SQLintersection

Session: 12/6, 11:30am - 12:30pm

Gems to Help You Troubleshoot Query Performance

Pedro Lopes
@SQLPedro

Speaker: Pedro Lopes

- Senior Program Manager
- SQL Server Tiger team owns in-market and vNext of SQL Server
- Focused on SQL Server Relational Engine (Query Processor, Query Perf)
- 8+ years at Microsoft
- 16+ years with SQL Server

@SQLPedro
pedro.lopes@microsoft.com
http://aka.ms/sqlserverteam

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 five-day SQLskills Immersion Event!
  - The first round of drinks is sponsored by SentryOne and SQLskills

SQL intersection

SQLskills immerse yourself in sql server & SentryOne™

Overview

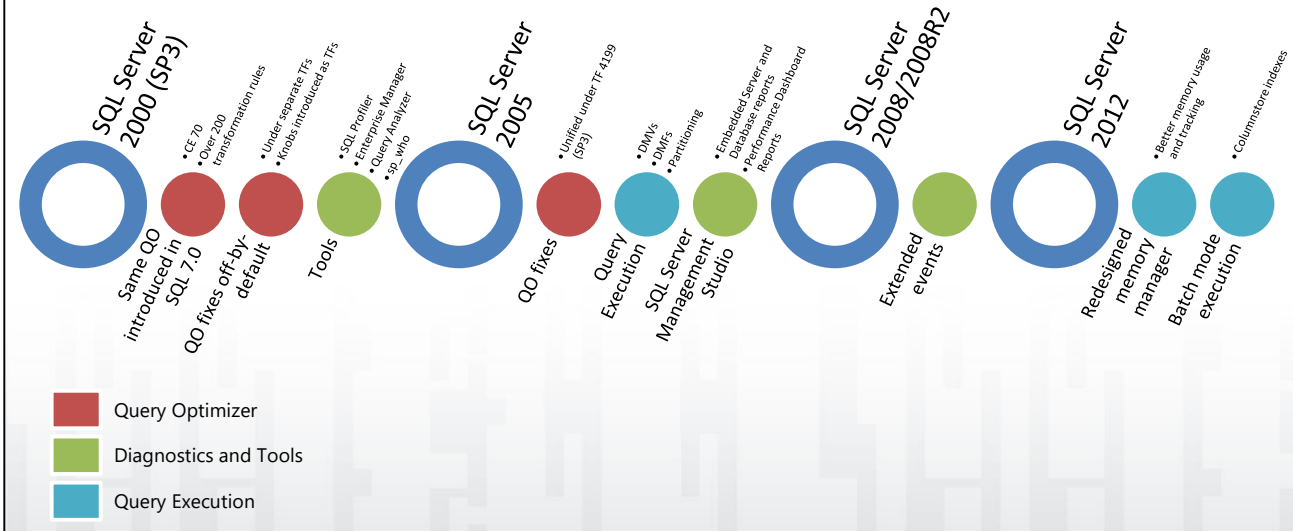
- A brief history of SQL Server query performance
- Diagnostics improvements
- Time permits: bonus improvement

SQL intersection

A brief history

SQL Server Query Performance

Query Performance Journey



Retired SQL 2000 docs available in PDF at
https://download.microsoft.com/download/5/4/A/54AFD350-6477-4910-9DF2-4C472C906684/SQL2000_release.pdf

