MSSQLTips Free SQL Server Conference Registration + Travel Expenses click to learn more

Tracing a SQL Server Deadlock

By: Greg Robidoux

Overview

A common issue with SQL Server is deadlocks. A deadlock occurs when two or more processes are waiting on the same resource and each process is waiting on the other process to complete before moving forward. When this situation occurs and there is no way for these processes to resolve the conflict, SQL Server will choose one of processes as the deadlock victim and rollback that process, so the other process or processes can move forward.

By default when this occurs, your application may see or handle the error, but there is nothing that is captured in the SQL Server Error Log or the Windows Event Log to let you know this occurred. The error message that SQL Server sends back to the client is similar to the following:

```
Msg 1205, Level 13, State 51, Line 3
Transaction (Process ID xx) was deadlocked on {xxx} resources with another process
and has been chosen as the deadlock victim. Rerun the transaction.
```

In this tutorial we cover what steps you can take to capture deadlock information and some steps you can take to resolve the problem.

Explanation

Deadlock information can be captured in the SQL Server Error Log or by using Profiler / Server Side Trace.

Trace Flags

If you want to capture this information in the SQL Server Error Log you need to enable one or both of these trace flags.

- 1204 - this provides information about the nodes involved in the deadlock
- 1222 - returns deadlock information in an XML format

You can turn on each of these separately or turn them on together.

To turn these on you can issue the following commands in a query window or you can add these as startup parameters. If these are turned on from a query window, the next time SQL Server starts these trace flags will not be active, so if you always want to capture this data the startup parameters is the best option.

```
DBCC TRACEON (1204, -1)
DBCC TRACEON (1222, -1)
```

Here is sample output for each of the trace flags.

Trace Flag 1222 Output

Source	Message
spid13s	waiter id=process15f708 mode=U requestType=wait
spid13s	waiter-list
spid13s	owner id=process8092c508 mode=X
spid13s	owner-list
spid13s	keylock hobtid=411043454976000 dbid=16 objectname=Northwind.dbo.Employees indexname=PK_Employees id=lock82d28180 mode=X associatedObjectId=411043454976000
spid13s	waiter id=process8092c508 mode=U requestType=wait
spid13s	waiter-list
spid13s	owner id=process15f708 mode=X
spid13s	owner-list
spid13s	keylock hobtid=289180401860608 dbid=16 objectname=Northwind.dbo.Products indexname=PK_Products id=lock82d34780 mode=X associatedObjectId=289180401860608
spid13s	resource-list
spid13s	update employees set firstname = 'Greg'
spid13s	inputbuf
spid13s	update employees set firstname = 'Greg'
spid13s	frame procname=adhoc line=1 sqlhandle=0x02000000cc00eb176bbab93d703b023de85137b27f0d665e
spid13s	UPDATE [employees] set [firstname] = @1
spid13s	frame procname=adhoc line=1 stmtstart=36 sqlhandle=0x020000009889e5073492f278180b42312796b34665be75dc
spid13s	executionStack
spid13s	process id=process15f708 taskpriority=0 logused=40720 waitresource=KEY: 16:411043454976000 (03000d8f0ecc) waittime=2729 ownerId=2247449 transactionname=user_transa
spid13s	update products set supplierid = 1
spid13s	update employees set firstname = 'Bob'
spid13s	begin tran
spid13s	inputbuf
spid13s	update products set supplierid = 1
spid13s	frame procname=adhoc line=3 stmtstart=108 sqlhandle=0x02000000d484532157c4e69fe7c0e64e8f08238b59fbf490
spid13s	frame procname=adhoc line=3 stmtstart=108 sqlhandle=0x0200000050877022ebb43bd41412d6078ec93299797b7a65
spid13s	executionStack
spid13s	process id=process8092c508 taskpriority=0 logused=1968 waitresource=KEY: 16:289180401860608 (010086470766) waittime=5514 ownerId=2247505 transactionname=user_tran
spid13s	process-list
spid13s	deadlock victim=process8092c508
spid13s	deadlock-list

