



C# Corner

SQL SERVER VIRTUAL CONFERENCE

JANUARY 29-30, 2021 | 07:00 AM - 07:00 PM EDT

RAISING MONEY FOR CHARITY HELPING CHILDREN AFFECTED BY COVID-19

[Donate Here](#)

Performance Features in SQL Server and Azure SQL





javier.ignacio.villegas@gmail.com



javier_vill



javiervillegas



sql-javier-villegas.blogspot.com.ar

Javier Villegas

IT Director – Data and BI at Mediterranean Shipping Company

Involved with the SQL Server since early versions. Azure SQL

Specialization in SQL Server Administration, Performance Tuning and High Availability

Microsoft MVP Data Platform

Microsoft Certified Trainer (MCT)

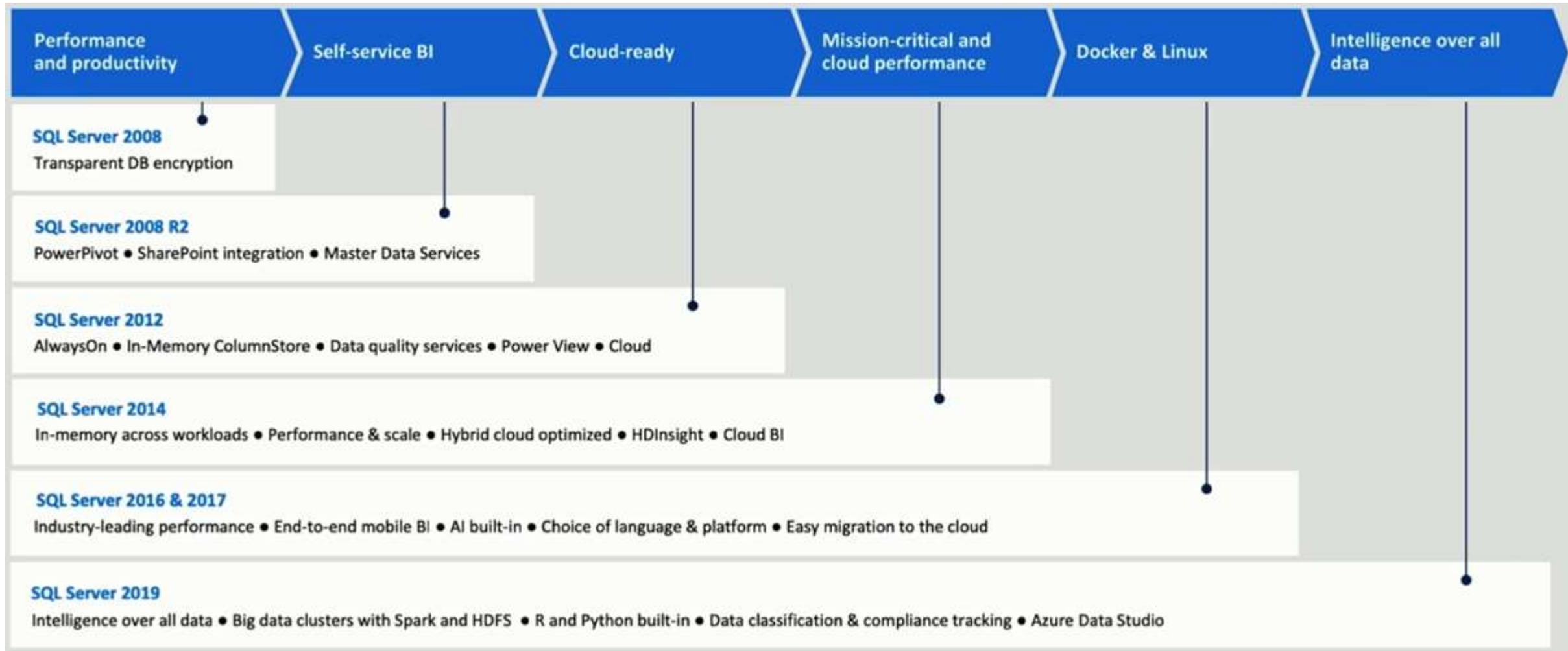
Technical Speaker

SQL PASS, 24 HOP, SQL Saturdays , PASS Marathon and PASS Virtual Groups,

vOpen, Microsoft AI+ Tour, GroupBy and DataPlatformGeeks



SQL Server





Azure SQL

Azure SQL

A unified SQL portfolio built on the industry-leading SQL Server engine

SQL Server on Azure Virtual Machines



Best for re-hosting and apps requiring OS-level access and control

Automated manageability features and OS-level access

Infrastructure as a Service

Azure SQL Managed Instance



Best for modernizing existing apps

Offers high compatibility with SQL Server and native VNET support

Platform as a Service

Azure SQL Database



Best for building new apps in the cloud

Pre-provisioned or serverless compute and Hyperscale storage to meet demanding workload requirements

SQL is everywhere

SQL Server 2019



Best for modernizing your databases in your private cloud

SQL on Linux



Best for a compatible SQL running on Linux

SQL in containers



Best for portable, consistent, and easy to patch SQL

SQL on Kubernetes



Best for database containers at scale with built-in HA

SQL virtual machines



Best for migrations and applications requiring OS-level access

SQL managed instances



Best for most lift-and-shift migrations to the cloud. Instance pool options are available

SQL databases



Best for modern cloud applications. Elastic pools, Hyperscale and Serverless options are available

SQL Edge



Best for data and machine learning applications on IOT Edge

Performance Capabilities and Tasks

Performance capabilities

- Memory, CPU, I/O capacities
- Indexes
- In-Memory OLTP
- Partitions
- Intelligent Performance

Performance tasks

- Configuring and Maintaining for Performance
- Monitoring and Troubleshooting Performance
- Performance Scenarios
- Accelerating and Tuning Performance

Azure SQL performance capabilities

Max capacities

- Azure SQL Database - Up to 128 vCores and 4TB Memory and 4TB Database (data)
- Hyperscale – 100TB Max Database Size
- Managed Instance – Up to 80 vCores, 400GB Memory, and 8TB Database (data)
- `sys.dm_os_job_object` for true capacities

Indexes

- All index types are supported
- Online and resumable indexes fully supported
- Columnstore Indexes available in almost all tiers

In-memory OLTP

- Available in Business Critical Tiers
- Memory Optimized FILEGROUP created with database creation
- Max memory a portion of overall memory limit

Partitions

- Supported for both Azure SQL Database and Managed Instance
- Placement on filegroups only supported for Managed Instance

SQL Server 2019 performance enhancements

- Not 100% in Azure SQL (today)
- Intelligent Query Processing (except for Scalar UDF Inlining)

Configuring for performance

Tempdb

- Always kept on local SSD drives
- For DB, # files scales with vCores
- You get 12 files with MI
- MIXED_PAGE_ALLOCATION IS OFF
- AUTOGROW_ALL_FILES is ON
- Tempdb Metadata Optimization not supported

Database

- Only full recovery supported so minimal logging for bulk operations not possible

Files and Filegroups

- MI supports adding files and sizes but not physical placement
- # files and file size can be used to tune I/O performance
- User defined FILEGROUP only supported for MI

MAXDOP

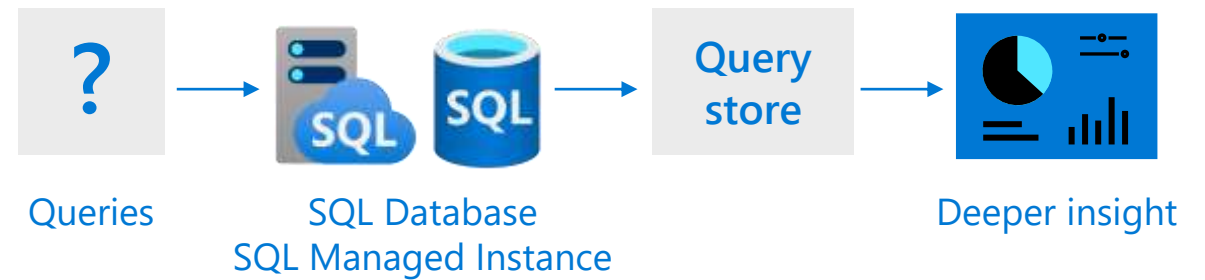
- ALTER DATABASE SCOPED CONFIGURATION
- sp_configure supported for MI
- Query hints allowed
- MI supports RG

Resource Governor

- User defined Managed Instance only

Monitoring and Troubleshooting Performance

- Azure Monitor Metrics, Alerts, and Logs
- Dynamic Management Views
- Extended Events
- Lightweight Query Profiling on by default
- Query Plan Debugging with SET statements
- Query Store on by default
- Performance Visualizations in portal



Dynamic Management Views



Azure SQL Managed Instance

- All SQL Server DMVs available
- sys.server_resource_stats

You will need these

- sys.dm_io_virtual_file_stats
- sys.dm_os_performance_counters
- sys.dm_instance_resource_governance
- sys.dm_user_db_resource_governance



Azure SQL Database

- Common SQL Server DMVs available
- sys.dm_db_resource_stats
- sys.elastic_pool_resource_stats

Geek out

- sys.dm_user_db_resource_governance_internal
- sys.dm_resource_governor_resource_pools_history_ex
- sys.dm_resource_governor_workload_groups_history_ex

Extended events with Azure SQL



Azure SQL Database

- Most common events and actions
- File (Azure Blob Storage), ring_buffer, and counter targets



Azure SQL Managed Instance

- All events, targets, and actions supported
- File targets to Azure Blob Storage
- Azure MI supports all SQL Server events plus Azure specific events

Intelligent Performance

- Query Store (2016)
- Automatic Tuning (2017)
- Adaptive Query Processing (AQP) (2017)
- Intelligent Query Processing (IQP) (2019)
- Columnstore and In-Memory OLTP
- Tempdb Metadata – “It just runs faster”

Industry leading performance



Challenge

DBA's spend most of their time tuning database performance, leading to high operational costs



Solution

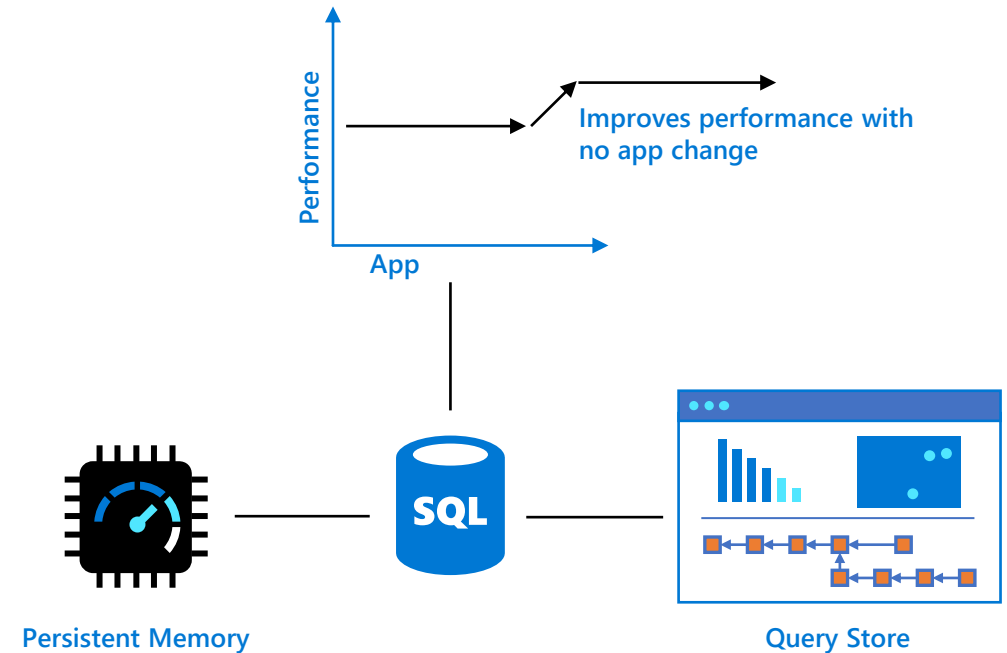
Automation with Intelligent Database capabilities while providing mission critical performance