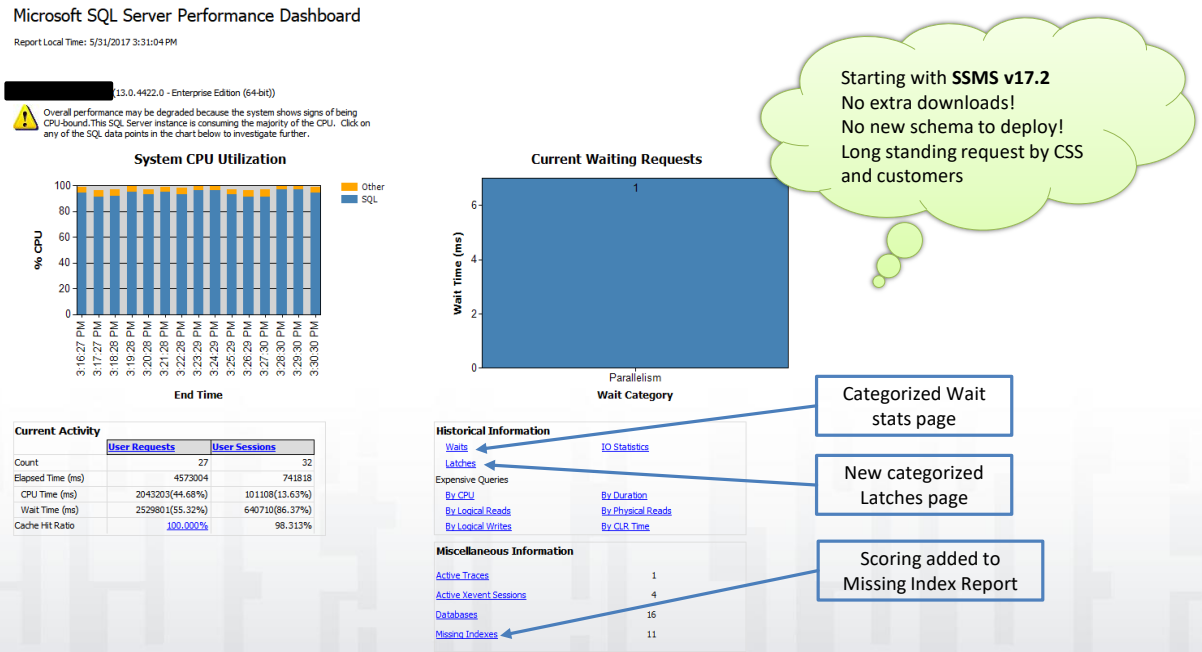
Query Performance Journey

Unlocking insights

Diagnostics Enhancements

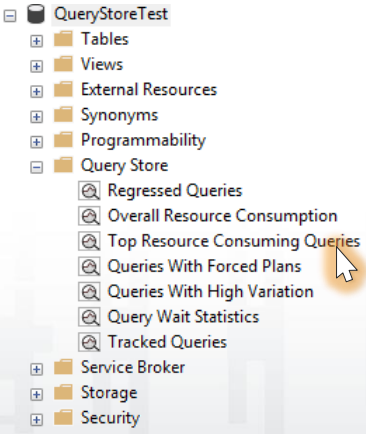
SQL intersection

Performance Dashboard in SSMS

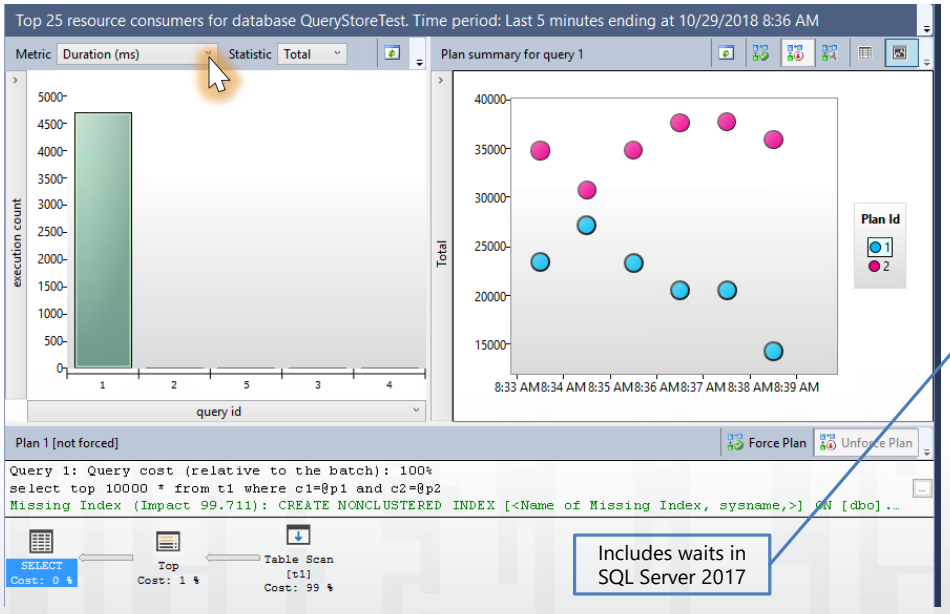


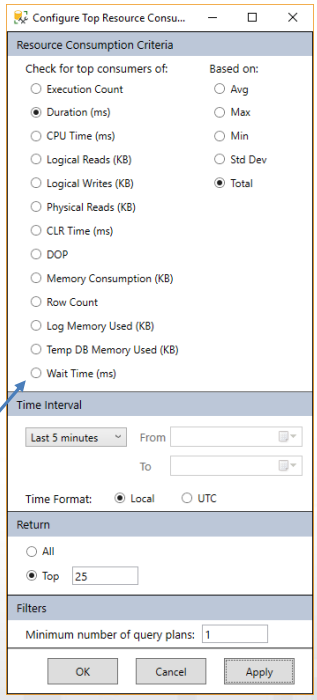
Query Store

Comprehensive query-performance information when you need it most!