Trace Flag 1204 Output

Source	Message
spid6s	ResType:LockOwner Stype:'OR'Xdes:0x0000000080716E90 Mode: U SPID:62 BatchID:0 ECID:0 TaskProxy:(0x000000008AB10538) Value:0x82d3c280 Cost:(0/1968)
spid6s	Victim Resource Owner:
spid6s	Log Viewer could not read information for this log entry. Cause: Data is Null. This method or property cannot be called on Null values.. Content:
spid6s	ResType:LockOwner Stype:'OR'Xdes:0x000000008B0CB740 Mode: U SPID:61 BatchID:0 ECID:0 TaskProxy:(0x000000008A2EC538) Value:0x82d3e200 Cost:(0/40720)
spid6s	Requested by:
spid6s	Input Buf: Language Event: begin tran update employees set firstname = 'Bob' update products set supplierid = 1
spid6s	SPID: 62 ECID: 0 Statement Type: UPDATE Line #: 3
spid6s	Owner:0x0000000080152100 Mode: X Flg:0x40 Ref:0 Life:02000000 SPID:62 ECID:0 XactLockInfo: 0x0000000080716ED0
spid6s	Grant List 1:
spid6s	KEY: 16:411043454976000 (03000d8f0ecc) CleanCnt:2 Mode:X Flags: 0x1
spid6s	Node:2
spid6s	Log Viewer could not read information for this log entry. Cause: Data is Null. This method or property cannot be called on Null values.. Content:
spid6s	ResType:LockOwner Stype:'OR'Xdes:0x0000000080716E90 Mode: U SPID:62 BatchID:0 ECID:0 TaskProxy:(0x000000008AB10538) Value:0x82d3c280 Cost:(0/1968)
spid6s	Requested by:
spid6s	Input Buf: Language Event: update employees set firstname = 'Greg'
spid6s	SPID: 61 ECID: 0 Statement Type: UPDATE Line #: 1
spid6s	Owner:0x00000000801524C0 Mode: X Flg:0x40 Ref:0 Life:02000000 SPID:61 ECID:0 XactLockInfo: 0x000000008B0CB780
spid6s	Grant List 0:
spid6s	KEY: 16:289180401860608 (010086470766) CleanCnt:2 Mode:X Flags: 0x1
spid6s	Node:1
spid6s	Log Viewer could not read information for this log entry. Cause: Data is Null. This method or property cannot be called on Null values.. Content:
spid6s	Wait-for graph
spid6s	Deadlock encountered .... Printing deadlock information

Profiler / Server Side Trace

Profiler works without the trace flags being turned on and there are three events that can be captured for deadlocks. Each of these events is in the **Locks** event class.

- Deadlock graph - Occurs simultaneously with the Lock:Deadlock event class. The Deadlock Graph event class provides an XML description of the deadlock.
- Lock: Deadlock - Indicates that two concurrent transactions have deadlocked each other by trying to obtain incompatible locks on resources that the other transaction owns.
- Lock: Deadlock Chain - Is produced for each of the events leading up to the deadlock.