Gain performance with no app changes with Intelligent Query Processing

Performance insights anytime anywhere with Query Store and lightweight query profiling

Accelerate I/O performance with Persistent Memory

Intelligent Database



Query Store

The SQL Server Query Store feature provides you with insight on query plan choice and performance.

It simplifies performance troubleshooting by helping you quickly find performance differences caused by query plan changes.

Query Store automatically captures a history of queries, plans, and runtime statistics, and retains these for your review.

It separates data by time windows so you can see database usage patterns and understand when query plan changes happened on the server.

Query Store

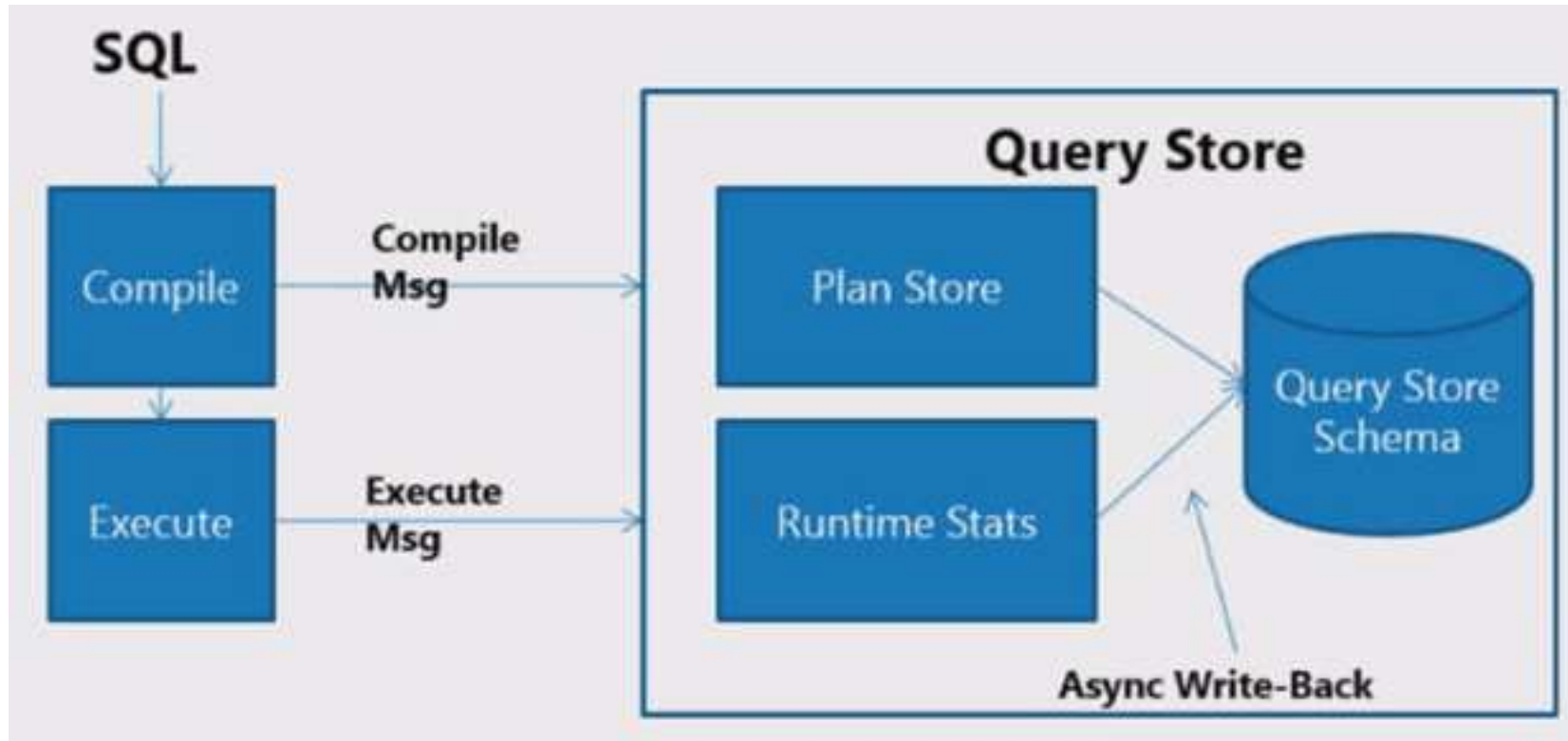
Execution plans for any specific query in SQL Server typically evolve over time due to a number of different reasons such as statistics changes, schema changes, creation/deletion of indexes, etc.

The procedure cache (where cached query plans are stored) only stores the latest execution plan. Plans also get evicted from the plan cache due to memory pressure. As a result, query performance regressions caused by execution plan changes can be non-trivial and time consuming to resolve.

