

Sincronização e transação

A sincronização é um tema importante pois a forma de operação em sistemas distribuídos é a execução de processos em diferentes máquinas, e, portanto, para o sucesso da execução, torna-se imprescindível conhecer a cronologia de um evento, ou seja, saber se determinado evento aconteceu antes ou depois, evitando assim erros de interpretação. Para que seja garantida a correta sequência lógica, é necessário focar no tempo, com a utilização de técnicas de sincronização, que veremos adiante.

A utilização de bancos de dados transacionais ilustra essa necessidade; os bancos de dados transacionais são otimizados para sistemas de produção em execução, destacando-se a necessidade de leitura e gravação rápida mantendo a integridade dos dados.

Para que possamos entender a relação de sincronização em bancos de dados transacionais, o conteúdo deste capítulo foi dividido em dois grandes tópicos: sincronização e transação em sistemas distribuídos e uso de bancos de dados transacionais.

1 Sincronização e transação em sistemas distribuídos

Sincronização quer dizer fazer a coisa certa na hora certa. O maior problema em sistemas distribuídos é que os processos em máquinas diferentes têm sua própria ideia do que é o tempo, e, portanto, o sincronismo torna-se importante.

Para Coulouris, Dollimore e Kindberg (2007), o tempo é um problema prático importante, pois cada computador tem seu próprio relógio físico, e, havendo vários computadores, os relógios podem diferir entre si, o que torna a necessidade da sincronização evidente. Portanto, em sistemas distribuídos, o objetivo é criar uma falsa sensação de que as aplicações estão sendo executadas em um único sistema, com a sincronização sendo realizada através de trocas de mensagens. Um exemplo real é o comércio eletrônico: os computadores distribuídos em diversas localidades devem mostrar a indicação de tempo das transações de forma consistente e íntegra.

Há vários modos de sincronizar relógios em um sistema distribuído, mas, em essência, todos os métodos são baseados em troca de valores de relógio considerando simultaneamente o tempo que leva para enviar e receber mensagens. Variações em atrasos de comunicação e o modo como essas variações são tratadas determinam, em grande parte, a precisão de algoritmos de sincronização de relógios. (TANENBAUM; VAN STEEN, 2008, p. 163)

A sobreposição geométrica é a ideia principal para resolver o problema de sincronização. Para isso, é utilizada a distância geométrica como

medida precisa para a latência entre dois nós. Pelo fato de utilizar coordenadas, esse processo torna-se muito parecido com o método aplicado para determinar a localização e a hora em GPS. Já a exclusão mútua distribuída é uma classe de algoritmos de sincronização que garante que, em um conjunto de processos distribuídos, apenas um processo por vez tenha acesso ao recurso compartilhado. Isso é alcançado utilizando-se um coordenador que monitora de quem é a vez. A sincronização, na maior parte das vezes, requer o uso de um coordenador para tomar essa decisão de quem é a vez de acessar o recurso compartilhado.



NA PRÁTICA

A Devovx é uma conferência anual organizada pela comunidade de desenvolvedores Belgian Java User Group e considerada um dos maiores eventos não patrocinados por fabricantes. No canal do YouTube da Devovx, é possível assistir a um vídeo com um exemplo prático de sincronização de transações. É só buscar por “transaction synchronization and spring application events”.

2 Bancos de dados transacionais

Um banco de dados pode ser definido como uma coleção organizada de dados passíveis de serem armazenados e acessados a partir de um sistema de computador, tendo aplicabilidade no controle de operações corporativas e no gerenciamento de informações (ELMASRI; NAVATHE, 2017). Conforme Puga, França e Goya (2014), existem diversos modelos de banco de dados. Entre eles, destacam-se: hierárquico, relacional, orientado a objetos e objeto-relacional. Neste capítulo, daremos ênfase para os bancos de dados transacionais, que têm como principal função garantir que o usuário perceba um conjunto de procedimentos como único. Os bancos de dados transacionais são projetados para serem

executados em sistemas de produção que requerem um tempo de resposta rápido, na ordem de milissegundos, e devem contemplar as propriedades ACID.

