Checklist Performance banco de dados.

Primeiramente precisamos entender que a necessidade de Tuning é natural para sistemas de banco de dados, ao longo do tempo a volumetria de dados cresce consideravelmente.

Para realizar um Tuning não existe um passo a passo gravado em pedras, ou exige conhecimento superior além do que já praticamos no dia a dia, tuning nada mais é que executar ações na infraestrutura, na instância do SQL Server ou no banco de dados, evitando a queda de performance.

Primeiramente precisamos olhar o incidente de forma macro, e isolar o máximo possível os afetados, para assim encontrar o ofensor, é importante que este trabalho de análise seja feito em diversas frentes, as ações tuning pode ser:

Otimização de querys;

Manutenção de Índices e Estatísticas;

Ajuste de recurso de hardware;

Configuração no banco de dados ou instância;

E etc...

2 Tipos de tuning - Proativo / Resolução de problema

Proativo, não temos nenhum incidente registrado, mas buscamos processos que estão com maior consumo e atuamos de forma preventiva para evitar que fique visível o problema de performance aos usuários.

Resolução de problema, temos um incidente e a query ofensora é mapeada e atuamos para um melhor desempenho.

Sugestões de etapas no Tuning

Etapa 1 - Coleta de informações

Precisamos coletar informações, verificar logs, conversar com o usuário, entender se é uma tela específica, ação, processo de importação etc. Este processo é essencial e indispensável, assim podemos canalizar o esforço e análise naquele ponto específico.

Etapa 2 – Indicadores na camada externa

Verificar os indicadores:

CPU

Consumo de Memória

Consumo de Rede

Leitura e Escrita de disco

Estes indicadores vão nos dar um norte para onde vamos focar nossa análise, por exemplo, se temos um alto consumo de CPU, na camada interna no nível da instância, vamos verificar quais são as querys com maior consumo deste recurso. Esta etapa vai nos ajudar a encontrar qual tipo de contador interno do SQL Server devemos avaliar no processo de tuning.

Etapa 3 – Ferramentas Windows

Task Manager – De forma simples podemos obter informações muito importantes, uma visão macro do consumo de memória, cpu, escrita e leitura de disco, vai nos ajudar a ter um panorama geral do ambiente.

Perfmon – Através dele é possível aprofundar mais estes contadores que você tem no task manager e muito mais, em geral, você consegue verificar contadores a nível de Windows e alguns do banco de dados.

Todas estas informações vão te ajudar a mapear pontos focais que precisam de uma atenção, e lhe dar base para atuar no processo de tuning, tanto na resolução de problemas ou proativa. Existe diversas formas de realizar tuning, com esta sugestão é possível ter a visão ampla para direcionar a sua análise.

Etapa 4 – Ferramentas SQL Server

Monitor atividades – Informações sobre processos ativos, querys recentes que consomem mais recurso, escrita e leitura de disco por arquivo de banco de dados, é um panorama geral da instancia.

SQL Server Profiler – Executar trace (rastreamento), é possível definir um layout e verificar tudo que está executando no banco de dados, favor solicitar ao departamento de banco de dados o manual de como configurar um layout de trace.

Etapa 5 – Recursos SQL Server

Extend Events – Ele tem a mesma funcionalidade do Profiler, porém tem informações a mais referente a hardware, é mais complicado de ser parametrizado e recomendo para coletas de baseline, para o dia a dia a forma mais rápida continua sendo o Profiler. Existe um boato na comunidade SQL Server que o mesmo será sucesso do Profiler que tende a ser descontinuado, por isso é importante conhecermos um pouco mais deste recurso do SQL Server.

Query Store - O recurso Repositório de Consultas fornece insights sobre a escolha e o desempenho do plano de consulta para o SQL Server, o Banco de Dados SQL do Azure, a Instância Gerenciada de SQL do Azure e o Azure Synapse Analytics. Ele simplifica a solução de problemas, ajudando você a identificar rapidamente diferenças de desempenho causadas por alterações nos planos de consulta. O Repositório de Consultas captura automaticamente um histórico das consultas, dos planos e das estatísticas de runtime e os mantém para sua análise. Ele separa os dados por janelas por hora, permitindo que você veja os padrões de uso do banco de dados e entenda quando as alterações aos planos de consulta ocorreram no servidor. Possibilitando ranquear e identificar quais querys consomem mais recursos.

Possíveis causas:

Alto consumo de CPU, iniciar a análise com as querys com maior consumo, e em seguida verificar a ausência de índices com os scripts auxiliares “Analise dos indices.sql” e index\_sugestao\_missing.sql.

Scripts Auxiliares

Análise dos indices.sql e index\_sugestao\_missing.sql

Com estas consultas você poderá visualizar as sugestões de índices do SQL Server baseado nas estatísticas de Missing Index. Muito cuidado com essas sugestões, pois nem sempre elas são a melhor opção para a criação de um índice. Analise as sugestões antes de criar no banco.

desempenho\_IO.sql

Verifica desempenho de I/O dos discos

eventos\_wait.sql

As estatísticas de espera são um dos indicadores mais importantes para identificar problemas de desempenho no SQL Server. Quando queremos solucionar qualquer problema de desempenho, primeiro precisamos diagnosticar o problema corretamente, pois diagnosticar corretamente os problemas ajuda com metade da solução.

<https://www.sqlshack.com/boost-sql-server-performance-with-wait-statistics/>

historico\_consumo\_cpu.sql

Histórico de utilização do CPU

identifica\_servico\_sql\_iniciado.sql

Indica quando o serviço foi iniciado.

Index Fragmentados.sql

Indica índices fragmentados, page count tem que ser maior que 1500 para considerarmos a fragmentação.

jobs\_em\_andamento.sql

Indica Jobs em execução.

lock\_recurso.sql

Identificar locks (sessões que estão travando recursos, mas não tem nenhuma sessão sendo travada ainda)

max\_server\_memory.sql

Demostra a quantidade máxima memória RAM configurada para a instância.

Memoria\_alocada\_p\_database.sql

Demostra a quantidade de memória RAM utilizada por banco de dados.

PLE.sql

Demostra quanto tempo uma página de dados permanece no Buffer, abaixo de 1000 pode indicar uma pressão de memória.

querys\_em\_execV2008-.sql

Demostra o texto das querys executadas no momento, script compatível com a versão 2008.

querys\_em\_execV2012+.sql

Demostra o texto das querys executadas no momento, script compatível com a versão 2012+.

query\_SPID.sql

Demostra o texto das querys executadas no momento.

statistics\_7dias.sql

Indica as estatísticas sem atualização por mais de 7 dias.

tamanho\_arquivos\_dados.sql

Indica o tamanho dos arquivos dos bancos de dados, indicando espaço reservado, espaço livre que possivelmente pode ser liberado para o S.O., porém, não é uma boa prática realizar shrink em ambientes produtivos, pois gera muita fragmentação.

tamanho\_banco\_de\_dados.sql

Indica o tamanho dos bancos de dados.

tamanho\_discos.sql

Indica o tamanho dos discos.

tamanho\_tabelas.sql

Indica o tamanho das tabelas.

Top100QuerysMaiorConsumoCPU.sql

Top 100 querys com maior consumo de CPU.

Top100QuerysMaiorConsumoCPU\_Plan.sql

Top 100 querys com maior consumo de CPU, com o plano de execução para análise.

Top100QuerysMaiorConsumoMemoria.sql

Top 100 querys com maior consumo de Memória.

Download scripts:

