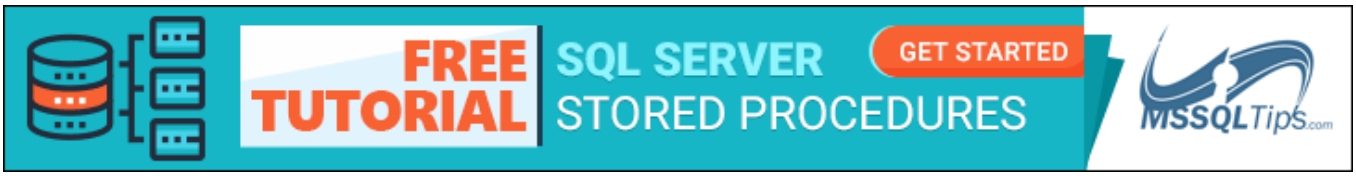
MSSQLTips Free SQL Server Conference Registration + Travel Expenses click to learn more

How to identify blocking in SQL Server



Free Webcast - Simplify SQL Server Administration

Problem

In our SQL Server environment, we have frequent locking and blocking across a few different versions of SQL Server. How can I find blocking and blocked SPID's in SQL Server 2005 and later versions? Is there only one way to find out which spids are blocking? Are there any commands that I can run against multiple SQL Server versions? Check out this tip to learn more about locking and blocking.

Solution

Whenever a user contacts the DBA team indicating a processes looks hung or a process is not proceeding checking the applicable database blocking makes a great deal of sense. Blocking happens when one connection from an application holds a lock and a second connection requires a conflicting lock. This forces the second connection to be blocked until the first connection completes. With this being said, locking is a natural occurrence in SQL Server in order to maintain data integrity. For more information about locking and blocking review these tips: Understanding SQL Server Locking and Understanding SQL Server Blocking.

There are number of ways to find out the details of the system processes IDs (spids) involved in blocking. I have tried to cover some of the options in this tip to include:

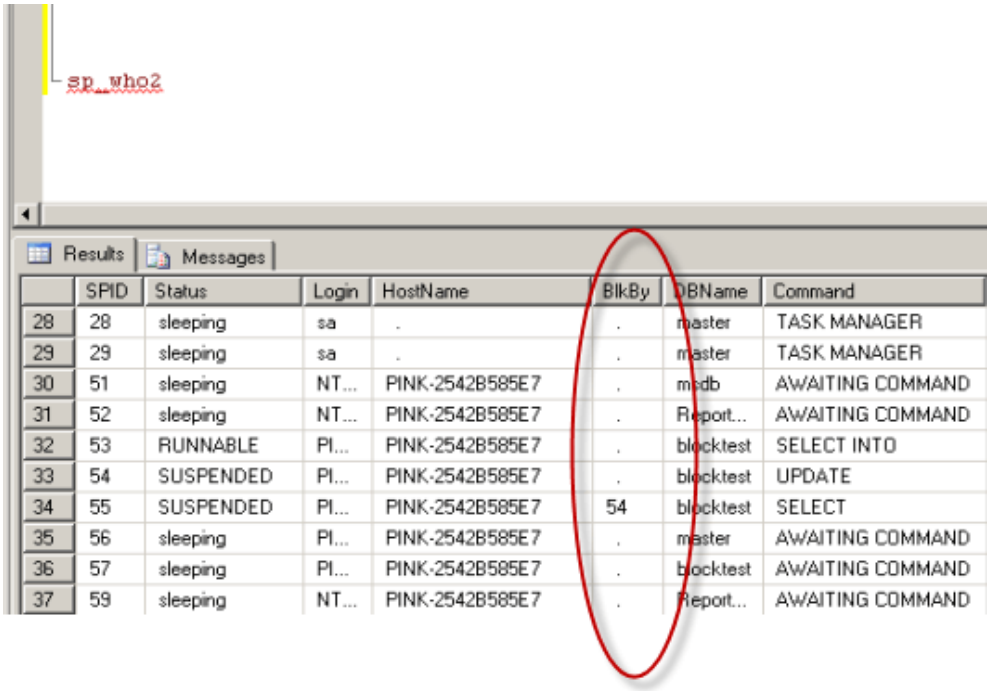
- sp_who2 System Stored Procedure
- sys.dm_exec_requests DMV
- Sys.dm_os_waiting_tasks
- SQL Server Management Studio Activity Monitor
- SQL Server Management Studio Reports
- SQL Server Profiler

