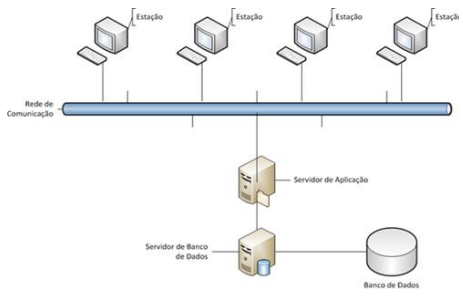


## Banco de Dados Distribuído

A necessidade de manipular e gerenciar grandes quantidades de dados de maneira confiável, junto com o interesse em descentralizar a informação, fez com que fossem impulsionados os estudos e desenvolvimentos na área de banco de dados distribuídos (BDD).

Diferente do banco de dados convencional, também conhecido como centralizado, no qual toda a massa de dados fica em um único local, no BDD, os dados ficam distribuídos em diversos servidores, e esses servidores podem estar fisicamente próximos ou, em alguns casos, geograficamente distantes.



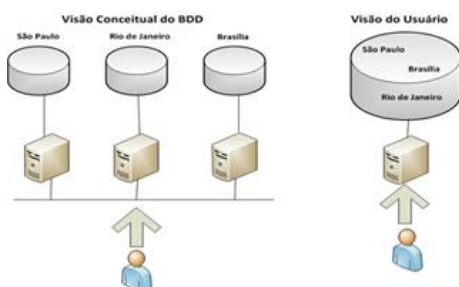
## Arquitetura de Banco de Dados Centralizado

Como os dados nesse ambiente muitas vezes estão replicados entre servidores, além de melhorar a eficiência no desempenho, consegue prover uma melhora na segurança dos dados em caso de desastres, pois não existe mais um único ponto de falha.



## Arquitetura de Banco de Dados Distribuída

A arquitetura de BDD é totalmente transparente para os usuários que utilizam aplicações nessa arquitetura distribuída, ou seja, para o usuário, os dados estão centralizados em um único servidor, mas na verdade eles podem estar “espalhados”, distribuídos em vários locais fisicamente separados.



## Arquitetura de Banco de Dados Distribuído

Na implementação desse ambiente distribuído, existem vários desafios, e manter a consistência dos dados certamente está entre os mais importantes e complexos. Garantir que os dados estão consistentes é altamente complexo, pois uma única transação pode manipular dados em diversos servidores. Para exemplificar, imagine uma instituição financeira na qual o banco de dados está distribuído em três grandes

cidades brasileiras – São Paulo, Brasília e Rio de Janeiro -, e transações distribuídas entre esses servidores são executadas o dia todo para armazenar informações e movimentações de seus clientes.

Agora imagine a situação em que o cliente José sacou R\$ 1.000,00 de sua conta corrente, e essa movimentação deve ser replicada para todos os servidores envolvidos na transação; caso contrário, o saldo de José ficará inconsistente, gerando informações diferentes em determinado servidor.

Isso pode causar grandes transtornos, pois teoricamente o usuário acessa os dados em servidores de acordo com a localidade, diminuindo assim a latência e retornando o resultado com mais rapidez, ou seja, se o usuário está acessando sua conta em São Paulo, o sistema busca os dados nos servidores de São Paulo, e assim por diante. Como no exemplo acima José sacou R\$ 1.000,00 de sua conta, imagine se ele estivesse em São Paulo e essa informação não fosse replicada para o servidor de Brasília? No servidor de São Paulo, ele estaria com determinado saldo e em Brasília com R\$ 1.000,00 a mais em sua conta.

Esse é um dos exemplos, mas poderia ser o estoque de um produto que foi vendido, porém não foi executada baixa em todos os servidores envolvidos na transação, gerando informações falsas sobre a quantidade em estoque desse produto. Claro que esse ambiente é só um exemplo simples para demonstrar a consistência dos dados.

A fim de garantir toda essa segurança e confiabilidade dos dados, o BDD preserva as propriedades ACID, onde:

- **A – Atomicidade:** Indica que a transação deve ter todas as suas ações concluídas ou não. Caso todas as ações da transação sejam terminadas com sucesso, então é executado o Commit na transação inteira; caso contrário, a transação inteira deve ser revertida, Rollback.
- **C – Consistência:** A execução de uma transação isolada preserva a consistência do banco de dados.
- **I – Isolamento:** Cada transação não toma conhecimento de outras transações concorrentes.
- **D – Durabilidade:** Depois que transação for completada com sucesso (Commit), as mudanças que ela faz no banco de dados persistem.