Query Store – Top Consumers

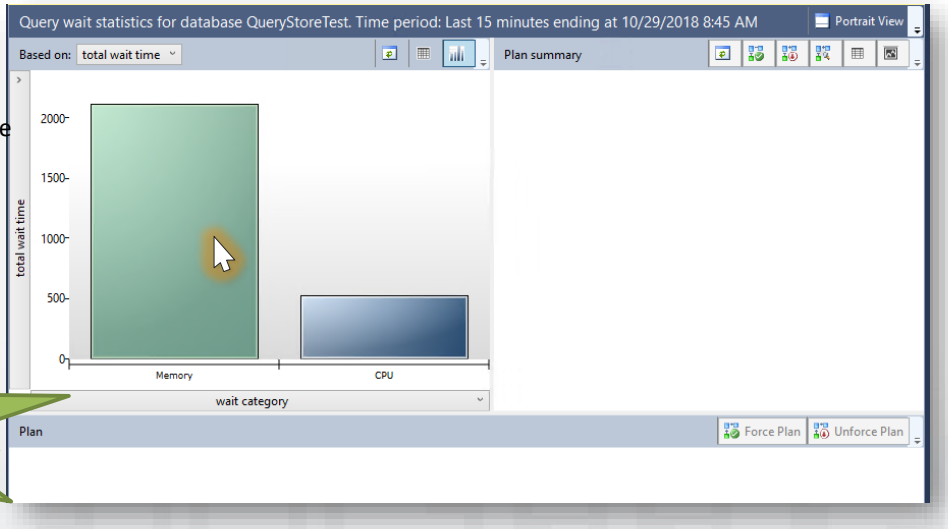




Includes waits in SQL Server 2017

Query Store – Wait Categories

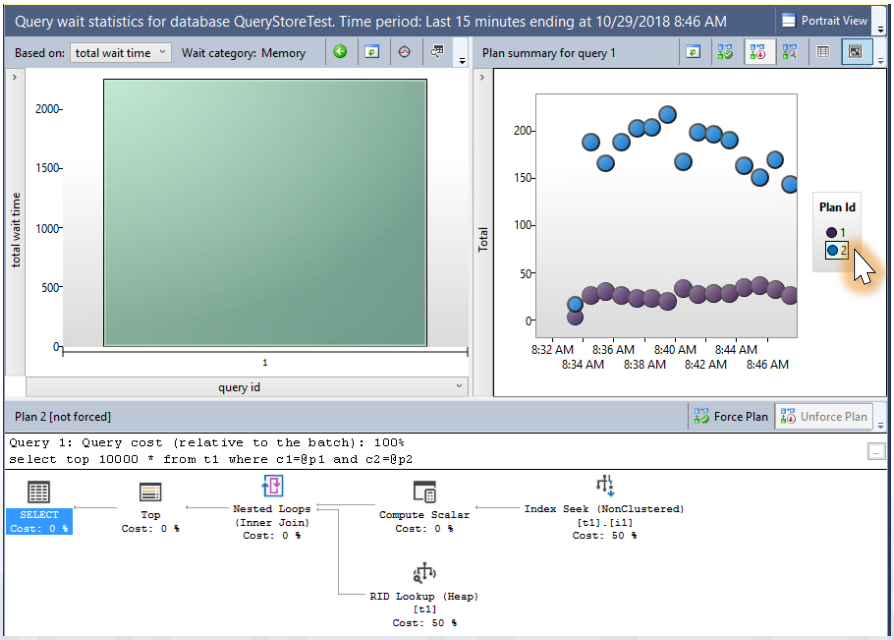
And now waits-based troubleshooting is also available in UI…



New for SSMS 18 Preview 5

Query Store – Wait Categories

Then, explore details on selected wait category



Using Query Store UI

Demo

5m – flow with wait stats, and then to Top resource consuming.

Query Analysis

Diagnostics Enhancements

Improved diagnostic data for stats blocking

Ever faced long-running queries that wait on synchronous statistics updates? Essentially a blocking scenario.

- **New wait type WAIT_ON_SYNC_STATISTICS_REFRESH in SQL Server 2019 is surfaced in sys.dm_os_wait_stats and sys.dm_exec_session_wait_stats.**

wait_type	waiting_tasks_count	wait_time_ms	max_wait_time_ms	signal_wait_time_ms
WAIT_ON_SYNC_STATISTICS_REFRESH	1	18781	18781	0

- **In sys.dm_exec_requests you see the command SELECT (STATMAN) if a SELECT is waiting for a synchronous stats update.**

session_id	request_id	start_time	status	command
59	0	2018-11-13 11:50:36.617	suspended	SELECT (STATMAN)

- **Not available in showplan waits – this is a pre-execution wait. Not in query store waits.**

Query plans: fundamental query perf diagnostics

- How data is accessed
- How data is joined
- Sequence of operations
- Use of temporary worktables and sorts
- Estimated rowcounts, iterations, and costs from each step
- Actual rowcounts and iterations
- How data is aggregated
- Use of parallelism
- Query execution warnings
- Query execution stats
- Hardware/Resource stats

SQL intersection

The query plan consists of physical operators which implement the operation described by logical operators.

The order of these determine how data is accessed. A query plan also determines what support objects must be created, such as worktables or workfiles in tempdb. Remember that these decisions rely heavily on statistics, which are somewhat shown in the execution plan in the form of estimated rows.

Some physical operators may perform other types of operations e.g. the Aggregate operator calculates an expression containing MIN, MAX, SUM, COUNT or AVG, for example.

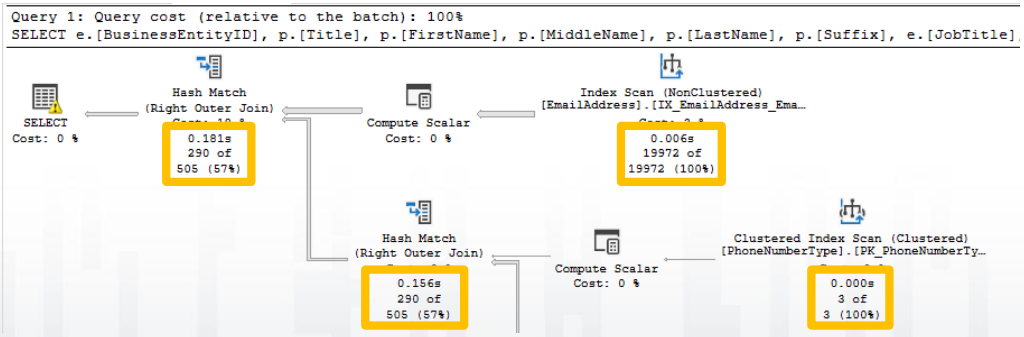
And we might have execution warnings.