---

sp_who2 System Stored Procedure

The sp_who2 system stored procedure provides information about the current SQL Server processes with the associated users, application, database, CPU time, etc. The information returned can be filtered to return only the active processes by using the ‘active' parameter. Below is some sample code and a screen shot with showing process 55 being blocked by process 54.

```
USE Master
GO
EXEC sp_who2
GO
```



	SPID	Status	Login	HostName	BlkBy	DBName	Command
28	28	sleeping	sa	.	.	master	TASK MANAGER
29	29	sleeping	sa	.	.	master	TASK MANAGER
30	51	sleeping	NT...	PINK-2542B585E7	.	modb	AWAITING COMMAND
31	52	sleeping	NT...	PINK-2542B585E7	.	Report...	AWAITING COMMAND
32	53	RUNNABLE	PI...	PINK-2542B585E7	.	blocktest	SELECT INTO
33	54	SUSPENDED	PI...	PINK-2542B585E7	.	blocktest	UPDATE
34	55	SUSPENDED	PI...	PINK-2542B585E7	54	blocktest	SELECT
35	56	sleeping	PI...	PINK-2542B585E7	.	master	AWAITING COMMAND
36	57	sleeping	PI...	PINK-2542B585E7	.	blocktest	AWAITING COMMAND
37	59	sleeping	NT...	PINK-2542B585E7	.	Report...	AWAITING COMMAND

Additional resources:

- SQL Server Command Line Tools To Manage Your Server
- Locking and Blocking Scripts in SQL Server 2000 vs SQL Server 2005
- Killing a SPID and Checking Rollback progress in SQL Server

---

sys.dm_exec_requests DMV

The sys.dm_exec_requests DMV provides details on all of the processes running in SQL Server. With the WHERE condition listed below, only blocked processes will be returned.

```
USE Master
GO
SELECT *
FROM sys.dm_exec_requests
WHERE blocking_session_id <> 0;
GO
```

Additional resources:

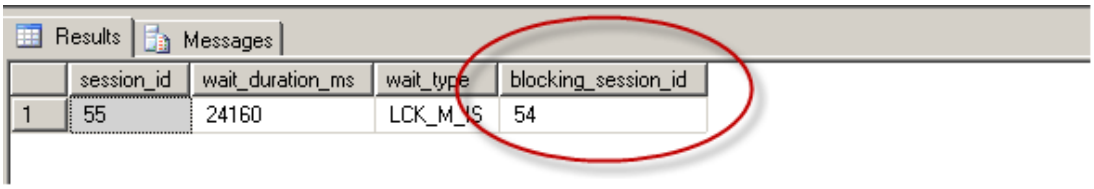
- How to isolate the current running commands in SQL Server
- Identifying the input buffer in SQL Server 2000 vs SQL Server 2005
- Finding a SQL Server process percentage complete with dynamic management views

---

sys.dm_os_waiting_tasks DMV

The sys.dm_os_waiting_tasks DMV returns information about the tasks that are waiting on resources. To view the data, users should have SQL Server System Administrator or VIEW SERVER STATE permissions on the instance.

```
USE Master
GO
SELECT session_id, wait_duration_ms, wait_type, blocking_session_id
FROM sys.dm_os_waiting_tasks
WHERE blocking_session_id <> 0
GO
```



Additional resources:

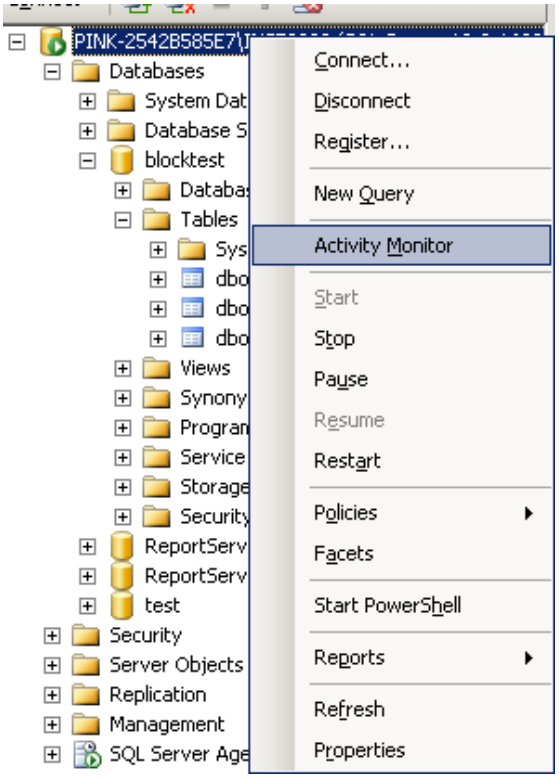
- SQL Server Tempdb Usage and Bottlenecks tracked with Extended Events

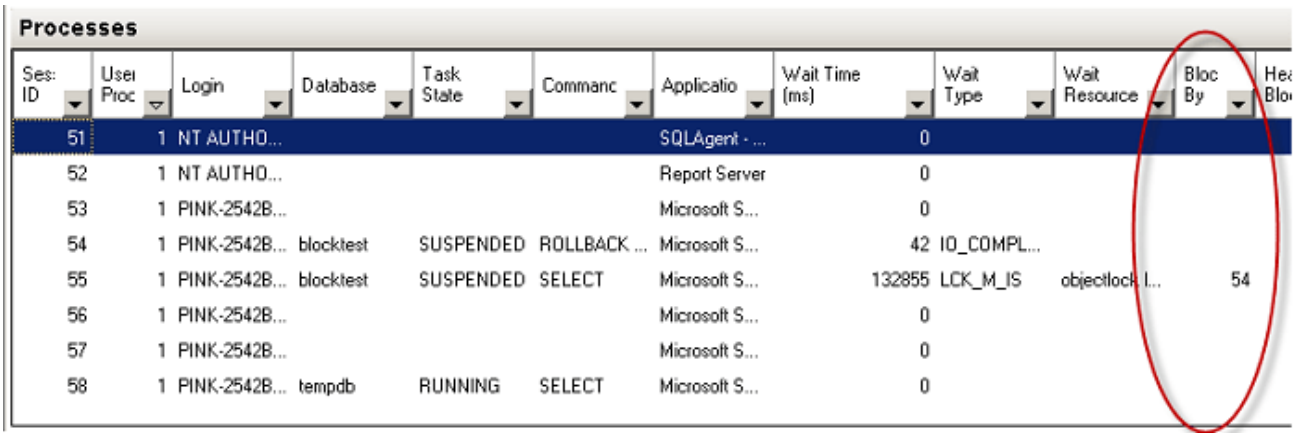
---

SQL Server Management Studio Activity Monitor

If you are more comfortable using SQL Server Management Studio to review locking and blocking as opposed to querying system objects or executing stored procedures, you are in luck. There are even a few different tools in SQL Server Management Studio you can use. The first option is the Activity Monitor, which can be accessed by navigating to the instance name | right click | select 'Activity Monitor'. To view the Activity

Monitor in SQL Server 2005 and SQL Server 2008, users should have SQL Server System Administrator or VIEW SERVER STATE permissions on the instance.





