

# Deadlocking in SQL Server



**Klaus Aschenbrenner**

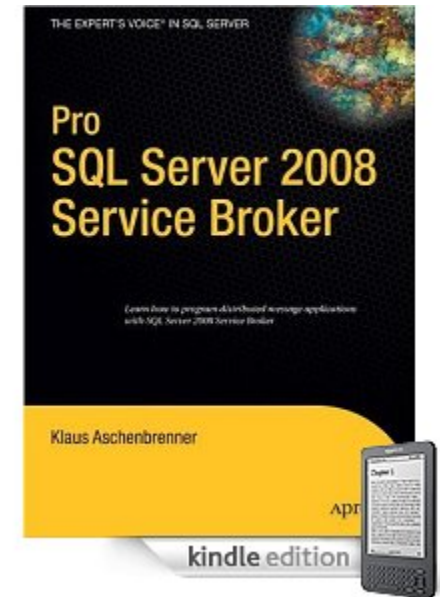
Microsoft Certified Master SQL Server 2008

[www.SQLpassion.at](http://www.SQLpassion.at)

Twitter: @Aschenbrenner

# About me

- CEO & Founder SQLpassion
- International Speaker, Blogger, Author
- SQL Server 2008 MCM
- "Pro SQL Server 2008 Service Broker"
- Twitter: @Aschenbrenner
- SQLpassion Academy
  - <http://www.SQLpassion.at/academy>
  - Free Newsletter, Training Videos



# Agenda

- Transactions & Isolation Levels
- Deadlocking

# Agenda

- Transactions & Isolation Levels
- Deadlocking

# Transactions

- 4 Properties (ACID)
  - Atomicity
  - Consistency
  - **Isolation**
  - Durability

# Concurrency Models

- Pessimistic Concurrency
  - Preventative Approach
  - Limited concurrent access
  - Uses blocking to avoid conflicts
  - Readers block writers
  - Writers block readers
- Optimistic Concurrency
  - Allows full concurrent access
  - Uses Row versioning
  - Conflict detection

# Pessimistic Isolation Levels

- Defines how readers acquire locks
- Read Uncommitted
  - Acquires no Lock during reading
  - Same as NOLOCK query hint
- Read Committed
  - Acquires Shared-Lock during Resource Access
- Repeatable Reads
  - Acquires Shared-Lock until the end of the Transaction
- Serializable
  - Acquires Shared Key-Range Locks until the end of the Transaction
  - Needs a good Indexing Strategy!



# Optimistic Isolation Levels

- Readers do not block writers
- Writers do not block readers
- 2 new Isolation Levels
  - Read Committed Snapshot Isolation (RCSI)
  - Snapshot Isolation (SI)
- Based on Row Versioning
  - Readers don't acquire Shared Locks
  - Readers get a previous committed version of the row
  - Previous versions are stored in TempDb
  - UPDATE, DELETE statements write old versions transparently to TempDb



# Transaction Isolation Levels

	Dirty reads	Non-Repeatable Reads	Phantoms	Update Conflict	Concurrency Model
READ UNCOMMITTED	Yes	Yes	Yes	No	
READ COMMITED					
• Locking	No	Yes	Yes	No	Pessimistic
• Snapshot	No	Yes	Yes	No	Optimistic
REPEATABLE READS	No	No	Yes	No	Pessimistic
SNAPSHOT	No	No	No	Yes	Optimistic
SERIALIZABLE	No	No	No	No	Pessimistic

# Lock Types

- Shared Lock
  - Many readers
- Update Lock
  - Reading with the intent to modify but has NOT yet modified
  - Allows better concurrency & avoids Deadlocks
- Exclusive Lock
  - Has made a modification and the transaction is still pending

# Lock Types

	Lock HELD		
	Shared (S)	Update (U)	Exclusive (X)
Shared (S)	Granted	Granted	WAIT
Update (U)	Granted	WAIT	WAIT
Exclusive (X)	WAIT	WAIT	WAIT

