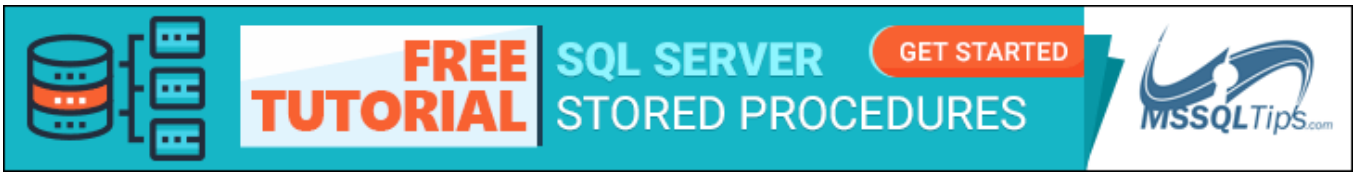
MSSQLTips Free SQL Server Conference Registration + Travel Expenses click to learn more

Troubleshooting Blocking



By: Greg Robidoux

Overview

In order for SQL Server to maintain data integrity for both reads and writes it uses locks, so that only one process has control of the data at any one time. There are serveral types of locks that can be used such as Shared, Update, Exclusive, Intent, etc... and each of these has a different behavior and effect on your data.

When locks are held for a long period of time they cause blocking, which means one process has to wait for the other process to finish with the data and release the lock before the second process can continue. This is similar to deadlocking where two processes are waiting on the same resource, but unlike deadlocking, blocking is resolved as soon as the first process releases the resource.

Explanation

As mentioned above, blocking is a result of two processes wanting to access the same data and the second process needs to wait for the first process to release the lock. This is how SQL Server works all of the time,

but usually you do not see blocking because the time that locks are held is usually very small.

It probably makes sense that locks are held when updating data, but locks are also used when reading data. When data is updated an Update lock is used and when data is read a Shared lock is used. An Update lock will create an exclusive lock on the data for this process and a Shared lock allows other processes that use a Shared lock to access the data as well and when two processes are trying to access the same data this is where the locking and blocking occurs.

Here are various ways you can identify blocking for your SQL Server instance.

sp_who2

In a query window run this command:

```
sp_who2
```

This is the output that is returned. Here we can see the BlkBy column that shows SPID 60 is blocked by SPID 59.

	SPID	Status	Login	HostName	BlkBy	DBName	Command	CPUTime	DiskIO	LastBatch	ProgramNam
41	59	sleeping	ED...	EDGEW...	.	test	AWAITING COMMAND	0	19	12/28 11:06:49	Microsoft S.
42	60	SUSPENDED	ED...	EDGEW...	59	test	UPDATE	0	4	12/28 11:06:56	Microsoft S.

Activity Monitor

In SSMS, right click on the SQL Server instance name and select Activity Monitor. In the Processes section you will see information similar to below. Here we can see similar information as sp_who2, but we can also see the Wait Time, Wait Type and also the resource that SPID 60 is waiting for.

Processes

Session ID	User Proce	Database	Task State	Command	Wait Time (ms)	Blocked By	Wait Type	Wait Resource
57	1	master			0			
58	1	master			0			
59	1	test			0			
60	1	test	SUSPENDED	UPDATE	315648	59	LCK_M_U	ridlock fileid=1 pageid=12...

Report - All Blocking Transactions

Another option is to use the built in reports in SSMS. Right click on the SQL Server instance name and select Reports > Standard Reports > Activity - All Block Transactions.

All Blocking Transactions

The description of transactions which are blocking other transactions.

Transaction ID	# Directly Blocked Transactions	# Indirectly Blocke Transactions	State	Transaction Type	Start Time	Resource Type	Session ID	Blocking SQL Statement
⊟ 789800	1	0	Active	Full Transaction	12/28/2011 11:06:49 AM	RID	59	⊞ --

Direct/Indirect	Blocked Transaction ID	Blocked Transacti	State	Transaction Type	Start Time	Resource Type	Session ID	Blocked SQL Statement
Direct	789800	user_transaction	Active	Full Transaction	12/28/2011 11:07:00 AM	RID	60	⊞ UPDATE [dbo].[Categories] set [Description] = @1 WHERE [CategoryName]=@2

Querying Dynamic Management Views

You can also use the DMVs to get information about blocking.

```
SELECT session_id, command, blocking_session_id, wait_type, wait_time, wait_resource, t.TEXT
FROM sys.dm_exec_requests
CROSS apply sys.dm_exec_sql_text(sql_handle) AS t
WHERE session_id > 50
AND blocking_session_id > 0
UNION
SELECT session_id, '', '', '', '', '', t.TEXT
FROM sys.dm_exec_connections
CROSS apply sys.dm_exec_sql_text(most_recent_sql_handle) AS t
WHERE session_id IN (SELECT blocking_session_id
                     FROM sys.dm_exec_requests
                     WHERE blocking_session_id > 0)
```

Here is the output and we can see the blocking information along with the TSQL commands that were issued.

	session_id	command	blocking_session_id	wait_type	wait_time	wait_resource	TEXT
1	59		0		0		BEGIN TRAN INSERT_FOO INSERT INTO dbo.Categ...
2	60	UPDATE	59	LCK_M_U	1157132	RID: 5:1:1217:0	(@1 varchar(8000),@2 varchar(8000))UPDATE [dbo].[Cat...

Additional Information

Here are some additional artilces about blocking.

- How to identify blocking in SQL Server 2005 and 2008
- Understanding SQL Server Blocking
- Understanding SQL Server Locking
- Understanding the SQL Server NOLOCK hint
- Using SQL Server Indexes to Bypass Locks
- Execute SQL Server SELECT command without locking

<< Previous Next >>

More SQL Server Solutions

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

Get Free SQL Tips

enter business email Join

Learning

DBAs

Developers

BI Professionals

Careers

Q and A

Today's Tip

Resources

Tutorials

Webcasts

Whitepapers

Tools

Copyright (c) 2006-2018 Edgewood Solutions, LLC All rights reserved

Some names and products listed are the registered trademarks of their respective owners.