SQL Server Waits and Queues

An introduction to waits and queues in SQL Server

Level 300

1 Hour

Who Am I?

- Passionate about understanding how databases work, performance and scalability
- Experience of large OLTP and DW estates
- Learning
- Working with SQL Server since 2005
- DBA since 2011
- Contact
  - LinkedIn: Rainer Unwin
  - Twitter: @RainerUnwin
  - Email: Rainer@RMUnwin.com

Overview

- Scheduling in SQL Server
- Tasks and workers
- Lists and queues in regards to waits in SQL Server
- Monitoring
- Examples of waits
- Resources

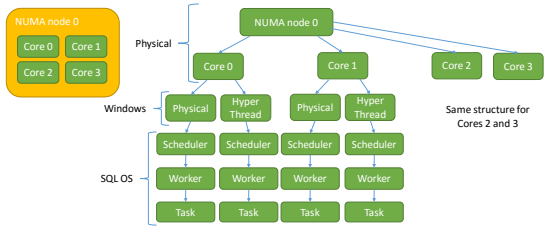
Wait stats and you

- What
  - What are they
  - What do they tell us
- Why
  - Why does any of this matter
- Who
  - Who is using them right now
  - How are you using them
  - What benefit are you getting

SQL Server Scheduling

- SQL Server uses co-operative scheduling
  - Workers will yield voluntarily
    - Yield on wait or quantum exhaustion
    - Differs to preemptive scheduling where tasks are externally interrupted with a scheduler determining what to run next
  - If a task needs to go external then the task goes pre-emptive (Windows scheduling)
    - File growth / creation
- Scheduler – 1 per logical CPU under normal circumstances
  - CAL Licencing or Standard edition limits available schedulers
  - Hidden schedulers
    - DAC
    - System (Resource Monitor, Log Writer, etc)

Schedulers in pictures



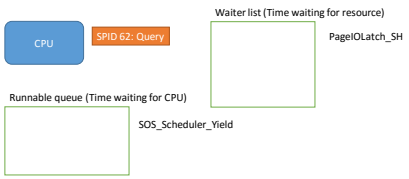
Tasks and Workers

- Tasks get created with incoming requests
  - Connection requests
  - Queries
  - Parallel queries have multiple tasks – control + 1 per thread
- Workers pick up tasks
  - Workers are created at start-up, more created or destroyed as needed
  - Task assigned to worker which itself is then assigned to a scheduler
  - Worker stays on that scheduler for life of task
  - Workers aren't re-assigned, even between tasks, if you use CPU affinity
  - Therefore CPU load is self-balancing as tasks come in

Task States

- Suspended – The task is waiting for an event to finish
  - In the waiter list waiting to be signalled that it can continue
- Runnable – The task is waiting to get back onto the CPU
  - In the runnable queue waiting for its turn on the scheduler
- Running – The task is executing on the CPU
  - The task will remain running until it needs to wait on a resource or exhausts its quantum (Which is 4ms in SQL Server)
- See DMV sys.dm_os_tasks on MSDN for more information

Scheduling in Action



Monitoring Waits

- DMVs
  - sys.dm_os_waiting_tasks
    - Shows currently waiting tasks (Snapshot in time)
  - sys.dm_os_wait_stats
    - Shows cumulative wait stats (Need to diff to get meaningful results)
    - Can be cleared with **DBCC SQLPERF ("sys.dm_os_wait_stats" , CLEAR);**
- Capture and store over time
- Eliminate benign waits
  - I use a table for ease of update, priority and time based filtering
- Paul Randal – List of benign waits and info on wait types

sys.dm_os_wait_stats

- Columns
  - wait_type
  - waiting_tasks_count
  - wait_time_ms - total amount of time spent waiting
  - max_wait_time_ms – single longest wait
  - signal_wait_time_ms – amount of time spent in runnable queue waiting for CPU
- Missing column
  - resource_wait_time_ms – amount of time spent in waiter list
  - calculated from (wait_time_ms- signal_wait_time_ms)

Interpreting Wait Times

- Is it relevant?
  - 100 seconds of PageIOLatch_SH?
  - Depends on your workload. You need historic values and trends
- Is it important
  - SQLTrace_Incremental_Flush_Sleep?
  - WriteLog
- High signal waits (>10-20%) - too little CPU
  - You need to work out if you need more CPUs or higher clock rate
- High resource waits
  - High depends on wait type (PageLatch_SH Vs Lck_M_Sh)
  - You need to investigate

Some Waits And What They Mean

- CXPacket
- OLEDB
- WriteLog
- PageIOLatch_XX
- Async_Network_IO
- SOS_Scheduler_Yield

Questions & Thank You

Resources

- Scheduling
  - Preemptive scheduling (Wikipedia)
  - Cooperative scheduling (Wikipedia)
- White papers
  - Microsoft whitepaper from 2006 (Microsoft.com)
  - Updated paper by SQLSkills.com (Erin Stellato & Jon Kehayias)
  - There are also Microsoft papers on latch stats and spinlock stats. These are very advanced and not properly documented so need deep knowledge
- Blogs
  - Paul Randal on wait stats (SQLSkills.com)
  - Wait type repository (MSDN)

