Module 8

Plan caching and recompilation

Module Overview

- Plan cache internals
- Troubleshooting the plan cache
- Query Store
- Automatic tuning

Lesson: Plan cache internals

- Query plan caching and retrieval
- Plan cache management
- Queries without plan cache entries
- Maximizing plan cache efficiency
- Examining the plan cache

Query plan caching and retrieval

- Three plan cache stores
  - Object Plans
  - SQL Plans
  - Bound Trees (algebrizer trees for views)
- Compiled plan is held in a hash bucket in the relevant plan cache store, uniquely identified by a plan handle
- A plan can be reused if bucket hash and plan cache key match
- An executable plan is a session-specific instance of a compiled plan
  - It includes working area for holding variable values and such

Demo Plan cache

SQL Server, Plan Cache object
https://docs.microsoft.com/en-us/sql/relational-databases/performance-monitor/sql-server-plan-cache-object

Examining the Performance Impact of an Adhoc Workload
https://sqlperformance.com/2019/05/sql-plan/perf-impact-adhoc-workload

Plan cache management

- Plan cache housekeeping will begin when the cache stores hit threshold values
- The least-expensive plans are removed first from the plan cache
- You can manually clear the plan cache in various ways

SQL Server Plan Cache Limits
https://www.sqlskills.com/blogs/erin/sql-server-plan-cache-limits/

Eight Different Ways to Clear the SQL Server Plan Cache
https://www.sqlskills.com/blogs/glenn/eight-different-ways-to-clear-the-sql-server-plan-cache/

Queries without plan cache entries

- Queries that cannot be cached:
  - Ad hoc and prepared T-SQL statements requiring object name resolution
- Queries marked for recompilation:
  - CREATE…WITH RECOMPILE
  - OPTION (RECOMPILE)
  - EXECUTE…WITH RECOMPILE
- Natively compiled procedures for memory-optimized tables

Maximizing plan cache efficiency

- Auto-parameterization
  - By default, only used for queries with an evenly distributed cardinality
- Optimize for ad hoc workloads
  - Available on both instance and database level
  - Ad hoc plans are cached only on second execution
- Methods used by various APIs and tools
  - Sp_executesql
    - Does parameter sniffing
  - Sp_prepare, sp_execute etc
    - Doesn't use parameter sniffing, it uses the same mechanism as when you use a T-SQL variable
- Object Plans (stored procedures, triggers, functions)
  - Does parameter sniffing

sp_executesql (Transact-SQL)
https://docs.microsoft.com/en-us/sql/relational-databases/system-stored-procedures/sp-executesql-transact-sql

sp_prepare (Transact SQL)
https://docs.microsoft.com/en-us/sql/relational-databases/system-stored-procedures/sp-prepare-transact-sql

optimize for ad hoc workloads Server Configuration Option
https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/optimize-for-ad-hoc-workloads-server-configuration-option

How to Reduce the CPU Overhead of Dynamic SQL
https://www.brentozar.com/archive/2018/05/using-dynamic-sql-frequently-executed-queries/

Specify Query Parameterization Behavior by Using Plan Guides
https://docs.microsoft.com/en-us/sql/relational-databases/performance/specify-query-parameterization-behavior-by-using-plan-guides

Blitz Result: Forced Parameterization
https://www.brentozar.com/blitz/forced-parameterization/

SQL Server Simple and Forced Parameterization
https://www.mssqltips.com/sqlservertip/2935/sql-server-simple-and-forced-parameterization/

Can Forced Parameterization Go Wrong?
https://www.brentozar.com/archive/2018/09/can-forced-parameterization-go-wrong/

Examining the Plan Cache

- sys.dm_exec_cached_plans
  - One row per cached plan
- sys.dm_exec_query_plan
  - Query plan in XML format
- sys.dm_exec_text_query_plan
  - Query plan in text format
- sys.dm_exec_plan_attributes
  - Plan attributes
- sys.dm_exec_query_stats and sys.dm_exec_procedure_stats
  - Plan statistics

Execution Related Dynamic Management Views and Functions (Transact-SQL)
https://docs.microsoft.com/en-us/sql/relational-databases/system-dynamic-management-views/execution-related-dynamic-management-views-and-functions-transact-sql