# Agenda

- Transactions & Isolation Levels
- Deadlocking

# Deadlock Handling

- SQL Server automatically detects deadlocks
  - Checks for cycles at regular intervals
  - Checks more often if there are frequent deadlocks
- Process with the cheapest transaction is the victim
  - Transaction rolled back
  - Error Message 1205
- Developer must check for 1205
  - Pause briefly
  - Resubmit
  - Keep track of recurrent deadlocks

# Deadlock Troubleshooting

- SQL Server Profiler
  - Deadlock Graph
  - XML format
- Trace Flag 1222
  - Logs Deadlocks to the SQL Server Error log
  - "XML" format
- Extended Events
  - Capturing historical Deadlock Information

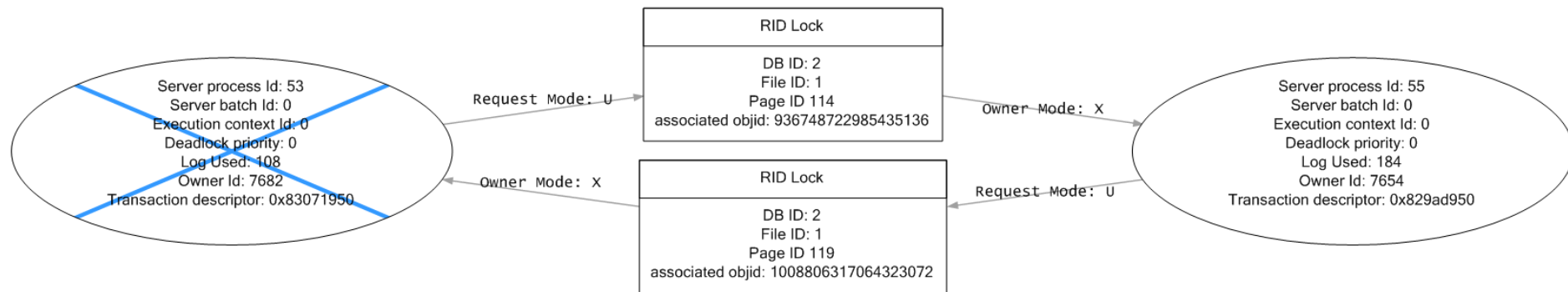
# SQL Server Profiler

- Deadlock Graph
  - XML Format
- Process List
  - Lists each process
  - Wait Resource
  - Execution Stack (T-SQL Statements)
- Resource List
  - Lists each resource
  - Owner List (Aquired Locks)
  - Waiter List

```
<deadlock-list>  
  <deadlock victim="process5e12e08">  
    <process-list>...  
    <resource-list>...  
  </deadlock>  
</deadlock-list>
```



# Deadlock Graph



# Deadlock XML

```
<deadlock-list>
  <deadlock victim="process5e12e08">
    <process-list>
      <process id="process5e12e08" taskpriority="0" logused="108" waitresource="RID: 2:1:114:0" waittime="611" ownerId="7682" transactionname="u
      <process id="process5e1c988" taskpriority="0" logused="184" waitresource="RID: 2:1:119:0" waittime="3604" ownerId="7654" transactionname="
      <executionStack>
        <frame procname="ad hoc" line="2" stmtstart="16" sqlhandle="0x020000000853a2804d2e29343d506b55985e405803b9c8fa6">
UPDATE [T2] set [C1] = @1      </frame>
        <frame procname="ad hoc" line="2" stmtstart="164" sqlhandle="0x02000000c258722ecc8c7dee6d812db98c8f720b56de59e3">
UPDATE T2 SET C1 = 2      </frame>
      </executionStack>
      <inputbuf>
-- Use the event 'Locks/Deadlock graph' from SQL Profiler to review the deadlock
UPDATE T2 SET C1 = 2
      </inputbuf>
    </process>
  </process-list>
  <resource-list>
    <ridlock fileid="1" pageid="114" dbid="2" objectname="tempdb.dbo.T1" id="lock636d500" mode="X" associatedObjectId="936748722985435136">
      <owner-list>
        <owner id="process5e1c988" mode="X"/>
      </owner-list>
      <waiter-list>
        <waiter id="process5e12e08" mode="U" requestType="wait"/>
      </waiter-list>
    </ridlock>
    <ridlock fileid="1" pageid="119" dbid="2" objectname="tempdb.dbo.T2" id="lock636d480" mode="X" associatedObjectId="1008806317064323072">
      <owner-list>
        <owner id="process5e12e08" mode="X"/>
      </owner-list>
      <waiter-list>
        <waiter id="process5e1c988" mode="U" requestType="wait"/>
      </waiter-list>
    </ridlock>
  </resource-list>
</deadlock>
</deadlock-list>
```