More information can be found here:

Showplan Logical and Physical Operators Reference (https://msdn.microsoft.com/en-us/library/ms191158.aspx)

Faster identification of heavy nodes

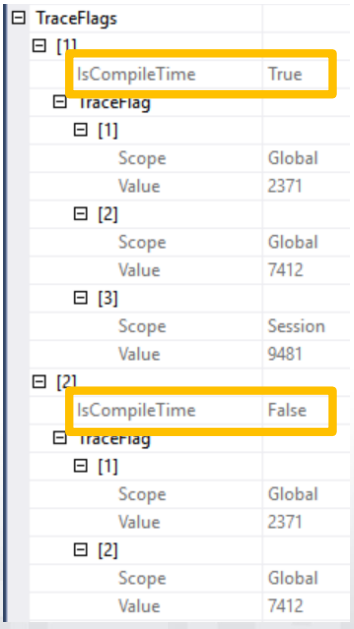
- **SSMS v18 showplan surfaces information on:**
  - Elapsed time per operator that consumes data
  - <actual rows> of <estimated rows> (percent of estimate)



Note: even on an Actual execution plan, the Cost pct is based on estimations. This is not an accurate measure of true operator cost.

Getting all context info in Showplan: Trace Flags

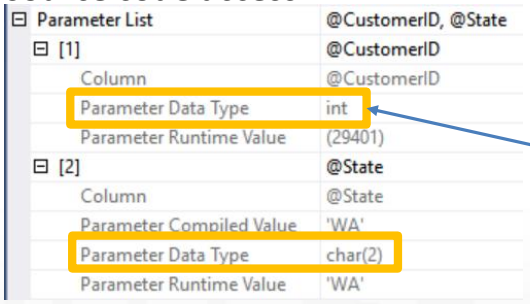
- **Shows list of active trace flags:**
  - Query
  - Session
  - Global
- **Useful to understand if active Trace Flags influence execution context**



In SQL 2012 SP4, SQL 2014 SP3, SQL 2016 SP1 and SQL Server 2017

Getting all context info in Showplan: Param Types

- **Easier detection of type conversion issues even when you don't have source code access.**



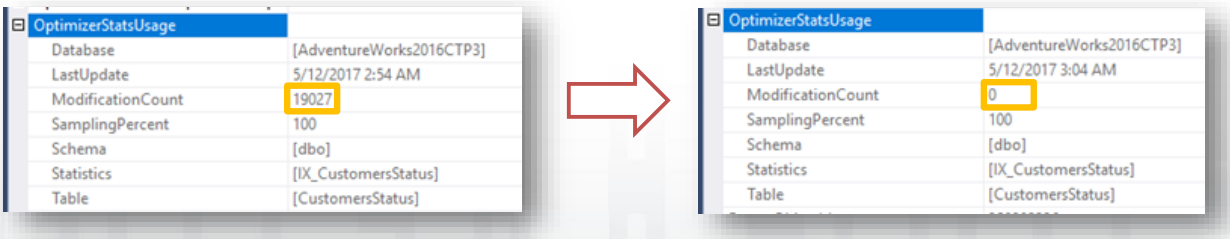
Look for conversion warnings

```
DECLARE @P1 int;
EXEC sp_prepare @P1 output,
    N'@P1 nvarchar(128)',
    N'SELECT database_id, name
            FROM sys.databases
            WHERE name=@P1';

EXEC sp_execute @P1, N'tempdb';

EXEC sp_unprepare @P1;
```

In SQL 2016 SP1 and SQL 2017

Getting all context info in Showplan: Statistics

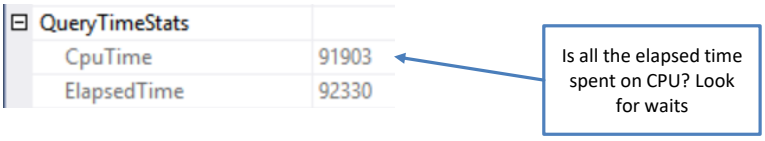
- **Identify which statistics were used by the Query Optimizer for a given compilation.**
- **Gain actionable insight to where estimations came from.**



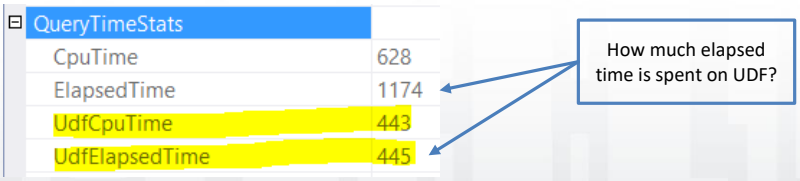
In SQL 2017 and SQL 2016 SP2

Getting all context info in Showplan: Times

- **Persisting information on elapsed and CPU times**



- **And Scalar UDF elapsed and CPU times**



QueryTimeStats: In SQL 2012 SP4, SQL 2014 SP3, SQL 2016 SP1 and SQL Server 2017.
UdfTimeStats: In SQL 2017 CU3, SQL 2016 SP2 and SQL 2014 SP3

Although very useful as a means to create a reusable routine, when using UDFs referenced in a query, you may be facing hidden costs that can be detrimental to performance, in a cursor like execution model.

