

Controle de concorrência em banco de dados distribuídos

Academico: Alisson Hanael Zini Gaio
Sistemas de Informação
Sistemas Distribuídos

Introdução

As organizações modernas contam com uma variedade de bancos de dados confiáveis para suportar suas operações em diferentes plataformas, incluindo mainframes, workstations e servidores com acesso à Intranet e à Internet. O uso estratégico desses recursos contribui para a eficiência e integração nas operações, independentemente do ambiente tecnológico. Essa diversidade de plataformas reflete a necessidade de adaptação das organizações às demandas em constante mudança do cenário empresarial global.

Banco de dados

Bancos de dados é uma coleção de dados operacionais, ou uma coleção de dados lógicos com algum significado ou relação entre os mesmos, sendo este um conjunto de dados integrados e organizados, constituindo assim uma representação natural de dados, sem impor restrições ou modificações para que sejam adaptados ao computador.

Controle de concorrência

O controle de concorrência em um sistema de banco de dados distribuído refere-se à capacidade do sistema, em um ambiente multiusuário, de permitir que várias transações acessem dados simultaneamente.

O grau de concorrência é definido pelo volume de transações executadas durante um intervalo de tempo.

Controle de concorrência

Garantir a consistência integral dos dados em um ambiente multiusuário implica que as alterações realizadas por uma transação não devem afetar outras transações em andamento.

A atividade de controle de concorrência coordena os acessos concorrentes de diferentes usuários ou processos, assegurando a integridade dos dados e evitando conflitos entre transações simultâneas.

Gerencia de controle de Concorrência

O módulo responsável pelo controle de concorrência em um SGBD é conhecido como escalonador (scheduler). Sua principal função é evitar interferências entre as transações que são executadas concorrentemente.

Gerencia de controle de Concorrência

Quando dois usuários submetem transações T1 e T2 ao SGBD aproximadamente ao mesmo tempo, existem duas possíveis maneiras de ordenar as operações das transações para execução: Executar todas as operações da transação T1 em sequência, seguida por todas as operações da transação T2 também em sequência. Executar todas as operações da transação T2 em sequência, seguida por todas as operações da transação T1 também em sequência. Se a intercalação das operações é permitido, várias ordens podem ser geradas para a execução de operações individuais das transações. Isso implica que o sistema tem a flexibilidade de intercalar as operações das transações de diversas maneiras, proporcionando diferentes resultados possíveis dependendo da política de controle de concorrência adotada.

Exemplo de execução de algoritmo

Transação T ₁	Transação T ₂
Read_item(X);	Read_item(X);
X := X - N;	X := X + M;
Write_item (X);	Write_item (X);
Read_item (Y);	
Y := Y + N	
Write_item (Y)	

Tabela 3.1. Duas transações simples. Transação T₁ e T₂.

Algoritmo Timestamp Ordering:

Cada transação é atribuída um carimbo de data/hora único.

Se uma transação T1 tem um carimbo anterior a T2, então T1 ocorre antes de T2.

T.O.

Cada nova operação é comparada com operações conflitantes que já tenham sido escalonadas.

- Se a nova operação for mais nova que as operações conflitantes já escalonada, será aceita.
- Do contrário será rejeitada obrigando a transação reiniciar com um novo timestamp.
- Um escalonador de TO tem a garantia de gerar escalonamentos serializáveis.
- Além do contador local, o tempo também pode ser usado para definir o timestamp

T.O.

As operações de cada transação são inseridas em buffers até ser possível estabelecer uma relação, tal que, não seja detectada possíveis rejeições, e elas possam ser executadas nessa ordem.