performance se os critérios fossem considerados desde o início do desenvolvimento.

A utilização de boas práticas em Banco de Dados traz benefícios para a organização, pois o tempo de resposta em uma consulta SQL é minimizado e a performance é garantida.

REFERÊNCIAS

NICOLAS, B. & CHAUDHURI, S. - **Automatic Physical Database Tuning: A Relaxation-based Approach**. In: PROCEEDINGS OF THE 2005 ACM SIGMOD INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA, 2005.

CUNHA, E. - **Estudo de viabilidade de uma Plataforma de Baixo Custo para Datawarehouse,** Master's thesis, Departamento de Informática, Universidade Federal do Paraná, 2004.

SHASHA, DENNIS, BONNET, PHILIPPE - **Database Tuning**, Morgan Kaufmann Publishers, 2003.

Conteúdo:

SAGAH
SOLUÇÕES EDUCACIONAIS INTEGRADAS