
sr.nimbus

SQL Server 2008

SQL07 – Recursos de Otimização para o desenvolvedor

Módulo 03 – Gerenciamento de concorrência

Propriedades ACID

- **Atomicidade**
  - Uma transação deve ser uma unidade atômica de trabalho: ou todas as modificações são efetuadas, ou nenhuma.
- **Consistência**
  - Quando efetivada, a transação deve deixar todos os dados e estruturas relacionadas em um estado consistente.

Propriedades ACID

- **Isolamento**
  - Uma transação não vê os dados modificados de outra transação enquanto ainda em execução. Somente podem ser acessados dados de antes ou depois da transação estar completa.
- **Durabilidade**
  - Transações devem ser persistidas, mesmo em caso de falha do sistema.

Transações

- **AutoCommit**
  - Padrão do SQL Server: toda instrução é uma transação
  - Erros de compilação resultam na falha do batch
  - Erros de runtime permitem parte do batch ser executado
  - Erro não realiza ROLLBACK – responsabilidade da aplicação
- **Implicit**
  - Instruções de modificação de dados iniciam automaticamente uma nova transação (SELECT, INSERT, UPDATE, DELETE, CREATE, ALTER, ...)
  - Aplicação deve executar COMMIT ou ROLLBACK

Transações

- **Explicit**
  - BEGIN TRAN
  - COMMIT TRAN
  - ROLLBACK TRAN
- **Outros recursos**
  - BEGIN TRAN <nome>
  - BEGIN TRAN WITH MARK <descricao>
  - SAVE TRAN <nomePonto>

Tipos de bloqueios

- Shared Lock (compartilhado)
- Exclusive Lock (Exclusivo)
- Update Lock (gateway para exclusive)
- Intent Lock
- Key Range Lock
- Metadata Lock
- BU Lock

Hierarquias de bloqueios

- Linha (Row)
- Chave (Key)
- Página (Page)
- Extent
- HOBT (partition lock)
- Objeto (Object)
- Banco de dados (Database)

Compatibilidade de bloqueios

	Bloqueio concedido								
Bloqueio requisitado	IS	S	U	IX	SIX	X	SchS	SchM	BU
IS	Sim	Sim	Sim	Sim	Sim	Não	Sim	Não	Não
S	Sim	Sim	Sim	Não	Não	Não	Sim	Não	Não
U	Sim	Sim	Não	Não	Não	Não	Sim	Não	Não
IX	Sim	Não	Não	Sim	Não	Não	Sim	Não	Não
SIX	Sim	Não	Não	Não	Não	Não	Sim	Não	Não
X	Não	Não	Não	Não	Não	Não	Sim	Não	Não
SchS	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Não	Sim
SchM	Não	Não	Não	Não	Não	Não	Não	Não	Não
BU	Não	Não	Não	Não	Não	Não	Sim	Não	Sim

DMVs relacionadas

- sys.dm_tran_locks
- sys.dm_tran_session_transactions
- sys.dm_tran_active_transactions
- Sys.syslockinfo (compatibilidade)
- (sp_lock)

Bloqueios dinâmicos

- **Estratégia de bloqueio é dinâmica**
  - O SQL Server utiliza a estratégia de menor custo (linha, página, tabela) em tempo de execução, baseado nos cálculos do otimizador de consultas.
  - Cada lock ocupa aproximadamente 100 bytes

Lock escalation

- **Utilizada para reduzir o número de bloqueios adquiridos por uma transação.**
  - Lock manager tenta substituir os bloqueios de linha ou página por um único bloqueio de tabela.
- **“De-escalation” nunca ocorre**
- **Baseado em thresholds internos**

Lock timeout

- **Configurável pela aplicação**
  - SET LOCK_TIMEOUT 10000
  - Faz o rollback da transação
- **Internal lock timeout**
  - Podem acontecer diversos timeouts internos que não são vistos por nós.
  - Não faz o rollback da transação

