SQLintersection

Tuesday October 31, 2017

3:45pm - 5:00pm

Advanced Query Store in SQL Server 2016/2017

David Pless

David.Pless@Microsoft.com

SQL intersection

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

SQL intersection

SQLskills immerse yourself in sql server & SentryOne™

Overview

- **Introduction**
  - Plan Guides
  - Query Store – Capabilities and Use Cases
  - Query Store in SQL Server 2017
  - Automatic Tuning and Plan Correction
  - Adaptive Query Processor

SQL intersection

"A Bad Plan is not the one which failed, but the one which succeeded at the Greatest Cost."

~Anonymous DBA

Plan Guides

SQL intersection

Why Use Plan Guides?

- Useful for tuning queries generated by 3rd party applications
- Plan guides work by keeping a list of queries on the server, along with the Hints you want to apply
- You need to provide SQL Server with the query you want to optimize and a query hint using the OPTION clause
- When the query is optimized, SQL Server will apply the hint requested in the plan guide definition

SQL intersection

Plan Guides Stored Procedures

- Use the sp_create_plan_guide stored procedure to create a plan guide
- Use sp_control_plan_guide to drop enable or disable plan guides
- You can see which plan guides are defined in your database using the sys.plan_guides catalog view
- **Note:** When Using Plan Guides, you must match Query Text and Parameter Names exactly

SQL intersection

Common Query Hints Used in Plan Guides

- OPTIMIZE FOR (Value, Unknown)
- RECOMPILE
- MAXDOP #
- FORCE ORDER
- USE PLAN
- NULL

SQL intersection

Demo

Plan Guides

Creating a Plan Guide

SQL intersection

Query Store

SQL intersection

Query and Query Plan Fingerprints

- Query Fingerprint
  - query_hash
  - Explicitly identifies a specific query in the cache.
  - *sys.dm_exec_requests*
  - *sys.dm_exec_query_stats*
- SQL Handle
  - sql_handle
  - Token for the SQL text that relates to a batch.
  - *sys.dm_exec_sql_text*
  - *sys.dm_exec_query_stats*
  - *sys.dm_exec_query_memory_grants*
- Query Plan Fingerprints
  - query_plan_hash
  - Useful to determine queries that share the same execution plan.
  - Can be used to determine if the query plan has changed.
  - *sys.dm_exec_requests*
  - *sys.dm_exec_query_stats*
- Plan Handle
  - plan_handle
  - Token for a cached execution plan.
  - *sys.dm_exec_query_plan*
  - *sys.dm_exec_cached_plans*

The pain of joining DMVs and xEvents

- ***query_hash* and *query_plan_hash* actions in xEvents**
  - Not the same data types as respective columns in DMVs *sys.dm_exec_requests* and *sys.dm_exec_query_stats*
  - That makes it difficult to correlate the information
- **In SQL Server 2016 RTM and 2014 SP2**
  - New actions *query_hash_signed* and *query_plan_hash_signed* allow you to join these DMVs with xEvents such as *rpc_completed* and *sql_batch_completed*

	Name	Description
☐	query_hash	Collect query hash. Use this to identify queries with similar logic. You can use the query hash to determine the aggregate resource usage for queries that differ only by literal
☑	query_hash_signed	Collect query hash. Use this to identify queries with similar logic. You can use the query hash to determine the aggregate resource usage for queries that differ only by literal
☐	query_plan_hash	Collect query plan hash. Use this to identify similar query execution plans. You can use query plan hash to find the cumulative cost of queries with similar execution plans
☑	query_plan_hash_signed	Collect query plan hash. Use this to identify similar query execution plans. You can use query plan hash to find the cumulative cost of queries with similar execution plans

When performance is not good…

- Database is not working

Website / App is down

- Impossible to predict / root cause

Temporary Perf. issues

- Regression caused by upgrade

System Upgrade

Plan choice change can cause these problems

What are you doing today?

- **Most solutions are reactive in nature**
  - Flush the bad plan from the cache with sp_recompile
  - Flush the entire plan cache with DBCC FREEPROCCACHE
  - Force the plan to recompile every time
  - Restart OS / SQL Server (It works for some reason?)
- **Proactive solutions are challenging**
  - Often takes a long time to even detect there is a plan problem
  - Only the latest plan is stored in the cache
  - Need to catch both the good and the bad plan in order to troubleshoot
  - Information is stored in memory only
    - Reboot or memory pressure causes data to be lost
    - No history or timing available – stats are aggregated for what is currently in cache

SQL intersection

Why Plan Changes Happen..

- SQL Query Optimizer considers many plans
- When a plan is chosen, it is cached and reused
- As your data changes, it might select a different plan as optimal.
- Volume and Data Distribution can affect plan choices
- Sometimes a rare plan choice will be cached (The Parameter Sensitive Plan Problem)

SQL intersection

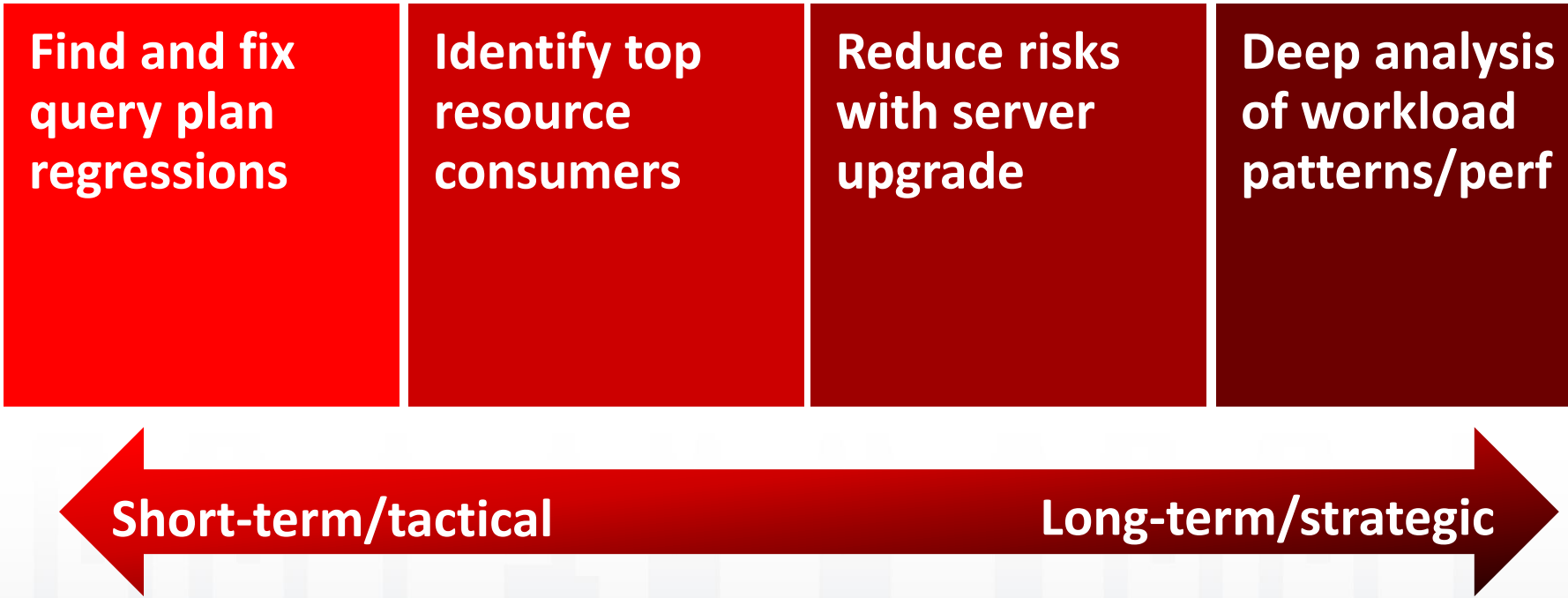
Tackling the Problem – What Could We Do?

1. Store the history of plans for each query
2. Baseline the performance of each plan over time
3. Identify queries that have “gotten slower recently”
4. Find a way to force plans quickly and easily
5. Make sure this works across server restarts, upgrades, and query recompiles

This is what the Query Store does for you!

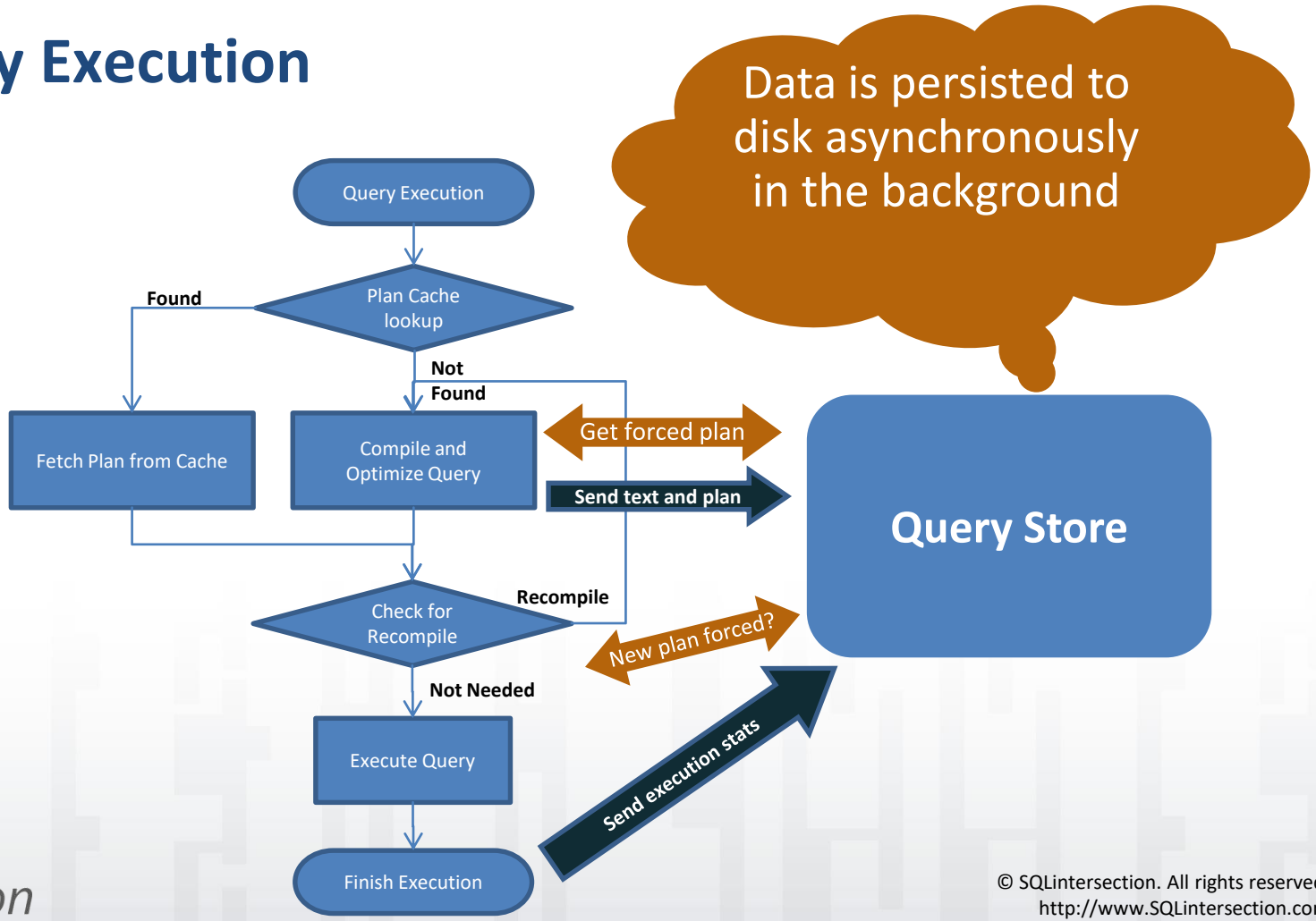
SQL intersection

Key Usage Scenarios



SQL intersection

SQL Query Execution



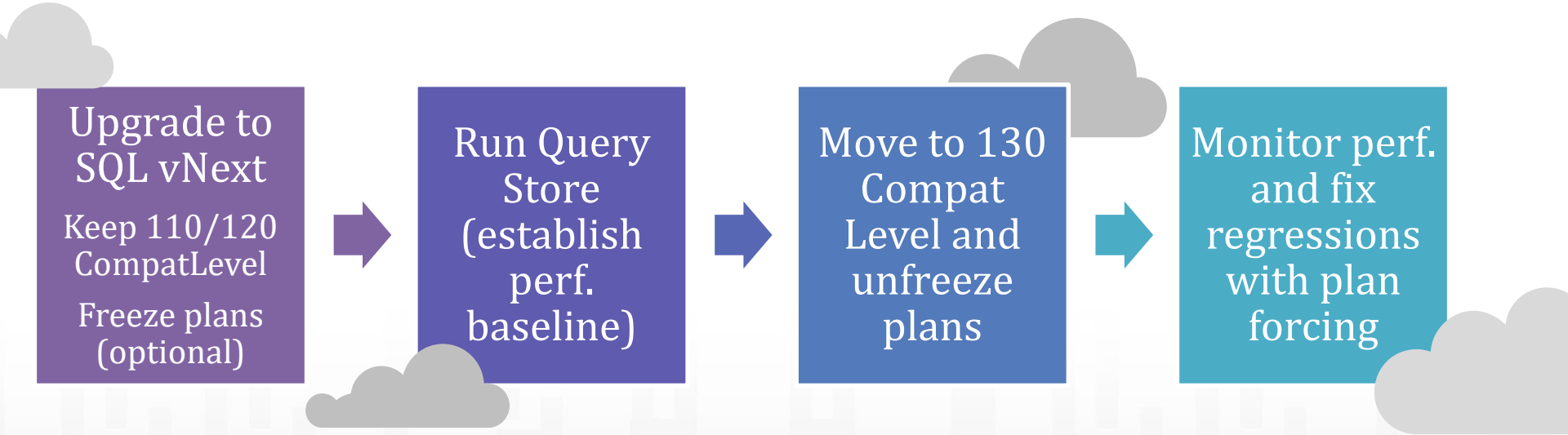
SQL intersection

What Gets Captured?

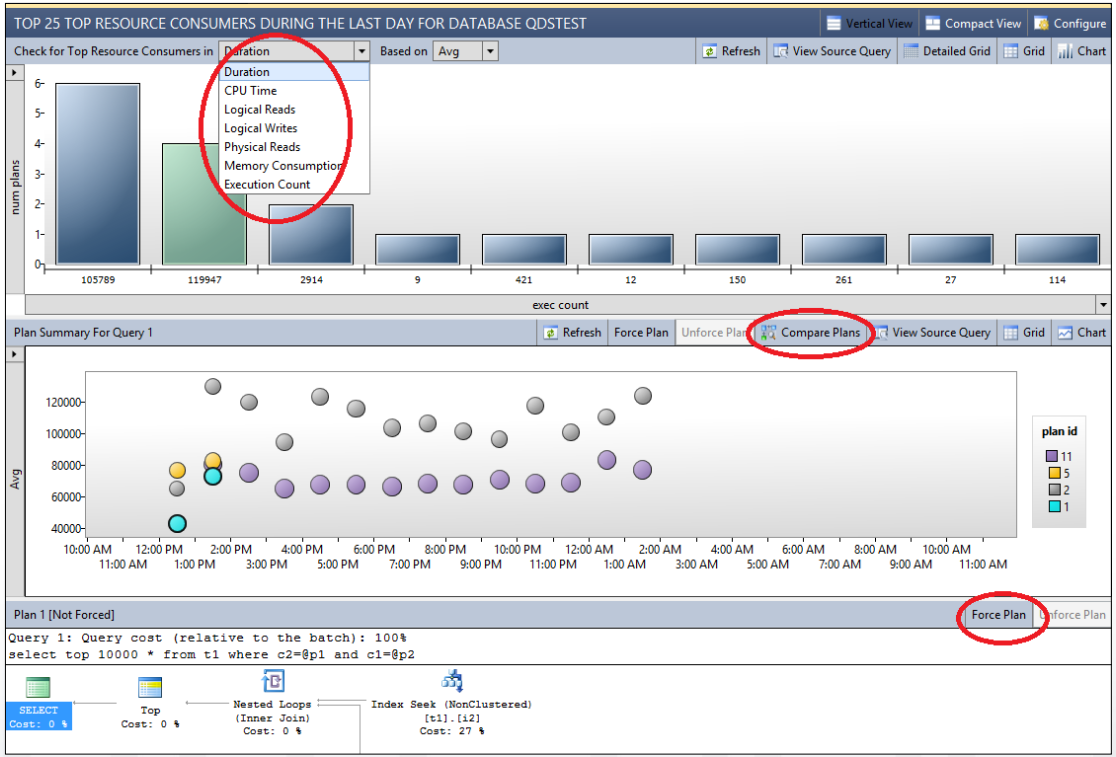
- **Query Texts and Query Plans**
- **Runtime Statistics (per unit of time, default 1 hour)**
  - Count of executions of each captured plan
  - For each metric: average, last, min, max, stddev
  - Metrics: duration, cpu_time, logical_io_reads, logical_io_writes, physical_io_reads, clr_time, DOP, query_max_used_memory, rowcount
  - Data is recorded when a query execution *ends*
- **Query Store is configurable**
  - Settings such as MAX_SIZE_MB, QUERY_CAPTURE_MODE, CLEANUP_POLICY allow you to decide how much data you want to store for how long
  - Can be configured either via the SSMS GUI or T-SQL scripts

SQL intersection

Keeping stability while upgrading to SQL Sever 2016 / 2017



Monitoring Performance with Query Store



- The Query Store feature provides DBAs with insight on query plan choice and performance

Demo

Using Query Store in SQL Server 2016 / 2017

SQL intersection

Working with Query Store DMVs

```
/* (6) Performance analysis using Query Store views*/
SELECT q.query_id, qt.query_text_id, qt.query_sql_text,
SUM(rs.count_executions) AS total_execution_count
FROM
sys.query_store_query_text qt JOIN
sys.query_store_query q ON qt.query_text_id =
q.query_text_id JOIN
sys.query_store_plan p ON q.query_id = p.query_id JOIN
sys.query_store_runtime_stats rs ON p.plan_id = rs.plan_id
GROUP BY q.query_id, qt.query_text_id, qt.query_sql_text
ORDER BY total_execution_count DESC

/* (7) Force plan for a given query */
exec sp_query_store_force_plan
12 /*@query_id*/, 14 /*@plan_id*/


);

/* (4) Clear all Query Store data */
ALTER DATABASE MyDB SET QUERY_STORE CLEAR;

/* (5) Turn OFF Query Store */
ALTER DATABASE MyDB SET QUERY_STORE = OFF;
```

- DB-level feature exposed through T-SQL extensions
- ALTER DATABASE
- Catalog views (settings, compile & runtime stats)
- Stored Procs (plan forcing, query/plan/stats cleanup)

