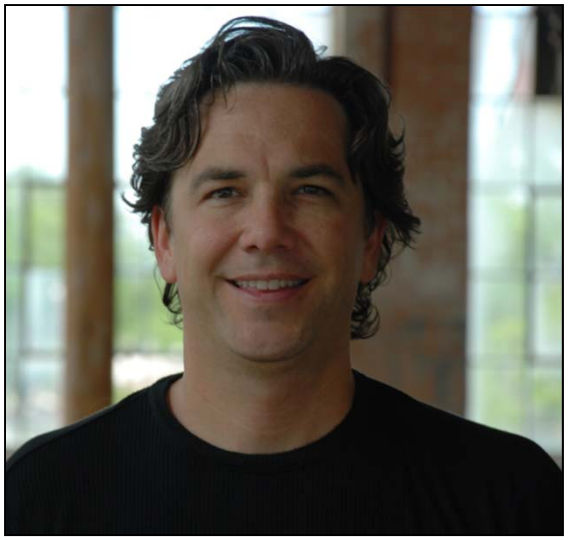
Chapter 10: Video 16

Waits and Queues



Chapter: Performance Monitoring
Course: SQL Server 2012 Database Administration
Course ID: 171
Instructor: Scott Whigham

LearnItFirst

DBAs need to be *effective* during crisis times

- Hardware/database failure isn't the only time you need to go into crisis mode
- Performance tuning is an art worth practicing!

Effective troubleshooting starts with identifying the problem accurately

- SQL Server's *wait stats* are often the best place to start

SQL Server's users request *resources*

- In a word: "data"
- SQL Server tracks any and all time spent waiting while it fulfills these requests
- These are your *wait stats*

Waits can be:

- … for disk I/O during a read or write
- … for memory pages to be read/write
- … for a GRANT (instead of a WAIT)
- … for a query plan to be created
- … for index maintenance to complete
- … for statistics to be created
- … for a thread to switch from one CPU to another (a.k.a. a “context switch”)
- … for a parallel query to complete
- … and much, much more

Waits can be seen with the DMVs/DMFs:

- `sys.dm_os_wait_stats` – shows the aggregate numbers for all waits
  - Defaults to showing aggregates since SQL Server service was started

Waits can be seen with the DMVs/DMFs:

- `sys.dm_os_waiting_tasks` – shows the currently suspended sessions
  - Correlate with `sys.dm_exec_request` and `sys.dm_exec_session` on `session_id`

When you have slowness on your system, check the wait stats

```
SELECT TOP(25) *
FROM sys.dm_os_wait_stats
ORDER BY wait_time_ms DESC
```

100 %

Results | Messages

	wait_type	waiting_tasks_count	wait_time_ms	max_wait_time_ms	signal_wa
1	LOGMGR_QUEUE	5398	683197	142	4
2	DIRTY_PAGE_POLL	6794	683114	116	3
3	LAZYWRITER_SLEEP	683	682610	1014	3
4	HADR_FILESTREAM_IOMGR_IOCOMPLETION	1364	682181	504	1
5	REQUEST_FOR_DEADLOCK_SEARCH	137	680045	5006	680045
6	SQLTRACE_INCREMENTAL_FLUSH_SLEEP	171	680034	4001	0
7	XE_TIMER_EVENT	147	650031	5022	650031
8	CHECKPOINT_QUEUE	8	638375	637507	1
9	XE_DISPATCHER_WAIT	6	570107	120000	0
10	BROKER_TO_FLUSH	334	341279	1037	1

LearnItFirst

Adding in a percentage column is helpful *(see query in .sql)*

wait_type	waiting_tasks_count	max_wait_time_ms	waits_ss	Percent
CXPACKET	21914	8616	4598.438000	62.87
ASYNC_IO_COMPLETION	1441	10049	513.246000	7.02
BACKUPBUFFER	213365	428	498.339000	6.81
BACKUPIO	123956	607	381.266000	5.21
WRITELOG	995663	718	341.857000	4.67
ASYNC_NETWORK_IO	273303	346	179.811000	2.46
LCK_M_U	64748	593	163.367000	2.23
PREEMPTIVE_OS_WAITFORSINGLEOBJECT	270771	324	161.668000	2.21
SOS_SCHEDULER_YIELD	1853909	701	79.219000	1.08
PREEMPTIVE_OS_AUTHENTICATIONOPS	552435	559	74.458000	1.02
MSQL_XP	41883	2119	68.734000	0.94
PAGEIOLATCH_SH	12775	229	37.809000	0.52
BACKUPTHREAD	6764	573	25.700000	0.35
PAGELATCH_EX	496128	503	21.636000	0.30
PREEMPTIVE_OS_QUERYREGISTRY	1939		20.047000	0.27

In the previous page, our top 5 waits:

- CXPACKET
- ASYNC_IO_COMPLETION
- BACKUPBUFFER
- BACKUPIO
- WRITELOG

CXPACKET waits

- Common, tricky
- Indicates parallel query usage
- High wait times (and high percentage of your waits) may be *ideal* for your database
  - If you also have high numbers of PAGEIOLATCH_.. waits, you need to fix it!
  - Experiment changing your MAXDOP to the number of cores in a single NUMA node to minimize switching

ASYNC_IO_COMPLETION waits

- Occurs when SQL Server has a task that is waiting on IO to complete
- Pay close attention to the other wait types here:
  - `BACKUPBUFFER`
  - `BACKUPIO`
  - `WRITELOG`
  - Notice a common thread?

This server would clearly benefit from a faster (or more) disks for the backup drive

- However, if backup times are within an acceptable range, you might be "okay"

Further reading:

- SQL Server 2005 Waits and Queues
- SQL Server Wait Type spreadsheet
- Books Online entry for wait stat types

In the next video…

– Using SQL Server Profiler to Spot Problems

"Those who dance are considered insane by those who can't hear the music."

- Friedrich Nietzche