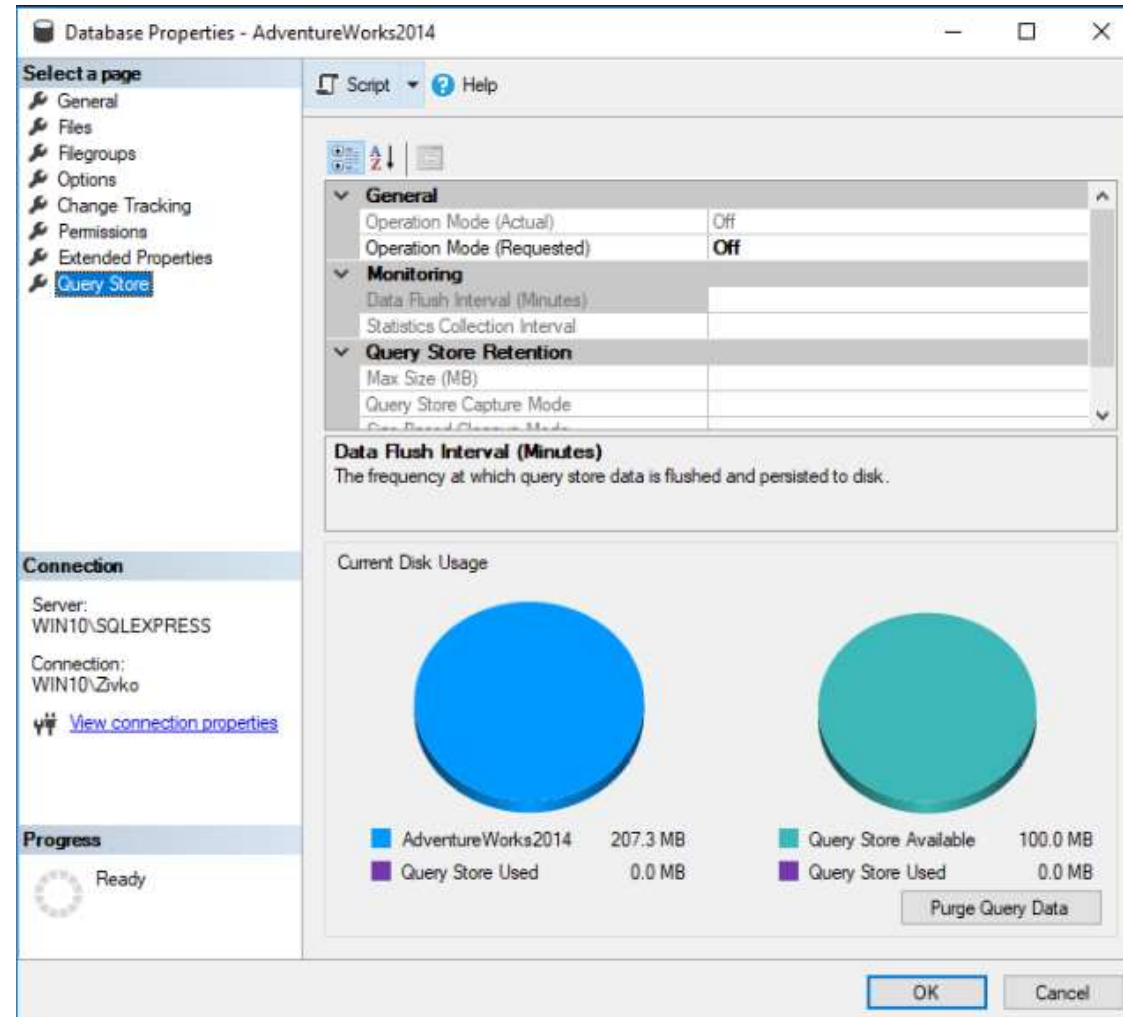
Since the query store retains multiple execution plans per query, it can enforce policies to direct the query processor to use a specific execution plan for a query. This is referred to as plan forcing.

Plan forcing can resolve a query performance regression caused by a plan change in a very short period of time.

Query Store



Query Store



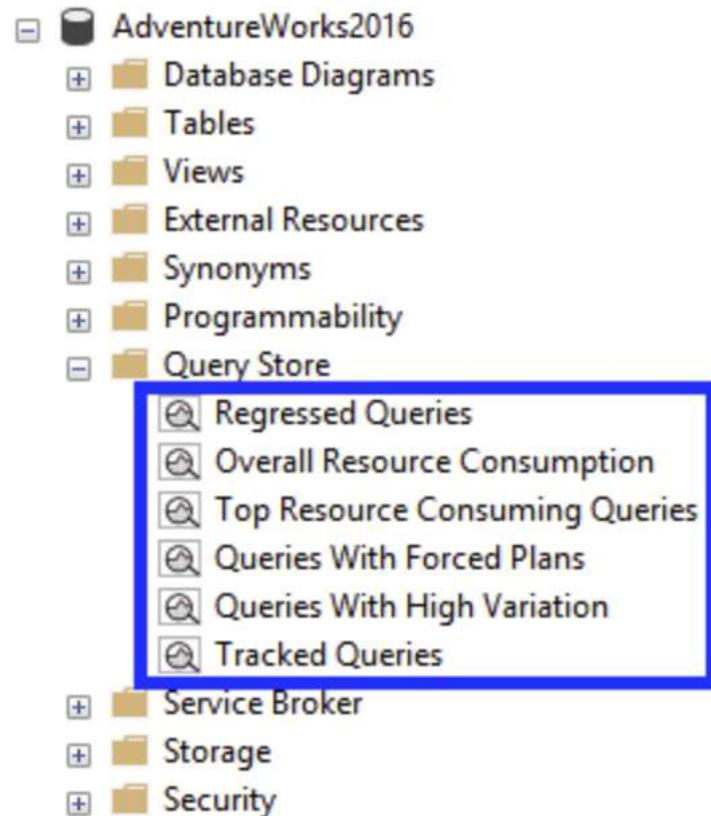
Common scenarios for using Query Store

- Quickly find and fix a plan performance regression by forcing the previous query plan. Fix queries that have recently regressed in performance due to execution plan changes.
- Determine the number of times a query was executed in a given time window, assisting a DBA in troubleshooting performance resource problems.
- Identify top n queries (by execution time, memory consumption, etc.) in the past x hours.
- Audit the history of query plans for a given query.
- Analyze the resource (CPU, I/O, and Memory) usage patterns for a particular database.
- Identify top n queries that are waiting on resources.
- Understand wait nature for a particular query or plan.

