SQLintersection

Session: Wednesday, 10:00am-11:00am

Introduction to Performance Troubleshooting Using Wait Statistics

Paul S. Randal
paul@SQLskills.com

SQL intersection

SQLskills
immerse yourself in sql server

Paul S. Randal

- **Consultant/Trainer/Speaker/Author**
- **CEO, SQLskills.com**
  - Email: Paul@SQLskills.com
  - Blog: http://www.SQLskills.com/blogs/Paul
  - Twitter: @PaulRandal
  - 5 years at DEC responsible for the VMS file-system and chkdsk
  - Almost 9 years as developer/manager in the SQL Storage Engine team through August 2007, ultimately responsible for Core Storage Engine
- **Instructor-led training (US, UK, Ireland, Australia), consulting (anything you need)**
- **Online training:** pluralsight hardcore developer training **http://pluralsight.com/**
- **Get our bi-weekly newsletter: http://www.sqlskills.com/Insider**

Reminder: Intersect with Speakers and Attendees

- **Tweet tips and tricks that you learn and follow tweets posted by your peers!**
  - □ Follow: #SQLintersection and/or #DEVintersection
- **Join us – Wednesday Evening – for SQLafterDark**
  - □ Doors open at 7:00 pm
  - □ Trivia game starts at 7:30 pm
    - *Winning team receives something fun!*
  - □ Raffle at the end of the night
    - *Lots of great items to win including a seat in a SQLskills Immersion Event!*
  - □ The first round of drinks is sponsored by SentryOne and SQLskills

Overview

- **Very common to see 'knee-jerk' performance tuning where someone jumps to a conclusion based on superficial analysis of performance data**
- **Interpreting wait statistics is not hard, but needs practice**

- **We're going to cover**
  - Introduction
  - Thread lifecycle
  - Waits and wait times
  - DMVs
  - Some common wait types

SQL intersection

Interpreting the Data

- **Don’t do ‘knee-jerk’ performance troubleshooting**
  - Work through the data to see what may be the root cause
  - You’ll end up spending less time overall
- **Proficiency in using wait statistics data comes from:**
  - Retrieving the data correctly
  - Understanding what common wait types mean
  - Recognizing patterns
  - Avoiding inappropriate Internet advice
  - Practice!
- **Better is to have a series of snapshots of wait statistics over time**

SQL *intersection*

What are Waits?

- **The term ‘wait’ means that a thread running on a processor cannot proceed because a resource it requires is unavailable**
  - It has to wait until the resource is available
- **The resource being waited for is tracked by SQL Server**
  - Each resource maps to a wait type
- **Example resources that may be unavailable:**
  - A lock (LCK_M_XX wait type)
  - A data file page in the buffer pool (PAGEIOLATCH_XX wait type)
  - Results from part of a parallel query (CXPACKET wait type)
  - A latch (LATCH_XX wait type)

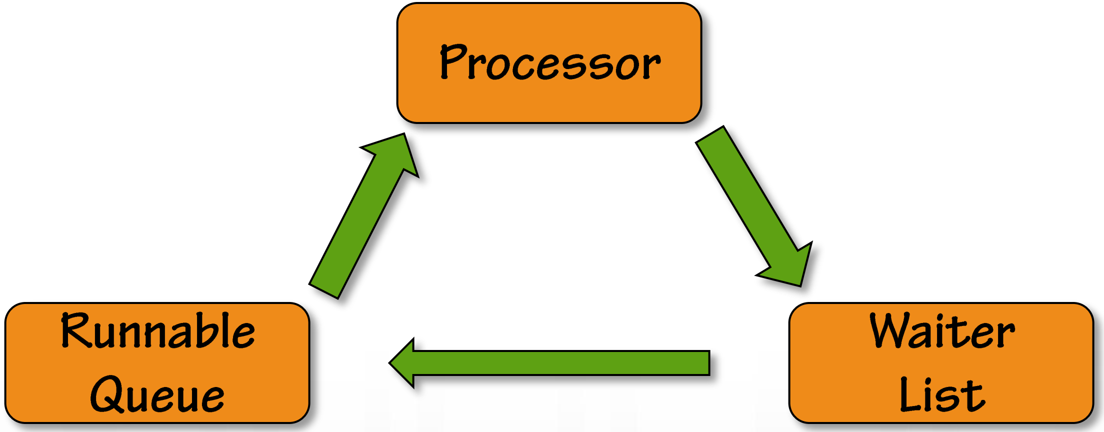
SQL intersection

Thread Scheduling

- **SQL Server performs its own thread scheduling**
  - Called non-preemptive scheduling
  - More efficient for SQL Server than relying on Windows scheduling
  - Performed by the SQLOS layer of the Storage Engine
- **Each processor core (whether logical or physical) has a scheduler**
  - A scheduler is responsible for managing the execution of work by threads
  - Schedulers exist for user threads and for internal operations
  - Use the sys.dm_os_schedulers DMV to view schedulers
- **When SQL Server has to call out to the OS, it must switch the calling thread to preemptive mode so the OS can interrupt it if necessary**