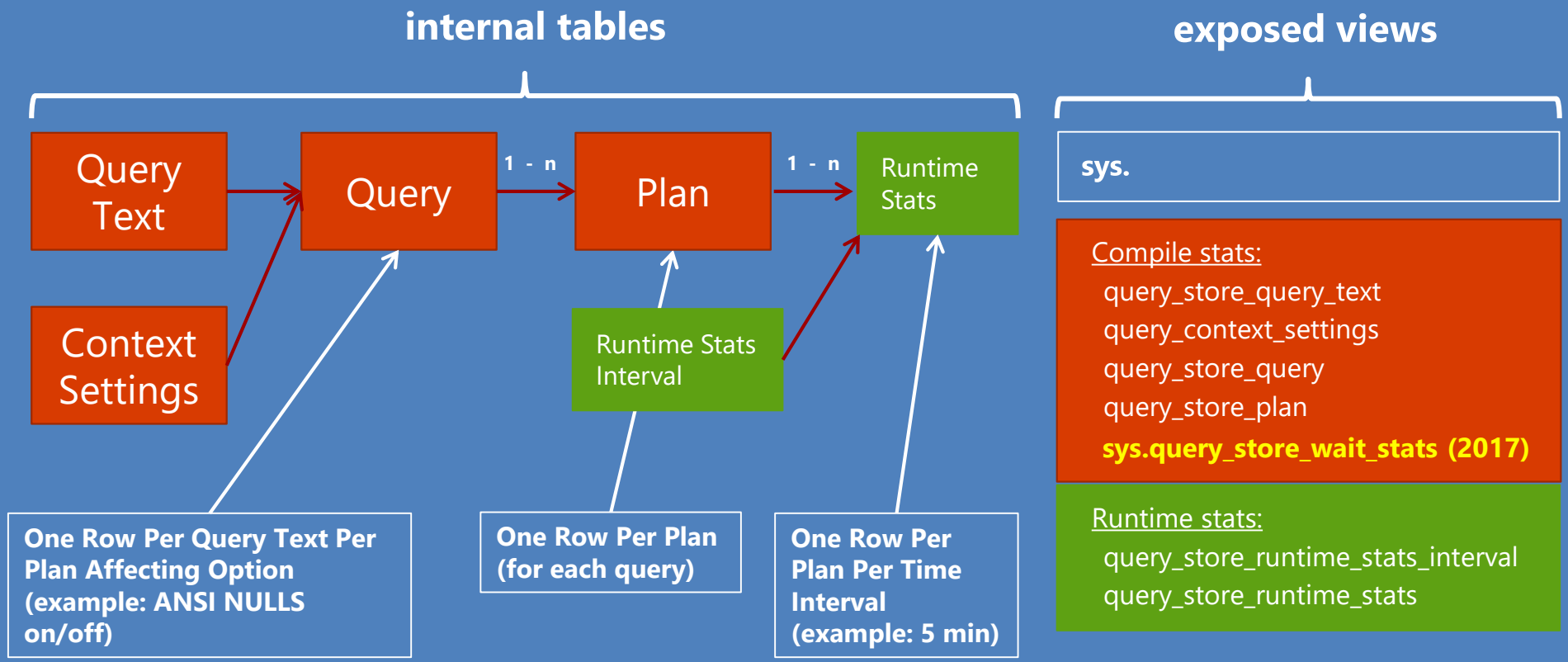
SQL intersection

Troubleshooting Query Store

- **Plan forcing does not always work**
  - Example: If you drop an index, you can't force a plan that uses it.
- **Query Store will revert to not forcing if it fails**
  - This keeps the application working if the hint breaks
- **You can see which plans are failing to force by looking at the Plan Table:**

```
SELECT * FROM sys.query_store_plan
WHERE is_forced_plan = 1 AND
force_failure_count > 0
```