The query store contains three stores:

- a **plan store** for persisting the execution plan information.
- a **runtime stats store** for persisting the execution statistics information.
- a **wait stats store** for persisting wait statistics information.

Query Store



```
ALTER DATABASE <DB Name> SET QUERY_STORE = ON;
```

```
EXEC sp_query_store_force_plan @query_id = 48, @plan_id = 49;
```

```
EXEC sp_query_store_unforce_plan @query_id = 48, @plan_id = 49;
```

Best Practices with Query Store

- Keep Query Store adjusted to your workload
- Verify Query Store is collecting query data continuously
- Set the optimal query capture mode
- Keep the most relevant data in Query Store
- Avoid using non-parameterized queries
- Avoid a DROP and CREATE pattern when maintaining containing objects for the queries
- Check the status of Forced Plans regularly
- Avoid renaming databases if you have queries with Forced Plans
- Use trace flags on mission critical servers to improve recovery from disaster

Automatic Query Tuning (IQP)

Automatic tuning is a database feature that provides insight into potential query performance problems, recommend solutions, and automatically fix identified problems.

Automatic tuning in SQL Server 2017 notifies you whenever a potential performance issue is detected, and lets you apply corrective actions, or lets the Database Engine automatically fix performance problems.

Automatic tuning in SQL Server 2017 enables you to identify and fix performance issues caused by **SQL plan choice regressions**.

Automatic tuning in Azure SQL Database creates necessary indexes and drops unused indexes.

Automatic Query Tuning

Database Engine monitors the queries that are executed on the database and automatically improves performance of the workload. Database Engine has a built-in intelligence mechanism that can automatically tune and improve performance of your queries by dynamically adapting the database to your workload. There are two automatic tuning features that are available:

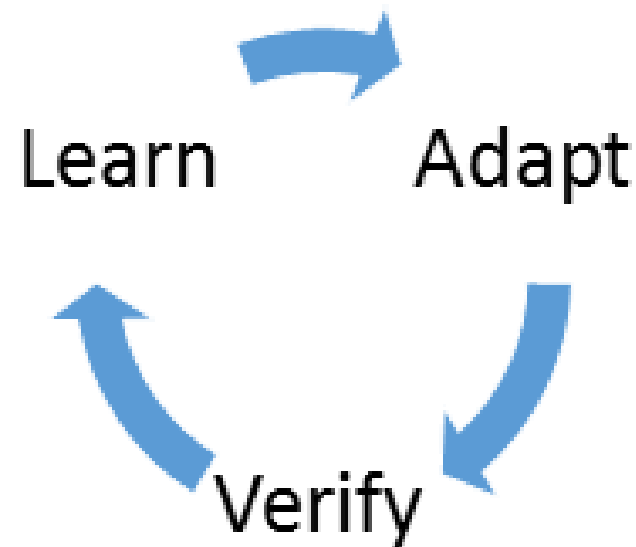
- **Automatic plan correction** (available in SQL Server 2017 and Azure SQL Database) that identifies problematic query execution plans and fixes SQL plan performance problems.
- **Automatic index management** (available only in Azure SQL Database) that identifies indexes that should be added in your database, and indexes that should be removed.

Why Automatic Query Tuning ?

One of the main tasks in classic database administration is monitoring the workload, identifying critical Transact-SQL queries, indexes that should be added to improve performance, and rarely used indexes. Database Engine provides detailed insight into the queries and indexes that you need to monitor. However, constantly monitoring database is a hard and tedious task, especially when dealing with many databases. Managing a huge number of databases might be impossible to do efficiently. Instead of monitoring and tuning your database manually, you might consider delegating some of the monitoring and tuning actions to Database Engine using automatic tuning feature.

How does automatic tuning works?

Automatic tuning is a continuous monitoring and analysis process that constantly learns about the characteristic of your workload and identify potential issues and improvements.

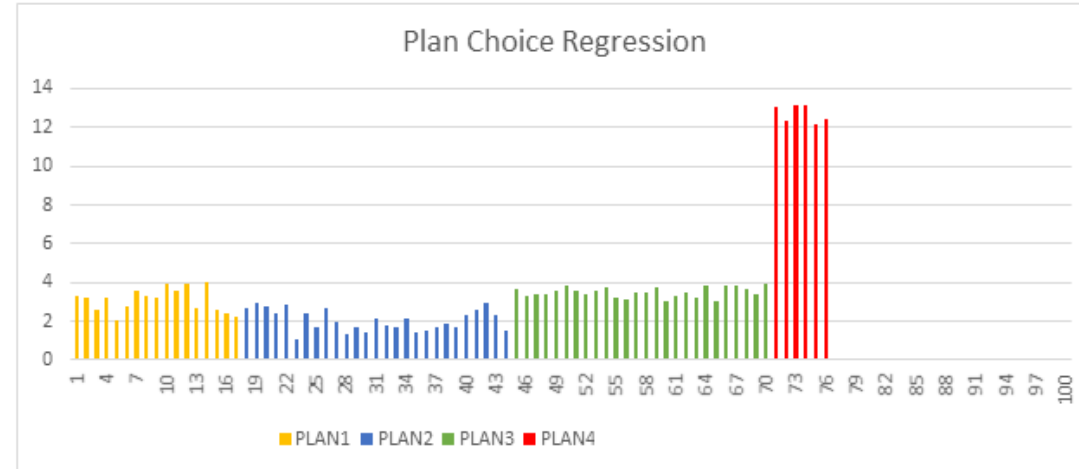


How does automatic tuning works?

