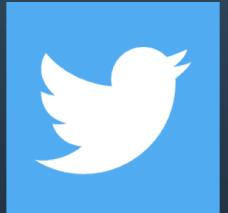


DBA MASTERY



Introduction to Query Store





DBA
MASTERY

Carlos Robles

Principal Consultant, DBA Mastery



/croblesdba



@dbamastery



dbamastery@gmail.com

Experience

Over 10 years of experience as DBA
Multi platform DBA
Linux, Windows, Virtual & Cloud environments
Oracle, SQL Server, MySQL, Azure SQL DB
MCSE Data management and analytics

Community

Speaker, Guatemala SSUG group leader,
blogger, author, mentor, volunteer, like to help
people in Twitter using #SQLHelp hashtag.

DBA Mastery

SQL Server tips, scripts, best practices and
more.

MAXDOP Calculator



AGENDA

- Performance issues scenarios
- Traditional troubleshooting tools
- What is Query Store
- Architecture Overview
- Setup and configuration
 - Demo
- Exploring Query Store
 - Demo
- Use cases
- Best Practices \ Considerations



PERFORMANCE ISSUES SCENARIOS

- Had issues with execution plan changes slowing down the front end application after upgrading a SQL Server instance to latest version
- Had a performance problem with a database and was unable to determine the root cause because someone decided to reboot the server
- Had a third party application experiencing performance problems but no changes to the database schema are not allowed



- Had an application down / slow because of the database is not performing well, upper management is expecting you to push the go faster button
- Had an application that only uses AD-HOC and dynamic SQL queries



SQL Server

Traditional troubleshooting tools



PERFORMANCE MONITOR

Built-in Windows tool, measures performance statistics while issues are occurring.

Setup data collector to log data into a file.



DMVs

```
297 .widget-area{  
347 .widget-area{  
348 font-size:  
349 }  
350  
351  
352 /x-menu  
353  
354  
355 #access {  
356 display: flex;  
357 height: 69px;  
358 float: right;  
359 margin: 11px;  
360 max-width:  
361  
362 .access ul {  
363 font-size: 1em;  
364 list-style-type: none;  
365 margin: 0 0 20px 0;  
366 padding-left: 0;  
367 &gt;-index: 999;  
368 text-align:  
369  
370 .access ul li {  
371 display: inline-block;  
372 width: 32px;  
373 height: 14px;  
374 float: left;  
375 margin-right: 2px;  
376 margin-top: 7px;  
377 &gt;#access .access ul li a {  
378 background: url(../img/mailbox.png);  
379 display: inline-block;  
380 width: 32px;  
381 height: 14px;  
382 float: left;  
383 margin-right: 2px;  
384 margin-top: 7px;  
385 &gt;#access .access ul li a:active {  
386 background: url(../img/phone.png);  
387 display: inline-block;  
388 width: 32px;  
389 height: 14px;  
390 float: left;  
391 margin-right: 2px;  
392 margin-top: 7px;  
393 &gt;#access .access ul li a:active:after {  
394 content: '</>';  
395 position: absolute;  
396 top: -10px;  
397 left: -10px;  
398 &gt;#access .access ul li a:active:after {  
399 content: '</>';  
400 position: absolute;  
401 top: -10px;  
402 left: -10px;
```

