DBCC SqlPerf

Being day 27 of the DBCC Command month at SteveStedman.com, today's featured DBCC Command is DBCC SQLPERF.

Description:

DBCC SQLPERF is used to do a couple different things.

- Show a list of all the wait stats on your SQL Server.
- Show a list of the transaction log and the space used in the transaction log.
- Clear the wait stats.
- Clear the latch stats.

DBCC SQLPERF Syntax:

Where option can be any of the following:

- WAITSTATS
- LOGSPACE
- 'sys.dm_os_latch_stats' , CLEAR
- 'sys.dm_os_wait_stats' , CLEAR

Example of Viewing Wait Statistics:

The following example shows the list of wait statistics on your SQL Server since the instance was last restarted, or the stats were last cleared.

SQLQuery1.sql - server2012.master (sa (58))*

Editor | Results | Messages

	Wait Type	Requests	Wait Time	Signal Wait Time
1	MISCELLANEOUS	0	0	0
2	LCK_M_SCH_S	0	0	0
3	LCK_M_SCH_M	22	6312	3
4	LCK_M_S	27	257589	328
5	LCK_M_U	1	0	0
6	LCK_M_X	5	67	0
7	LCK_M_IS	0	0	0
8	LCK_M_IU	0	0	0
9	LCK_M_IX	0	0	0
10	LCK_M_SIU	0	0	0
11	LCK_M_SIX	0	0	0
12	LCK_M_UIX	0	0	0
13	LCK_M_BU	0	0	0
14	LCK_M_RS_S	0	0	0
15	LCK_M_RS_U	0	0	0

Query executed succes... | server2012 (11.0 RTM) | sa (58) | master | 00:00:00 | 650 rows

The results include 650 rows of wait statistics, many of which are 0.

A better way to format this would be to dump the results into a table variable, then you can run operations like sorting on the result set, or filtering out the 0 values.

SQLQuery1.sql - server2012.master (sa (58))*

Editor | Results | Messages

	WaitType	Requests	WaitTime	SignalWaitTime
1	DIRTY_PAGE_POLL	3842319	386156800	22
2	REQUEST_FOR_DEADLOCK_SEARCH	77223	386156512	386156512
3	LAZYWRITER_SLEEP	385818	386145984	17328
4	LOGMGR_QUEUE	3773502	386131680	8018
5	XE_TIMER_EVENT	86864	386115072	386115072
6	HADR_FILESTREAM_IOMGR_IOCOMPLETI	771479	386092800	395
7	CHECKPOINT_QUEUE	11075	385772288	7935
8	FT_IFTS_SCHEDULER_IDLE_WAIT	6383	382926208	23
9	WAITFOR	51564	202647632	13
10	SLEEP_TASK	1400930	193053136	5223
11	BROKER_TO_FLUSH	188417	193046240	71
12	BROKER_TASK_STOP	390	3900405	8
13	DISPATCHER_QUEUE_SEMAPHORE	173807	3849135	4340
14	BACKUPTHREAD	359854	2507732	6931
15	BACKUPIO	2829466	1303315	50428
16	WRITELOG	291984	767240	46099

Query executed successfully. | server2012 (11.0 RTM) | sa (58) | master | 00:00:00 | 44 rows

If you are interested in Wait Stats, another way to track Wait Stats over time is with the Database Health Reports application which includes real time reporting as well as historic (over time) wait stat reporting. With the Database Health Reports application you can even track down the query that is causing the most wait stats.

Example of Clearing Wait Statistics:

The following example will clear the wait stat counters on your SQL Server.

Example of Clearing Latch Statistics:

The following example clears all the latch statistics on your SQL Server.

Example of Viewing Transaction Log Utilization:

The following example lists the raw output from DBCC SqlPerf (logspace), which show the overall usage of the log files for your database.

SQLQuery1.sql - server2012.master (sa (58))*

Editor | Results | Messages

	Database Name	Log Size (MB)	Log Space Used (%)	Status
1	master	0.7421875	65.52631	0
2	tempdb	0.9921875	54.37992	0
3	model	0.7421875	67.89474	0
4	msdb	19.61719	35.14038	0
5	ReportServer	6.867188	18.44425	0
6	ReportServerTempDB	1.054688	50.97222	0
7	Northwind	3.367188	27.59571	0
8	pubs	3.367188	25.53654	0
9	DBHealthHistory	2599.117	1.167877	0
10	cte_demo	0.7578125	54.63918	0
11	JProCo	44.55469	17.75162	0
12	dbBasics	0.7578125	46.06959	0
13	dbTester	0.7578125	54.83247	0
14	dbSkillCheck	0.7578125	51.54639	0
15	dbMovie	0.7578125	57.15206	0
16	tsql2012	18.11719	12.54043	0
17	dbcc_corruption	101.5547	4.414762	0

Query executed successfully. | server2012 (11.0 RTM) | sa (58) | master | 00:00:00 | 17 rows

If we wanted to clean up the output, we could do something similar to the waitstats above by inserting the results into a table variable, and doing some sorting or filtering. How you sort or filter may be determined by where you are having trouble. For instance if you were running low on disk space, you could sort on LogSize MB.

SQLQuery1.sql - server2012.master (sa (58))*

Editor | Results | Messages

	DB	LogSizeMB	PercentUsed	Status
1	DBHealthHistory	2599	1.18127179145813	0
2	dbcc_corruption	101	4.41476249694824	0
3	JProCo	44	17.7581977844238	0
4	msdb	19	35.6282348632813	0
5	tsql2012	18	12.5565977096558	0
6	ReportServer	6	18.4442539215088	0
7	Northwind	3	27.5957069396973	0
8	pubs	3	25.5365428924561	0
9	ReportServerTempDB	1	50.9722213745117	0
10	master	0	65.5263137817383	0
11	tempdb	0	58.6614189147949	0
12	model	0	67.8947372436523	0
13	cte_demo	0	55.0257720947266	0
14	dbBasics	0	46.456184387207	0
15	dbTester	0	55.2190704345703	0
16	dbSkillCheck	0	51.932991027832	0
17	dbMovie	0	57.5386581420898	0

Query executed successfully. | server2012 (11.0 RTM) | sa (58) | master | 00:00:00 | 17 rows

Here we can see that one log file is about two and a half GB, and significantly larger than any other database log file. This one was caused by some load testing that I was doing on the Database Health Reports application. The interesting thing here is that on 1.18% of the log file is being used, so most of that two and a half GB is being wasted.

Posts related to unusually large log files:

- Unusually large log file at the Database Health Reports website.
- Blog: Steps to shrink an unusually large log file at SteveStedman.com.

Notes:

For more information see TSQL Wiki DBCC sqlperf.

DBCC Command month at SteveStedman.com is almost as much fun as cleaning up someone a mess left in the database by someone else.