Coulouris, Dollimore e Kindberg (2007) informam que o mnemônico “ACID” está relacionado com:

- **Atomicidade:** garantir que toda operação em uma transação seja efetuada ou que nenhuma delas seja efetuada.
- **Consistência:** garantir a integridade do dado durante e após a transação.
- **Isolamento:** garantir o controle sobre os dados para que uma transação não impacte outra transação.
- **Durabilidade:** controlar a persistência, garantindo que após o sucesso da transação o dado esteja íntegro e disponível.



PARA SABER MAIS

O site Database.Guide oferece recursos práticos e teóricos sobre bancos de dados. Busque pelo termo “ACID” para se aprofundar no assunto e adicionalmente visualize o roteiro de instalação do PostgreSQL em servidor Linux que oferece suporte ao ACID.

Em banco de dados, uma transação pode ser definida como uma unidade ou sequência de trabalho realizada em ordem lógica. Por exemplo, se você estiver criando, atualizando ou excluindo um registro da tabela, você está executando uma transação nessa tabela. É importante controlar essas transações para garantir a integridade dos dados e lidar com os erros do banco de dados.

Para que esse objetivo possa ser alcançado, temos os comandos para controlar as transações:

- COMMIT: comando utilizado para salvar transações no banco de dados.
- ROLLBACK: comando utilizado para desfazer transações que ainda não foram salvas no banco de dados.
- SAVEPOINT: ponteiro utilizado para reverter a transação para determinado ponto sem reverter toda a transação.
- SET TRANSACTION: especifica se uma transação pode ser somente leitura ou leitura e gravação.

Para ilustrar alguns desses comandos transacionais, considere o exemplo a seguir para a tabela de carros. A saída do comando SELECT mostra o conteúdo original da tabela com 4 carros. Um ponteiro com a referência dos dados da tabela original foi criado através do comando SAVEPOINT. Posteriormente, um comando de remoção foi enviado para apagar os carros com o ano de fabricação igual a "2017". E então um pedido para desfazer a transação através do comando ROLLBACK foi emitido, de modo que a tabela voltou para a sua condição original.

```
SQL> SELECT * FROM CARROS;
```

ID	NOME	ANO
1	Gol	2019
2	Voyage	2018
3	Palio	2019
4	Celta	2017

(cont.)

4 rows selected.

```
SQL> SAVEPOINT SP1;
```

Savepoint created.

```
SQL> DELETE FROM CARROS WHERE ANO=2017;
```

1 row deleted.

```
SQL> SELECT * FROM CARROS;
```

+	+	+	+
	ID	NOME	ANO
+	+	+	+
	1	Gol	2019
	2	Voyage	2018
	3	Palio	2019
+	+	+	+

3 rows selected.

```
SQL> ROLLBACK TO SP1;
```

Rollback complete.

```
SQL> SELECT * FROM CARROS;
```

(cont.)

+---+-----+-----+		
ID	NOME	ANO
+---+-----+-----+		
1	Gol	2019
2	Voyage	2018
3	Palio	2019
4	Celta	2017
+---+-----+-----+		
4 rows selected.		

Considerações finais

Neste capítulo, aprendemos que a sincronização em sistemas distribuídos é fundamental para assegurar a consistência da transação, e o uso de relógios é uma forma de garantia para esse correto funcionamento.

Vimos também as características principais de um banco de dados e como um banco de dados transacional é de suma importância para garantir a integridade dos dados, a baixa latência e a operação e atualização em tempo real, assegurando os requisitos das aplicações críticas que demandam esse tipo de comportamento. Um exemplo real foi demonstrado, fazendo-se uso dos comandos de controle de transações em banco de dados.

Referências

COULOURIS, George; DOLLIMORE, Jean; KINDBERG, Tim. **Sistemas distribuídos: conceitos e projeto**. 4. ed. Porto Alegre: Bookman, 2007.

ELMASRI, Ramez; NAVATHE, Shamkant. **Sistemas de banco de dados**. 7. ed. São Paulo: Pearson Education do Brasil, 2017.

PUGA, Sandra; FRANÇA, Edson; GOYA, Milton. **Banco de dados: implementação em SQL, PL/SQL e Oracle 11g**. São Paulo: Pearson Education do Brasil, 2014.

TANENBAUM, Andrew S.; VAN STEEN, Maarten. **Sistemas distribuídos: princípios e paradigmas**. São Paulo: Prentice Hall, 2008.