

Comparison of Tools for Collecting Information About Query Performance in Microsoft SQL Server

Mitko Radoev*

Summary:

The query performance is a very important issue in every Database Management System. The slow execution of a query in an information system based on the online transaction processing (OLTP), for example, can prompt potential customers to give up making an order. The efforts of information system developers are aimed at achieving optimal performance. However, the actual results do not always go to plan.

It is therefore necessary to use some tools for collecting and analysing information for the execution of the queries. As a result of the findings of this analysis, actions to improve the queries' performance are recommended. The improvement of the performance, however, should be targeted mainly at the problematic queries.

There are many different resources and tools for collecting information and improving the performance of the queries in Microsoft SQL Server. This paper aims to present the capacity of these tools and make a comparison of them. It further outlines options for using them in view of collecting and analysing information about performance of the queries.

Key words: query performance, performance analysis, performance monitoring, performance tuning

JEL Classification: C8

1. Introduction

Everyone who has dealt with the query performance analysis is familiar with the tools that Microsoft SQL Server provides. Starting from Microsoft SQL Server 2005, many execution-related Dynamic Management Objects (Views, Functions, and Procedures) became available.

In SQL Server 2008 two new tools were introduced - Extended Events (Radoev, 2014) and Data Collector (Radoev, 2014).

The Query Store is a new feature that is added in the Microsoft SQL Server 2016 (Radoev, 2015).

The capabilities of these tools as well as their advantages or disadvantages will be presented in this study. Then they will be compared using a set of indicators and, on this basis, recommendations will be made for their use with regard to collecting and analysing information about query performance in different situations.

2. Dynamic Management Objects

In Microsoft SQL Server 2005 numerous new system views and functions were introduced. Direct or indirect information about query execution is provided mainly in two groups of such functions - Execution Related Dynamic Management Views

* Chief Assistant Professor, Department Information Technologies and Communications, Faculty Applied Informatics and Statistics, UNWE; e-mail: mradoev@unwe.bg

Articles

and Functions and SQL Server Operating System Related Dynamic Management Views.

The most important of the Execution Related Dynamic Management Views and Functions are:

- **sys.dm_exec_requests:** Returns information about each request that is executing within SQL Server;
- **sys.dm_exec_query_stats:** Returns aggregate performance statistics for cached query plans. The view contains one row per query statement within the cached plan;
- **sys.dm_exec_sql_text:** Returns the text of the query that is identified by the specified `sql_handle`;
- **sys.dm_exec_query_plan:** Returns the execution plan in XML format for the specified plan handle;
- **sys.dm_exec_text_query_plan:** Returns the execution plan in text format for the specified plan handle;
- **sys.dm_exec_cached_plans:** Returns a row for each query plan that is cached by SQL Server for faster query execution.
- In Microsoft SQL Server 2008 two new system views in this group were introduced:
- **sys.dm_exec_procedure_stats:** Returns aggregate performance statistics for cached stored procedures. The view returns one row for each cached stored procedure plan;
- **sys.dm_exec_trigger_stats:** Returns aggregate performance statistics for cached triggers. The view contains one row per trigger.
- In more recent versions of the Microsoft SQL Server, apart from adding new system views, there are some changes in the parameters of existing objects. For example, the system view `sys.dm_exec_query_stats` in version 2008 R2 includes the following new parameters:

Comparison of Tools for Collecting Information About Query Performance in Microsoft SQL Server

- **total_rows:** Total number of rows returned by the query;
- **last_rows:** Number of rows returned by the last execution of the query;
- **min_rows:** Minimum number of rows returned by the query;
- **max_rows:** Maximum number of rows returned by the query.

The most important of the SQL Server Operating System Related Dynamic Management Views are:

- **sys.dm_os_tasks:** Returns one row for each task that is active in the instance of SQL Server;
- **sys.dm_os_waiting_tasks:** Returns information about the wait queue of tasks that are waiting on some resource;
- **sys.dm_os_wait_stats:** Returns information about the waits including wait type.
- The system views and functions are sources of valuable information for the execution of the queries and the state of the system in general. They can be used to detect different types of problematic queries such as:
 - Queries that run for a long time;
 - Queries that require a lot of resources (CPU, memory, disk I/O). Such queries block server by consuming a significant portion of its resources and thus worsen the performance of the other queries that are waiting to be executed;
 - Queries that do not use too many resources but are executed very often for a short period of time.
- Use of the system views and functions has several disadvantages such as:
 - System views contain information only about queries with cached plans. Getting information about ad-hoc queries is problematic;
 - The information is stored temporarily, whereas the query execution plan is stored in the memory;

Articles

- It is not possible to compare the performance of the same query with different plans;
- A graphical representation of collected information is missing;
- They are only sources of information but they are not instruments for query performance tuning.

3. Extended Events

SQL Server Extended Events is a general event-handling system. It was first introduced in SQL Server 2008. Extended Events is a tool that can be configured to monitor and capture different types of data, including performance data.