Components of a Scheduler

- All schedulers are composed of three 'parts'



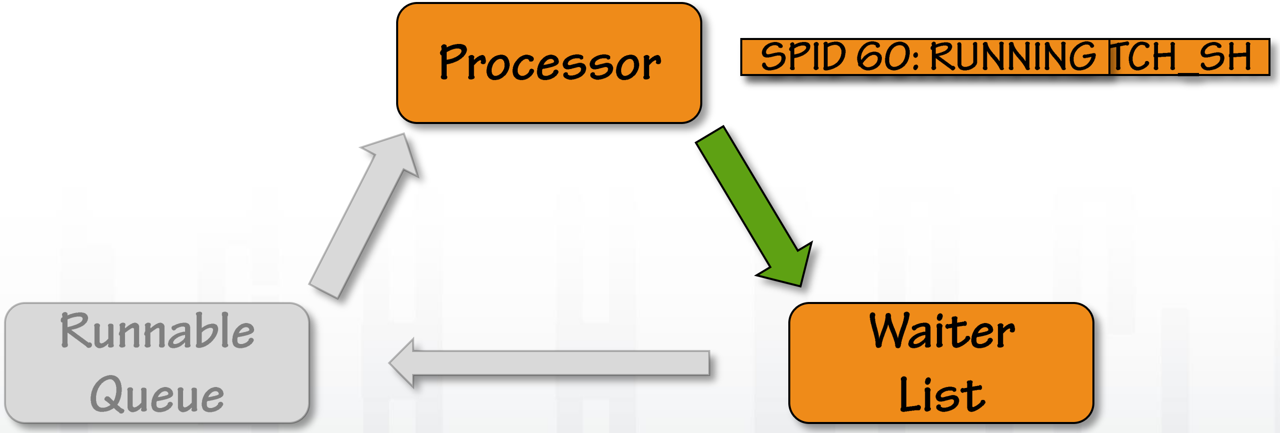
- Threads transition around these parts until their work is complete

Thread States

- **A thread can be in one of three states when being actively used as part of processing a query**
- **RUNNING**
  - The thread is currently executing on the processor
- **SUSPENDED**
  - The thread is currently on a Waiter List waiting for a resource
- **RUNNABLE**
  - The thread is currently on the Runnable Queue waiting to execute on the processor
- **Threads transition between these states until their work is complete**