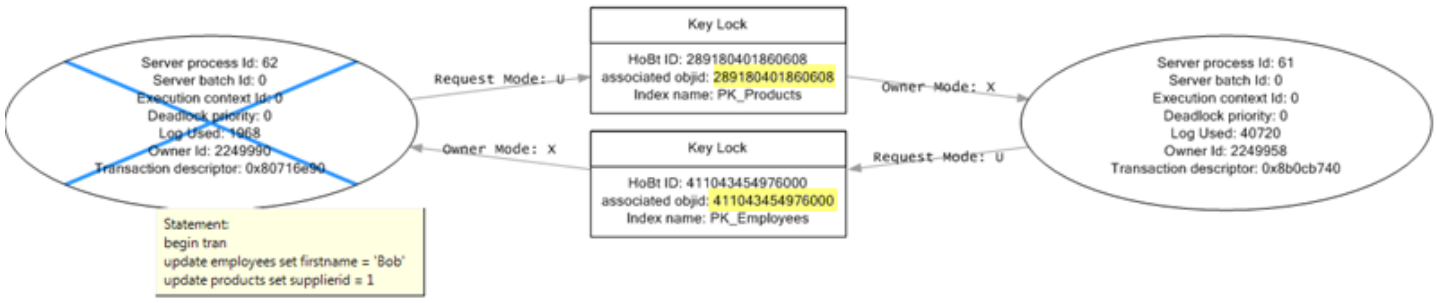
Event Output

In the below image, I have only captured the three events mentioned above.

EventClass	TextData	DatabaseID	ObjectID2
Trace Start			
Lock:Deadlock Chain	Deadlock Chain SPID = 62 (010086470766) ...	16	289180401860608
Lock:Deadlock Chain	Deadlock Chain SPID = 61 (03000d8f0ecc) ...	16	411043454976000
Lock:Deadlock	(010086470766)	16	289180401860608
Deadlock graph	<deadlock-list> <deadlock victim="proce...		

Deadlock Graph Output

Below is the deadlock graph which is the output for the Deadlock graph event. We can see on the left side that server process id 62 was selected as the deadlock victim. Also, if you hover over the oval with the X through it we can see the transaction that was running.



Finding Objects Involved in Deadlock

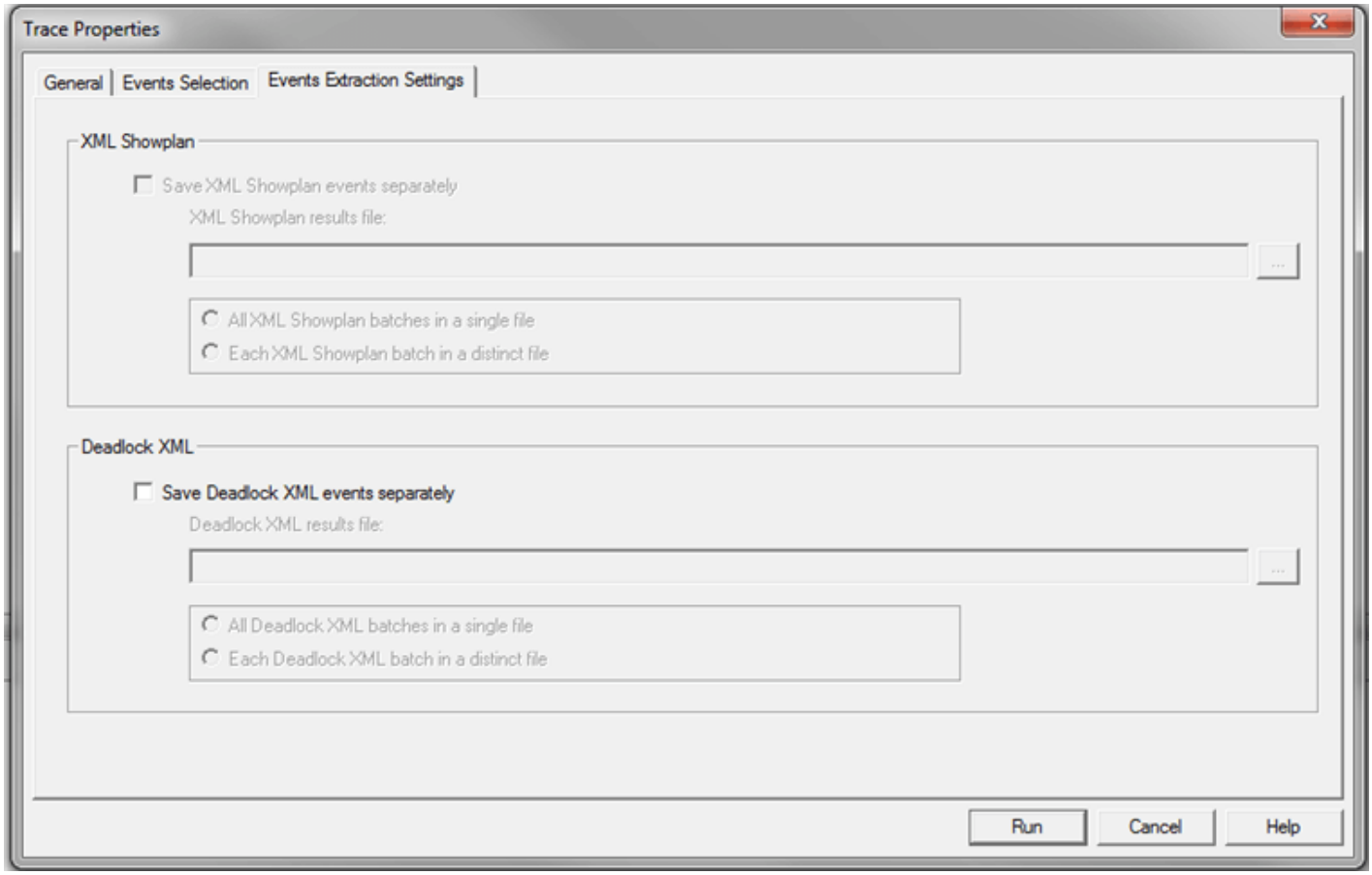
In all three outputs, I have highlighted the object IDs for the objects that are in contention. You can use the following query to find the object, substituting the object ID for the partition_id below.

```
SELECT OBJECT_SCHEMA_NAME([object_id]),
OBJECT_NAME([object_id])
FROM sys.partitions
WHERE partition_id = 289180401860608;
```