There are several option to replace UDFs, such as Inline expressions for simple functions or derived tables if possible.

The internal execution of these functions may also be hidden from the query plan of the calling query, as we will see in the upcoming demo.

If you are using simple T-SQL UDFs that do not touch any tables (meaning do not access data), make sure you specify the SCHEMABINDING option during creation of the UDFs. This will make the UDFs schema-bound and ensure that the query optimizer does not generate any unnecessary spool operators for query plans involving these UDFs.

For UDFs that are schema-bound, any attempt to change the underlying objects' schema will raise an error. Using this option ensures that the UDF will not inadvertently break due to changes of an underlying object's schema.

Getting all context info in Showplan: Waits

- **Shows top 10 waits from sys.dm_exec_session_wait_stats**

Correlate waits with overall query times

QueryTimeStats	
CpuTime	1045
ElapsedTime	3010

Misc	
Cached plan size	160 KB
CardinalityEstimationModelV	130
CompileCPU	11
CompileMemory	728
CompileTime	136
DatabaseContextSettingsId	3
Degree of Parallelism	12
Estimated Number of Rows	121308
Estimated Operator Cost	0 (0%)
Estimated Subtree Cost	4.48002
Memory Grant	80448
MemoryGrantInfo	
Optimization Level	FULL

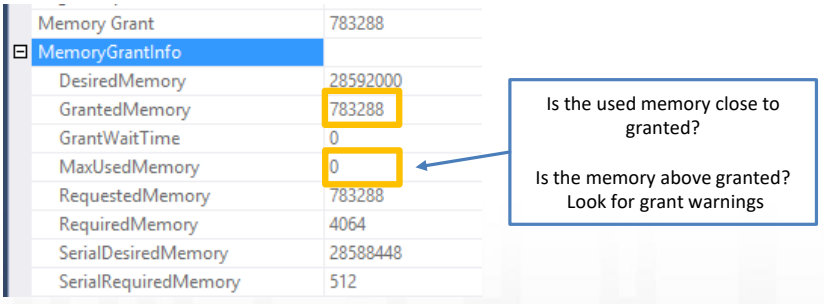
WaitStats	
[1]	
WaitCount	98
WaitTimeMs	3
WaitType	LATCH_SH
[2]	
WaitCount	50
WaitTimeMs	761
WaitType	PAGEIOLATCH_SH
[3]	
WaitCount	67
WaitTimeMs	1942
WaitType	LATCH_EX
[4]	
WaitCount	129
WaitTimeMs	2509
WaitType	ASYNC_NETWORK_IO
[5]	
WaitCount	2220
WaitTimeMs	30622
WaitType	CXPACKET

Note: Parallelism waits available in SQL Server 2017 CU3 and 2016 SP2

SQL 2016 SP1 and SQL 2017

Getting all context info in Showplan: memory

- **Showplan extended to include grant usage per thread and iterator**



- **Also found in sys.dm_exec_query_stats**

SQL 2017 and SQL 2016 SP2

Memory grant info:

SerialRequiredMemory: Required memory in KB if the query runs in serial mode. The query will not start without this memory.

SerialDesiredMemory: Memory estimated to fit intermediate results in KB if the query runs in serial mode.

RequiredMemory: Required memory in KB for the chosen degree of parallelism. If the query runs in serial mode, this is the same as SerialRequiredMemory.

DesiredMemory: Memory estimated to fit intermediate results in KB for the chosen degree of parallelism. If the query runs in serial mode, this is the same as SerialDesiredMemory.

RequestedMemory: Memory in KB which the query requests the memory manager to grant. This can be smaller than sum of RequiredMemory and DesiredMemory if it exceeds the maximum allowed for single query.

GrantWaitTime: Time in seconds if the query has to wait for successful memory grant.

MaxUsedMemory: Maximum memory in KB used by the query.

MaxQueryMemory: Maximum memory in KB allowed for single query.

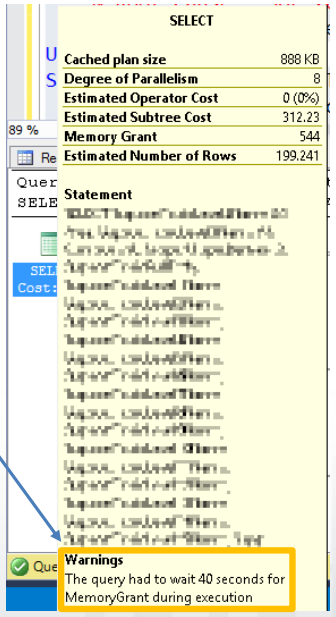
Notice the warnings

Analyzing query plan properties

Warning: Memory Grant Wait

- **Occurs when a T-SQL statement or stored procedure waits more than one second for a memory grant or when the initial attempt to get memory fails.**

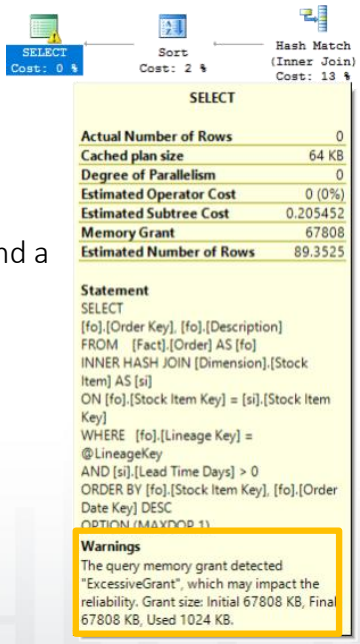
RESOURCE_SEMAPHORE waits may indicate excessive number of concurrent queries, or excessive memory request amounts



In SQL 2012 +

Ex. Use this event in combination with events that identify waits to troubleshoot contention issues that impact performance. For excessive mem grant, you can see how to improve estimations (perhaps update stats), or administratively by using MIN_PERCENT_GRANT and /or MAX_PERCENT_GRANT query hints.