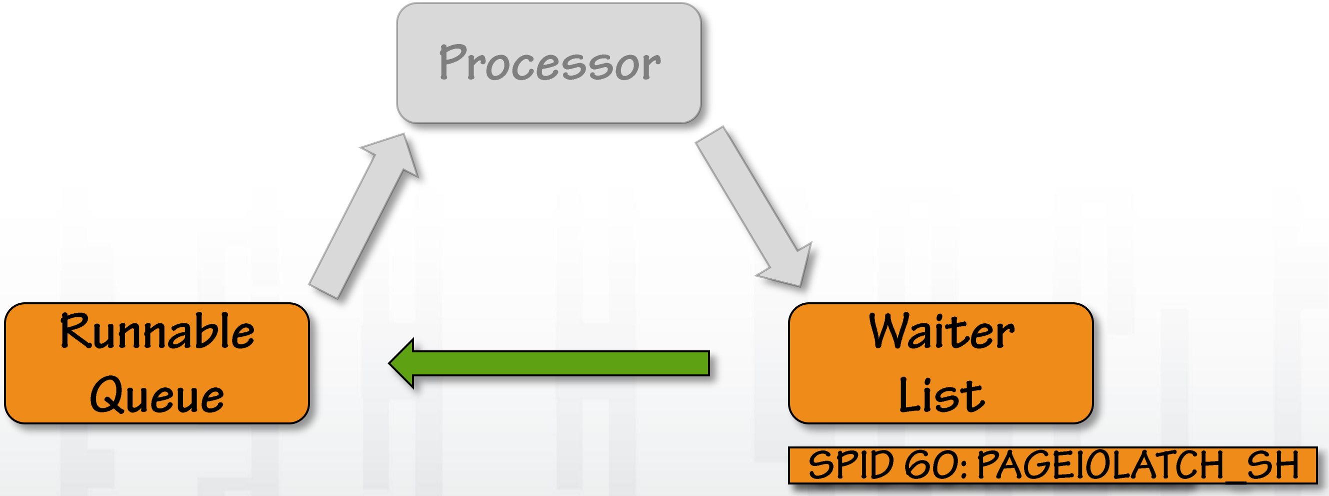
Transition: RUNNING to SUSPENDED

- **A thread continues executing on the processor until it must wait for a resource to become available**
  - The thread's state changes from RUNNING to SUSPENDED
  - The thread has been 'suspended' and moves to a Waiter List



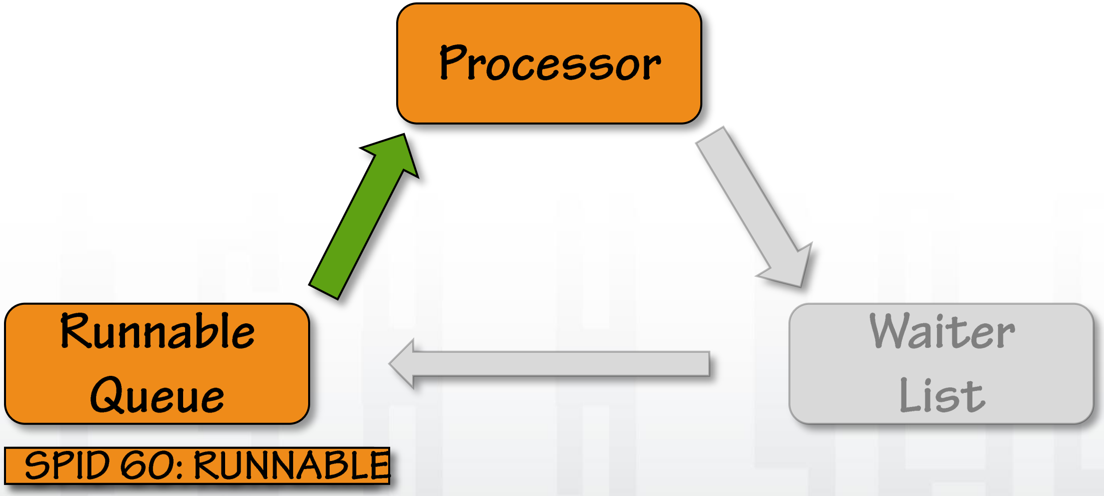
Transition: SUSPENDED to RUNNABLE

- **A thread continues to wait until it is told that the resource is available**
  - The thread's state changes from SUSPENDED to RUNNABLE
  - The thread moves to the Runnable Queue
  - This is called being 'signaled'

