



Query Store

What is it all about?

Andrew J. Kelly, SolidQ



Andrew J. Kelly

Mentor, SolidQ

Andrew J. Kelly is a Mentor with SolidQ and a Data Platform MVP with over 20 years' experience with relational databases and application development specializing in Performance, Scalability and Maintainability of large scale SQL Servers. He is a regular speaker each year and a contributing editor and writer for SQL Server Pro.

http://sqlblog.com/blogs/andrew_kelly/default.aspx

 @GunneyK

Agenda

- Query Store Overview
 - Differences between the QDS and the Query & Procedure Stats
- Why do we need it?
- How does it work?
 - How do you enable it?
 - How do you configure it?
 - Where is the data stored
- How do you use it?
- Demos
- Questions & Answers

Plan Cache Overview

Memory resident collection of query & procedure plans for the entire instance

- `sys.dm_exec_cached_plans`
- `sys.dm_exec_plan_attributes(Plan_Handle)`
- `sys.dm_exec_sql_text(Plan_Handle)`
- `sys.dm_exec_query_plan(Plan_Handle)`

Query Store (QDS) Overview

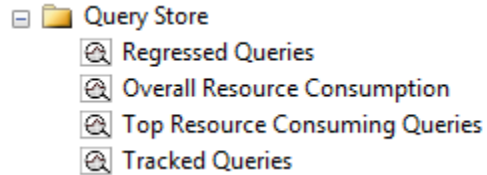
Persisted store of query plans and their associated performance metrics

- These plans are a persisted copy of the plan that is/was in the proc cache
- Per Individual database and uniquely configurable as such
- Retains a history of plans and their associated metrics
- Tracks both run time and compile time metrics
- Allows for forcing of a particular plan for a given query
- Graphical interface in SSMS to
 - Compare plans
 - Force / Un-Force plans
 - Report on QDS related queries
 - Of course everything can be accessed programmatically as well

Difference Between QDS and (Query & Procedure Stats (Q&P))

- Q&P Stats are memory resident only
 - Stats can be flushed out for various reasons
- Q&P Stats don't track past plans
 - If dropped or recompiled the Q&P stats are lost
- QDS tracks all
 - If recompiled the new metrics are aggregated with the old ones
 - If dropped and recreated they are get a new query ID
- They track or report slightly different but over lapping metrics
 - IE: Q&P reports averages where as QDS reports totals (both can be calculated though)
 - Q&P tracks thread usage where as QDS does not
- Q&P is instance wide, QDS is database specific

Built in Reports (SSMS)



Regressed Queries - Queries that received a worse Execution Plan than it had previously

Overall Resource Consumption - Summary of the aggregated query runtime statistics for a given time interval

Top Resource Consuming Queries - Most expensive resource consuming queries for the given options over the given time interval

Tracked Queries - Shows the runtime statistics for a specific query along with any Execution Plans

- <https://www.simple-talk.com/sql/database-administration/the-sql-server-2016-query-store-built-in-reporting/>

Performance Impact

- Microsoft predicts a 3% to 5% impact on average
- As always it depends on several factors
 - How many DBs with the Query Store enabled
 - Individual DB Query Store settings
 - Transaction / Compile / Recompile rate
 - ...
- Several additions to help monitor the impact
 - 4 new Query Store specific perfmon counters
 - 19 new Query Store specific Wait Types as of SQL2016 CU1
 - There are a ton of Extended Events as well

Query Store Perfmon Counters

Available counters

Select counters from computer:

<Local computer> Browse...

- MSSQL\$SQL2016DEV:Plan Cache
- MSSQL\$SQL2016DEV:Query Store
- Query Store CPU usage
- Query Store logical reads
- Query Store logical writes
- Query Store physical reads
- MSSQL\$SQL2016DEV:Replication Agents
- MSSQL\$SQL2016DEV:Replication Dist.

