

Plan Stability through Plan Forcing

Erin Stellato

Erin@SQLskills.com





Erin Stellato

Principal Consultant, SQLskills



Erin@SQLskills.com



[@erinstellato](https://twitter.com/erinstellato)



www.sqlskills.com/blogs/erin

Trainer/Speaker

In addition to consulting, I teach content for our IE0: Accidental DBA course, and our IEPTO2: Performance Tuning and Optimization course

PASS Volunteer

I was a member of the PASS Nomination Committee this past year, have previously served on the board of my local user group (ONSSUG) and supported the Performance Virtual Chapter

Data Platform MVP

I have been fortunate to be recognized as an MVP by Microsoft since 2012



- Team of SQL Server consultants:
 - Kimberly Tripp (@KimberlyLTripp)
 - Jonathan Kehayias (@SQLPoolBoy)
 - Glenn Berry (@GlennAlanBerry)
 - Paul Randal (@PaulRandal)
 - Erin Stellato (@ErinStellato)
 - Tim Radney (@Tim Radney)
- Instructor-led training: Immersion Events and onsite
- Online training through Pluralsight (www.pluralsight.com)
- Consulting: Health checks, design, performance, upgrades, Azure
- Remote DBA: system monitoring and troubleshooting
- Conferences: PASS Summit, SQLintersection, SQLBits
- Get our newsletter: <https://www.sqlskills.com/Insider>



2019 Classes and Discounts

- In-depth, instructor-led, technical training for SQL Server
 - Held in the US in 2019 – IEPTO1/2, IE0, IEAzure, IEPowerBI, IEUpgrade
- Online, live, Immersion Events throughout the year
 - Columnstore, Transactions/locking/blocking, Query performance, Upgrade, VLTs and Partitioning
- Always announced first, and with a special discount, in our Insider newsletter
 - <https://www.sqlskills.com/Insider>
- For more information: <https://www.sqlskills.com/training/>
- Services information: <https://www.sqlskills.com/services/>



- Email paul@sqlskills.com with the subject line: User Group Pluralsight code to get a FREE (no catches, no credit card) 30-day trial of over 150+ hours of SQLskills content (and more)
- For example:
 - <https://www.pluralsight.com/courses/sqlserver-query-store-introduction>
 - 3 hours providing an introduction to Query Store
 - <https://app.pluralsight.com/library/courses/sqlserver-azure-database/>
 - 1.5 hours on Automatic Tuning in SQL Server 2017 and Azure SQL Database
 - <http://www.pluralsight.com/courses/sqlserver-optimizing-stored-procedure-performance>
 - 7 hours (!) on stored procedure performance tuning (Kimberly)

Scripts for this session, along with a PDF of the slides, will be provided for download here:

<https://www.sqlskills.com/sql-server-resources/sql-server-user-groups/>

Abstract

Intermittent performance issues due to parameter-sensitive queries are a DBA's nemesis. By the time you're notified of an issue, sometimes it's gone away...or someone else has restarted the instance or cleared the plan cache, and any data you could use is gone. Capturing the different plans for those queries can be tricky, but with Query Store it becomes very simple - the hard part is deciding what plan you like best. The Query Store feature, introduced in SQL Server 2016 and also available in Azure SQL Database, captures query performance data, and it allows you to manually force a plan for a query. In SQL Server 2017 and Azure SQL DB, the Automatic Plan Correction feature will force plans for you. Sound scary? It's not. In this session we'll explore both manual and automatic plan forcing through demos so you understand how it works, what to expect, and what to watch out for when you implement it in your environment.

Overview

- Query Store and Plan Variability
- Plan Forcing
- Automatic Plan Correction

Query Store Details

Enabled at the
database level

Data persisted in
internal tables

Cannot be enabled
for master,
model*, or
tempdb

Requires VIEW
DATABASE STATE
to view data

Requires
db_owner to
force/un-force
plans

Data is not
captured on
readable
secondaries

Data Captured by Query Store

Plan Store

- Compile time and duration
- Last execution time
- Query text
- Query plan

Runtime Stats Store

- Execution counts
- Duration
- CPU
- Logical reads
- Physical reads
- Write
- Memory use
- DOP
- Log bytes/used
- tempdb