This process enables database to dynamically adapt to your workload by finding what indexes and plans might improve performance of your workloads and what indexes affect your workloads. Based on these findings, automatic tuning applies tuning actions that improve performance of your workload. In addition, database continuously monitors performance after any change made by automatic tuning to ensure that it improves performance of your workload. Any action that didn't improve performance is automatically reverted. This verification process is a key feature that ensures that any change made by automatic tuning does not decrease the performance of your workload.

What is SQL plan choice regression?

SQL Server Database Engine may use different SQL plans to execute the Transact-SQL queries. Query plans depend on the statistics, indexes, and other factors. The optimal plan that should be used to execute some Transact-SQL query might be changed over time. In some cases, the new plan might not be better than the previous one, and the new plan might cause a performance regression.



Whenever you notice the plan choice regression, you should find some previous good plan and force it instead of the current one using **sp_query_store_force_plan** procedure. Database Engine in SQL Server 2017 provides information about regressed plans and recommended corrective actions. Additionally, Database Engine enables you to fully automate this process and let Database Engine fix any problem found related to the plan changes.

Automatic plan choice correction

Database Engine can automatically switch to the last known good plan whenever the plan choice regression is detected.



Database Engine automatically detects any potential plan choice regression including the plan that should be used instead of the wrong plan. When the Database Engine applies the last known good plan, it automatically monitors the performance of the forced plan. If the forced plan is not better than the regressed plan, the new plan will be unforced and the Database Engine will compile a new plan. If Database Engine verifies that the forced plan is better than regressed one, the forced plan will be retained until a recompile (for example, on next statistics or schema change) if it is better than the regressed plan.

Note: Any plans auto forced do not persist on a restart of the SQL Server instance.

Adaptive Query Processing

- Before SQL Server 2017, the behavior of the SQL Server query-processing engine was to analyze the query first, create the plan and then execute it.
- If the plan was somehow not appropriate, the query-processing engine was not able to change it while executing the query or even after it.
- Sometimes the query execution plans made by SQL Server are not appropriate.

Adaptive Query Processing

There are number of reasons behind poorly designed execution plan.

- Lack of appropriate indexes
- Outdated statistics
- In-appropriate query execution plans cached with outdated values are stored
- Poorly written codes

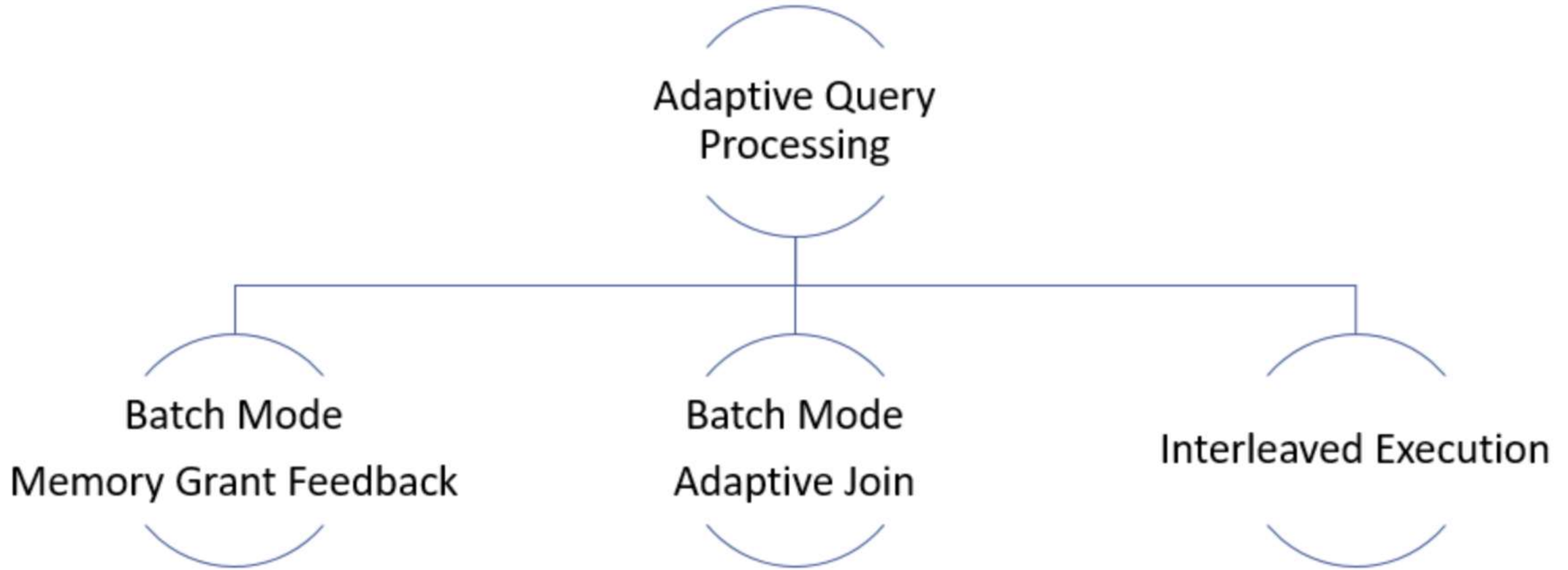
There are two ways to fix these problems.

- Provide more relevant and accurate information about the query
- re-write the code so that it may perform in a better way.

Adaptive Query Processing

- The Adaptive Query Processing breaks the barrier between query plan optimization and actual execution.
- Now, we can have the optimization done while the actual query is executing or even after the actual query execution is completed to benefit the later executions.