Instances of selected object:

- Total
- <All instances>
- AdventureWorks
- WideWorldImporters

Search

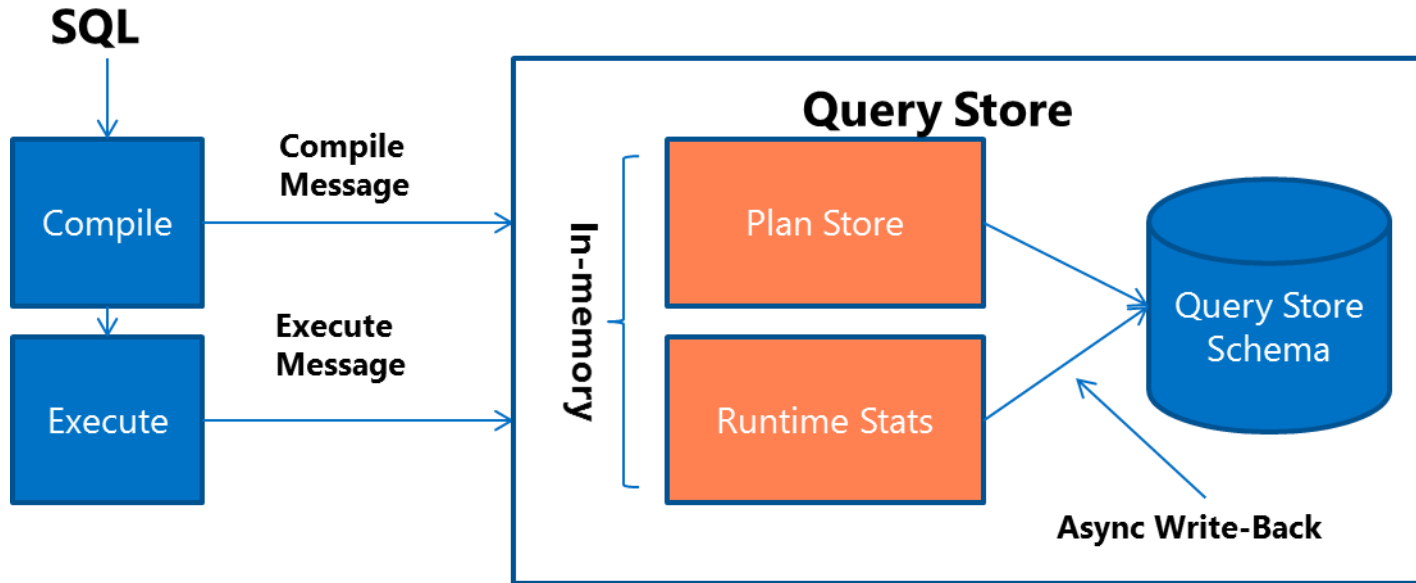
Add >>

Query Store Wait Types (SQL2016 SP1)