Níveis de isolamento

- **Níveis de isolamento (isolation levels): como uma transação impacta em outra**
- **5 níveis no SQL Server**
  - Read Uncommitted
  - Read Committed
  - Repeatable Read
  - Serializable
  - Snapshot

Níveis de isolamento

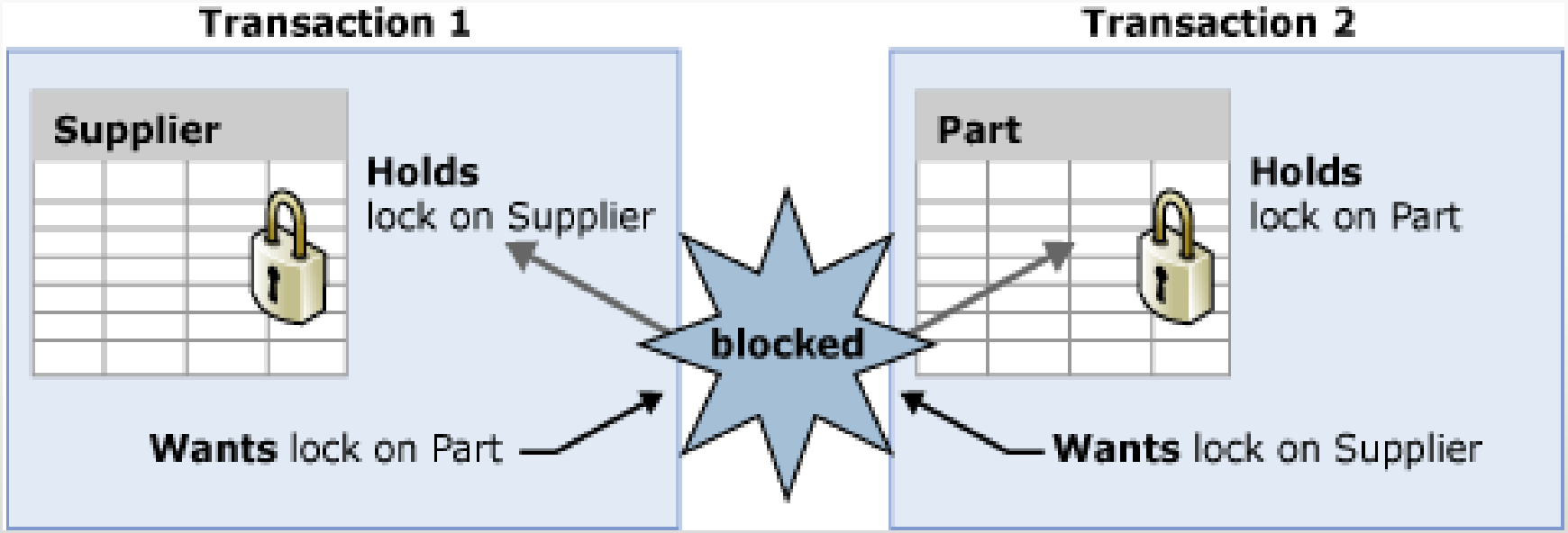
- Identificados através dos fenômenos relacionados.
- Duração dos bloqueios variam

Isolation Level	Dirty Read	Non-Repeatable Read	Phantom
Read Uncommitted	Sim	Sim	Sim
Read committed	Não	Sim	Sim
Repeatable read	Não	Não	Sim
Serializable	Não	Não	Não

Deadlocks

- **Sem tradução!**
- **Exemplo:**
  - User01 consegue bloqueio X na página 100
  - User02 consegue bloqueio X na página 200.
  - User01 requisita bloqueio S na página 200
  - Fica esperando User02 finalizar
  - User02 requisita bloqueio S na página 100
  - Fica esperando User01 finalizar
  - Espera infinita...

