

Banco de Dados Distribuídos

Prof. Renato Fileto
INE/CTC/UFSC

Banco de Dados

Conjunto de dados organizados, compartilhados e integrados, que seguem um conjunto de regras de formação e conteúdo.

- Normalmente este conjunto de dados se encontra centralizado, ou seja, os dados estão armazenados em apenas um local.

Sistema de Banco de Dados Distribuído (SBDD)

Coleção de nós, onde cada nó mantém um banco de dados local, mas cujo conjunto atua como um único sistema de banco de dados lógico.

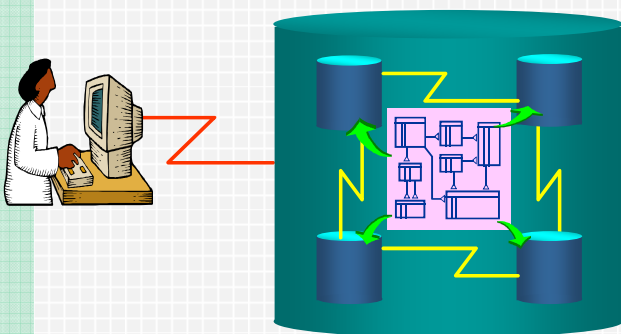
- Dados locais, mas em acordo com um esquema de banco de dados distribuídos integrado.
- Os nós podem executar consultas locais, assim como consultas que acessam dados localizados em outros nós da rede.

Independência de localização

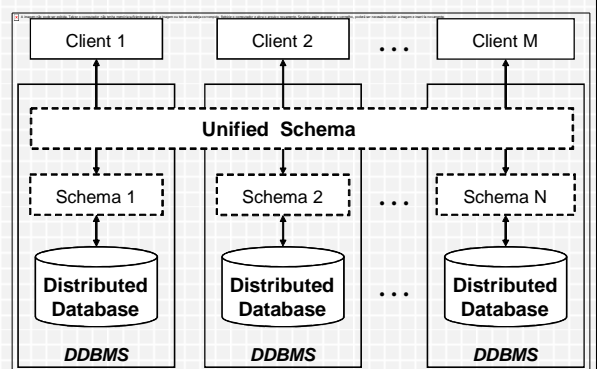
Independência de localização

Apesar de existir fisicamente a distribuição dos dados, esta deve estar em um nível lógico para o usuário, permitindo o acesso aos dados de maneira transparente.

Esquema de um SBDD

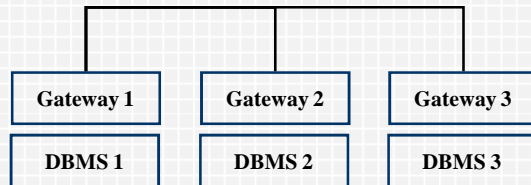


Arquitetura de um SBDD



Tipos de Sistemas de SBDD

- **Homogêneo:** Todos os nós rodando o mesmo SGBD.
- **Heterogêneo:** Diferentes nós rodando diferentes SGBDs (relacionais ou mesmo não relacionais).



Sistema de Gerenciamento de BD Distribuído (SGBDD)

- **Em 1987, C. J. Date, um dos primeiros projetistas de bancos de dados relacionais, junto com o Dr. E. F. Codd, autor da teoria relacional, propôs 12 regras que um SGBDD completo deveria seguir.**

As 12 Regras para SGBDD

- 1. Autonomia local:** Cada nó participante de um sistema distribuído deve ser independente dos outros nós. Cada nó deve prover mecanismos de segurança, bloqueio, acesso, integridade e recuperação após falha.
- 2. Não dependência de um nó central:** Um sistema de banco de dados distribuído não deve depender de um nó central, pois isso acarretaria um único ponto de falha, afetando todos os outros nós. Um nó central também poderia ficar sobrecarregado resultando em perda de desempenho do sistema.
- 3. Operação contínua:** Um sistema de banco de dados distribuído nunca deve precisar ser desativado. As operações de backup e recuperação devem ser suportadas on-line. Essas operações devem ainda ser rápidas o bastante para não afetarem o funcionamento do sistema (backup incremental, por exemplo).