Adaptive Query Processing



Interleaved Executions

- SQL Server Query Optimizer learns that if the estimates are way off than the actual ones, it adjusts the execution plan by actually executing a part of the query execution plan first and re-design the Query Execution Plan based on the actual amount of the rows.
- This leads to a much better plan, which is created and adjusted while the query is executing.

Batch Mode Memory Grant Feedback

- This feature enables the SQL Server Query Processing engine to learn that if the memory grants are not sufficient then the engine will change the cached execution plan and update the memory grants so that the later executions should benefit with the new grants.

Batch Mode Adaptive Joins

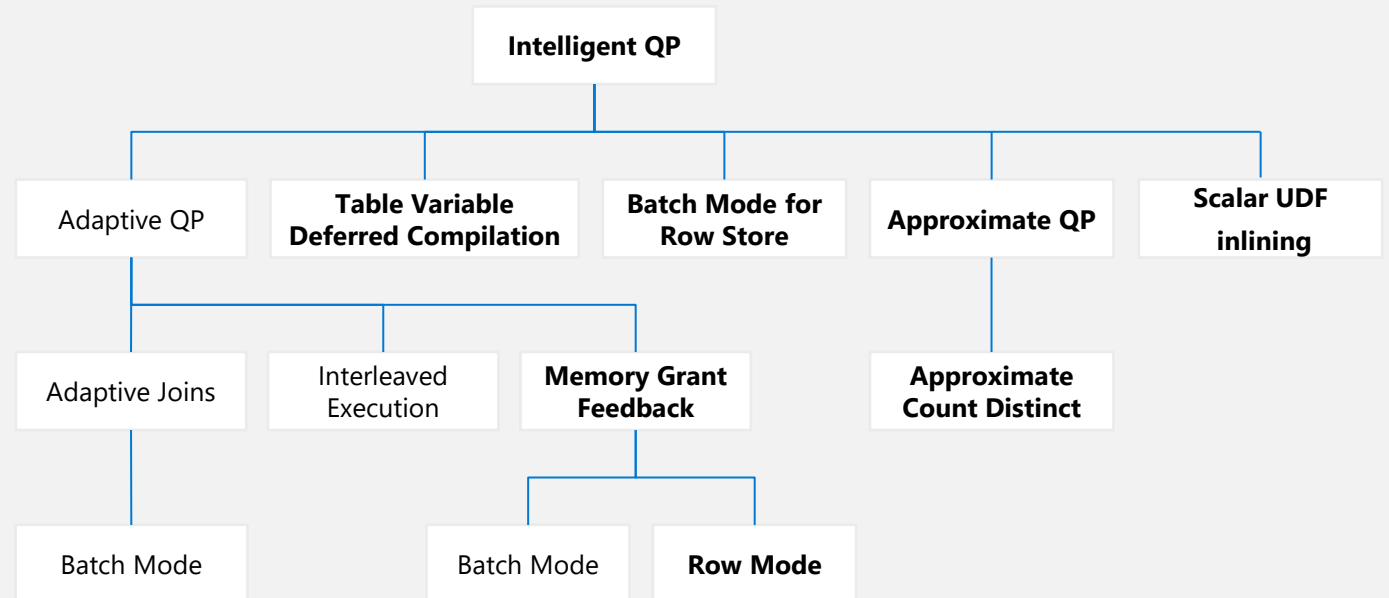
- This is a way to improve the query performance based on the number of rows flowing through the actual execution plan.
- The execution engine defers the choice of a Hash Join or a Nested Loop Join until the first join in the query execution plan is made.
- After that first join has been executed and based on the number of records fetched, the SQL Server Query Processing engine decides whether to choose Hash Join or Nested Loop Join.

Mission critical performance

The intelligent database

- Intelligent Query Processing
- Gain performance insights anytime and anywhere with Lightweight Query Profiling
- Accelerating I/O performance with Persistent Memory
- Hybrid Buffer Pool
- Tempdb: It Just Runs Faster

The Intelligent Query Processing feature family



Bold indicates new and improved features in SQL Server 2019

Intelligent Query Processing

dbcompat to enable

140	SQL Server 2017
150	SQL Server 2019

The problem

Build a query processor to work and adapt to any type of workload with no application changes required

The Solution(s)

- Build intelligent, adaptable operators
- Modify query plans in cache based on previous execution
- Expand batch mode
- Execution data drives downstream compilation
- Smarter query processing

Feature Suite

Adaptive Join 140

140 150
Memory Grant Feedback for Batch and Row

Batch Mode for Row Store 150

Interleaved Execution 140
Table Variable Deferred Compilation 150

Approximate Count Distinct
Scalar UDF Inlining 150

Lightweight Query Profiling

The problem

I want to see the details of a query plan at the operator level for any **active** executing query