Wait Stats Store

- Wait statistics per plan

SQL Server
2017 and
Azure SQL
Database

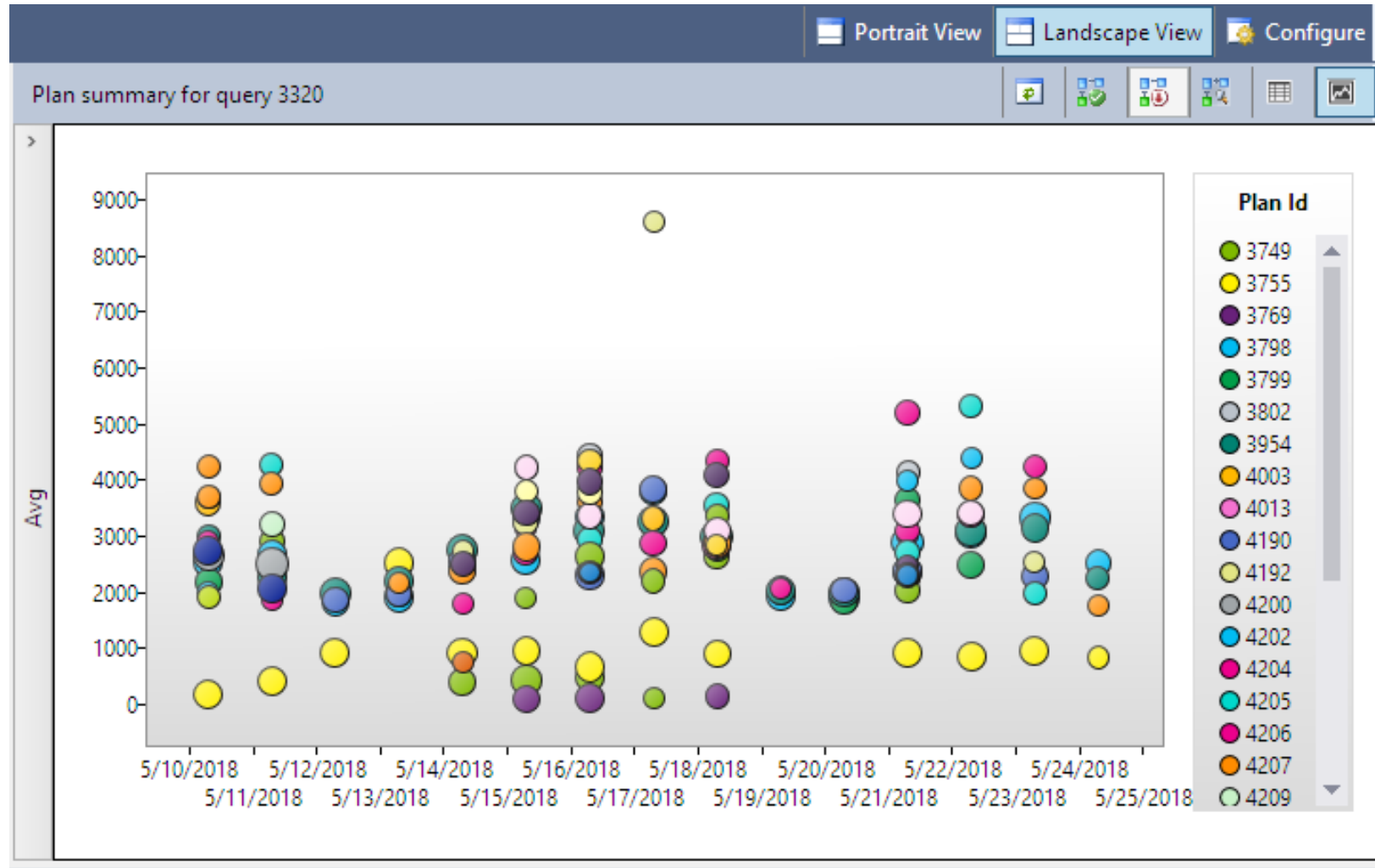
Query Store Settings₍₁₎

- Query Store is *not* enabled by default in SQL Server 2016 or SQL Server 2017
 - Query Store *is* enabled by default for new databases in Azure SQL Database (included those in Managed Instances)
- There are nine settings related to Query Store configuration, and they affect what data is collected and how it is stored
- Improper configuration of these settings can cause data to be removed from Query Store before expected, or Query Store can stop collecting data entirely
 - <https://www.sqlskills.com/blogs/erin/query-store-settings/>

Summary of Common Performance Problems

- Queries that are slow/take too long to execute
- Queries that consume significant resources
- Queries that execute with high frequency
- Queries with high variability in performance

Plan Variability



How do you stabilize query performance?