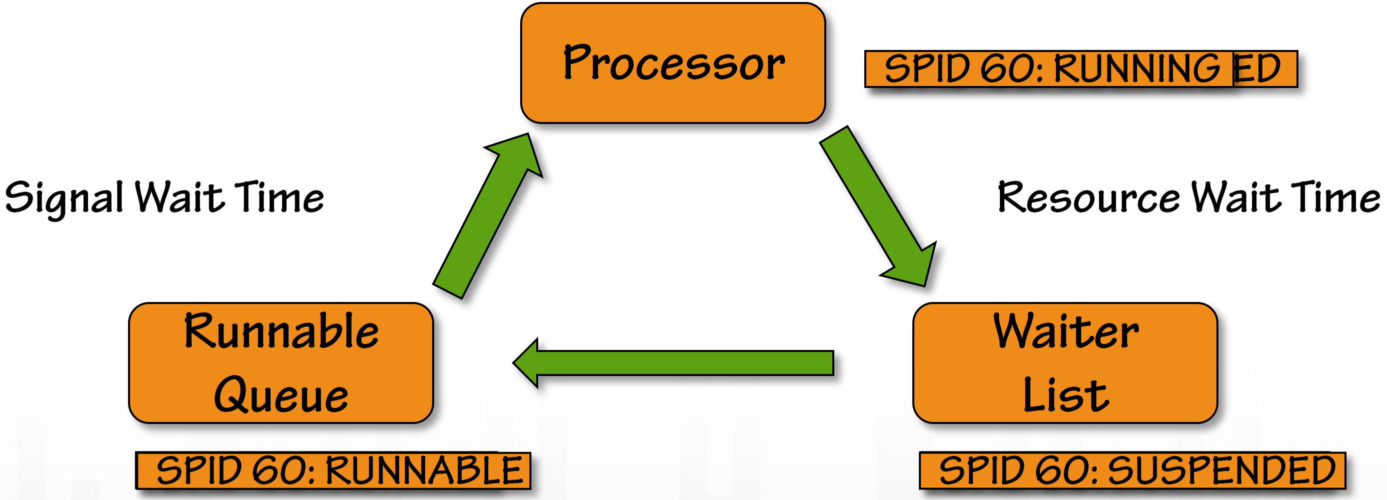


Transition: RUNNABLE to RUNNING

- **The thread waits on the Runnable Queue until it is picked as the next thread when the processor becomes available**
  - The thread’s state changes from RUNNABLE to RUNNING



Wait Times Definition



Wait Time = Resource Wait Time + Signal Wait Time

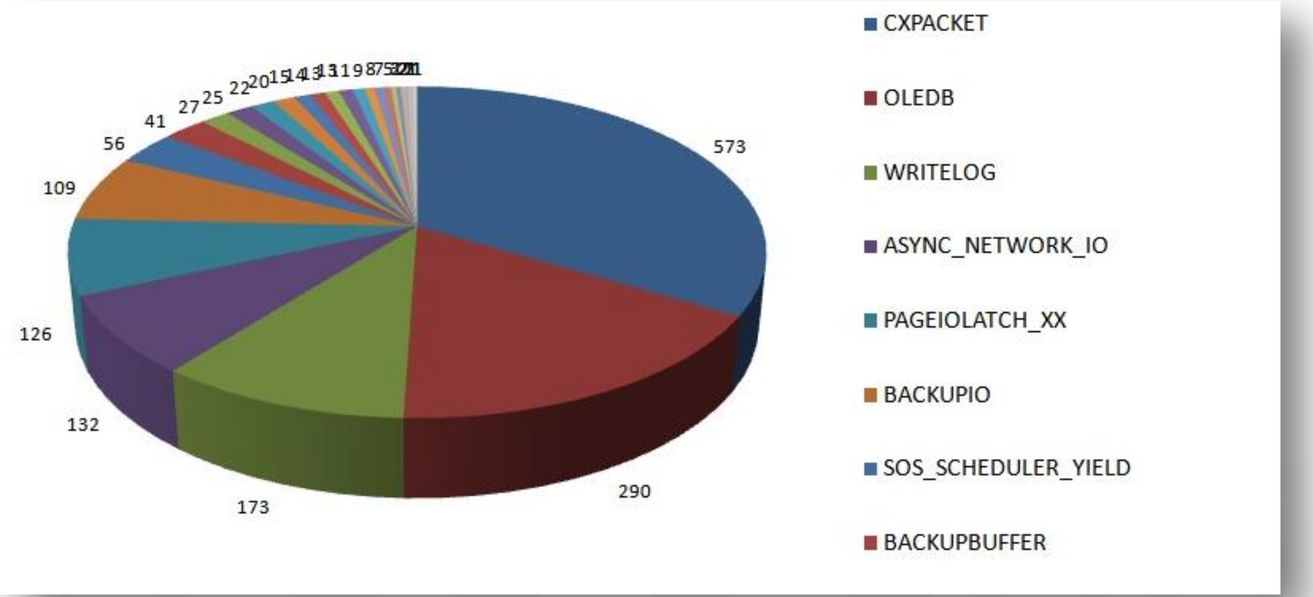
SQL intersection

DMVs

- **sys.dm_os_waiting_tasks**
  - Shows all threads that are currently suspended
  - Think of it as the ‘what is happening right now?’ view of a server
    - Usually very first thing to run when approaching a ‘slow’ server
- **sys.dm_os_wait_stats**
  - Shows aggregated wait statistics for all wait types
    - Aggregated since the server started or the wait statistics were cleared
  - Think of this as the ‘what has happened in the past?’ view of a server
- **Need to filter out benign waits and background tasks**
- **Need to use script around each of these (see demos…)**