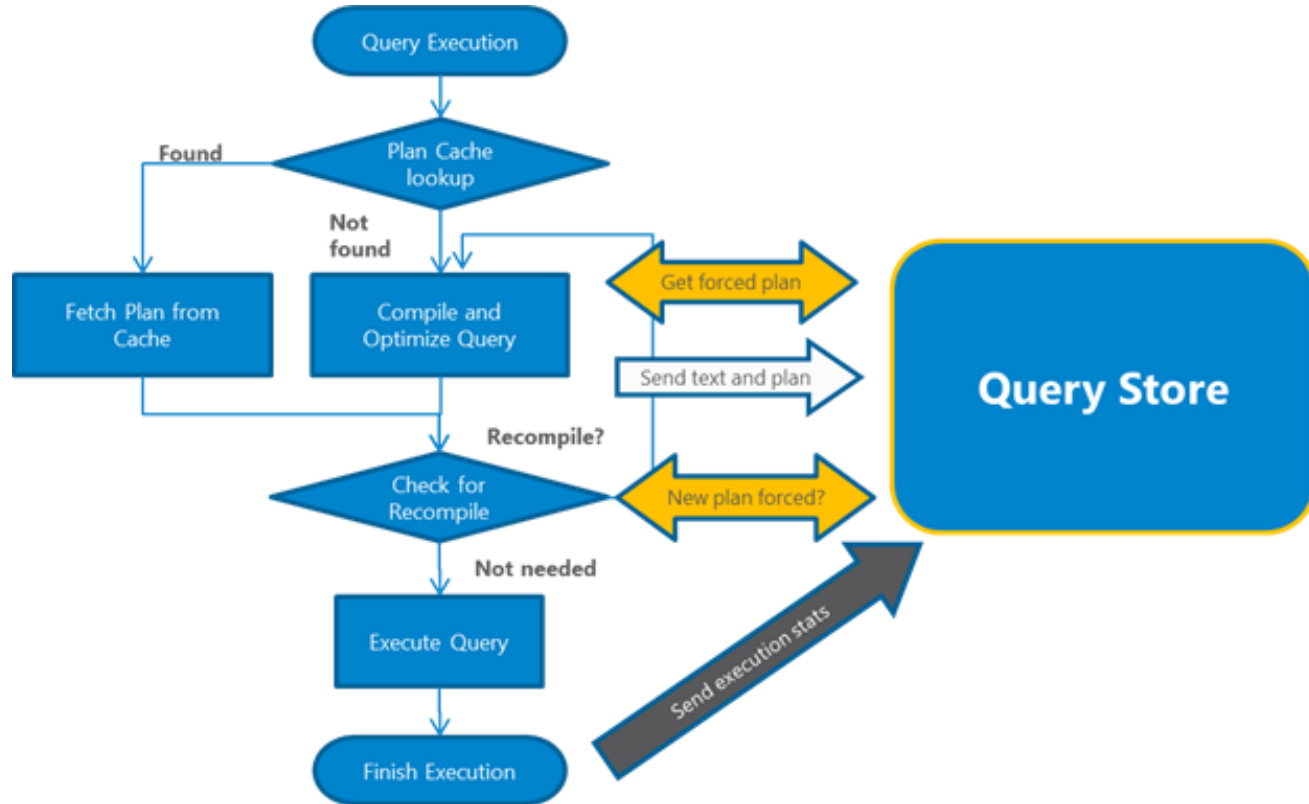
Wait Type
QDS_DYN_VECTOR
QDS_STMT
QDS_CTXS
QDS_BCKG_TASK
QDS_DB_DISK
QDS_STMT_DISK
QDS_ASYNC_PERSIST_TASK
QDS_LOADDB
QDS_ASYNC_PERSIST_TASK_START
QDS_ASYNC_CHECK_CONSISTENCY_TASK
QDS_TASK_START
QDS_PERSIST_TASK_MAIN_LOOP_SLEEP
QDS_TASK_SHUTDOWN
QDS_SHUTDOWN_QUEUE
QDS_EXCLUSIVE_ACCESS
QDS_CLEANUP_STALE_QUERIES_TASK_MAIN_LOOP_SLEEP
QDS_ASYNC_QUEUE
QDS_BLOOM_FILTER
QDS_QDS_CAPTURE_INIT

```
SELECT * FROM sys.dm_os_wait_stats  
WHERE wait_type LIKE 'qds%'
```

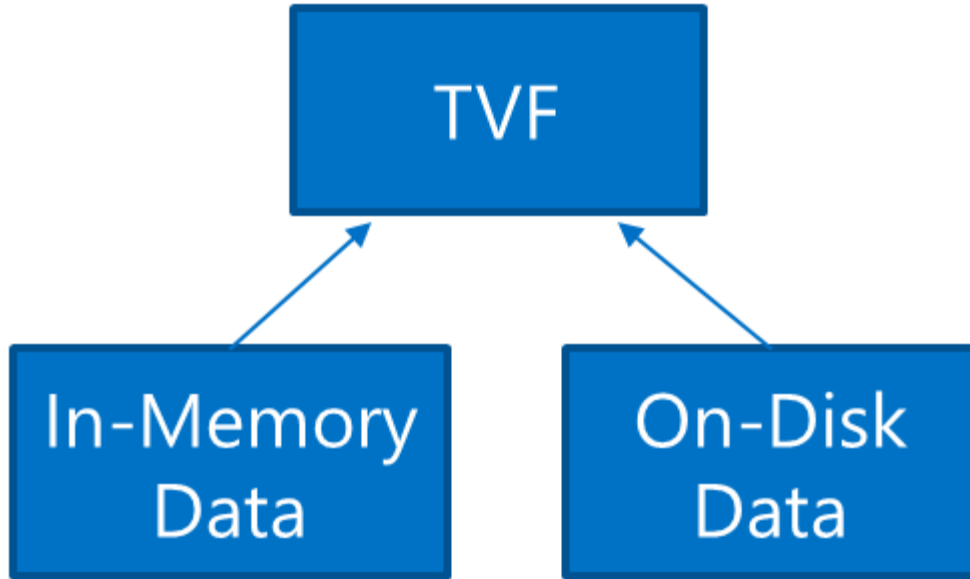
High Level Overview



Plan Caching and Execution



Query Store Data Retrieval



So why do we need this?