Lesson: Troubleshooting the plan cache

- Execution plan recompilation
- Recompilation issues
- Problems of the plan cache
- Using the plan cache to guide optimization
- T-SQL query anti-patterns

Execution plan recompilation

- Recompilation occurs when a cached plan is invalidated and a new plan is compiled
- A cached plan may become invalid because:
  - It becomes incorrect (like a schema change)
  - It becomes non-optimal (like a statistics update)
- Recompilation can be tracked using
  - Extended Events
  - Windows Performance Monitor

Understanding SQL Server Recompilations
https://www.mssqltips.com/sqlservertip/5308/understanding-sql-server-recompilations/

Recompilation issues

- Parameter sniffing—fewer recompiles than expected. Address with:
  - OPTION(RECOMPILE)
  - OPTION(OPTIMIZE FOR...)
- Statistics changes
  - KEEP PLAN query hint reduces recompilations caused by statistics changes
  - KEEPFIXED PLAN query hint prevents recompilations caused by statistics changes

Hints (Transact-SQL) – Query
https://docs.microsoft.com/en-us/sql/t-sql/queries/hints-transact-sql-query

KEEP PLAN Demystified
https://straightforwardsql.com/posts/keep-plan-demystified/

Problems of the plan cache

- Plan cache bloat:
  - Caused by multiple copies of the same query execution plan entering the cache
  - Addressed by:
    - Code changes
    - Optimize for ad hoc workloads setting (server or database level)
    - FORCED PARAMETERIZATION database level

Using the Plan Cache to Guide Optimization

- Many performance measures are captured for cache plans in sys.dm_exec_query_stats and sys.dm_exec_procedure_stats
- You can use these DMVs to identify candidate statements for optimization and performance tuning
  - Query store is way superior for these purposes, though

T-SQL query anti-patterns

- Avoid below if you care about performance
  - Cursors
  - Using DISTINCT indiscriminately
  - Scalar functions
    - Unless you are on 2019 with database compatibility level 2019
      - Verify that UDF inlining work for you, though
  - Calculations on the column-side
    - YEAR(OrderDate) = 2022
  - Non-matching data types
    - Pass the value from the client with the same type as the column is defined in the table

T-SQL Anti-Patterns: SQL User Defined Functions (UDFs) that turn your set operation into a cursor
https://sqlsolutionsgroup.com/t-sql-anti-patterns-user-defined-functions/

Scalar functions and improvements in SQL Server 2019
https://sqlblog.karaszi.com/scalar-functions-and-improvements-in-sql-server-2019/

Scalar functions in SQL server 2019, part 2
https://sqlblog.karaszi.com/scalar-functions-in-sql-server-2019-part-2/

Match those types!
https://sqlblog.karaszi.com/match-those-types/

AddWithValue is Evil
https://www.dbdelta.com/addwithvalue-is-evil/

Lesson: Query Store

- What is Query Store?
- Enabling Query Store
- Configuring Query Store
- Accessing Query Store Data
- Forcing Query Execution Plans

What is Query Store?

- Stores queries, plans, and run-time statistics
- Highlights database pattern usage
- Helps troubleshoot query performance issues
- Compatible with on-premises and Azure SQL databases

Monitoring performance by using the Query Store
https://docs.microsoft.com/en-us/sql/relational-databases/performance/monitoring-performance-by-using-the-query-store

Enabling Query Store

- Switch on using ALTER DATABASE
  - `ALTER DATABASE dbname SET QUERY_STORE = ON;`
- To use SSMS, right-click database, click Properties, and then on the Query Store page, change Operation Mode
- Cannot be turned on for system databases
- Rumored to be on by default in SQL Server 2022
  - For new databases
- Should we turn it on?
  - Run sp_BlitzCache, check 2:nd resultset.
  - If we have priority 1 warning, Query Store might not be for us.