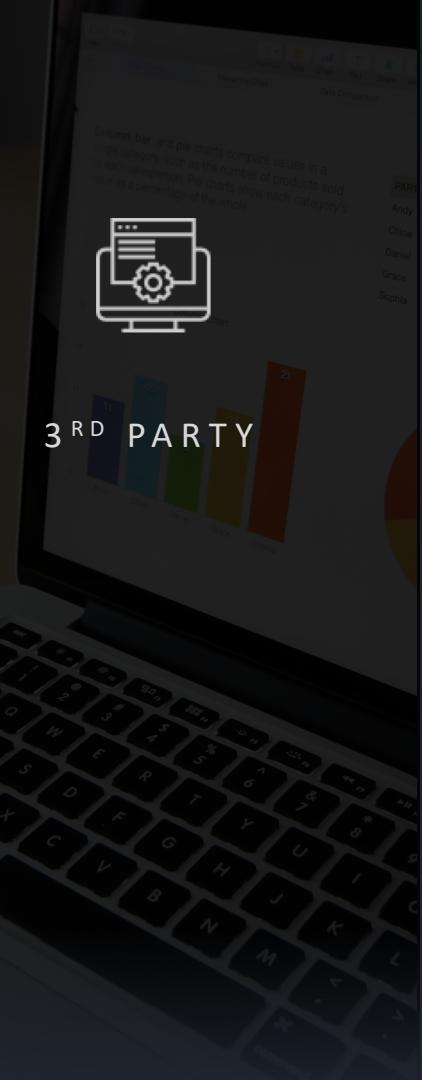


EXTENDED EVENTS

```
125  
126  
127  
128  
129 }  
130  
131  
132 em.well {  
133 background: url(../img/mailbox.png);  
134 display: inline-block;  
135 width: 32px;  
136 height: 14px;  
137 float: left;  
138 margin-right: 2px;  
139 margin-top: 7px;  
140 &gt;#access .access ul li a {  
141 background: url(../img/phone.png);  
142 display: inline-block;  
143 width: 32px;  
144 height: 14px;  
145 float: left;  
146 margin-right: 2px;  
147 margin-top: 7px;  
148 &gt;#access .access ul li a:active {  
149 background: url(../img/mailbox.png);  
150 display: inline-block;  
151 width: 32px;  
152 height: 14px;  
153 float: left;  
154 margin-right: 2px;  
155 margin-top: 7px;  
156 &gt;#access .access ul li a:active:after {  
157 content: '</>';  
158 position: absolute;  
159 top: -10px;  
160 left: -10px;  
161 &gt;#access .access ul li a:active:after {  
162 content: '</>';  
163 position: absolute;  
164 top: -10px;  
165 left: -10px;
```



3RD PARTY



SQL Server

Traditional tools



PERFORMANCE MONITOR

Built-in Windows tool, measures performance statistics while issues are occurring.

Setup data collector to log data into a file.



DMVs

Dynamic management views and functions return server state information that can be used to monitor the health of a server instance, diagnose problems, and tune performance.



EXTENDED EVENTS

```
125  
126  
127  
128  
129  
130  
131  
132 .menu-item {  
133   background: url(../img/mail-  
134   display: inline-block;  
135   width: 32px;  
136   height: 14px;  
137   float: left;  
138   margin-right: 2px 7px 0 0;  
139 }  
140 .access {  
141   display: flex;  
142   height: 60px;  
143   float: right;  
144   margin-top: 10px;  
145   max-width: 300px;  
146 }  
147 .access ul {  
148   font-size: 14px;  
149   list-style-type: none;  
150   margin: 0 0 20px 0;  
151   padding-left: 0;  
152   &gt; li {  
153     border-bottom: 1px solid #ccc;  
154     padding: 10px 0 10px 20px;  
155     position: relative;  
156   }  
157   &gt; li::before {  
158     content: '</>';  
159     font-size: 1.5em;  
160     position: absolute;  
161     top: -10px;  
162     left: 20px;  
163   }  
164 }  
165 .access ul li {  
166   &gt; a {  
167     color: inherit;  
168     text-decoration: none;  
169   }  
170 }  
171 .access ul li::before {  
172   content: '</>';  
173 }
```

3RD PARTY



SQL Server

Traditional tools



PERFORMANCE MONITOR

Built-in Windows tool, measures performance statistics while issues are occurring.

Setup data collector to log data into a file.



DMVs

Dynamic management views and functions return server state information that can be used to monitor the health of a server instance, diagnose problems, and tune performance.



EXTENDED EVENTS

Light weight performance monitoring system that uses very few performance resources.



3 RD PARTY



SQL Server

Traditional tools



PERFORMANCE MONITOR

Built-in Windows tool, measures performance statistics while issues are occurring.

Setup data collector to log data into a file.



DMVs

Dynamic management views and functions return server state information that can be used to monitor the health of a server instance, diagnose problems, and tune performance.



EXTENDED EVENTS

Light weight performance monitoring system that uses very few performance resources.



3 RD PARTY

Idera - SQL Diagnostic Manager
Quest - Foglight
RedGate - SQL Monitor
SentryOne
Solarwinds - DB Performance

WHAT IS QUERY STORE

- Flight recorder \ black box of SQL Server
- Captures query text, estimated execution plans, execution runtime statistics and wait statistics
- It aggregates data by time intervals
- It simplifies troubleshooting by helping to find performance issues and query regressions easy
- Introduced in SQL Server 2016, existed before on Azure SQL DB