Deadlocks


Transaction 1
Transaction 2
Supplier
Holds
lock on Supplier
Part
Holds
lock on Part
blocked
Wants lock on Part
Wants lock on Supplier

Deadlocks

- **O SQL Server escolhe como vitima a transação onde o rollback é mais barato.**
- **É feito rollback na transação vítima**
- **Vítima é notificada pelo erro 1205:**
  - Error 1205: Your transaction (process ID #%d) was deadlocked with another process and has been chosen as the deadlock victim. Rerun your transaction

Deadlocks

- **SQL Server profiler**
  - TF-1204 (SQL2000)
  - TF-1222 (SQL2005)
- **Como resolver problemas de deadlock?**
  - Minimizar tamanho de transações
  - Acessar recursos compartilhados na mesma ordem
- **E se o código não é seu ou o deadlock for by-design?**
- **SET DEADLOCK PRIORITY High/Medium/Low**

Hints para bloqueios

- Granularity hints: ROWLOCK, PAGLOCK, TABLOCK
- Isolation LEVEL hints:
- READUNCOMMITTED (NOLOCK), READCOMMITTED, REPEATABLEREAD, SERIALIZABLE (HOLDLOCK)
- UPDLOCK: utiliza update lock ao invés de shared quando ler registros
- XLOCK: usa um lock exclusivo
- TABLOCKX: lock exclusivo de tabela
- READPAST: “pula” registros bloqueados

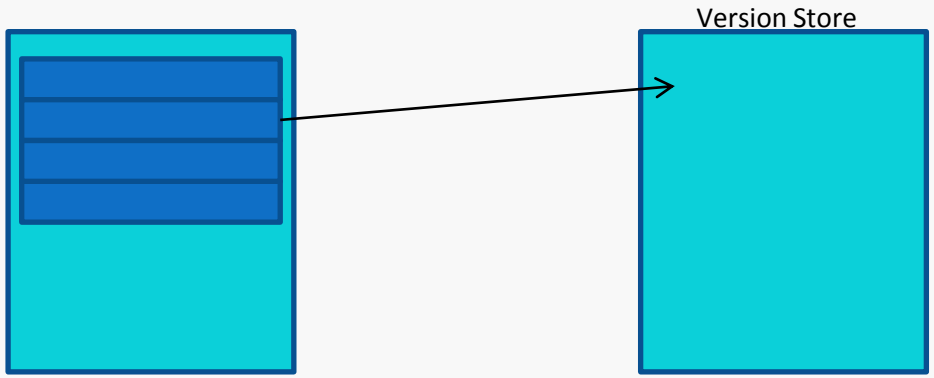
Isolamento Snapshot

- Snapshot Isolation Level é o novo nível de isolamento disponível a partir do SQL Server 2005 que utiliza o recurso de versionamento de registros para permitir que múltiplas leituras / única escrita possa ser realizado sobre um registro. Diferente do read uncommitted, esse nível de isolamento garante consistência transacional para um momento específico no tempo e não sofre com os possíveis problemas de concorrência do nível de isolamento padrão, o read committed.
- É utilizado por triggers (tabelas inserted e deleted), MARS e online index rebuilding.

Isolamento Snapshot

- **Mecanismo usado pelo SQLServer para implementar o nível de isolamento Snapshot.**
- **Área de armazenamento na tempdb que mantém páginas com o histórico dos registros e seu XSN. Cada registro na tabela possui um ponteiro (RID) para outro registro na version store.**
  - XSN (Transaction Sequence Number): número incremental único para toda a instância. Volta para 1 quando a instância é reiniciada.

Isolamento Snapshot



Begin transaction
Update set col1 = v1.2
Commit transaction

Col1 = v1.2 XSN: 50

Begin transaction
select * from tabela
select * from tabela
Commit transaction

XSN: 51
XSNs: 50

Begin transaction (RCSI)
select * from tabela
select * from tabela
Commit transaction

XSN: 52
XSNs: 50, 51
XSN: 53
XSNs: 51

Col1 = v1.2 XSN: 50

Isolamento Snapshot

- **Somente são armazenadas as versões enquanto as transações estão abertas**
  - Thread em background entra freqüentemente para limpar as versões que não estão mais sendo utilizadas.
- **Operações de leitura recuperam a última versão de cada registro que estavam comitadas no momento que a transação snapshot iniciou.**