### **Protocolos de Efetivação**

Em ambiente distribuído, podem existir transações locais e globais. Transações locais seriam aquelas que iniciam e terminam no mesmo nó, não afetando outros servidores para realizar determinada transação. Já as globais seriam aquelas que iniciam em determinado nó e terminam em outro nó, ou iniciam e terminam no mesmo nó, porém no meio da transação foi necessário acessar outro nó para realizar alguma operação, afetando assim outro servidor em algum momento do processamento.

Os protocolos de efetivação garantem integridade de transação distribuída que atinge mais de um servidor, fazendo com que a transação só seja efetuada (Commit) ou abortada (Rollback) quando todos os servidores entram em um acordo.

- **Two-Phase Commit**

O protocolo de efetivação Two-Phase Commit é composto por duas fases, em que todos os servidores envolvidos na transação T entram em acordo se efetivaram (Commit) ou abortaram (Rollback) a transação.

- **Therree-Phase Commit**

O protocolo Therree-Phase Commit é uma continuação do protocolo de duas fases em que é adicionada uma terceira fase que busca a diminuição de falhas do coordenador. Com a implementação da terceira fase, a troca de dados entre coordenador e gerenciadores de transação aumenta o grau de complexidade e proporciona um maior tráfego na rede (overhead).

### **Vantagens na utilização de BDD**

- A confiabilidade assim como a disponibilidade visam ao funcionamento do sistema mesmo com falhas, mas sem violar a consistência do banco de dados.

- Autonomia local, um departamento pode controlar seus dados (já que é o mais familiarizado com eles).
- O aumento de desempenho é uma vantagem em um sistema de banco de dados distribuídos, a consulta pode ser subdividida e realizada paralelamente.
- Maior disponibilidade – uma falha em um banco de dados afetará somente um fragmento, em vez do banco de dados inteiro.



Falha em um servidor

### Desvantagens na utilização de BDD

- A complexidade dos problemas é uma desvantagem, porque os problemas são mais complexos do que em bancos de dados centralizados, pois além de haver os problemas que são comuns em banco de dados centralizados, que normalmente ocorrem nos servidores locais, haverá os problemas que surgem com a comunicação entre esses servidores locais.
- Implantação mais cara – o aumento da complexidade e uma infraestrutura mais extensa significam custo extra de trabalho.
- Falta de padrões – ainda não há metodologias e ferramentas para ajudar usuários a converterem um SGBD centralizado para um SGBD distribuído.

Com o crescente avanço tecnológico em comunicação de dados, BDD passou a ser uma possibilidade para trabalhar com banco de dados em grande escala, possibilitando a distribuição geográfica (WAN). Como foi demonstrado, existem algumas desvantagens devido à complexidade e aos custos, mas com certeza os pontos positivos as superam.

Essa área é muito ampla, demanda muito estudo e, conseqüentemente, vários pontos de dúvidas interessantes, pois a construção de uma arquitetura como essa envolve vários departamentos, como infraestrutura, segurança da informação e banco de dados.

Um sistema de banco de dados distribuído (BDD) consiste em uma relação de nós, cada qual podendo participar na execução de transações que acessam dados em um ou mais nós. Em um sistema de banco de dados distribuído, o banco de dados é armazenado em diversos computadores (nós).

Os computadores, em um sistema distribuído, comunicam-se uns com os outros por intermédio de vários meios de comunicação, tais como: redes de alta velocidade, redes sem fio ou linhas telefônicas, eles não compartilham a memória principal e o relógio.

A diferença principal entre sistemas de banco de dados centralizados e distribuídos é que no primeiro os dados estão localizados em um único lugar, enquanto que no outro os dados residem em diversos locais. Esta distribuição de dados é motivo de muitas preocupações e dificuldades.

Os processadores em um sistema distribuído podem variar em tamanho e função, podendo incluir microcomputadores, estações de trabalho, minicomputadores e sistemas de computadores de uso em geral.

Estes processadores são geralmente chamados de nós, dependendo do contexto no qual eles estejam mencionados. Usa-se principalmente o termo nó (lugar, posição), a fim de enfatizar a distribuição física destes sistemas. Veja o exemplo na Figura 1:

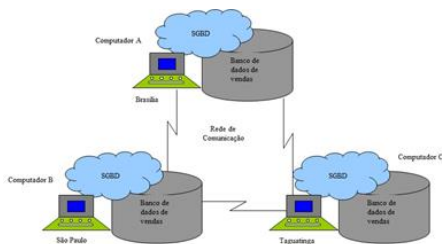


Figura 1: Figura demonstrando o sistema de banco de dados distribuído.