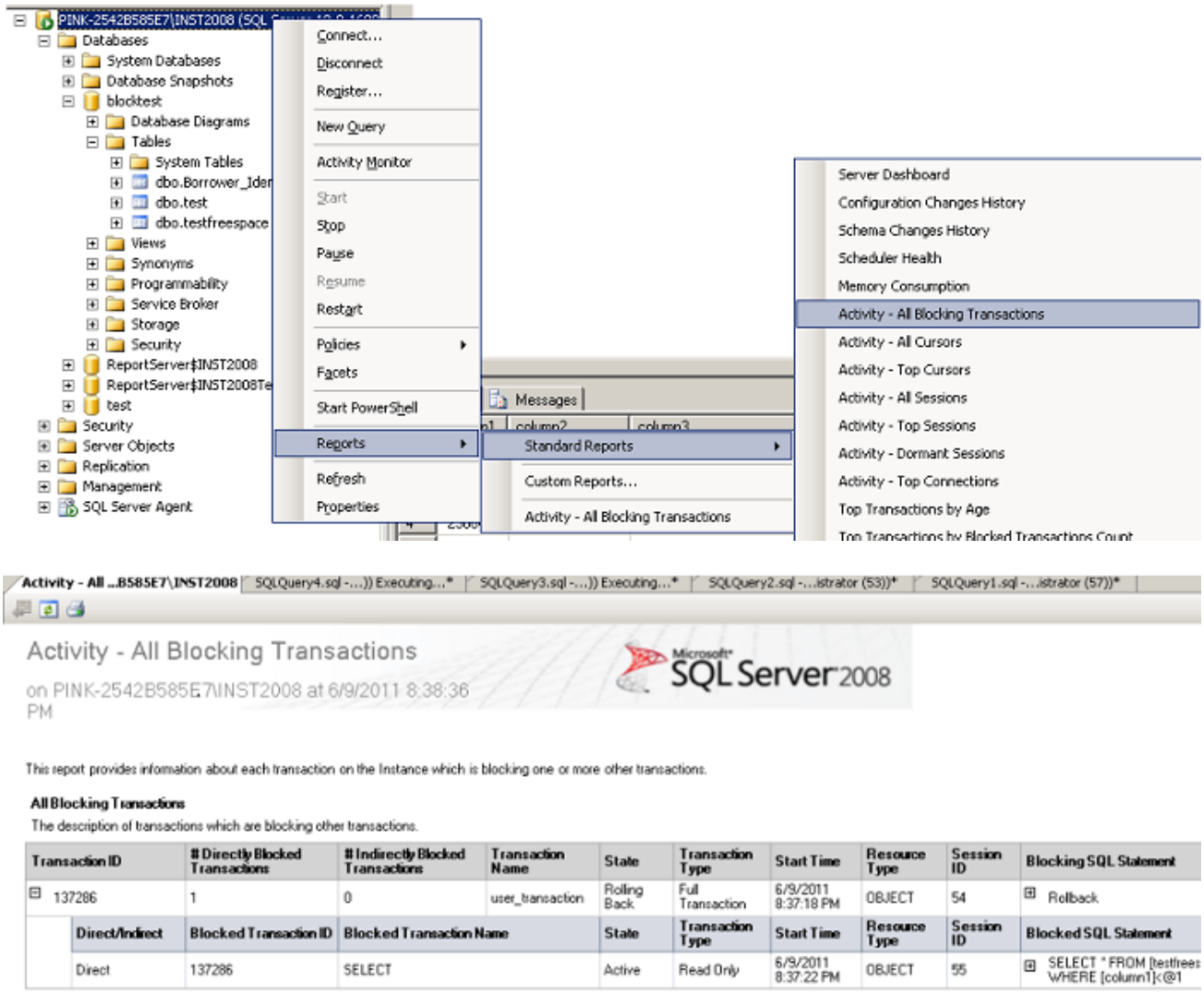
Additional resources:

- Performance Analysis Using SQL Server 2008 Activity Monitor Tool

---

SQL Server Management Studio Reports

The second option in SQL Server Management Studio to monitor blocking is with the standard reports, which can be accessed by navigating to the instance name | right click | Reports | Standard Reports | Activity - All Blocking Transactions. Once again, users should have SQL Server System Administrator or VIEW SERVER STATE permissions on the instance.