Query Store Schema Explained

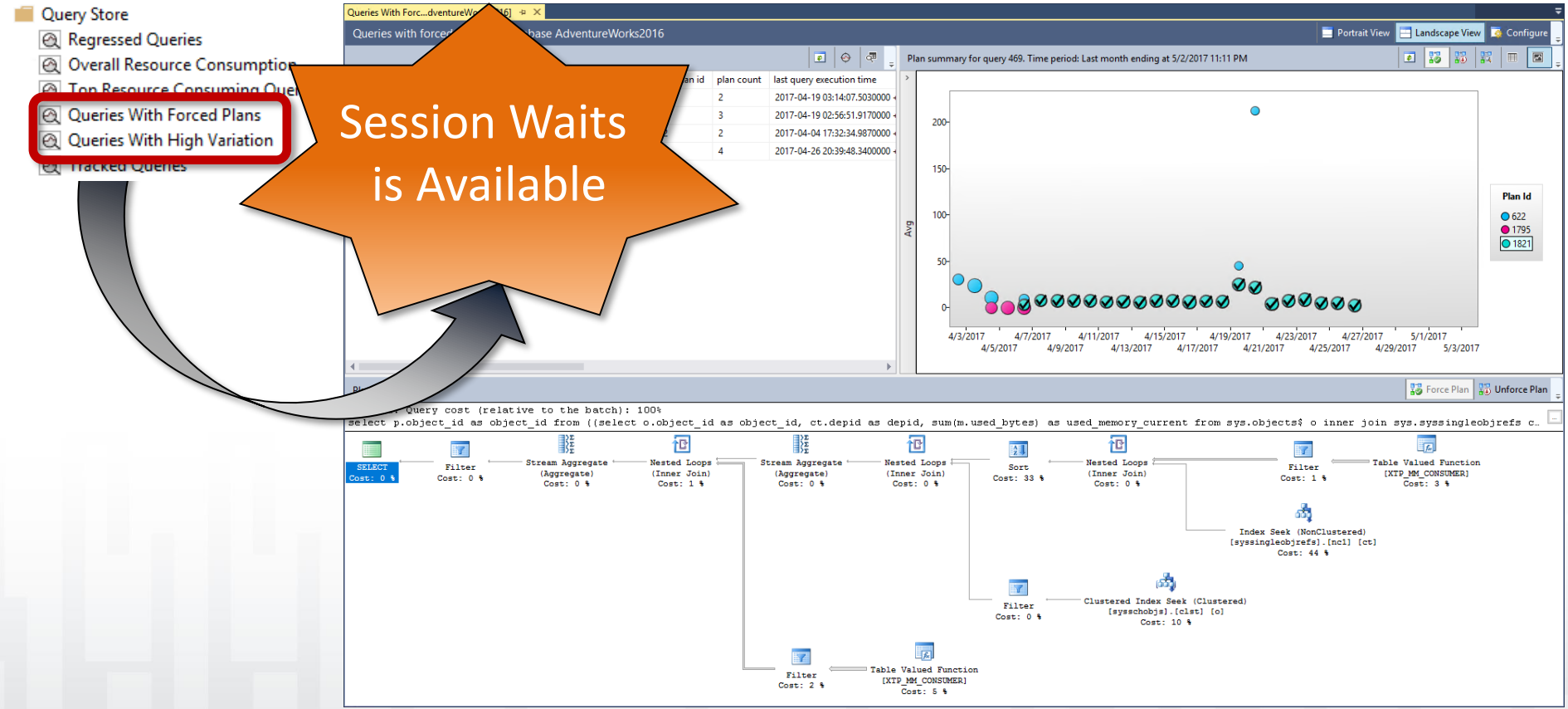


Demo

Using Query Store DMVs in SQL Server 2016 / 2017

SQL intersection

Queries with Forced Plans in SQL Server 2017



Demo

Query Store in SQL Server 2017

- Query Store Waits
- Queries with Forced Plans
- Queries with High Variations

SQL intersection

SQL Server 2017 – Query Store Improvements

- **New Query Store Reports**
- **Automatic Tuning Feature Support**

```
ALTER DATABASE AdventureWorks2017
    SET AUTOMATIC_TUNING ( FORCE_LAST_GOOD_PLAN = ON );
```

- **DBCC CLONEDATABASE flushes statistics while cloning to avoid missing query store runtime statistics**
- **New DMVs**
  - sys.query_store_wait_stats
  - sys.dm_db_tuning_recommendations
  - sys.database_automatic_tuning_mode
  - sys.database_automatic_tuning_options

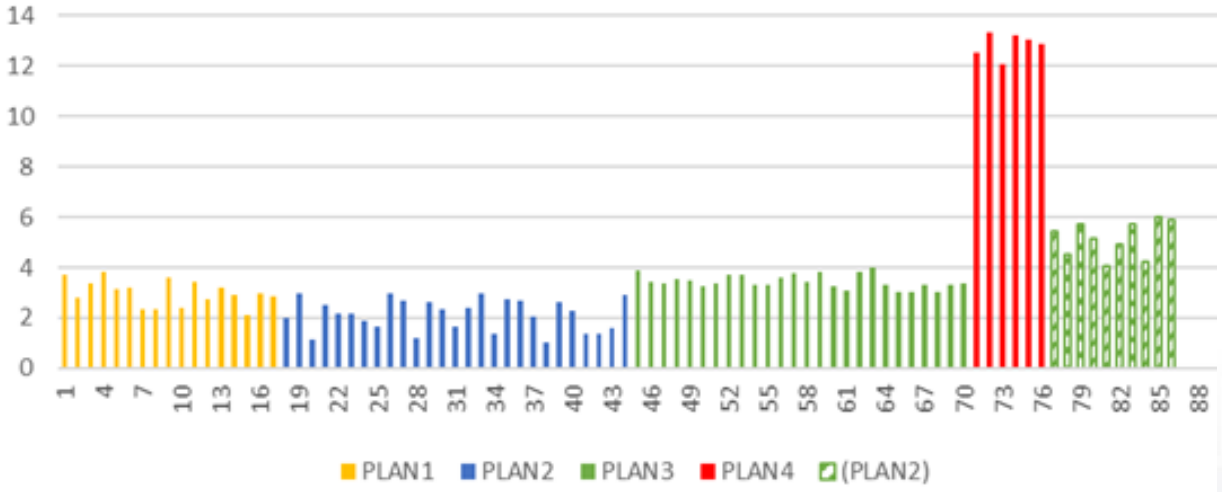
SQL Server 2017 Automatic Tuning

Detect with **dm_db_tuning_recommendations** and force manually

Turn on Auto and system corrects

Reverts back to "Last Known Good"

```
ALTER DATABASE CURRENT
SET AUTOMATIC_TUNING (FORCE_LAST_GOOD_PLAN = ON);
```



Perfect to help with parameter sniffing

Demo

Automatic Tuning in SQL Server 2017

SQL intersection

Adaptive Query Processing

SQL intersection

Query Processing and Cardinality Estimation

- Query performance is dependent on query plan quality
- Plan quality means that we are making good decisions around the order of operations and the physical algorithm selection
- Plan quality is heavily dependent on properly estimating the number of rows flowing through the query
  – The process is known as cardinality estimation (CE)
- CE is an involved process that uses a combination of statistical techniques and assumptions to provide an estimated number of rows to the optimization process

SQL intersection

Risks of Misestimation

Slow Query Response Time Due to Bad Plans

Excessive Resource Utilization (CPU, Memory, IO)

Reduced Throughput and Concurrency

T-SQL Refactoring for Off-Model Statements

Adaptive Query Processing (SQL Server 2017)

Interleaved Execution

- Materialize estimates for multi-statement table valued functions (MSTVFs)
- Downstream operations will benefit from the corrected MSTVF cardinality estimate

