Locking and Blocking

Kalen Delaney
www.SQLServerInternals.com

Kalen Delaney

Background:

- MS in Computer Science from UC Berkeley
- Working exclusively with SQL Server for over **28** years
- Contracted by both Sybase and Microsoft to develop and teach internals courses to Tech Support staff

- Author: *SQL Server Internals: In-memory OLTP* (Red Gate, 2014)
- Primary Author*: SQL Server 2012 Internals* (O'Reilly, 2013)
- Author*: SQL Server Concurrency* (Red Gate, 2011)
- Primary Author*: SQL Server 2008 Internals* (MS Press, 2009)
- Primary Author: *Inside SQL Server 2005* (MS Press, 2007)
- *SQL Server Magazine* columnist and contributing editor
- Editor for Red-Gate: *SQL Server Stairways*

Topics

- Transactions in SQL Server
- Aspects of Locking
- Blocking
- Controlling Locking
- Troubleshooting Tools
- Best Practices

Transaction Control in TSQL

- Autocommit Transaction
  - Statement level implicit transaction
- Explicit Transaction
  - BEGIN TRANSACTION
  - COMMIT / ROLLBACK TRANSACTION
- Implicit Transaction Mode (not recommended)
  - SET IMPLICIT_TRANSACTIONS ON
  - sp_configure ‘user options’, 2

Nesting Transactions

- Nesting is only possible syntactically
  - There is at most ONE open transaction
- Successive BEGIN TRAN statements increment @@trancount
- Each COMMIT TRAN decrements @@trancount
  - When @@trancount reaches 0, COMMIT occurs
- ROLLBACK TRAN sets @@trancount to 0
- Useful for transaction control in stored procedures

Aspects of Locking

- Type of Lock
- Duration of Lock
- Granularity of Lock

Types of Locks

- Shared Lock
- Exclusive Lock
- Update Lock

Lock Duration

Duration is dependent on 'owner' (scope)

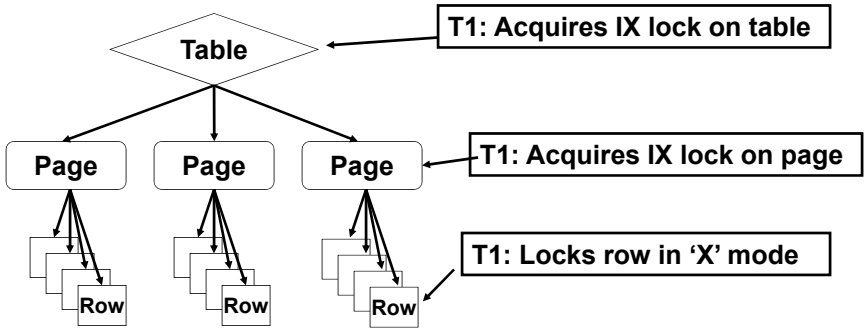
- SHARED_TRANSACTION_WORKSPACE (Resource = DATABASE)
  - Held as long as connection is using DB context
  - Prohibit some status changes
- EXCLUSIVE_TRANSACTION_WORKSPACE (Resource = DATABASE)
  - Acquired during DROP DATABASE, RESTORE
  - Acquired during status changes
- Transaction Locks
  - Shared locks held until done reading
  - Exclusive locks held until end of transaction
- Cursor Locks
  Scroll locks held until next FETCH
- Session Locks (Resource = DATABASE)
  Only used with Application Locks

Granularity of Locks: Resources

- Row
  - RID – row of a heap
  - KEY – row of an index
- Page
- Table
- Partition
  - After escalation
  - During partition operations such as SWITCH or REBUILD
- Extent
  - During allocation or deallocation
- Database

Multi-Granular Locking

- To lock a fine level, SQL server places intent locks at higher levels



Another Transaction, T2, cannot obtain X lock on page or table

Locking Granularity and Escalation

- Default is RID or KEY lock
- After optimization, SQL Server may decide to lock pages or the whole table
  - If too much memory is used for locks
  - If optimizer has decided to do a scan
- RID/KEY/PAGE locks can be escalated during execution

  Tradeoff is resources used vs. concurrency
- By default, escalation is to a table; can be overridden to escalate to partition, or to never escalate (2008)

```
ALTER TABLE <table_name>
   SET ( LOCK_ESCALATION = {TABLE | AUTO | DISABLE })
```

Key Range Locking

To support SERIALIZABLE transactions:

- Lock sets of rows controlled by a predicate

  WHERE salary between 35000 and 45000
- Need to lock data that doesn't exist!