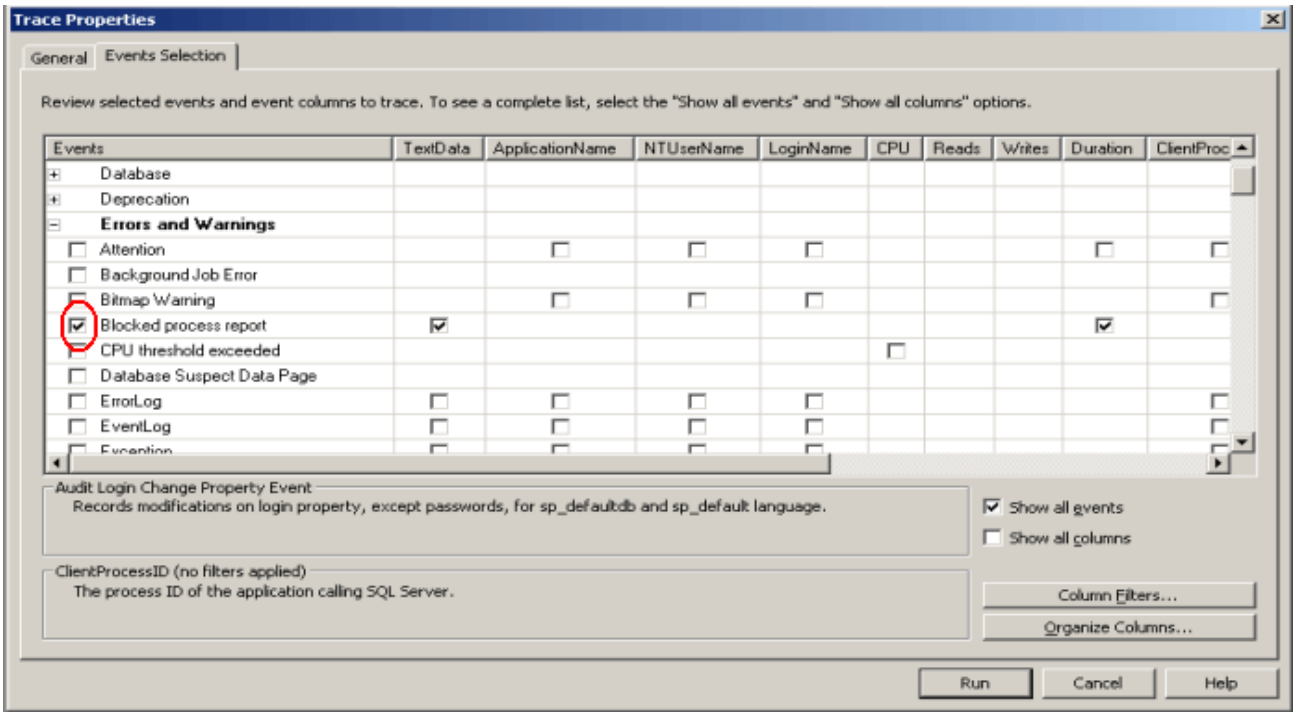
Additional resources:

- Built-In Performance Reports in SQL Server 2005

SQL Server Profiler

To capture blocking related data on a continuous basis, one option is to run SQL Server Profiler and save the data to a table or file for analysis purposes. In order to configure Profiler to capture blocking related data, execute Profiler, configure the general properties then navigate to Event Selection tab | select Show all events | Errors and Warnings | check the Blocked process report and then run the application. In addition, be sure to configure the 'blocked process threshold' before you start Profiler using this code:

```
sp_configure 'show advanced options', 1
GO
RECONFIGURE
GO
sp_configure 'blocked process threshold', 20
GO
RECONFIGURE
GO
```



Additional resources:

- Tip category: Profiler and Trace
- Capturing SQL Server Deadlock Information in XML Format
- Finding and troubleshooting SQL Server deadlocks
- SQL Server Profiler Graphical Deadlock Chain

Next Steps

- If you are faced with a blocking situation, be sure to consider all of your options in the short and long term. To resolve the immediate issue, you may need to KILL some spids, but to resolve the issue you may need to change your database design, change your data access, add NOLOCK hints to particular queries, etc.
- Check out these tips to learn more about locking and blocking
  - Understanding SQL Server Locking
  - Understanding SQL Server Blocking
  - Tip category: Locking and Blocking Tips

Last Update: 2011-07-15

You're invited
to our next free
SQL Server webcast!

click to learn more

Next Tip

Click Here

About the author

Nitansh Agarwal is a lead with 4+ years of extensive experience in database administration where he handles large critical databases.

View all my tips

Related Resources

- Identify the cause of SQL Server blocking...
- How to identify blocking in SQL Server...
- Understanding SQL Server Blocking...
- Different techniques to identify blocking in SQL S...
- More SQL Server DBA 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