Saving Deadlock Graph Data in XML File

Since the deadlock graph data is stored in an XML format, you can save the XML events separately. When configuring the Trace Properties click on the Events Extraction Settings and enable this option as shown

below.



Additional Information

Here are some additional artilces about deadlocks.

- Finding and troubleshooting SQL Server deadlocks
- SQL Server Deadlock Priority Configuration
- Finding SQL Server Deadlocks Using Trace Flag 1222
- Capturing SQL Server Deadlock Information in XML Format
- Using a Clustered Index to Solve a SQL Server Deadlock Issue

<< Previous Next >>

More SQL Server Solutions

SQL Monitor
Get your free ROI calculation
Start now

FREE
SQL Server Webcasts
Live and On-demand

Get Free SQL Server Tips

Post a comment or let the author know this tip helped.

All comments are reviewed, so stay on subject or we may delete your comment. Note: your email address is not published. Required fields are marked with an asterisk (*).

*Name *Email Notify for updates

*** NOTE *** - If you want to include code from SQL Server Management Studio (SSMS) in your post, please copy the code from SSMS and paste the code into a text editor like NotePad before copying the code below to remove the SSMS formatting.

p

Send me SQL tips:

I'm not a robot

reCAPTCHA
Privacy - Terms

Save Comment

Tuesday, April 26, 2016 - 9:05:12 AM - Greg Robidoux Back To Top

Hi Krunal,

take a look at this tip that shows how to create and detect a deadlock. This might give you more information.

https://www.mssqltips.com/sqlservertip/1036/finding-and-troubleshooting-sql-server-deadlocks/

-Greg

Tuesday, April 26, 2016 - 8:27:45 AM - Krunal Patel Back To Top

Hello ! Gerg

I'm facing deadlock since couple of days, can you please tell me what are the cases for occurance of deadlocks and how can i resolve it?

Thanks,

Krunal Patel

Get Free SQL Tips

enter business email Join