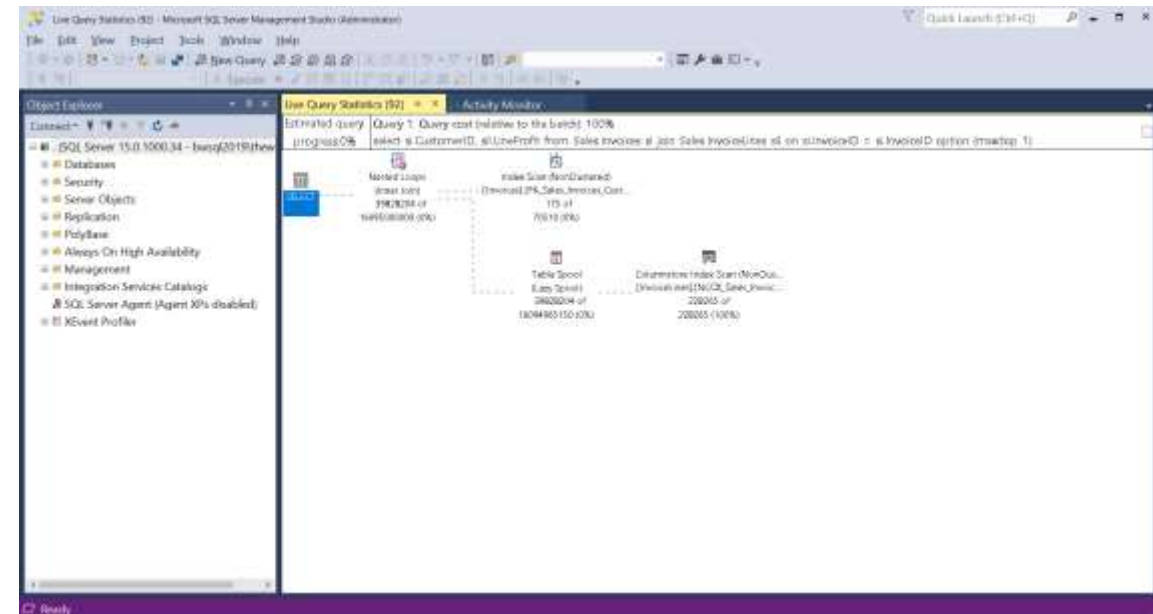
SQL Server 2016 SP1/2017

- Trace flag 7412
- XEvent query_thread_profile

SQL Server 2019

- On by default. No knobs required

```
dm_exec_query_profiles
dm_exec_query_statistics_xml
```



Tempdb Just Runs Faster

The problem(s)

High multi-user rates of tempdb usage can lead to latency due to....

allocation page latch waits – Multiple users needing to allocate pages for temp tables

system table page latch waits – High rates of create/drop require system table modifications

The Solution

- Create multiple files to partition allocation pages
- SQL Server 2016+ creates multiple tempdb files during setup
- Start with 8 and add by 4 until concurrency alleviated
- But...what about system tables?

The SQL 2019 Solution

- Key tempdb system tables become SCHEMA_ONLY memory optimized tables
- Latch and lock free
- Turn on with ALTER SERVER CONFIGURATION
- This is NOT user data just metadata so memory requirements small

Managed Instance

- App compatibility:
AAD logins, Trace Flags
Distributed Transactions (preview), ML services (preview),
SSRS catalog in MI
- Network and connectivity:
Public Endpoint, Service Endpoints, Subnet Delegation, Inbound and
outbound subnet lockdown
Global VNet peering
- PaaS features and manageability:
Failover Groups, TDE BYOK, Accelerated DB recovery, PITR (deleted DBs ,
cross subscriptions), AAD password hash/pass-through auth, SCOM MP
Testing application fault resiliency
Management operations progress indicator
Backups – compression, configurable retention, zone redundant storage
Long term retention (limited preview)
- Perf:
tempdb, log write throughput and IOPS improvements

Hyperscale

- Change Tracking (GA)
- Geo DR (Preview)
- DB Copy (Preview)

Serverless

- Maximum vCore limit: 40 vCores
- Minimum auto-pause delay: 1 hour
- Broader API support including PS & CLI

Platform

- Private Link
- PAUSED_RESUMABLE_INDEX_ABORT
- UTF-8 support
- New memory and compute optimized SKUs
- Backup storage redundancy options
- Zone redundancy for GP tier
- AAD Server Principals

Virtual Machines

- SQL VM Resource Provider
- Free DR replica for on-premises
- Free HA or DR Azure SQL Server
- Automated patching
- SQL 2019 Linux images
- SQL FCI using Azure Shared Disk
- Configure Always On AG through portal

Get started today!

Web pages

[Azure SQL family](#)

[SQL Server on Azure Virtual Machines](#)

[Azure SQL Managed Instance](#)

[Azure SQL Database](#)

[Free SQL Server and Azure training](#)

[Choose Your Azure SQL database tool](#)

[Azure Hybrid Benefit for SQL Server](#)

[Azure SQL Blogs](#)

[Migration guide](#)

[Microsoft Learn: Azure SQL fundamentals](#)

[YouTube: Azure SQL for beginners](#)

[YouTube: Data Exposed](#)

3rd party studies

[GigaOM price-performance study](#)

[Forrester Consulting Total Economic Impact™ study](#)

Resources

[Azure SQL family](#)

[Azure SQL Database serverless](#)

[Azure SQL customer stories](#)



Key takeaways

Azure SQL managed databases take your workloads and apps to the next level

- ✓ Evergreen SQL
- ✓ Industry's highest SLA on Azure SQL Database
- ✓ ML-based performance and security
- ✓ Streamlined app modernization
- ✓ Hyperscale demanding workloads
- ✓ Optimize price-performance with serverless compute
- ✓ Save with the best total cost of ownership



Mediterranean Shipping Company

"With the combination of Intelligent Query Processing in SQL Server 2019 as the second generation of Adaptive Query Processing as well as Memory-Optimized TempDB Metadata, we are now able to achieve incredible performance in a more predictable way for all our business-critical processes. We are really happy with SQL Server 2019; key features such as scalar UDF Inlining and Table Variable Deferred Compilation enhance our developers' productivity and empower MSC in providing improved solutions for our customers."

Javier Villegas, Global Database Administrator & Design Coordinator
Mediterranean Shipping Company



MEDITERRANEAN SHIPPING COMPANY



Learn more

What's new in SQL Server 2019

QR: <https://cutt.ly/khnFloC>



Training

Azure SQL Fundamentals

QR: <https://cutt.ly/ngnX9ob>



Performance Features in SQL Server and Azure SQL



javier_vill



javiervillegas



sql-javier-villegas.blogspot.com.ar



javier.ignacio.villegas@gmail.com





THANK YOU!!

GRACIAS!!