# Trace Flag 1222

- Deadlocks are logged in the Error Log
- Not very handy for further analysis
- Contains
  - Process List
  - Resource List

# Trace Flag 1222

Log File Viewer - localhost

Select logs

- ☐ Database Mail
- ☒ SQL Server
  - ☒ Current - 28.04.2011 12:42:56
  - ☐ Archive #1 - 28.04.2011
  - ☐ Archive #2 - 27.04.2011
  - ☐ Archive #3 - 27.04.2011
  - ☐ Archive #4 - 26.04.2011
  - ☐ Archive #5 - 26.04.2011
  - ☐ Archive #6 - 25.04.2011
- ☐ SQL Server Agent

Log file summary: No filter applied

Date	Source	Message
28.04.2011 12:42:56	spid24s	waiter id=process5e1c988 mode=U requestType=wait
28.04.2011 12:42:56	spid24s	waiter-list
28.04.2011 12:42:56	spid24s	owner id=process5e12e08 mode=X
28.04.2011 12:42:56	spid24s	owner-list
28.04.2011 12:42:56	spid24s	ridlock fileid=1 pageid=119 dbid=2 objectname=tempdb.dbo.T2 id=lock636d480 mode=X associatedObjectId=1008806317064323072
28.04.2011 12:42:56	spid24s	waiter id=process5e12e08 mode=U requestType=wait
28.04.2011 12:42:56	spid24s	waiter-list
28.04.2011 12:42:56	spid24s	owner id=process5e1c988 mode=X
28.04.2011 12:42:56	spid24s	owner-list
28.04.2011 12:42:56	spid24s	ridlock fileid=1 pageid=114 dbid=2 objectname=tempdb.dbo.T1 id=lock636d500 mode=X associatedObjectId=936748722985435136
28.04.2011 12:42:56	spid24s	resource-list
28.04.2011 12:42:56	spid24s	UPDATE T2 SET C1 = 2
28.04.2011 12:42:56	spid24s	-- Use the event "Locks/Deadlock graph" from SQL Profiler to review the deadlock
28.04.2011 12:42:56	spid24s	inputbuf
28.04.2011 12:42:56	spid24s	UPDATE T2 SET C1 = 2
28.04.2011 12:42:56	spid24s	frame procname=ad hoc line=2 stmtstart=164 sqlhandle=0x02000000c258722ecc8c7dee6d812db98c8f720b56de59e3
28.04.2011 12:42:56	spid24s	UPDATE [T2] set [C1] = @1
28.04.2011 12:42:56	spid24s	frame procname=ad hoc line=2 stmtstart=16 sqlhandle=0x020000000853a2804d2e29343d506b55985e405803b9c8fa6
28.04.2011 12:42:56	spid24s	executionStack
28.04.2011 12:42:56	spid24s	process id=process5e1c988 taskpriority=0 logused=184 waitresource=RID: 2:1:119:0 waittime=3604 ownerid=7654 transactionname=user_transaction last...

Selected row details:

Date: 28.04.2011 12:42:56  
Log: SQL Server (Current - 28.04.2011 12:41:00)  
Source: spid24s  
Message:

Close

# Extended Events

- Always On Event Session "system\_health"
  - Runs always in the background
  - Captures historical deadlocks
- Returns Deadlock information as XML

# Demo

## Deadlock Examples

# Summary

- Transactions & Isolation Levels
- Deadlocking



# SQL Server Performance Tuning Workshop

- June 1 – 5 in London
- Content
  - Database Internals
  - Execution Plans
  - Indexing
  - Statistics
  - Locking, Blocking, Deadlocking
  - Performance Troubleshooting
- More Information
  - <http://www.SQLpassion.at/academy/perftuning>
  - 10% Discount!

