

A green background with a grid of lines that curve and warp, creating a sense of depth and perspective.

CURSO

Ajuste (Tuning) de Servidores e SGBDs

Aula 7



departamento
de engenharia
elétrica





Máquinas Virtuais versus SGBDs

- **Virtual Machine (VM):** Consiste em um software de ambiente computacional, que executa programas como um computador real, também chamado de processo de virtualização.
- “Máquinas virtuais são extremamente úteis no dia a dia, pois permitem ao usuário rodar outros sistemas operacionais dentro de uma única máquina física, tendo acesso a outros software existentes que podem ser instalados dentro da própria máquina virtual.”

Máquinas Virtuais versus SGBDs



Fonte: <https://www.ufsexplorer.com/articles/storage-technologies/virtual-machines-data-organization.php>

- **Vantagens:**

- Permite testar um sistema operacional antes de adotá-lo;
- Permite executar programas específicos em um ambiente virtualmente isolado;
- **Permite a movimentação de uma VM de um *host* para outro de forma facilitada;**
- **Permite a consolidação de vários serviços/servidores em um mesmo *host* aproveitando seu *hardware*;**

- **Desvantagens:**

- Há uma perda de desempenho no sistema operacional, mesmo que pequena. Esta perda depende do SO e do recurso avaliado;
- O canal de acesso à disco (I/O) é serial e geralmente apresenta concorrência entre as VMs no mesmo *host*;
- O acesso à memória é compartilhado entre os processadores, o que impacta no desempenho quando há muitos processos ativos em várias VMs no mesmo *host*;
- Quando há sobreposição na configuração de núcleos de processadores entre as VMs configurados no mesmo *host*, a utilização intensa pode afetar o desempenho de cada VM.

Máquinas Virtuais versus SGBDs

- **Vantagens em uso de SGBDs:**

- Quando o SGBD possui baixa carga de demanda e não há previsão de aumento dessa carga;
- Economia com hardware.
- Além de todas as vantagens listadas anteriormente.

- **Desvantagens em uso de SGBDs:**
 - O planejamento sobre o crescimento do SGBD, tanto em espaço utilizado como a carga de processamento, pode ser comprometida uma vez que tem que se adaptar ao planejamento das demais VMs no mesmo *host*;
 - A concorrência entre as diversas VMs em um mesmo host pode impactar significativamente a análise de problemas de desempenho, dado que aumenta-se bastante os pontos de problemas potenciais;
 - A ausência de uma análise de desempenho periódica sobre o *host* pode tornar mais complexa a análise de desempenho do SGBD;
 - Além de todas as desvantagens apresentadas anteriormente sobre a virtualização de máquinas.

Máquinas Virtuais versus SGBDs

- **Recomendação:**
 - Para SGBDs em ambiente de produção, deve-se evitar ao máximo sua instalação em ambientes virtualizados, a fim de se obter a melhor performance possível do *hardware* utilizado.

Tipos de dados:

- Os tipos de dados de cada coluna devem ser pensados de forma a dar preferência aos tipos que ocupam menos espaço em disco.
- Exemplo:
 - DATE vs DATETIME
 - DATE vs TIMESTAMP
 - CHAR vs VARCHAR
 - VARCHAR vs NUMBER
 - BIGINT vs INT vs SMALLINT

Utilize sempre o modelo de dados como referência no desenvolvimento

- O modelo de dados representa a estrutura de armazenamento de dados. Traz informações sobre nome, tipos, tamanho de colunas, suas tabelas e os relacionamento entre elas.
- É o guia para evitar equívocos durante a implementação de um sistema e agilizar seu desenvolvimento.

Loops de aplicação com SQL:

Situação: Quando um programa utiliza uma consulta no banco de dados, e para cada linha retornada, executa uma segunda consulta complementar.

