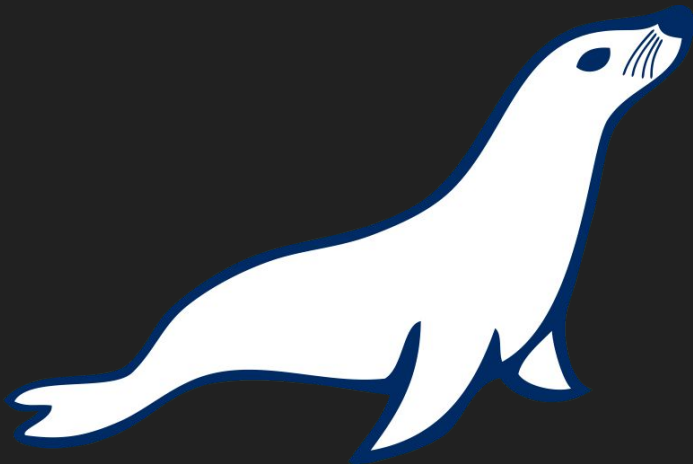


Trabalho de Banco de Dados II

Bloqueios no MariaDB e PostgreSQL

Daniel Sant' Anna Andrade
Luan Gadioli Mendonça Berto

MariaDB



Isolamento

- Transações isoladas uma das outras, de modo que cada uma tenha a impressão de que nenhuma outra transação está sendo executada concorrentemente a ela.
- O módulo de controle de concorrência cuida desse problema, controlando a forma como as operações das transações são intercaladas durante a execução concorrente.
- Ele tem por base a propriedade de serialização

Lock

- um bloqueio (lock) é uma variável associada a um item de dado
- descreve a condição do item em relação às possíveis operações que podem ser aplicadas a ele.
- geralmente há um bloqueio para cada item de dado
- deadlock -> quando eu tenho duas transações uma esperando a outra liberar o lock.

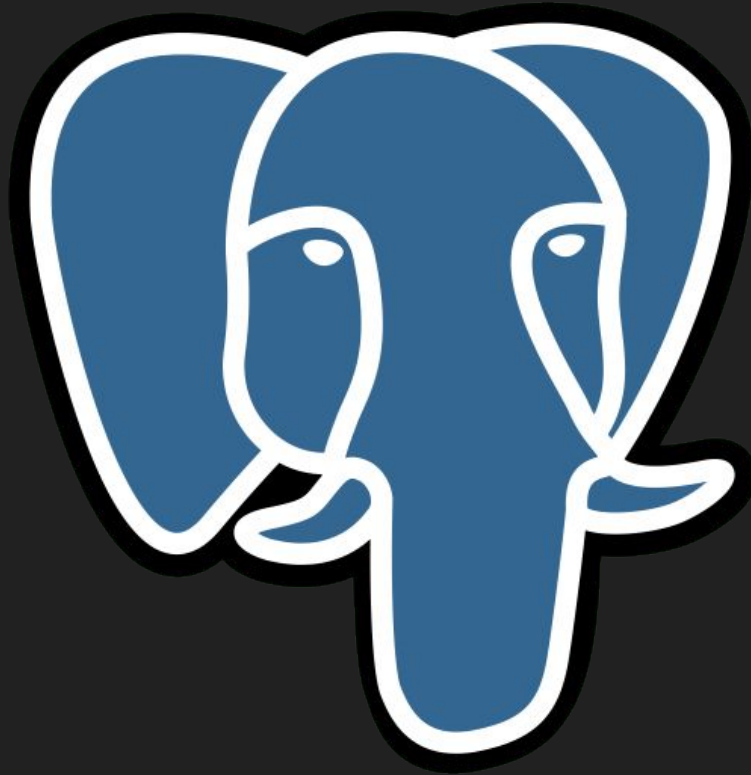
T1	T2
<code>rlock(X)</code> <code>ler(X)</code>	<code>rlock(Q)</code> <code>ler(Q)</code>
<code>wlock(Q)</code> ** espera **	<code>wlock(X)</code> ** espera **

- MariaDB permite as sessões cliente adquirir bloqueios de tabela explicitamente com a finalidade de cooperar com outras sessões no acesso de tabelas, ou para impedir com que outras sessões alterem tabelas durante períodos onde uma sessão requer acesso exclusivo à elas. Uma sessão pode adquirir ou liberar bloqueios somente por si mesma.
- `LOCK TABLES` explicitamente, adquire bloqueios de tabela para a sessão cliente atual.
- `UNLOCK TABLES` explicitamente libera qualquer bloqueio de tabela realizado pela sessão atual.

bloqueios no MariaDB

- livre de deadlock
- Método de bloqueio para escrita
 - Se não existirem travas na tabela, coloque uma bloqueio de escrita na mesma
 - Caso contrário, coloca a requisição de trava na fila de bloqueios para escrita.
- Método de bloqueio para leitura
 - Se não existirem travas na tabela, coloca um bloqueio de leitura na mesma
 - Caso contrário, coloca a requisição de trava na fila de bloqueios para leitura.

