SQLintersection

Session: Tuesday, 1:30 PM – 2:30 PM

Answering the Question “What Happened” with Query Store

Erin Stellato
Erin@SQLskills.com

SQL intersection

SQLskills immerse yourself in sql server

Erin Stellato

- **Consultant/Trainer/Speaker**
- **Principal Consultant, SQLskills.com**
  - Email: Erin@SQLskills.com
  - Blog: http://www.SQLskills.com/blogs/Erin
  - Twitter: @ErinStellato
- **SQL Server MVP since January 2012**
- **PASS volunteer since 2010**
- **12 years working for software vendors, helping to support customer implementations in a variety of ways**

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 – Tuesday 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

Abstract

One of the most highly anticipated new features in SQL Server 2016 is Query Store. It's been referred to as the "flight recorder" for SQL Server because it tracks query information, namely query plans and runtime statistics. If you've ever had to drop everything to troubleshoot a sudden change in performance, then you've probably already realized the value of this feature.

In this session, we'll step through how Query Store works - you'll understand what information it captures and how to access the historical data through the UI and the system views. Then, we'll look at the data collected to identify queries that don't perform well and those that have regressions in performance due to different plans. Finally, we'll see how to force a specific plan for a query, discover what happens when the optimizer cannot use that plan, and examine how plan forcing compares to using plan guides. Expect to learn how to make troubleshooting easier with this feature that's included in all editions of SQL Server 2016!

SQL intersection

Overview

- **Implementing Query Store**
- **Finding Information**
- **Forcing Plans**

What do you do all day?

- No really…
- What are your job responsibilities (DBA vs. Dev vs. other)?
- What's a typical day like?

What tools do you use?

- Performance Monitor
- DMVs
- Extended Events/Trace
- Query plans
- DBCC commands
- Xperf
- CU/SP Release notes
- Others?

- What about baseline data?

SQL intersection

Sources for Query Performance Data

Performance Monitor

- SQL Compilations/sec
- SQL Re-Compilations/sec

DMVs

- sys.dm_exec_cached_plans
- sys.dm_exec_query_stats
- sys.dm_exec_sql_text
- sys.dm_exec_query_plan

Extended Events

- sp_statement_completed
- sql_statement_completed

SSMS

- Estimated and actual plans

SQL intersection

Defining Query Store

- **Billed as a flight data recorder**
- **Captures information about query execution**
  - Query text
  - Query plan
  - Compilation time
  - Last execution time
  - Duration, CPU, logical reads, physical reads, writes
- **Available in *all* editions of SQL Server**
  - Make sure you’re running SP1 of 2016 if using Standard or Express Edition
  - https://support.microsoft.com/en-us/kb/3178297

Query Store Details

- **Enabled at the database level**
- **Data persisted in internal tables *in* the database which has Query Store enabled**
- **Cannot be enabled for master, tempdb, or model**
  - Default settings for Query Store are taken from the model database
  - http://www.sqlskills.com/blogs/erin/sql-server-query-store-default-settings/
- **Requires VIEW DATABASE STATE to view Query Store data**
- **Requires db_owner to force/unforce plans**
- **Data can be viewed in Management Studio for each database**
  - Information is also accessible via catalog views using TSQL

Demo

Query Store in Action

The data we know, but in Query Store

Manual/Third-Party Capture

SQL Compilations/sec
SQL Re-Compilations/sec
sys.dm_exec_sql_text
sys.dm_exec_query_plan
sys.dm_exec_cached_plans

sys.dm_exec_query_stats
sp_statement_completed
sql_statement_completed

Query Store

Plan Store

*Captured plans contain estimates

Runtime Stats Store

Data Captured by Query Store

Plan Store

Compile time and duration
Last execution time
Query text
Query plan

Runtime Stats Store

Execution times and counts
Duration
CPU
Logical reads
Physical reads
Writes
DOP
Memory use