  If "where salary between 35000 and 45000" doesn't return any rows the first time, it shouldn't return any on subsequent scans
- Earlier version locked larger units to prevent phantoms

  Prior to SQL Server 7.0, SQL Server used page and table locks

Blocking

- Occurs when one process requests a lock on the same resource held by another process in an incompatible mode
- What locks are compatible?

	Lock Mode Already Granted				
Mode Requested	IS	S	U	IX	X
IS	Yes	Yes	Yes	Yes	No
S	Yes	Yes	Yes	No	No
U	Yes	Yes	No	No	No
IX	Yes	No	No	Yes	No
X	No	No	No	No	No

Schema Locks

- Sch-S (Schema Stability) – compatible with all locks except Sch-M

  Acquired when compiling a query or scan with NOLOCK

- Sch-M (Schema Modification) -- not compatible with anything

  Acquired during certain ALTER TABLE operations

- Low Priority Waits for Schema Locks
  - New in SQL Server 2014
  - Online Index Rebuild and Partition Switching
  - Other processes will not 'stack up behind'
  - After <n> minutes, low priority waiter can:
    - Give up
    - Kill the blocker
    - Revert to regular wait status

Controlling Locking

- Lock Hints (use WITH keyword)
  - Unit
  - Duration
  - Type
  - READPAST
- Lock Timeout
  - Value set in milliseconds, can't be set globally
  - Transaction does NOT rollback
  - Check for error 1222, or use SET XACT_ABORT
- Isolation Level

Tools for Troubleshooting Blocking

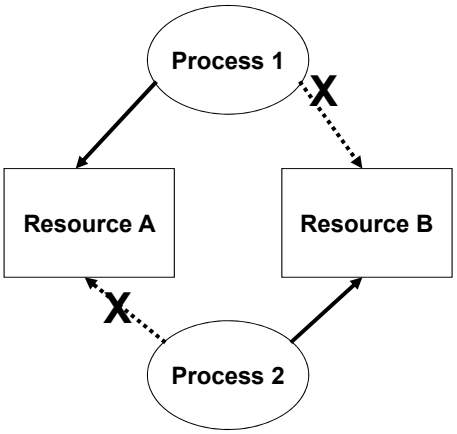
- Extended Events
- SQL Trace
- Performance Monitor
- Dynamic Management Objects
  - sys.dm_tran_locks
  - sys.dm_tran_active_transactions
  - sys.dm_tran_current_transaction
  - sys.dm_os_waiting_tasks
  - sys.dm_db_index_operational_stats
  - sys.dm_exec_requests

Deadlock

- What is Deadlock?
- Handling Deadlock
- Minimizing Blocking and Deadlocking

What is Deadlock?

- Two processes mutually blocking each other



Handling Deadlock

- SQL Server automatically detects deadlock
  - Checks for cycles at regular intervals
  - Checks more often if there are frequent deadlocks
- Process with cheapest transaction is chosen as victim
  - Transaction rolled back
  - Error message 1205
- Developer must check for 1205
  - Pause briefly
  - Resubmit
  - Keep track of recurrent deadlocks
  - Occasional deadlocks are not a major problem

How to Minimize Blocking and Deadlocking

- Keep transactions short and in one batch
- Identify and tune long running queries
- No user interaction during transactions
- Process results quickly and completely
- Reduce isolation level to READ COMMITTED
- Stress test at maximum projected user load
- Rollback when canceling, or on any error or timeout

Check *sys.dm_exec_sessions* to view transaction nesting depth

Check *sys.dm_tran_database_transactions* and *sys.dm_tran_active_transactions* for info on transaction state

Summary

- Transactions in SQL Server
- Aspects of Locking
- Blocking
- Controlling Locking
- Troubleshooting Tools
- Best Practices

Thank You!