Top Wait Types

- **Survey results from 1700+ SQL Server instances across Internet**



Source: my blog at http://bit.ly/1n1m1IF

PAGEIOLATCH_XX Wait and Solutions

- **Waiting for a data file page to be read from disk into memory**
  - Common modes to see are SH and EX
- **Do not assume the I/O subsystem or I/O path is the problem**
- **Further analysis:**
  - Determine which tables/indexes are being read
  - Analyze I/O subsystem latencies with sys.dm_io_virtual_file_stats
    - Move the affected data files to faster I/O subsystem?
  - Correlate with CXPACKET waits, suggesting parallel scans
    - Create appropriate nonclustered indexes and/or update statistics
  - Examine query plans for parallel scans and implicit conversions
  - Investigate buffer pool memory pressure and Page Life Expectancy
    - If data volume has increased, consider increasing memory

SQL intersection

PAGELATCH_XX Wait and Solutions

- **Waiting for access to an in-memory data file page**
  - Common modes to see are SH and EX
- **Do not confuse these with PAGEIOLATCH_XX waits**
- **Does not mean add more memory or I/O capacity**
- **Further analysis:**
  - Determine the page(s) that the thread is waiting for access to
  - Classic tempdb contention?
    - Add more tempdb data files, enable trace flag 1118, reduce temp table usage
  - Analyze the table and index structures involved
    - Excessive page splits occurring in indexes
    - Insert-point hotspot in a clustered index with an ever-increasing key

LCK_M_XX Wait and Solutions

- **A thread is waiting for a lock that cannot be granted because another thread is holding an incompatible lock**
- **Do not assume that locking is the root cause**
- **Further analysis:**
  - Follow blocking chain to see what the lead blocking thread is waiting for
  - Use blocked process report to capture info on queries waiting for locks
    - Michael Swart's blog post (http://bit.ly/ki3bYI)
  - Lock escalation from a large update or table scan?
    - Consider a different indexes, snapshot isolation, a different isolation level, or locking hints
  - Something preventing a transaction from releasing its locks quickly?
    - E.g. synchronous DBM/AG, DTC, or log throughput problems

Demo

Insert hotspot and using the waits DMVs

WRITELOG Wait and Solutions

- **Waiting for a transaction log block buffer to flush to disk**
- **Do not assume that log file I/O system has a problem (can be the case)**
- **Do not create additional transaction log files**
- **Further analysis:**
  - Write queue limited to 31/32? Upgrade to 2012+
  - Move the log to a faster I/O subsystem
  - Increase size of transactions to prevent many tiny log block flushes
    - Or used delayed durability in 2014+
  - Reduce logging from unused nonclustered indexes, page splits
  - Potentially split the workload over multiple databases or servers
    - Consider In-memory OLTP in 2014+