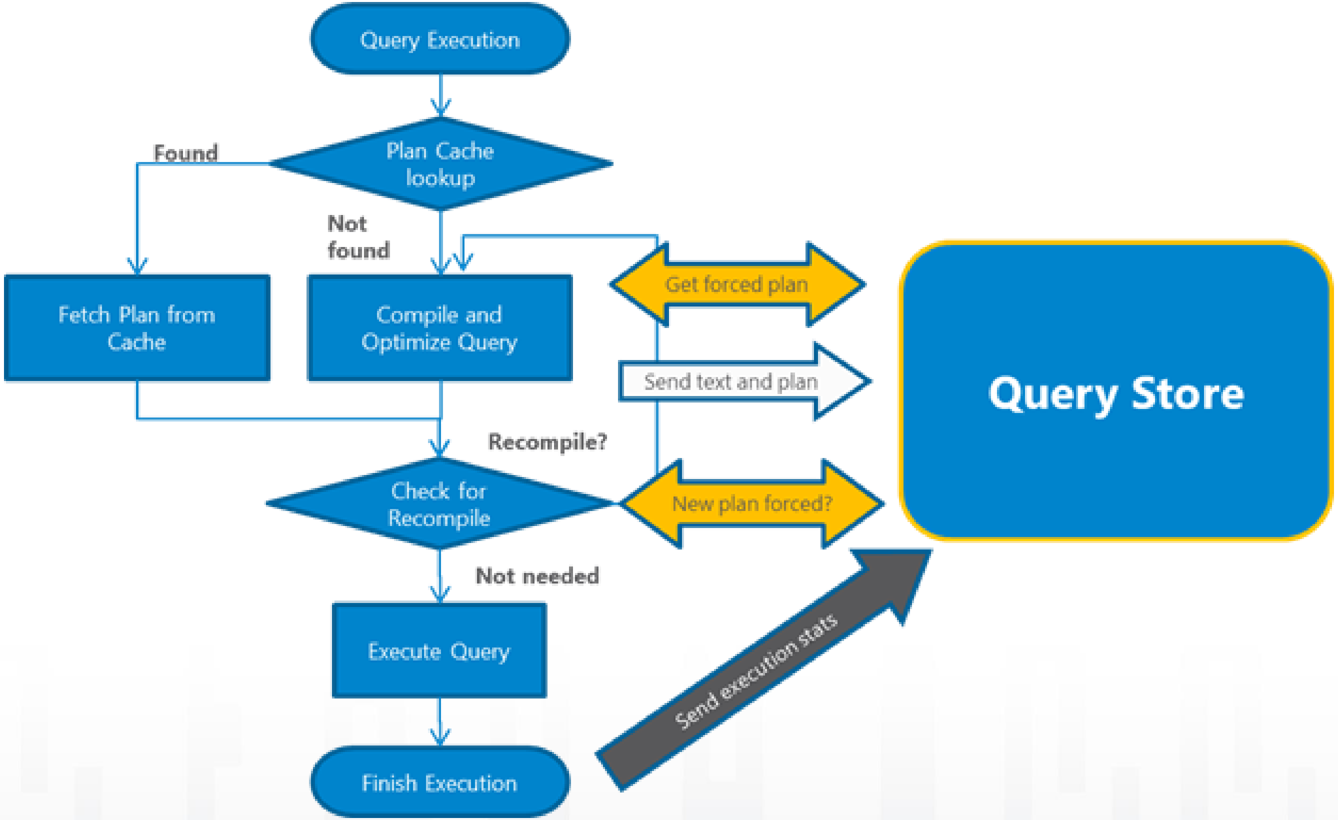

Image attribution: https://msdn.microsoft.com/en-us/library/mt631173.aspx

Query Store Settings

- OPERATION_MODE = [READ_WRITE | READ_ONLY]
- QUERY_CAPTURE_MODE = [ALL | AUTO | NONE]
- MAX_PLANS_PER_QUERY = #
- MAX_STORAGE_SIZE_MB = #
- CLEANUP_POLICY = (STALE_QUERY_THRESHOLD_DAYS = #)
- SIZE_BASED_CLEANUP_MODE = [AUTO | OFF]
- DATA_FLUSH_INTERVAL_SECONDS = #
- INTERVAL_LENGTH_MINUTES = #

Finding Information in Query Store

- Data can be viewed in Management Studio for each database
- Information is also accessible via catalog views using TSQL

Demo

Using Query Store

How do you fix a Poorly-Performing Query?

Change code and/or schema

Manually get the “best” plan in cache

Use a plan guide

Force a plan in Query Store

Forcing Plans with Query Store

- **Query Store allows you to easily find queries with multiple plans and force one plan**
- **DDL changes can cause forcing failures**
  - Monitor failures with Extended Events (query_store_plan_forcing_failed)
  - Can also check sys.query_store_plan
- **If a plan is no longer optimal, Query Store *may* continue to use it unless you remove it**
- **Adding hints changes the query text which creates a new query (and query_id) in Query Store**

Demo

Forcing Plans

Plan Guides

- **Multiple options exist for plan guide creation**
  - OBJECT, SQL, and TEMPLATE guides
- **Plan guides allow you to add hints without changing existing code, and allow you to parameterize ad-hoc queries**
  - Done via T-SQL
- **Plan guides can be tricky to implement**
- **Not always easy to verify that a plan guide is being used**

Plan Guides vs. Forcing Plans in QS

Plan Guides

You can force a plan for an ad-hoc query or stored procedure query

You can add hints with a plan guide without altering code

Plan guides can be tricky to implement

Failures can be difficult to find

Forced Plans

You can force a plan for an ad-hoc query* or stored procedure query

Adding a hint to query code *changes* the query text

Forcing a plan is very easy

Failures can be tracked with XE

SQL intersection

Points to Remember

- It may not always be obvious that a plan is forced – check the plan and Query Store to see if it is
- Query performance can be different across environments for multiple reasons – including forced plans!
- If object_id changes, a forced plan will no longer be tied to the object
- If an index name changes, a forced plan cannot be used
- Pay attention to forced plans when testing code and schema changes
- A forced plan overrides a plan guide*

SQL intersection

Uses of Query Store

- **Troubleshooting query performance**
  - Find and address regressions in query plans
  - Identify queries that consume the most resources
  - Find differences or issues with DOP, memory-grants, or execution variability
- **Reduce risks of upgrades**
  - Testing query performance **before** an upgrade (hardware, software, application)
- **Testing changes in the Cardinality Estimator**
- **Proactively analyze workload patterns**
- **Force query plans for specific queries**

SQL *intersection*

Review

- Implementing Query Store
- Finding Information
- Forcing Plans

Questions?

SQLskills
immerse yourself in sql server

Don’t forget to complete an online evaluation!

Answering the Question “What Happened?” with Query Store

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

SQL intersection

Thank you!

Save the Date

www.SQLintersection.com

Oct 30-Nov 2, 2017

We’re back in Vegas baby!

SQL
intersection