As 12 Regras para SGBDD (cont.)

- 4. Transparência/independência de localização:** Os usuários do sistema não devem precisar saber o local onde estão localizados os dados; devem se comportar como se os dados estivessem armazenados localmente.
 - A transparência de localização pode ser alcançada pela utilização de sinônimos estendidos e pelo extenso uso do dicionário de dados.
 - A transparência de localização permite que aplicações sejam portadas de um nó da rede para outro sem a necessidade de modificações.
- 5. Independência de fragmentação:** As tabelas que fazem parte de um sistema de banco de dados distribuído podem estar divididas em fragmentos, localizados fisicamente em diferentes nós, de forma transparente para o usuário.
- 6. Independência de replicação:** Dados podem estar replicados em vários nós da rede, de forma transparente. As réplicas de dados devem ser mantidas sincronizadas automaticamente pelo SGBDD.

As 12 Regras para SGBDD (cont.)

- 7. Processamento de consultas distribuído:** O desempenho de uma consulta deve ser independente do local onde a mesma é submetida. Um SGBDD deve possuir um otimizador capaz de selecionar não apenas o melhor caminho para o acesso a um determinado nó da rede, mas também otimizar o desempenho de uma consulta distribuída, levando em conta a localização dos dados, utilização de CPU, I/O e o tráfego na rede.
- 8. Gerenciamento de transações distribuídas:** Um SGBDD deve suportar transações atômicas. As propriedades ACID (Atomicidade, Consistência, Independência e Durabilidade) das transações e a serialização devem ser suportadas não apenas para transações locais, mas para transações distribuídas também.
- 9. Independência de hardware:** Um SGBDD deve poder operar e acessar dados em uma variedade de plataformas de hardware. Um SGBDD verdadeiro não deve depender de uma determinada

As 12 Regras para SGBDD (cont.)

- 10. Independência de sistema operacional:** Um SGBDD deve poder executar em sistemas operacionais diferentes. Assim como na regra anterior, um SGBDD não deve depender de um sistema operacional em especial.
- 11. Independência de rede:** Um SGBDD deve ser projetado para executar independentemente do protocolo de comunicação e da topologia de rede usada para interligar os vários nós que fazem parte da rede.
- 12. Independência de SGBD:** Um SGBDD ideal deve possuir capacidades para se comunicar com outros sistemas de gerenciamento de banco de dados executando em nós diferentes, mesmo se estes sistemas de bancos de dados são diferentes (heterogêneos). Todos estes sistemas devem usar APIs

Recent Trends (Ramakrishnan and Gehrke, 1999)

- *Users have to be aware of where data is located, i.e., Distributed Data Independence and Distributed Transaction Atomicity are not supported.*
- *These properties are hard to support efficiently.*
- *For globally distributed sites, these properties may not even be desirable due to administrative overheads of making the location of data transparent.*

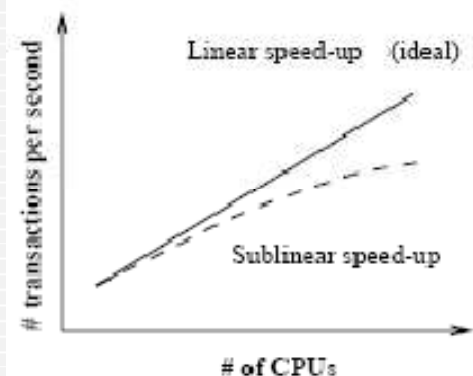
Vantagens de SGBDD (em relação a SGBD centralizado)

- **Compartilhamento de dados e controle distribuído:** Existe um administrador global, responsável pelo sistema como um todo, mas parte das responsabilidades são delegadas a administradores locais que gozam de certa autonomia.
- **Maior confiabilidade:** O sistema funciona conforme o projeto.
- **Maior disponibilidade:** O sistema está disponível por maior percentual de tempo.
- **Melhor desempenho no processamento de consultas:** Sub-consultas podem ser executadas em paralelo.
- **Maior escalabilidade:** É mais fácil acrescentar um nó, desde que os mesmos sejam autônomos, do que substituir um sistema centralizado existente por um maior.