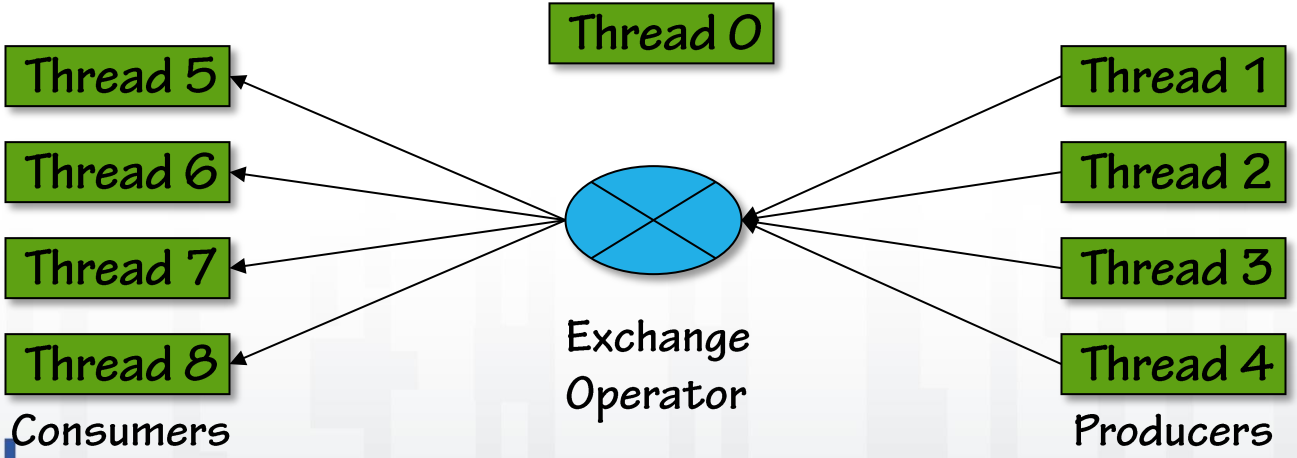
SQL intersection

Demo

Slow transaction log

Parallel Threads Example

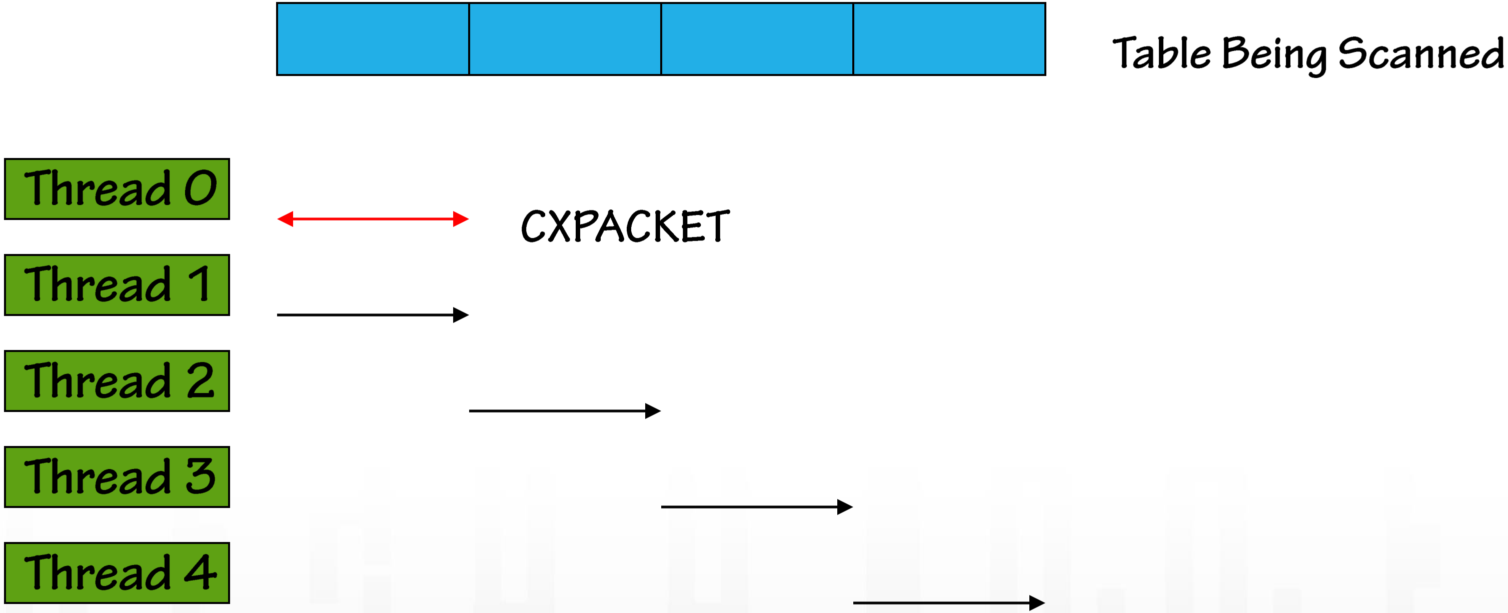
- **In a query plan, you may see the operator, for example**
- **This is a Repartition Streams operation**
  - Uses producer and consumer threads, plus a control thread
- **For a degree-of-parallelism = 4 operation, the threads would look like:**