POSTGRESQL



MVCC

(Multiversion Concurrency Control)

- Melhorar a performance das transações no ambiente multiusuário;
- Processos de leitura não bloqueiam processos de escrita e vice-versa;
- Cada transação trabalha com uma imagem dos dados;
- Em cada registro de cada tabela existem dois campos utilizados no controle de transações: xmin, xmax;
 - Xmin informa o id da transação;
 - Xmax determina o id da transação a alterou;
- Necessidade de realizar vacuum de tempos em tempos.



MVCC helps me deal with concurrency. This means I store multiple versions of a row instead of over-writing.

xmin : stores XID of the transaction that inserted a row

xmax : stores XID of the transaction that deleted or updated a row



XID=1

INSERT (A,1)



XID=2

INSERT (B,2)



XID=3

DELETE (A,1)



XID=4

UPDATE (B,3)

A	1
B	2
A	1
B	3

After XID=1

xmin	xmax
1	

After XID=2

xmin	xmax
2	

After XID=3

xmin	xmax
1	3

After XID=4

xmin	xmax
2	4

After XID=3

xmin	xmax
3	

After XID=4

xmin	xmax
4	

Modos de Bloqueio

- Os modos de bloqueios são vinculados a ações específicas para controlar o acesso simultâneo aos dados;
- Podem ser automáticos (MVCC controla) ou explícita (MVCC não possui bloqueio);
- As operações que alteram a estrutura da tabela precisam bloquear operações concomitantes.
- Os bloqueios podem ser por linha ou por tabela
 - Bloqueio de linha: É obtido quando a tupla é atualizada ou excluída.
 - Bloqueios de tabela: Dependem das operações realizadas.

Bloqueios de Tabela

Access Share	Solicitado por instruções SELECT e ANALYZE apenas permitindo simples leitura. Conflita com Access Exclusive.
Row Share	Solicitado por instrução SELECT FOR UPDATE. Conflita com: Exclusive e Access Exclusive.
Row Exclusive	Solicitado por instruções INSERT, UPDATE, DELETE.
Share Update Exclusive	Bloqueio do comando VACUUM sem a opção FULL. Bloqueia a tabela inteira.
Share	Bloqueia toda a tabela e é requerido por instruções CREATE INDEX.
Share Row Exclusive	Similar ao Exclusive, mas apenas no nível de linha. Não há instrução que automaticamente solicite este bloqueio.
Exclusive	É requerido automaticamente apenas em tabelas de catálogo em determinadas operações.
Access Exclusive	Conflita com todos os outros. Solicitado por instruções ALTER TABLE, DROP TABLE, TRUNCATE, REINDEX, CLUSTER e VACUUM FULL.

Impasses ou DeadLocks

- Bloqueios explícitos pode causar impasses;
- Ocorre quando dois ou mais processos bloqueados tentam obter um bloqueio no objeto que o outro bloqueou;
- O PostgreSQL detecta automaticamente as situações de impasse;
- A resolução ocorre abortando um dos processos e enviando uma mensagem para o arquivo erro log do SQL Server.

Seção 1 bloqueia a Tabela A
Seção 2 bloqueia a Tabela B
Seção 1 tenta bloquear a Tabela B
Lock da Seção 1 pois a Tabela B está bloqueada pela Seção 2
Seção 2 tenta bloquear a Tabela A
Lock da Seção 2 pois a Tabela A está bloqueada pela Seção 1
DeadLock

Impedindo Deadlocks

- Verifique a modelagem do banco de dados;
- Não permitir interferência de usuário nas transações;
- Transações curtas no SQL;
- Utilizar procedures para manter as transações dentro de um único lote, diminuindo a conexão com o SQL Server;
- Reduzir a quantidade de leituras;
- Diminuir ao máximo o tempo de bloqueio. Definir bloqueios como último caso e liberá-los o mais rápido possível.