Problema:

- Quando a consulta principal retorna poucos registros, não é percebida lentidão no tempo de resposta (Acontece normalmente quando o sistema está em desenvolvimento e as tabelas possuem poucos registros).
- Mas quando a consulta principal retorna milhares de registros, obriga a aplicação a executar milhares de consultas a gerando problema de desempenho.

Solução:

Junta a consulta principal com a consulta executada para cada linha transformando em apenas um consulta principal, dispensando a necessidade de executar milhares de consultas.

Essa junção de consultas requer certa análise para garantir mesmo resultado.

Uma consulta que retornar **milhares** de registros é muito mais performática que **milhares** de consultas retornando **uma** registros cada uma.

Garanta que haja registros suficientes no teste para detectar gargalos:

Bases de desenvolvimento ou teste com poucos registros mascara eventuais problemas de desempenho que seriam percebidos no ambiente de produção. Identificar gargalos antecipadamente leva à construção de sistemas mais robustos com uma experiência que leva o usuário a uma maior confiança no sistema.

Variáveis de substituição:

Os bancos de dados geralmente possuem solução de submissão de comandos SQL utilizando variáveis de substituição.

Exemplo:

```
SELECT CAMPO1, CAMPO2  
FROM TABELA1  
WHERE CAMPO3 = :VAR1
```

Variáveis de substituição:

Entre as vantagens de se utilizar variáveis de substituição:

- Evita invasões através do ataque chamado “SQL Injection”
- Permite que o banco de dados otimize o desempenho identificando consultas idênticas com variáveis diferentes;
- Economiza memória no servidor de banco de dados.

Arquitetura de aplicações - Regra de negócio no código ou no banco de dados?

Antes de decidir, vale a pergunta:

Há expectativa de troca de SGBD a curto ou médio prazo?

Lembrando que: A troca de SGBD geralmente representa a adaptação do código da aplicação. Ou seja, significa um alto investimento em termos de tempo e dinheiro!

Arquitetura de aplicações - Regra de negócio no código ou no banco de dados?

Se não perspectiva de troca de SGBD, recomenda-se:

- Ações que não dependem de interação direta do usuário (como rotinas batch) devem ser desenvolvidas em banco de dados;
- Ações que dependem da ação direta do usuário (como interação com telas) devem ser desenvolvidas na aplicação;
- Transformações de dados devem ficar preferencialmente localizada nas consultas, fazendo uso de funções definidas pelo usuário, se for o caso.

Arquitetura de aplicações - Aplicações Meta-modeladas

Conceito:

São aplicações onde a estrutura de dados é aberto de forma que a própria aplicação possa simular a criação de novas tabelas e colunas.

Problema:

- Enquanto a aplicação está começando, possui poucos registros, o ambiente aparenta funcionar corretamente.
- Mas quando a quantidade de registros aumenta, o problema de desempenho aparece.

Solução: Mudar para modelagem clássica ou comprar um super computador.

Arquitetura de aplicações - SGBDR vs NoSQL

Em linhas gerais:

- Se o dado é estruturado, independente da quantidade/volume, bancos de dados relacionais costumam ser a melhor opção na maioria dos casos;
- Se o dado é desestruturado (Vídeos, áudios, textos abertos, medição de sensores, etc.) e haverá análise/tratamento desse dados, bancos de dados NoSQL podem ser indicados na maioria dos casos.
- Se o dado é desestruturado e será apenas armazenado e recuperados (sem tratamento do seu conteúdo), nem sempre NoSQL é indicado.

Arquitetura de aplicações - Armazenamento de arquivos em FileSystem ou em SGBD

Embora os SGBDs possuam recurso para armazenamento de arquivos (Imagem, Word, Excel, PDF, Áudio, Vídeo, etc.), isto traz algumas desvantagens com relação a armazenamento em sistema de arquivos:

- Concorre em I/O, memória, processador e rede com consultas transacionais comuns.
- Ocupa mais espaço em disco, uma vez que o tamanho do bloco no SGBD é maior que o bloco no Sistema Operacional.