- Most of us should be able to gain at least some benefit from QDS
 - Many will gain a LOT of benefit
 - However not every database may need this
- QDS makes it much easier to track and report on query performance
 - Especially at the DB level
- QDS allows you to easily see why performance may have dropped for a given query / procedure due to plan changes
 - Did you ever have an issue with performance after an upgrade?
- Do you suffer from Parameter Sniffing?
- I predict 3rd party tool vendors will do great things with this feature

Query Store Stored Procedures

- `sp_query_store_flush_db`
- `sp_query_store_reset_exec_stats`
- `sp_query_store_force_plan`
- `sp_query_store_unforce_plan`
- `sp_query_store_remove_plan`
- `sp_query_store_remove_query`

Query Store Catalog View's

- `sys.database_query_store_options`
- `sys.query_store_query`
- `sys.query_store_plan`
- `sys.query_store_query_text`
- `sys.query_store_runtime_stats`
- `sys.query_store_runtime_stats_interval`

Database Query Store Options

- **Desired & Actual State** – The desired operation mode of the Query Store
 - OFF / READ_ONLY / READ_WRITE (ERROR for Actual)
- **ReadOnly Reason** – bitmap that indicates why the QS is in ReadOnly mode
- **Current Storage Size** – MB's of the current QS on disk
- **Max Storage Size** – Maximum MB's of the QS on disk (Dflt = 100MB)
- **Flush Interval** - # of seconds for the regular flushing of data to disk (Dflt = 900 Seconds / 15 Minutes)

Database Query Store Options

- **Interval Length** - # of minutes for RunTime stats intervals. Must be one of these 1,5,10,15,30,60,1440 (Dflt = 60)
- **Stale Query Threshold** - # of days that queries with no set policy are kept. (Dflt = 30)
- **Max Plans Per Query** – Limits the # of plans for a given query. (Dflt = 200)
- **Query Capture Mode** – Current query capture mode (All (Dflt), Auto, None)
- **Size Based Cleanup Mode** – Determines if data will automatically be cleaned up when Size reaches 90% of max size. Off or Auto (Dflt)

Where is the Data Physically Stored?

Who wants to guess?

- It is stored in the Primary File Group of each database
- You don't have an option to change that
- This may change how you setup your primary FG

Querying the Query Store

Use the view's mentioned earlier but:

- Remember the run time stats should always be aggregated when looking for totals across each of the relevant intervals
- Even within a specific interval there can be multiple rows for a given runtime_stats_id or query_id & plan_id combination
 - Runtime_stats_id is no longer a PK
- The majority of the ID's are unique only to a given QDS where as Query_Hash and Plan_Hash are not

Things to Keep in Mind

Remember:

- Query Store stats are at the plan (statement) level not batch or procedure
- Stats are stored in the database you executed the query from
 - If you execute a stored procedure in another DB that DB has context
 - If you run them from a db that does not have QS enabled or a system db such as tempdb or master they are not tracked
- Run time stats are based on UTC datetime
- Always aggregate the DMV results when looking at totals
- You may not get the plan you expect when you force it



Query Store

Demo

References

Channel 9 video on Query Store

<https://channel9.msdn.com/Shows/Data-Exposed/Query-Store-in-SQL-Server-2016>

Monitoring Performance by Using the Query Store

<https://msdn.microsoft.com/en-US/library/dn817826.aspx>

Query store Usage Scenarios

<https://msdn.microsoft.com/en-us/library/mt614796.aspx>

5 Part Simple Talk series on Query Store by Enrico van de Laar

<https://www.simple-talk.com/sql/database-administration/the-sql-server-2016-query-store-overview-and-architecture/>

Query Store Best Practices

<https://msdn.microsoft.com/en-us/library/mt604821.aspx>

How the Query Store Collects Data

<https://msdn.microsoft.com/en-us/library/mt631173.aspx>

Using the Query Store with In-Memory Objects

<https://msdn.microsoft.com/en-us/library/mt590480.aspx>



Thank You

Learn more from
Andrew J. Kelly

akelly@solidq.com or follow [@gunneyk](https://twitter.com/gunneyk)