WHAT MAKES QUERY STORE SO GOOD

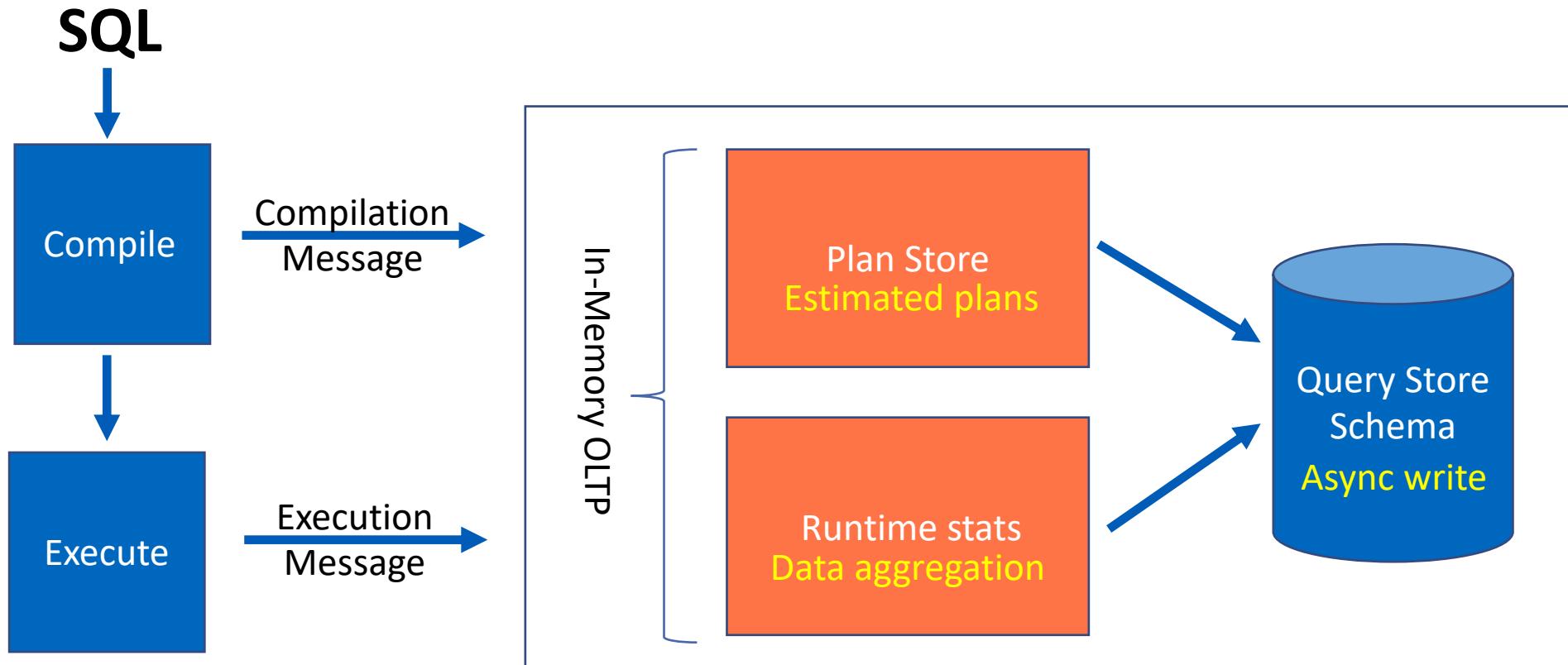
- What is the added value? Why it is so attractive for DBAs?
- It saves time!!!
- Makes query tuning less complex
- The previous solutions were not part of SQL Server engine
- Tracks information independent of what is cached
- Stores every SQL statement from a batch
- Works at database level



- Easy identification of top consuming queries
- Resource usage patterns
- Easy plan forcing (it depends ...)



ARCHITECTURE OVERVIEW



Aggregated data flush → Async checkpoint mechanism

CONFIGURATION

- Available in all SQL Server editions
- It is enabled at database level (but master and TempDB)
- Permissions
 - Access Query Store data \ reports → VIEW DATABASE STATE
 - Query plan forcing → DB_OWNER
- To enable it from T-SQL

```
ALTER DATABASE MyDatabase SET QUERY_STORE = ON;
```



DEMO



OPTIONS RELATED TO DATA COLLECTION

- Operation mode
 - Default Read-Write
 - Read Only
- Query capture mode
 - Default ALL for SQL Server 2016+
 - Default AUTO for Azure SQL Server DB
 - NONE
- Max plans per query
 - Default 200



OPTIONS RELATED TO DATA STORE

- Max storage size MB
 - Default 200 MBs
- Size based cleanup mode
 - AUTO
 - OFF
- Cleanup policy
 - Stale query threshold in days
 - Default 30 days