Warning: Memory Grant



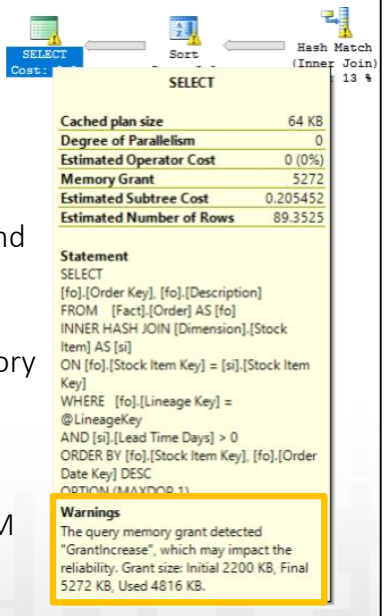
3 conditions:

- Excessive Grant: when max used memory is too small compared to the granted memory. This scenario can cause blocking and less efficient usage when large grants exist, and a fraction of that memory was used.

Also in SQL Server 2014 SP2 and SQL Server 2016 SP1.

Ex. For excessive mem grant, you can see how to improve estimations (perhaps update stats), or administratively by using MIN_PERCENT_GRANT and /or MAX_PERCENT_GRANT query hints.

Warning: Memory Grant

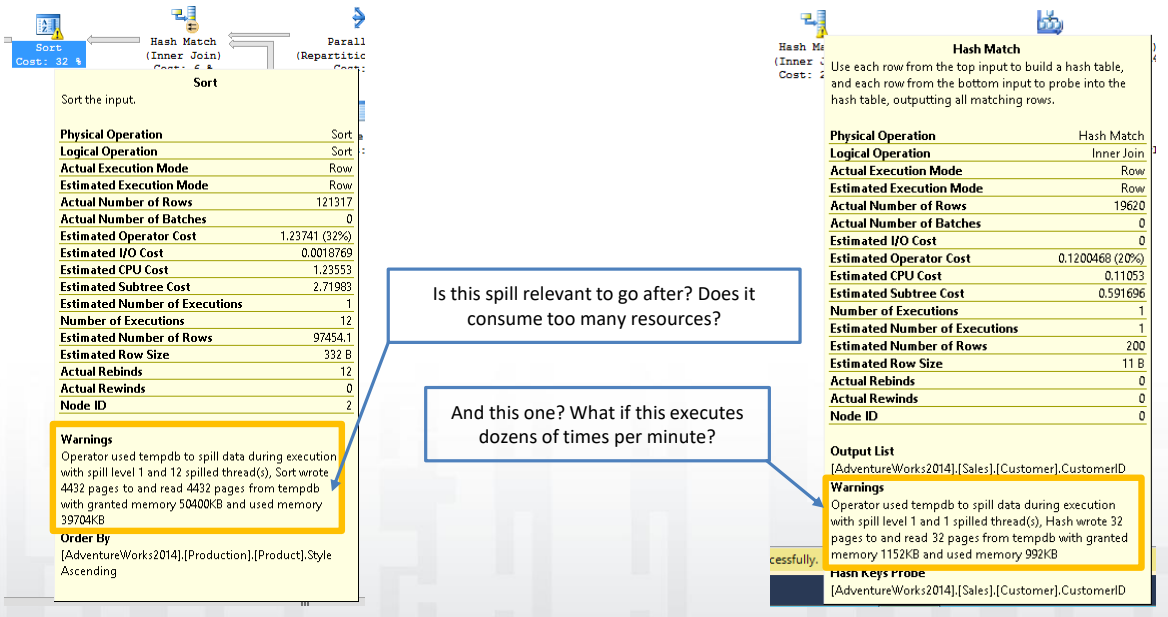


3 conditions:

- Excessive Grant: when max used memory is too small compared to the granted memory. This scenario can cause blocking and less efficient usage when large grants exist, and a fraction of that memory was used.
- Grant Increase: when the dynamic grant starts to increase too much, based on the ratio between the max used memory and initial request memory. This scenario can cause server instability and unpredictable workload performance.
- Used More Than Granted: when the max used memory exceeds the granted memory. This scenario can cause OOM conditions on the server.

Ex. For excessive mem grant, you can see how to improve estimations (perhaps update stats), or administratively by using MIN_PERCENT_GRANT and /or MAX_PERCENT_GRANT query hints.

Warning: Spills



In SQL 2014 SP2; SQL 2016

Sort Spills = sort operations that do not fit into memory

Supported by a Worktable in TempDB

Spill level 1

Means one pass over the data was enough to complete the sort.

