Using sp_who2

Posted on **December 20, 2009** by Derek Dieter 12 7

« Cannot Log in Remotely to SQL Server Express

Best SQL Server Pagination Method »

This article shows the usage of sp_who2. To diagnose system slowdowns, see (Troubleshooting SQL Slowness).

One of the first lines of defense in determining the causes of database slowdowns is to use sp_who2. sp_who2 shows all the sessions that are currently established in the database. These are denoted as SPID's, or Server process Id's. Running sp_who2 is easy, all that is required is to type sp_who2 and execute it, however it's a good idea to call it with EXEC.

[cc lang="sql"]
EXEC sp_who2
[/cc]

Results | Messages

	SPID	Status	Login	HostName	BlkBy	DBName	Command	CPUTime	DiskIO	LastBatch	ProgramName	SPID	REQUESTID
1	1	BACKGROUND	sa	.	.	NULL	RESOURCE MONITOR	406656	0	11/22 22:13:12		1	0
2	2	BACKGROUND	sa	.	.	NULL	XE TIMER	15	0	11/22 22:13:12		2	0
3	3	BACKGROUND	sa	.	.	NULL	XE DISPATCHER	359	0	11/22 22:13:12		3	0
4	4	BACKGROUND	sa	.	.	NULL	LOG WRITER	827062	0	11/22 22:13:12		4	0
5	5	BACKGROUND	sa	.	.	NULL	LOCK MONITOR	6953	0	11/22 22:13:12		5	0
6	6	BACKGROUND	sa	.	.	master	SIGNAL HANDLER	0	0	11/22 22:13:12		6	0
7	7	BACKGROUND	sa	.	.	NULL	LAZY WRITER	1956640	0	11/22 22:13:12		7	0
8	8	sleeping	sa	.	.	master	TASK MANAGER	0	0	11/22 22:13:12		8	0
9	9	BACKGROUND	sa	.	.	master	TRACE QUEUE TASK	12109	0	11/22 22:13:12		9	0
10	10	BACKGROUND	sa	.	.	master	BRKR EVENT HNDLR	0	2	11/22 22:13:12		10	0
11	11	BACKGROUND	sa	.	.	master	BRKR TASK	12641	0	11/22 22:13:12		11	0
12	12	BACKGROUND	sa	.	.	master	TASK MANAGER	0	0	11/22 22:13:12		12	0
13	13	BACKGROUND	sa	.	.	master	CHECKPOINT	33191062	3057673	11/22 22:13:12		13	0
14	14	BACKGROUND	sa	.	.	master	BRKR TASK	0	0	11/22 22:13:12		14	0
15	15	BACKGROUND	sa	.	.	master	BRKR TASK	0	0	11/22 22:13:12		15	0
16	16	sleeping	sa	.	.	master	TASK MANAGER	0	0	11/22 22:13:12		16	0
17	17	sleeping	sa	.	.	master	TASK MANAGER	0	38	11/22 22:13:12		17	0

Results of sp_who2

The first 50 results are system SPIDs. Generally these do not effect slowdowns of the system. These system events include the Checkpoint writer, Log writers and task schedulers. User processes are SPID numbers 50 and over. In diagnosing slowdowns it is these SPIDs that are the potential resource hogs.

sp_who2 also takes a optional parameter of a SPID. If a spid is passed, then the results of sp_who2 only show the row or rows of the executing SPID.

[cc lang="sql"]
EXEC sp_who2 243
[/cc]

There are four main things to look for when when diagnosing slowdowns.

1. Blocking
2. High CPU usage
3. High IO usage
4. Multiple entries for the same SPID representing parallelism.

When a number is shown in the column named BlkBy, this represents the SPID that is currently stopping the SPID in the row shown. Sometimes many rows will show SPID numbers in the BlkBy column. This is because there is a chain of blockers. The way this occurs usually starts with one "lead" blocker blocking another process. In turn, the process that is being blocked, blocks others. This occurs down a chain. It can be a messy situation. In order to rectify, you may have to kill the lead blocker. If it happens often, you will want to research why this particular process is blocking. So, before you kill any process, find out what statement it is running first. To do this, execute DBCC Inputbuffer.

High CPUTime or High DiskIO time is usually spotted by comparing the relative CPUTime to the DiskIO time. It should be noted that CPUTime and DiskIO time represent the **sum** of all executions since the SPID has been active. It may take some training before you are able to spot a high number here. At times, you will see very high CPUTimes and almost no corresponding DiskIO. This is usually indicative of a bad execution plan. For more information see this article (slow performance).