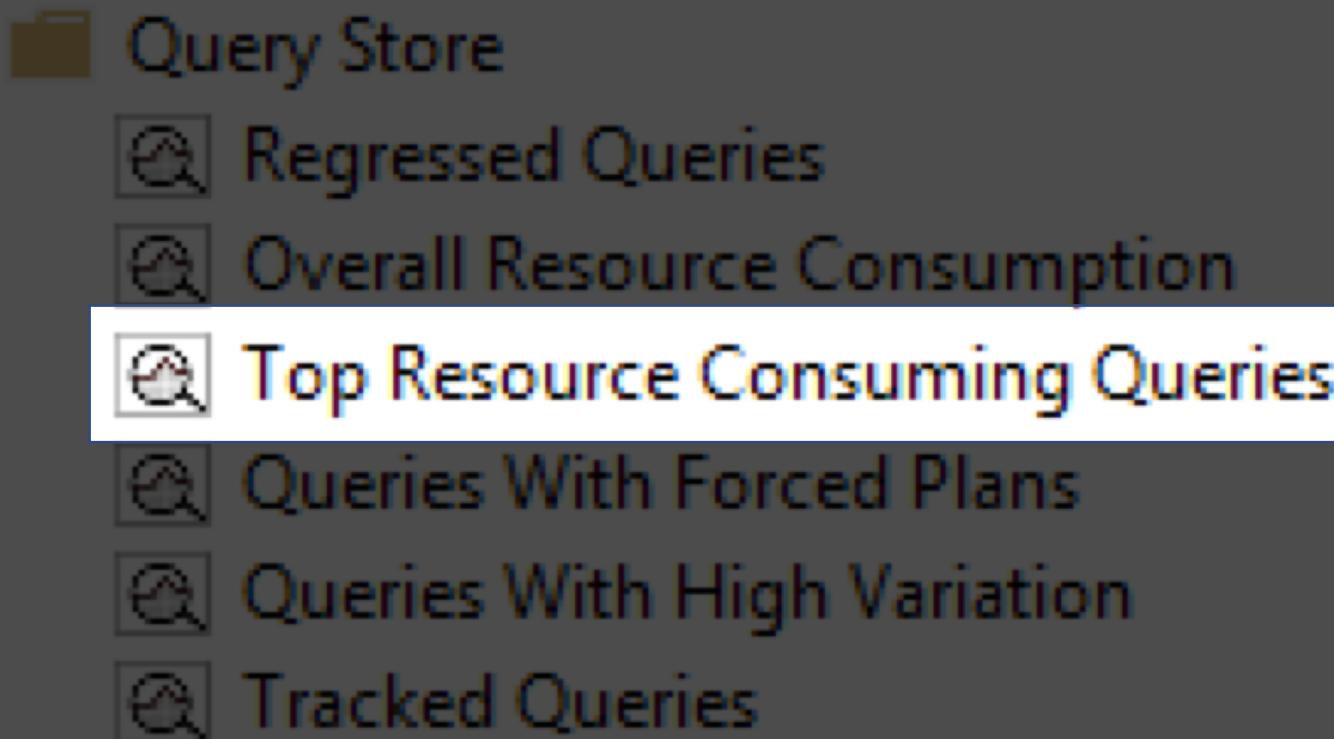
OPTIONS RELATED TO DATA AGGREGATION

- Data flush interval
 - Default 15 minutes
 - Flushing runtime and wait stats data from memory to disk
 - In case of memory pressure data will be flushed to disk immediately
- Interval length minutes
 - Default 60 minutes
 - 1, 5, 10, 15, 30, 60, 1,440 (24 hours)
- Wait stats capture



HOW TO USE QUERY STORE

- SQL Server Management Studio (SSMS)
- T-SQL through DMVs



DMVs

- Plan store
 - Query plan & query text
 - Query context settings
- Runtime stats store
 - Compilation time, duration & last execution
 - CPU & DOP
 - Logical reads & physical reads
 - Writes
 - Wait statistics (SQL Server 2017)



Plan Store

`sys.query_store_query`
`sys.query_store_plan`
`sys.query_store_query_text`
`sys.query_context_settings`

`sys.query_store_runtime_stats`
`sys.query_store_wait_stats`
`sys.query_store_runtime_stats_interval`

Runtime Statistics Store


Table Valued Function
[QUERY_STORE_RUNTIME_STATS]
Cost: 3 %

TVF

In-Memory
Data

On-Disk
Data

T-SQL Queries

DEMO



USE CASES

- Pinpoint and fix queries with plan choice regressions
- Identify and tune top resource consuming queries
- A/B testing, application changes
- Keep performance stability during the upgrade to newer SQL Server



BEST PRACTICES \ CONSIDERATIONS

- SSMS latest version
 - Update file in case SSMS is already installed
- XE monitoring
- T-SQL monitoring
- Persisted after backup \ restore
- CPU additional load ~3-5%



- PerfMon
 - Query Store CPU usage
 - Query Store logical reads, writes
 - Query Store physical reads (ASYNC writer)
- Database datafile distribution
 - PRIMARY (For internal objects and Query Store) MAIN (User objects)



QUESTIONS ?



- For more information

[Best practices according to Microsoft](#)

[How Query Store Collects data](#)

[Monitoring performance by using Query store](#)

[Query Store Usage Scenarios](#)





-  /croblesdba
-  @dbamastery
-  dbamastery@gmail.com

THANKS !!!