Spill level 2

Means multiple passes over the data are required to sort the data

Hash Spills = hash recursion or cessation of hashing (hash bailout) has occurred during a hashing operation

Supported by a Workfile in TempDB

Spill level 1 = Hash recursion

Occurs when the build input does not fit into available memory, resulting in the split of input into multiple partitions that are processed separately.

If any of these partitions still do not fit into available memory, it is split into sub-partitions, which are also processed separately. This splitting process continues until each partition fits into available memory or until the maximum recursion level is reached.

Spill level 2 = Hash bailout

Occurs when a hashing operation reaches its maximum recursion level and shifts to an alternate plan to process the remaining partitioned data.

Faster insights

The middle-of-the-night call

- You're on call for supporting the data tier of a mission-critical SQL Server instance.
- Key business processes are being delayed when ETL is running.
- You get a call asking to mitigate the issue and then determine the root cause.

Defining the problem

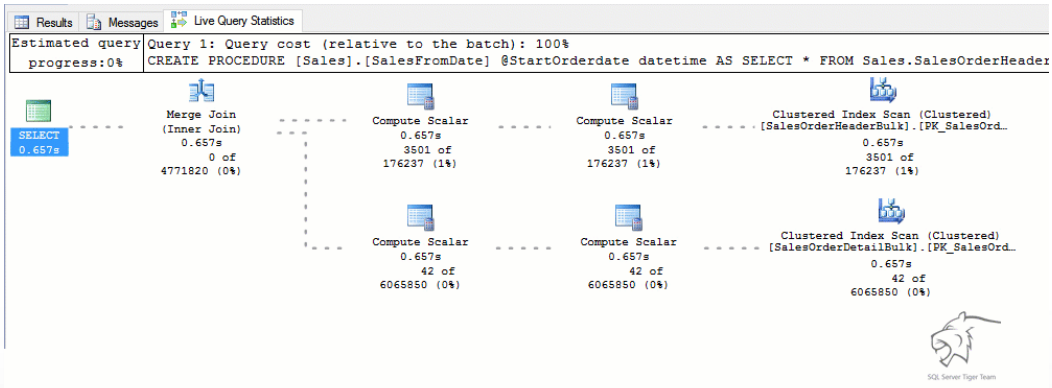
Reasonable hypothesis: a long running query.

Query completion is a prerequisite for the availability of an actual query plan.

Actual query plans unsuitable for troubleshooting complex performance issues:

- **Long running queries**
- **Queries that run indefinitely and never finish execution.**

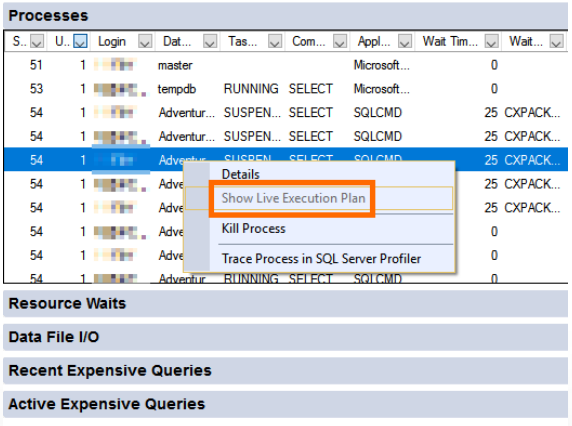
What if I could do live query troubleshooting?



What if I could do live query troubleshooting?

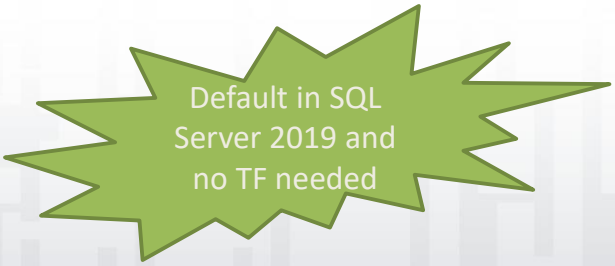
Ok, but to have in-flight query execution visibility, the *query execution statistics profile infrastructure* must be enabled on demand:

- **Overhead goes up to 75% with TPC-C like workload.**
- **It makes bad things worse if running all the time…**



Query progress – anytime, anywhere

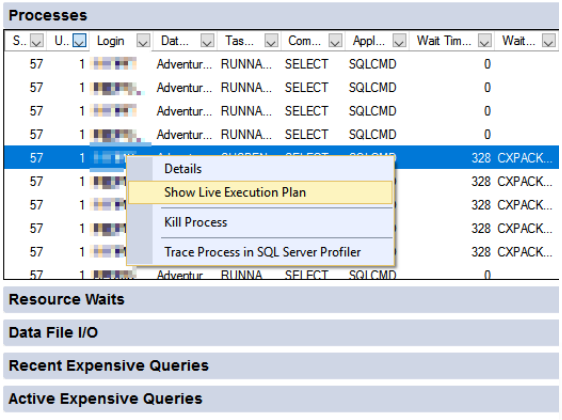
- **Starting with SQL Server 2016 SP1* and 2017, the new *lightweight query execution statistics profile infrastructure* allows continuous collection of per-operator query execution statistics**
  - Using global TF 7412
  - Enabling *query_thread_profile* and *query_plan_profile* extended event
  - Using query hint USE HINT('query_plan_profile') in SQL Server 2017 CU11 and 2016 SP2 CU3 (KB 4458593)



* Also available in SQL 2014 SP2 and 2016 RTM as a less optimized versions than SQL 2016 SP1 and 2017. Becomes default ON for SQL 2019.