### Armazenamento Distribuído Dos Dados

Uma relação  $r$  (ou tabela) possui diversos enfoques para o armazenamento em um banco de dados distribuído (BDD):

- Replicação: o sistema mantém réplicas idênticas da relação, onde cada réplica é armazenada em sites diferentes, resultando na replicação dos dados
- Fragmentação: a relação é particionada em vários fragmentos, onde cada fragmento é armazenado em um site diferente
- Replicação e fragmentação: a relação é particionada em vários segmentos, e o sistema mantém diversas réplicas de cada fragmento

#### Replicação de dados

A replicação de dados significa que um determinado objeto de dados lógico pode possuir diversos representantes armazenados, em nós. O grau de suporte para a replicação é um pre-requisito para atingir o verdadeiro potencial de um sistema distribuído.

#### Fragmentação de dados

Uma relação é dividida em fragmentos, onde cada fragmento contém informação suficiente para permitir a reconstrução da relação original.

Existem duas formas de fazer a fragmentação:

- Fragmentação Horizontal: divide a relação separando as tuplas de  $r$  em dois ou mais fragmentos.
- Fragmentação Vertical: divide a relação pela decomposição do esquema  $R$  da relação  $r$ .

#### Fragmentação e Replicação de Dados

As técnicas de fragmentação e replicação podem ser aplicadas sucessivamente a uma mesma relação. Um fragmento pode ser replicado, e as réplicas podem ser fragmentadas novamente e assim por diante.

### Processamento De Consultas Distribuídas

A transparência para leitura é mais fácil de se conseguir e manter do que a transparência para atualização. O maior problema para a atualização é garantir que todas as réplicas e fragmentos sejam atualizados, após uma atualização em uma das réplicas ou fragmentos. A atualização deve ser prolongada para todas as cópias (réplicas e fragmentos) existentes no sistema.

Um dos fatores mais importantes no desempenho de uma consulta, em uma base centralizada, é a quantidade de acesso a disco necessária para atingir o resultado. Em um banco distribuído os problemas aumentam, pois existe também a preocupação com a transmissão de dados na rede. Um fator interessante para a consulta realizada em uma base distribuída é que para os diversos sites podem processar partes da consulta em paralelo.

Na realização de uma consulta simples (trivial), como consultar todas as tuplas da relação CONTA, pode caracterizar um processamento não tão trivial, pois CONTA pode estar fragmentada, replicada ou ambas.

### Transações

O acesso a diversos itens de dados em um sistema distribuído é normalmente acompanhado de transações que têm de preservar as propriedades ACID:

- A: Atomicidade
- C: Consistência
- I: Isolamento
- D: Durabilidade

### Características da ACID

- Atomicidade: Todas as operações da transação são refletidas corretamente no BD ou nenhuma será.
- Consistência: A execução de uma transação isolada preserva a consistência do banco de dados.
- Isolamento: Cada transação não toma conhecimento de outras transações concorrentes.
- Durabilidade: Depois da transação completar-se com sucesso, as mudanças que ela faz no banco de dados persistem.

### Tipos de transação

- Locais: mantem acesso e atualizam somente a base de dados local.
- Globais: mantem acesso e atualizam diversas bases de dados locais.

### Funções adicionais

- Rastreamento de dados.
- Processamento de consultas distribuídas.
- Gerenciamento de transações distribuídas.
- Gerenciamento de dados replicados.
- Recuperação de banco de dados distribuído.
- Segurança.
- Gerenciamento do diretório distribuído

### Vantagens

- Gerenciamento de dados distribuído com níveis diferentes de transparência.
  - Transparência de distribuição ou de rede.
  - Transparência de replicação.
  - Transparência de fragmentação.
- Melhoria da confiabilidade e na disponibilidade.
- Melhoria no desempenho.
- Expansão mais fácil.

## Falhas

Em um sistema de banco de dados distribuído pode sofrer os mesmos tipos de falhas que ocorrem em um sistema centralizado, porém existem falhas adicionais que podem ocorrer em um (BDD), tais como: falha de comunicação entre eles, perda de mensagens e o particionamento da rede, cada um desses problemas devem ser considerados no projeto de recuperação de um BDD. Para um sistema ser robusto, ele precisa detectar qualquer uma dessas falhas, reconfigurar-se enquanto a falha é recuperada.

Com isso finalizo este artigo, espero que o mesmo ajude mesmo que de forma básica com essas informações sobre este tipo de banco de dados.

[illegible]