Batch-mode Memory Grant Feedback

- Adjust memory grants based on execution feedback
- Remove spills and improve concurrency for repeating queries

Batch-mode Adaptive Joins

- Defer the choice of hash join or nested loop until after the first join input has been scanned
- Uses nested loop for small inputs, hash joins for large inputs

Demo

Adaptive Query Processing

- Interleaved Execution
- Batch-Mode Memory Grant Feedback
- Batch-Mode Adaptive Join

SQL intersection

Review

- **SQL Server 2016 / 2017 Performance Features:**
  - Plan Guides
  - Query Store – Capabilities and Use Cases
  - Query Store in SQL Server 2017
  - Automatic Tuning and Plan Correction
  - Adaptive Query Processor
  - Live Query Statistics

SQL intersection

References

- **Query Store in SQL Server 2016**
  - https://channel9.msdn.com/Shows/Data-Exposed/Query-Store-in-SQL-Server-2016
- **Exploring Query Store**
  - https://vlabs.holsystems.com/vlabs/technet?eng=VLabs&auth=none&src=vlabs&altadd=true&labid=14030
- **Live Query Statistics**
  - https://msdn.microsoft.com/en-us/library/dn831878.aspx
- **Exploring Live Query Statistics**
  - https://vlabs.holsystems.com/vlabs/technet?eng=VLabs&auth=none&src=vlabs&altadd=true&labid=14027

SQL intersection

Questions?

Don’t forget to complete an online evaluation!

Advanced Query Store in SQL Server 2016/2017

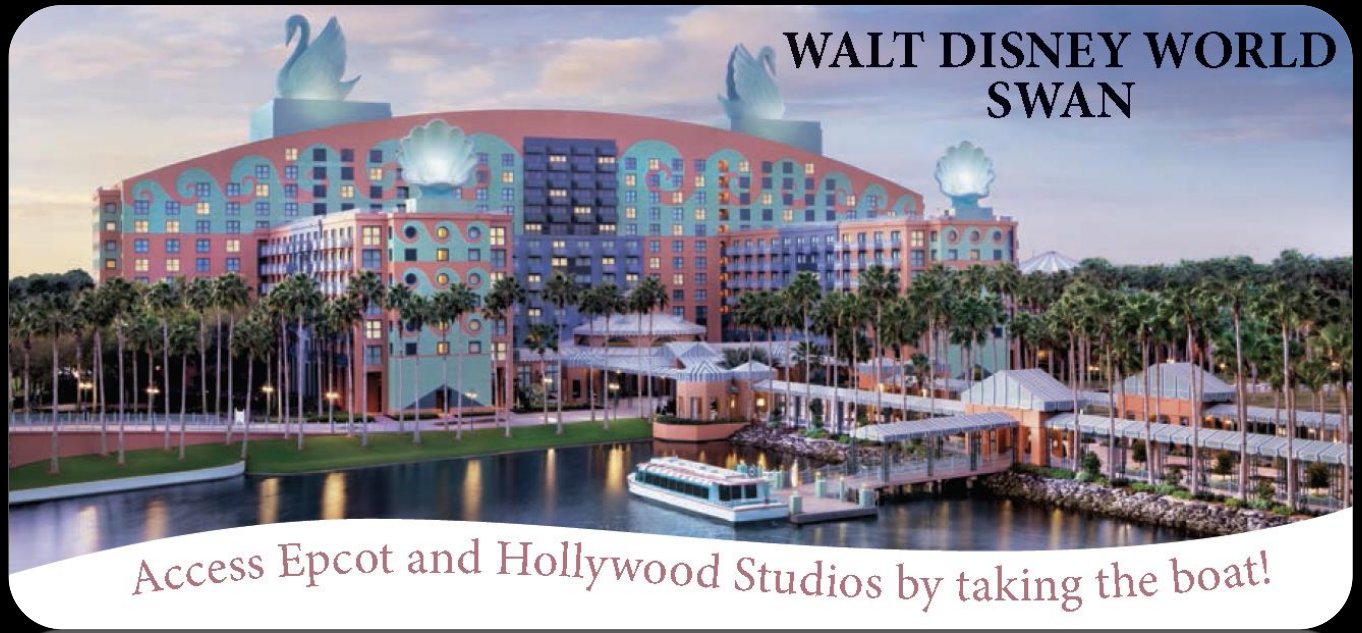
Your evaluation helps organizers build better conferences and helps speakers improve their sessions.

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!