Query progress – anytime, anywhere

- **When lightweight profiling is on, *sys.dm_exec_query_profiles* is also populated for all sessions**
  - Enables new LQS feature in SSMS (including Activity Monitor)
  - New DMF sys.dm_exec_query_statistics_xml
- **The following still use regular profiling infra**
  - SET STATISTICS XML (or Include Actual Plan)
  - query_post_execution_showplan extended event



* Also available in SQL 2014 SP2 and 2016 RTM as a less optimized versions than SQL 2016 SP1 and 2017.

See more information in https://blogs.msdn.microsoft.com/sql_server_team/query-progress-anytime-anywhere/ and https://docs.microsoft.com/en-us/sql/relational-databases/performance/query-profiling-infrastructure

Bringing it all together with live troubleshooting + new xEvents

Demo

3m demo on finding offending workload with LQS and fixing it with TF OFF.

What is the impact of Lightweight Profiling?

Query Execution Statistics Profiling Infrastructure tests with TPC-C like workloads

	Overhead percent (up to)	
Infra Type	**no active xEvents**	**Active xEvent query_post_execution_showplan**
Regular	**75.5**	93.17
Lightweight in SQL Server 2014 SP2/2016	3.5	62.02
Lightweight in SQL Server 2016 SP1 and above	**2**	14.3

Reference: https://blogs.msdn.microsoft.com/sql_server_team/query-progress-anytime-anywhere/

The infamous truncation error

Loading data into a table, and hitting this?

```
Msg 8152, Level 16, State 30, Line 13
String or binary data would be truncated.
The statement has been terminated.
```

Starting with SQL Server 2019 CTP 2.1, you get this:

```
Msg 2628, Level 16, State 1, Line 14
String or binary data would be truncated in table
'AdventureWorks2016CTP3.Sales.SalesOrderHeaderTest', column
'CreditCardApprovalCode'. Truncated value: '1231736Vi8604'.
The statement has been terminated.
```

Also in SQL Server 2017 CU12 and upcoming SQL Server 2016 SP2 CUx

Needs TF 460

See more in blog post @ http://aka.ms/sqlserverteam

Link to blog post: https://blogs.msdn.microsoft.com/sql_server_team/string-or-binary-data-would-be-truncated-replacing-the-infamous-error-8152/

Bookmarks

SQL Server Team (Tiger) Blog	http://aka.ms/sqlserverteam
Tiger Toolbox GitHub	http://aka.ms/tigertoolbox
SQL Server Release Blog	http://aka.ms/sqlreleases
SQL Server Performance Center	http://aka.ms/sqlperfcenter
BP Check	http://aka.ms/bpcheck
SQL Server Standards Support	http://aka.ms/sqlstandards
Trace Flags	http://aka.ms/traceflags
SQL Server Support lifecycle	http://aka.ms/sqllifecycle
SQL Server Updates	http://aka.ms/sqlupdates
SQL Server Guides	http://aka.ms/sqlserverguides
SQL Feedback (New “Connect”)	http://aka.ms/sqlfeedback
T-SQL Syntax Conventions	http://aka.ms/sqlconventions
SQL Server Error Codes	http://aka.ms/sqlerrors
Database Compatibility	http://aka.ms/dbcompat

Questions?

Microsoft

Don’t forget to complete an online evaluation!

Practical guidance to make your tier-1 SQL Server ROAR!

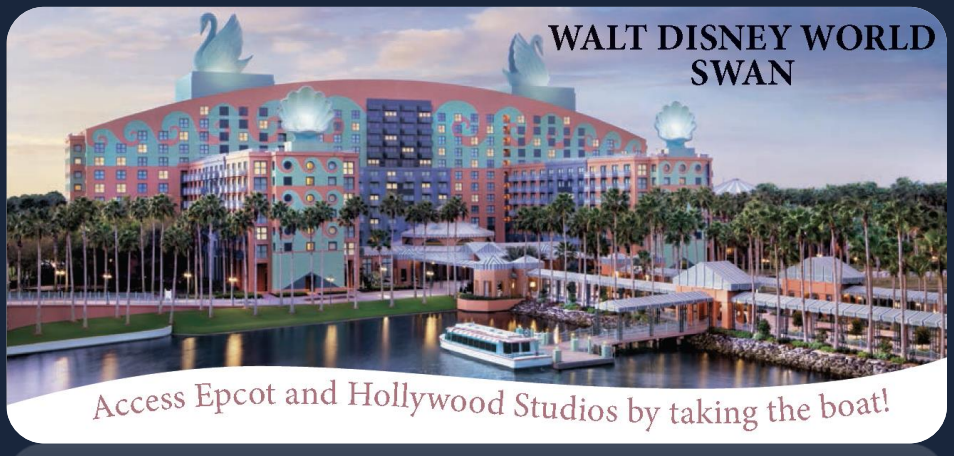
Your evaluation helps organizers build better conferences and helps speakers improve their sessions.

SQL *intersection*

Thank you!

Save the Date!

www.SQLintersection.com



2019

June 9-14

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!