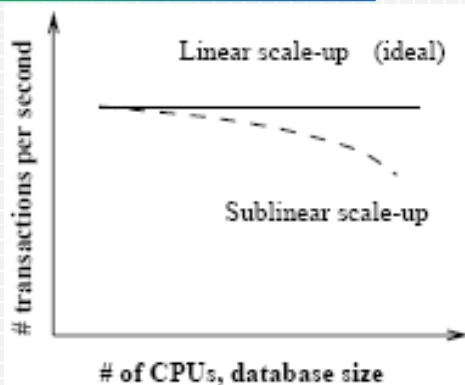
Desvantagens de SGBDD (em relação a SGBD centralizado)

- **Custo de desenvolvimento do software:** A alta complexidade torna mais difícil implementar um SGBDD, tornando-o mais caro.
- **Grande potencial para bugs:** Ocorrência de erros muito sutis na colaboração entre os nós do SGBDD.
- **Aumento do overhead de processamento:** Devido à troca de mensagens e à computação adicional para obter a coordenação entre os nós.
- **Questões de projeto específicas:** Por exemplo, replicação e fragmentação de dados.
- **Dificuldades para obter conhecimento global:** Por exemplo, controle de concorrência entre transações distribuídas e detecção de *deadlock*.

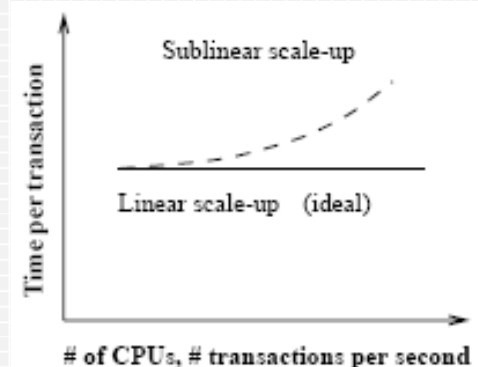
Melhoria de desempenho



Escalabilidade no tamanho da base



Escalabilidade na carga de trabalho



Fragmentação de dados

Uma relação é dividida em fragmentos, que são armazenados em diferentes nós.

- **Fragmentação horizontal:** os fragmentos são definidos por **seleção** de tuplas.
- **Fragmentação vertical:** os fragmentos são definidos por **projeção** de atributos.

Replicação de dados

- **Réplicas dos dados em diferentes nós**
- **Aumento do paralelismo**
- **Aumento da disponibilidade de dados**
- **Elevação do *overhead* de atualização**
- **Melhora desempenho das operações de leitura, enquanto acarreta maior *overhead* nas atualizações**

Exemplo de Fragmentação

TID	eid	name	city	age	sal
t1	53666	Jones	Madras	18	35
t2	53688	Smith	Chicago	18	32
t3	53650	Smith	Chicago	19	48
t4	53831	Madayan	Bombay	11	20
t5	53832	Guldu	Bombay	12	20

Vertical Fragment

Horizontal Fragment

Atualização dos dados replicados

- **Replicação síncrona:** Todas as cópias de uma relação modificada (fragmentos) deve ser atualizadas antes da transação modificante fazer *commit*
 - A distribuição de dados fica transparente para o usuário
- **Replicação Assíncrona:** As cópias da relação modificada só são atualizadas periodicamente; réplicas podem ficar inconsistentes por algum tempo
 - Os usuários devem estar cientes da distribuição e replicação
 - Produtos atuais seguem esta abordagem