Multiple rows that have the same SPID are known as Parallel processes. This happens when SQL Server has determined that a particular query is going to take a long time to run (according to the parallel settings at the server level). When this happens, SQL Server will launch multiple threads for the same procedure. When this happens often, it can have a devastating effect on SQL Server. Particularly IO. To rectify, either raise the query threshold for parallelism or turn down the maximum processors that parallelism can use (MAXDOP) Max degree of parallelism, a setting at the server level.

sp_who2 does provide limited information regarding slowdowns. For a more in depth procedure look at sp_who3

Filed under DBA

中国进出口商品交易会
CHINA IMPORT AND EXPORT FAIR
Since 1957
The 123rd Canton Fair is almost here!
Apr 15-19 | Apr 23-27 | May 1-5, 2018

Related Posts:

- Using sp_who
- SQL Server SPID – What is it?
- What is Parallelism?
- A Better sp_who2 using DMVs (sp_who3)
- Configuring Parallelism
- SQL Server Slow Performance
- Using DBCC INPUTBUFFER
- Dynamic Numbers Table
- Using sp_spaceused
- Blocking Processes (lead blocker)

Post a comment

Comments (RSS) | Trackback | Permalink

Name:

Email:

Comment:

Submit comment

12 comments

Ketan
06 Jun 2017 at 4:35 pm

This is really great article. will definitely going to try it.

↳ Reply to this comment

Anup
03 Jul 2015 at 9:19 am

These articles are really helpful,I appreciate your Effort Derek.....

↳ Reply to this comment

Erdem
28 Aug 2014 at 7:55 am

Thank you so much....

↳ Reply to this comment

Da Bamz
30 Jan 2013 at 4:18 pm

Can you go more into DBCC Inputbuffer?
Regarding this article you mentioned but left no details.

Thx
Da Bamz

↳ Reply to this comment

Derek Dieter
31 Jan 2013 at 6:01 pm

Hi Da Bamz,

Thanks for the improvement suggestion. I added a link to an article about DBCC Inputbuffer

↳ Reply to this comment

MS Enterprise Library data access – Understanding SQL 'user connections' management | Jisku.com
15 Oct 2012 at 11:41 pm

[…] run SQL profiler over the login / logout events, and taken snapshots with the sp_who2 command, showing who owns the connections. The latter shows that indeed the web site (seen by […]

↳ Reply to this comment

SharePoint – SQL Interaction – performance troubleshooting from SQL Side « sharepointking
12 Sep 2012 at 3:42 am

[…] sp_who2 (system stored procedure) or sp_who3 to find out which SPID is taking a lot of […]

↳ Reply to this comment

Neeraj Kumar
14 Mar 2012 at 10:49 pm

Thanks for nice article…..please again do me a favor…can we find all information about all DDL and DML operation like Insert, delete and update on a particular host with login id and command??

↳ Reply to this comment

krishna
12 Aug 2011 at 2:37 pm

Currently on my serverwe have following replprov.dll version 5.2.3790.3959

To upgrade that we try to install following hotfix and service pack
SQLServer2008R2-KB2494086-x64.exe
SQLServer2008R2-KB2494088-x64.exe

SQLServer2008R2SP1-KB2528583-x64-ENU.exe
still replprov.dll remain same nothing replaced
Beacase of this my servers are not passing through company auditing
Please help where i can find latest one, atleast that particular DLL (preferebly from microsoft source)

↳ Reply to this comment

Derek Dieter
12 Aug 2011 at 10:05 pm

Hi Krishna,

That's interesting. I have never heard of a single DLL preventing a company from passing an audit. That seems pretty strict. There must be something I don't know regarding what the audit is looking for. What I do know, is that typically you cannot just replace a single DLL. Typically I've seen audits maybe look at the service pack version or hot fixes applied, but never to the level of individual DLL versions. Just curious, but who is requiring this specific DLL version?

Thanks,
Derek

↳ Reply to this comment

Top 10 Obsucre T-SQL commands « SteveFibich.Net
05 May 2011 at 6:41 pm

[…] in my repertoire but it hasn't faded completely so sp_who2 gets an honorable mention. (http://sqlserverplanet.com/dba/using-sp_who2/ […]

↳ Reply to this comment

Leonardo Raygoza
06 Apr 2011 at 2:19 pm

a really nice article I have resolved my issue with you help thanks a lot 🙂 blog added to my favorites

↳ Reply to this comment

© 2011 - 2015 SQL Server Planet. All Rights Reserved