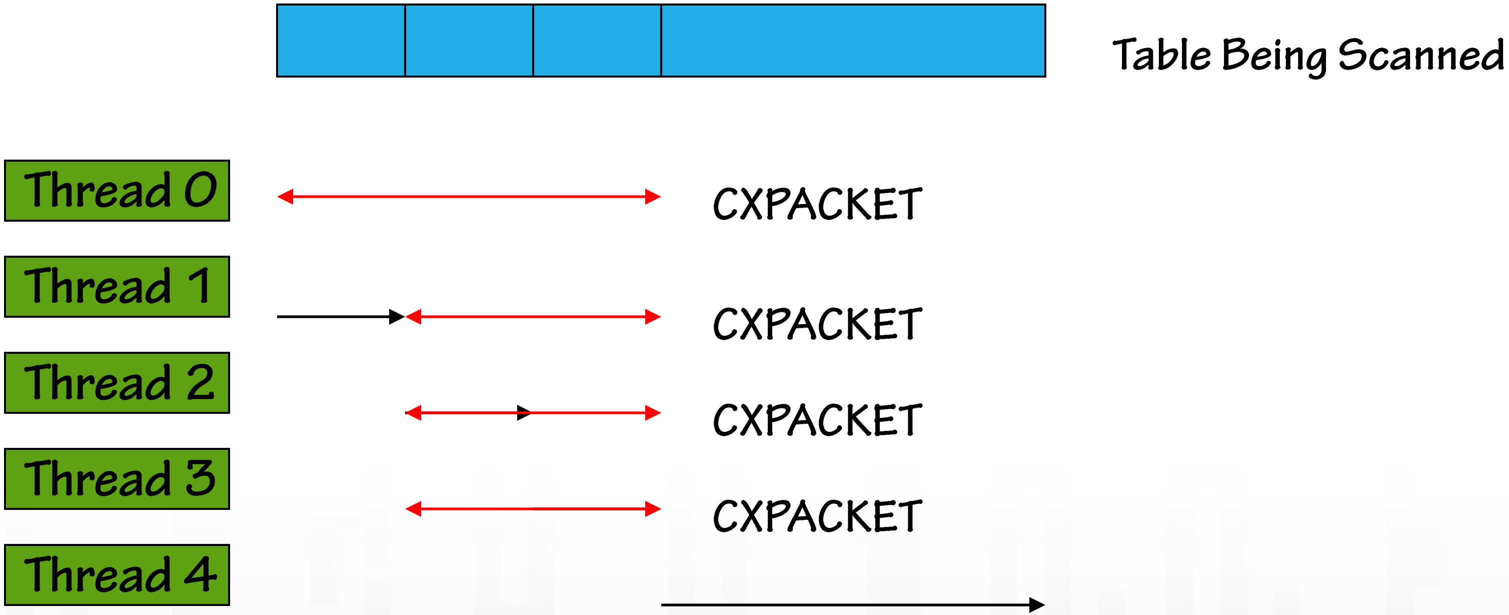
CXPACKET Wait Explanation

- **What does it mean:**
  - Parallel operations are taking place
  - Accumulating very fast implies skewed work distribution amongst threads or one of the workers is being blocked by something
- **Avoid knee-jerk response:**
  - Do not set server-wide MAXDOP to 1, disabling parallelism
- **Further analysis:**
  - Correlation with PAGEIOLATCH_SH waits? Implies large scans
  - Examine CXPACKET query plan to see if the query plans make sense
  - What is the wait type of the non-control parallel thread that is taking too long? (i.e. the thread that does not have CXPACKET as its wait type)

CXPACKET Wait Example (1)



CXPACKET Wait Example (2)



SQL intersection

CXPACKET Wait Solutions

- **Possible root-causes:**
  - Just parallelism occurring (e.g. http://bit.ly/1kPymkZ from CSS)
  - Table scans due to missing nonclustered indexes or incorrect query plan
  - Out-of-date statistics causing skewed work distribution
- **If there is actually a problem:**
  - Make sure statistics are up-to-date and appropriate indexes exist
  - Consider MAXDOP = physical cores per NUMA node, or 8 for non-NUMA
  - Consider MAXDOP for the instance, but beware of mixed workloads
  - Consider MAXDOP for the query or using Resource Governor for MAX_DOP
  - Consider setting 'cost threshold for parallelism' higher than query cost
    - Jon's blog post at http://bit.ly/1rTs9UX

Demo

Parallelism

ASYNC_NETWORK_IO Wait

- **What does it mean:**
  - SQL Server is waiting for a client to acknowledge receipt of sent data
- **Avoid knee-jerk response:**
  - Do not assume that the problem is network latency
- **Further analysis:**
  - Analyze client application code, client app server, network latencies
- **Possible root-causes and solutions:**
  - Usually poorly-coded application that is doing RBAR (Row-By-Agonizing-Row)
    - Very easy to show using a large query and SSMS on same machine as SQL Server
  - Could be from using MARS with large result sets
    - Otherwise look for network hardware issues, incorrect duplex settings, or TCP chimney offload problems (see http://bit.ly/aPzoAx)

Summary: Methodology

- **Gather information about exactly when the performance problem arose and the user-visible characteristics of the problem**
- **Gather information about what changed before the problem arose**
- **Examine the output from sys.dm_os_waiting_tasks**
  - What is happening on the server right now?
- **Examine the output from sys.dm_os_wait_stats**
  - What has happened in the past?
- **Look at the top 3-4 relevant waits**
- **Avoid temptation to knee-jerk and equate symptoms with root-cause**
- **Gather further information from relevant sources to pin-point problems**
  - DMVs, query plans, performance counters, code analysis