Best practices with Query Store
https://docs.microsoft.com/en-us/sql/relational-databases/performance/best-practice-with-the-query-store

Query Store Best Practices
https://www.sqlskills.com/blogs/erin/query-store-best-practices/

Configuring Query Store

- View Query Store options using sys.database_query_store_options
- Configure Query Store options using ALTER DATABASE SET QUERY STORE (OPTION = VALUE)
- View and configure Query Store options using SSMS

Accessing Query Store data

- Access query stored data through catalog views or SSMS
- SSMS:
  - Regressed Queries
  - Overall Resource Consumption
  - Top Resource Consuming Queries
  - Tracked Queries
  - …
- Consider using Erik Darling's sp_QuickieStore

Demo Query store

The SQL Server 2016 Query Store: Built-in Reporting
https://www.red-gate.com/simple-talk/databases/sql-server/database-administration-sql-server/the-sql-server-2016-query-store-built-in-reporting/

Does Query Store's "Regression" Always Catch Nasty Parameter Sniffing?
https://www.littlekendra.com/2016/01/21/query-store-regression-parameter-sniffing/

Introducing sp_QuickieStore: Find Your Worst Queries In Query Store
https://www.erikdarlingdata.com/sp_quickiestore/introducing-sp_quickiestore-find-your-worst-queries-in-query-store-fast/

List of blog posts, including some that give tips on using sp_QuickieStore for various purposes
https://www.brentozar.com/archive/2022/10/erik-darlings-month-of-free-tools-training/

Forcing Query Execution Plans

- SSMS:
  - Click Force Plan button when viewing a query plan in the Query Store
  - Click Unforce Plan button to undo
- Transact-SQL:
  - Use sp_querystore_force_plan to force a plan
  - Use sp_querystore_unforce_plan to unforce a plan
- Regularly check the performance of forced plans

sp_query_store_force_plan (Transact-SQL)
https://docs.microsoft.com/en-us/sql/relational-databases/system-stored-procedures/sp-query-store-force-plan-transact-sql

Why You Need Query Store, Part II: Plan forcing
https://www.sqlskills.com/blogs/erin/why-you-need-query-store-part-ii-plan-forcing/

Lesson: Automatic tuning

- What is Automatic Tuning?
- sys.dm_db_tuning_recommendations
- Plan choice correction

What is Automatic Tuning?

- Requires Enterprise Edition
- Changes in query execution plans that impact performance can be time consuming to find and fix
- Plan choice regression is a recompiled plan that results in poorer performance
- Automatic tuning
  - Collects data about query performance and execution plans
  - Finds links between plan changes and reduced performance
  - Generates a script to force the previous plan to be used
  - Can be configured to automatically apply the previous script

Automatic tuning
https://docs.microsoft.com/en-us/sql/relational-databases/automatic-tuning/automatic-tuning

Understanding automatic tuning in SQL Server 2017
https://www.sqlshack.com/understanding-automatic-tuning-in-sql-server-2017/

sys.dm_db_tuning_recommendations

- sys.dm_db_tuning_recommendations returns:
  - A score between 0 and 100 to indicate the anticipated performance impact
  - A JSON string containing the recommendations, including:
    - Metrics used to identify plan choice regression
    - Commands used to apply the recommendation
- Contents of sys.dm_db_tuning_recommendations are retained until the instance is restarted

sys.dm_db_tuning_recommendations (Transact-SQL)
https://docs.microsoft.com/en-us/sql/relational-databases/system-dynamic-management-views/sys-dm-db-tuning-recommendations-transact-sql

Sql Server automatic tuning and sys.dm_db_tuning_recommendations
https://www.scarydba.com/2017/12/26/sql-server-automatic-tuning-and-sys-dm_db_tuning_recommendations/

Plan choice correction

- Automatic plan choice correction:
  - Enabled at the database level
  - Plan choice recommendations are forced when:
    - CPU time gain > 10 seconds
    - Fewer errors in the recommended plan
  - Results are monitored
    - If performance does not improve, the plan is recompiled
  - Manual plan choice correction
    - Manually apply recommendations from the DMV
    - Manually monitor performance and act on findings
- Azure SQL Database also has auto create and drop index

Demo Automatic tuning

Lab 8: Plan caching and recompilation

- Ex 1: Fixing stored procedure with plan issues
- Ex 2: Server is gradually slowing down

Estimated Time: 30 minutes