Isolamento Snapshot

- A situação da version store e das transações pode ser acompanhada através das DMVs (Dynamic Manangement Views)
- sys.dm_tran_active_snapshot_database_transactions
- sys.dm_tran_version_store
- sys.dm_tran_active_transactions
- sys.dm_tran_current_snapshot
- sys.dm_tran_top_version_generators

Demo

Snapshot isolation e version store

Olhando o version store em detalhes

Blocker script

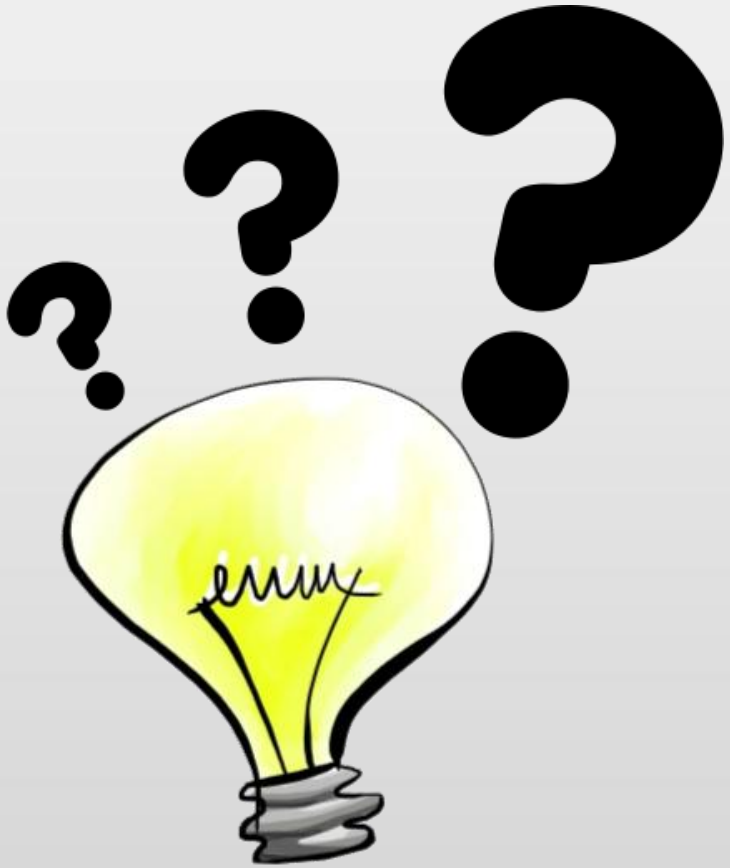
- **Blocking scripts poll the sysprocesses table to look for blocking. When blocking is detected they collect additional information.**
  - DBCC PSS
  - DBCC INPUTBUFFER
  - Sp_lock
- **Before using:**
  - Consider script's sample interval (WAITFOR DELAY), volume of information collected, etc
  - Test any scripts before using

Conclusão

- **O uso de transações impede o acesso a dados inconsistentes, além de proteger suas modificações de falhas no servidor.**
- **É comum a ocorrência de bloqueios e deadlocks decorrente do uso de locks nas transações.**
  - Minimizados através de conceitos como tipo e granularidade de locks.

Conclusão

- **O desenvolvedor tem papel fundamental em evitar estes problemas:**
  - Uso consciente de transações.
  - Seleção do melhor nível de isolamento transacional.
  - Uso das ferramentas de monitoração para detecção e resolução de bloqueios e deadlocks.
- **Novos níveis de isolamento: SNAPSHOT e READ COMMITED SNAPSHOT**
  - Aumentam a concorrência do nível default de isolamento (READ COMMITED), sem comprometer sua consistência.



Dúvidas

Recursos

- http://blogs.msdn.com/sqlserverstorageengine/archive/2006/05/17/Lock-escalation.aspx
- SQL Server 2005 Row Versioning-Based Transaction Isolation
- http://sqlblogcasts.com/blogs/tonyrogerson/archive/2006/11/16/1345.aspx