Resources

- **Comprehensive waits/latches library**
  - https://www.SQLskills.com/helps/waits
- **Whitepapers:**
  - SQL Server Performance Tuning Using Wait Statistics: A Beginners Guide
    - https://www.sqlskills.com/help/sql-server-performance-tuning-using-wait-statistics/
  - Diagnosing and Resolving Latch Contention on SQL Server
  - Diagnosing and Resolving Spinlock Contention on SQL Server
    - Gnarly links – see our whitepapers page at http://bit.ly/19j0cOd (zero then oh)
- **Blog post categories**
  - https://www.sqlskills.com/blogs/paul/category/wait-stats/ and /latches/ and /spinlocks/
- **Pluralsight:** SQL Server: Performance Tuning Using Wait Statistics

SQL intersection

Questions?

SQLskills
immerse yourself in sql server

Don’t forget to complete an online evaluation!

Introduction to Performance Tuning Using Wait Statistics

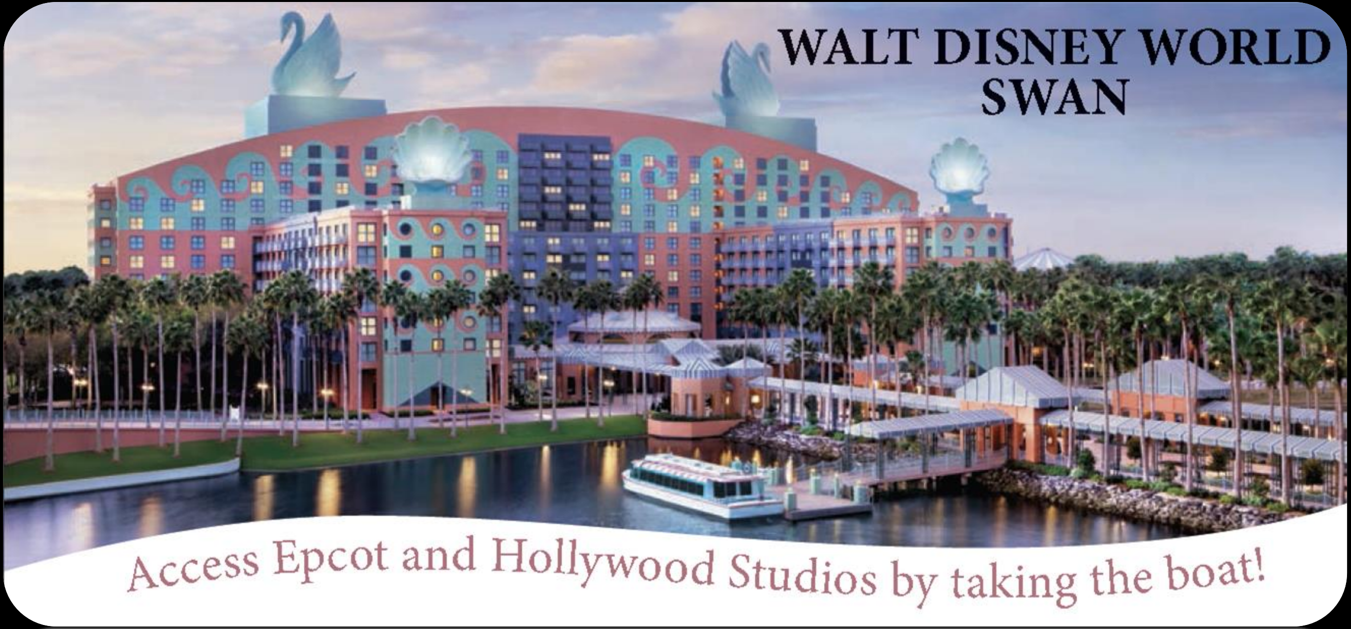
Your evaluation helps organizers build better conferences and helps speakers improve their sessions.

SQL intersection

Thank you!

Save the Date!

www.SQLintersection.com



2018

Mar 25-28

We're back in Orlando!

SQL intersection

Leave the every day behind and enter a world of wonder and enchantment at the Walt Disney World® Resort. Located in the heart of the most magical place on earth, the Walt Disney World Swan and Dolphin Resort provides a truly extraordinary backdrop for our event! Beautiful tropical landscaping, tranquil waterways, and classic art and architecture work together to create a stunning landmark!