Change code
and/or schema

Add RECOMPILE

Manually get the
“best” plan in
cache

Update statistics

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
- Not schema-bound
- Monitor failures with Extended Events
- If a plan is no longer optimal, Query Store will continue to use it, unless you un-force it or forcing fails

A bad plan is not the one which failed, but the one which succeeded at the greatest cost.

-Anonymous DBA

Demo: Creating Plan Stability

Typical Reasons Forcing Can Fail

- Drop an index (NO_INDEX)
- Change an index name (NO_INDEX)
- Remove columns from an index (NO_PLAN)
- Change the object_id due to DROP/CREATE rather than ALTER

Points to Remember with Plan Forcing₍₁₎

- It may not always be obvious that a plan is forced – check the actual plan and Query Store to determine
- Query performance can be different across environments for multiple reasons – including forced plans!
- Pay attention to forced plans when testing code and schema changes
 - Changing index names
 - Changing object names

Points to Remember with Plan Forcing (2)

- You cannot force a plan on read-only secondary by forcing the plan on the primary
- You cannot force a plan for a query if it hasn't been generated by that query
- Turning off Query Store negates the ability to use forced plans
- Forced plans will not be aged out of Query Store

Automatic Tuning

- Reduces manual intervention required from data professionals
- Monitors workload performance, makes changes, continues to monitor and make additional changes if needed (e.g. revert)
 - Query Store must be enabled
- Two components:
 - Automatic plan correction
 - Automatic index management

Automatic Tuning

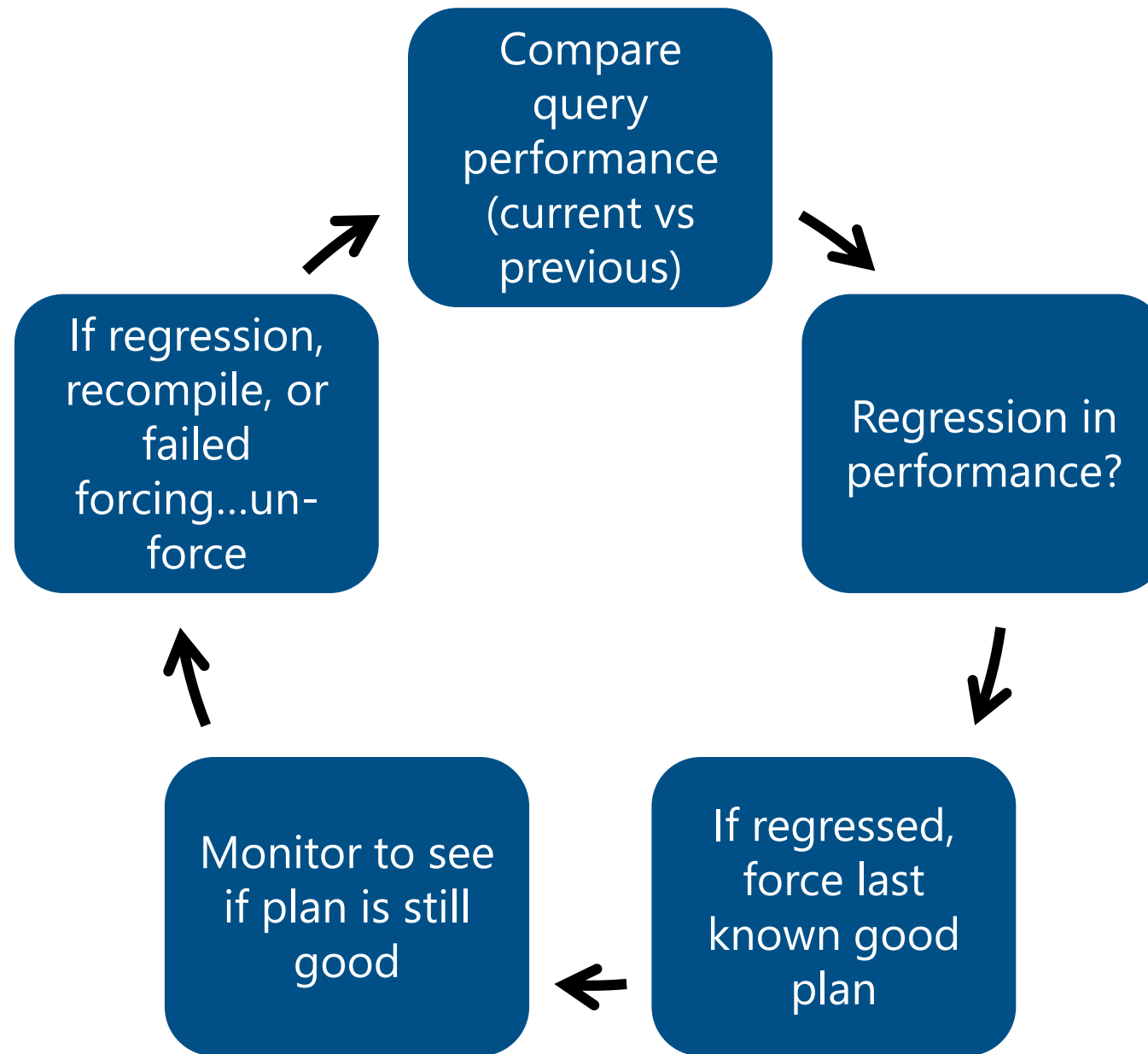
- SQL Server 2017
 - Automatic Plan Correction
 - Disabled by default
- Azure SQL Database
 - Automatic Plan Correction
 - Automatic Index Management
 - Starting January 15, 2018 a change was rolled out to start to enable Automatic Tuning for all existing Azure SQL DBs
 - Users notified in advance
 - Eventually will be enabled by default for all new databases