Processamento de consultas em SGBDD

Deve levar em conta:

- **Réplicas dos dados**
- **Reconstrução de relações a partir de fragmentos**
- **Tempo de recuperação**
- **Tempo de processamento**
- **Transmissão de dados via rede**

Consultas Distribuídas

```
SELECT AVG(sal)
FROM TopModels M
WHERE M.age > 20
AND M.age < 35
```

- **Replicação:** cópias em vários nós
 - Escolha baseada em custos locais e de transmissão
- **Fragmentação horizontal:** Modelos com idade < 25 no Rio e idade >= 25 em SP
 - Pode-se computar **AVG(salary)** em ambos o nós
 - Como calcular a média geral dos salários
 - Pode-se trazer ambas os dados de ambas as relações
- **Fragmentação vertical:** *mid* e *age* em SP, *mid*, *name* and *sal* no Rio
 - Pode-se reconstruir a relação de modelo através de junção no atributo *mid*, e então processar a consulta

Controle de Transações em SGBDD (garantindo a Atomicidade)

Protocolo *two-phase commit*

1. Preparação:

- a) Um dos participantes é designado coordenador do *two-phase commit* e envia a cada nó envolvido na transação uma solicitação para se preparar para fazer *commit*.
- b) Uma vez preparado, cada participante escreve uma marca no seu *log*, avisa o coordenador e não pode mais abortar.

2. Commit:

- a) Se todos os participantes estão prontos, o coordenador envia um sinal de *commit* para cada participante, que o executa.
- b) Se algum dos participantes falhar na preparação para o *commit*, este notifica o coordenador, que envia um sinal de *rollback* para todos os participantes.

Controle de Transações em SGBDD (garantindo a Consistência)

Escalonamentos seriais: entrelaçamento da execução de operações de diferentes transações, com resultado idêntico ao da execução serial das transações

■ Exclusão mútua por bloqueio (*lock*)

- Troca de mensagens para solicitar bloqueios locais
- *Deadlock* é mais difícil detectar em um SGBDD

■ Métodos otimistas

- Livre execução das operações
- Monitoramento de conflitos (via grafos/árvores de execução e/ou relógios lógicos)
- Em caso de conflito, a transação mais nova pode ser desfeita, para depois ser realizada novamente

Cópia primária / secundária para locks distribuídos

- **Cópia Primária** é a que coordena a atribuição de *locks* ao item replicado no sistema distribuído
- **Cópia Secundária** atualizada de maneira síncrona com a cópia primária para assumir em caso de falha da primeira
- **Método do Site Primário / coordenador único:** todas as cópias primárias em um mesmo site
- **Método da Cópia Primária / coordenadores múltiplos:** cópias primárias em sites distintos
- **Eleição** para definição de novo coordenador em caso de falha

Sincronização de Operações de requisição de locks em SGBDD

- **Cópia primária / secundária**
 - Permite tratar dados replicados como não replicados
- **Coordenador único**
 - Simples, mas com um ponto de falha e gargalo
- **Protocolo desviado**
 - Bloqueio partilhado – leitura (somente local)
 - Bloqueio exclusivo – escrita (todas as réplicas)
- **Protocolo de eleições**
 - Gerenciamento local e trocas de requisições
- **Marcador de tempo** (cada transação recebe um marcador)
 - Centralizado (relógio local ou contador)
 - Distribuído (relógios lógicos)

Tratamento de *deadlocks* em SGBDD

- **Deteção de ciclos centralizada:** Os grafos locais são enviados para um nó central para concatenação e deteção de ciclos
- **Deteção de ciclos hierárquica:** Organização dos nós em uma hierarquia, com envio do grafo de espera de cada nó para o nó pai
- **Timeout:** Abortar uma transação quando esta estiver esperando por muito tempo, para recomeça-la logo depois

Grafos de espera

