IDERA® | Database Administration

DBA automation in SQL Server and Azure SQL Database

DBA AUTOMATION IN SQL SERVER AND AZURE SQL DATABASE

Database administrators (DBAs) have been hearing about how robots in SQL Server and Azure are going to be taking their jobs by automating performance tuning. However, most of these claims are just marketing hype that Microsoft is using to sell the latest versions of their products. This article blows the marketing smoke away on these solutions by explaining their latest features and identifying the ones that are ready for prime time. This information will better prepare you for discussions about whether you should upgrade and how you can use the robots in SQL Server and Azure to do your job better.

BATCH MODE EXECUTION

The reporting style for queries in SQL Server 2017 involves lots of reads and processing. In comparison, SQL Server 2019 performs the same number of reads faster and with less processing. The primary reason for this difference is that v2017 plans have two parallelism zones, whereas v2019 plans only use one.

Version 2017 and earlier also use row-mode execution such that each plan operator is a mini-program that runs in isolation and passes data to the next operator as rows. Version 2019 uses columnstore indexes such that each column has its own index. This change provides many benefits such as smaller query results, since v2019 queries only store unique values. Losing duplication also allows v2019 to group, sum, and sort results more with less processing.

The result of columnstore indexes is that columnstore tables get batch mode in SQL Server 2019 Standard Edition, whereas rowstore tables only get it in Enterprise Edition. This feature is most useful for reports, but it makes less sense for online transaction processing (OLTP).

TVF INTERLEAVED EXECUTION

Multi-statement table-valued functions (MSTVTs) create several problems for SQL Server, such as row estimates that are too inaccurate to be useful. Version 2017 resolves this problem with interleaved execution, which runs the MSTVFs first and then passes the row estimates to the rest of the plan. As a result, a query that uses MSTVTs takes 42 seconds to execute in v2016, while the same query runs in seven seconds in v2017. Interleaved execution notifies the rest of the plan that it will receive many rows, thus improving memory grant accuracy.

SCALAR FUNCTION INLINING

Scalar function inlining is one of the most ambitious new features in SQL Server 2019. This feature takes scalar user-defined functions (UDFs) that run on a row-by-row basis and rewrites them into set-based functions. It also rewrites the queries that call those functions to use the set-based versions of these functions.

Other new features in v2019 like big data clusters, common language runtime (CLR), Hekaton, and Polybase are more complex and difficult to implement. They are also less useful since they require users to change their code, tables and indexes. Scalar function inlining is not supposed to require the user to change anything, although Microsoft keeps disabling it as they continue finding and fixing bugs with it. This function may not be useful to you since it does not necessarily speed up queries, even when it is available.

ADAPTIVE MEMORY GRANTS

DBAs often create poor queries and indexes, making it difficult for SQL Server to estimate query results accurately. It tracks the frequency of spills for query plans, although it does nothing with this information. You can illustrate this problem by building a stored procedure that is vulnerable to sniffing and running it repeatedly.

A test procedure run on v2017 spilled 83,481 pages to TempDB. The same query had 167,303 spills on the first run with v2019, but none on a second run with @Rep = 1. This result shows the benefit of adaptive memory grants, which involves SQL Server reviewing its memory use after a query.

If the query spills to disk, SQL Server increases its memory grant for the next run. If it did not use its existing random-access memory (RAM), then SQL Server reduces its memory grant. However, a third run with @Rep = 2 shows a spill of 167,115 pages. That means SQL Server adapts its memory grants to the parameters of the last user parameters rather than remembering results. Adaptive grants for rowstore queries are therefore of questionable value at this time.

ADAPTIVE JOINS

A test of the adaptive join capability of SQL Server requires you to join two tables without using indexes. It makes sense to perform nested loops when the driving table returns only a few rows, such that each operation functions on one row. A hash join is better when the driving table returns many rows, which involves scanning the other table and checking all of its rows.

Adaptive joins rarely occur in SQL Server. They require a columnstore index in v2017, even if it is empty. They also work with rowstore indexes in v2019. Both versions require SELECT queries with joins that can work with nested loops or hash joins. Adaptive joins can be beneficial, although they are rarely game changers by themselves. Running queries with adaptive joins in succession with different @Reputation values can cause dramatically slower queries because of the way adaptive memory grants work in SQL Server.

AUTOMATIC TUNING

Automatic Tuning implies that SQL Server is improving your query, but that is not even close to what this feature does. It allows you to store multiple query plans by using Query Store. SQL Server can then measure their performance to determine which one is better. However, automatic tuning does not help if you never develop a good plan, or if the query changes at all.

SUMMARY

Microsoft is doing well with its latest versions of SQL Server, which can reduce its hosting costs for Azure if done right. This aim is crucial for making Azure more cost competitive with Amazon Relational Database Service (RDS), Aurora MySQL, and PostgreSQL besides other on-premises databases. Most organizations now require the Enterprise Edition of SQL Server, but Microsoft may need to make some features in this edition available in less expensive versions of SQL Server. This scenario is likely if open-source database servers begin competing with Microsoft.

IDERA'S SOLUTION

IDERA's SQL Diagnostic Manager for SQL Server

IDERA's SQL Diagnostic Manager for SQL Server helps database administrators to find and fix SQL Server performance problems in physical, virtual, and cloud environments. Unlike its competition, it provides effective scalability, advanced SQL query analysis and optimization, prescriptive analysis with corrective SQL scripts, powerful automated alert responses, broad PowerShell integration, complete customization, and extensive support for current and legacy SQL Server and Windows.

Start for FREE

IDERA

IDERA.com