Automatic Plan Correction

- Available in SQL Server 2017 and Azure SQL Database
- Only available in Enterprise Edition
- Disabled by default

```
ALTER DATABASE <database_name_here>  
SET AUTOMATIC_TUNING ( FORCE_LAST_GOOD_PLAN = ON );
```

- Can use the information captured to make corrections manually
 - Stored in sys.dm_db_tuning_recommendations
 - Recommendations will not appear in Standard Edition



Demo: Automatic Plan Correction

Can I trust it?

- It is not perfect, but it has been developed with **operational telemetry** from Azure SQL Database implementations
- It may not “catch” every regression you expect, and it may make a not-so-great decision
- Its ability to recover from any “bad decision” is highly reliable as there is continual validation of forced plans and automatic back-off logic built-in

Notes About Forcing

- If you manually force a plan – either because you determined it was needed or based on a recommendation from `sys.dm_db_tuning_recommendations` – it will never be automatically un-forced
- Only plans that are forced with the Automatic Plan Correction feature will be automatically un-forced

Monitoring Automatic Plan Correction

- Information in `sys.dm_db_tuning_recommendations` is lost on instance restart for plans not currently forced
 - Snapshot to a table if you want to retain information
- Create an Extended Events session that captures automatic tuning events, writes to an `event_file` target, and starts when the instance starts (always running)
 - `automatic_tuning_error`
 - `automatic_tuning_plan_regression_detection_check_completed`
 - `automatic_tuning_plan_regression_verification_check_completed`
 - `automatic_tuning_recommendation_expired`

Creating Plan Stability

- Manually:
 - Identify plans with multiple queries
 - Determine the “best” plan
 - *Temporarily*, force the “best” plan
 - Work with developers to change code and/or schema
- Automatically:
 - Automatic Plan Correction feature
 - Capture recommendations and *temporarily* force the “best” plan
 - Work with developers to change code and/or schema

Review

- Query Store and Plan Variability
- Plan Forcing
- Automatic Plan Correction

Resources

- SQL Server: Introduction to Query Store
 - <https://www.pluralsight.com/courses/sqlserver-query-store-introduction>
- Automatic Tuning in SQL Server 2017 and Azure SQL Database
 - <https://app.pluralsight.com/library/courses/sqlserver-azure-database/table-of-contents>
- Blog:
 - <https://www.sqlskills.com/blogs/erin/category/query-store/>
- Query Store Requests
 - <https://www.sqlskills.com/blogs/erin/query-store-requests/>

Thank you!