When using Extended Events, collected data can be used for identifying the problematic queries. Events data is collected to a file. The data can be loaded into a table or query it directly, using this system function:

- `sys.fn_xe_file_target_read_file`

Another possible approach is to query the data from the following Extended Events related dynamic management views:

- `sys.dm_xe_sessions`
- `sys.dm_xe_session_targets`
- Discovering the slow-running queries is crucial because they have the biggest impact on the end-users' opinion about the information system. One can identify the slow running queries by analysing a session output file and sort the trace output on the duration data column. To identify queries that require a lot of resources, one can just change the sort order on physical reads data column, for instance.

Use of the Extended Events system has several advantages and drawbacks, such as:

- The collected information is stored permanently;
- The information is stored in XML format in an external file;

- Querying information from this external file is harder than from the system views;
- There is no information about plan changes;
- A graphical representation of collected information is missing;
- Extended Events are only sources of data but they cannot be used for query performance tuning.

4. Data Collector

The Data Collector is a component of Microsoft SQL Server that collects different sets of data. The Data Collector stores the collected data in a relational database or rather data warehouse. The Data Collector provides one point for data collection across database servers. This collection point can obtain data from different sources and is not limited to performance data (Microsoft Corporation, 2014).

Data Collector was introduced in SQL Server 2008 and is available in the Standard, Enterprise, Business Intelligence, and Web editions. SQL Server Integration Services must be installed.

System behaviour can be either tracked in real time in the form of graphs or captured as a log for offline analysis. The preferred mechanism on production servers is to use the log (Fritchey, 2013).

Data Collector collects performance information and saves it in a data warehouse. It has three built-in data collecting specifications designed to collect the most important performance metrics.

The information collected by default is about disk usage, query statistics, and server activity. Default data collectors are:

- **Disk Usage:** Collects information about disk space used by data and log files for all databases on the SQL Server instance, growth trends, and average day growth. By default, the Disk Usage data collector collects data every 60 seconds and uploads data into the data warehouse every 6 hours.

Articles

- **Query Statistics:** Collects information about query statistics, activity, execution plans and text on the SQL Server instance. It does not store all executed statements, only 10 worst performing ones.
- **Server Activity:** Collects information about resource usage (CPU, memory, disk I/O, and network), SQL Server waits, instance activity (batch requests, logouts, SQL compilations and recompilations, transactions, user connections, and logins), and operating system activity.

The Query Statistics data collection set has a single collection item that collects information about query statistics, activity, execution plans, and code on the SQL Server instance. By default, it collects data every 10 seconds and uploads it in the cached mode every 15 minutes (Petrovic, 2014).

SQL Server includes three built-in reports, corresponding to the predefined data collections:

- Disk Usage Summary Report
- Query Statistics History Report
- Server Activity History Report

Each of these reports provides more detailed information in the form of sub-reports. In addition to these standard reports, custom reports can be created using the SQL Server Business Intelligence Development Studio, or with any of the various Reporting

Comparison of Tools for Collecting Information About Query Performance in Microsoft SQL Server

Services tools (McGehee, 2008).

Query Statistics History Report presents the Top Queries that use the most resources, such as:

- Total CPU
- Duration
- Total I/O
- Physical Reads
- Logical Writes

An example of the Query Statistics History Report can be seen in Figure 1.

The table below the graph provides detailed statistical information about the execution of queries. This report contains two sub-reports:

- Query Details sub-report
- Query Plan Details sub-report

The Query Details sub-report presents the text of the query. It shows a graph of the query plans and average execution time. Query Plan Details sub-report contains the information about the query plan. There is also a hyperlink that opens a graphical representation of the execution plan for the current query. Query Statistics History Report is especially useful in analysing the performance of queries because it contains detailed information about each query using significant resources over a period of time.

Data Collector has some advantages and disadvantages, such as:

Performance - Top Queries by Average IO

This report identifies the queries currently residing in the plan cache that have caused the most IO on average each time they execute aggregated over the lifetime of the plan in the cache and is available only for plans currently in the cache.

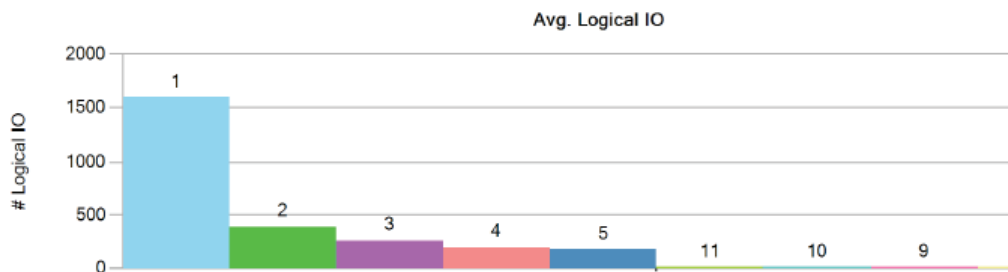


Fig. 1. An example of the Query Statistics History Report

Articles

- The collected information is stored permanently;
- The data is stored in two places: the msdb database and the management data warehouse;
- Querying information from these sources is quite natural;
- It does not store all executed queries, but only the 10 worst performing ones;
- Data Collector provides reports with graphical representation of collected data;
- Data Collector is only a source of data but it cannot be used for query performance tuning.

5. Query Store

The Query Store is a new feature, which is added in the Microsoft SQL Server 2016. It stores information about all executed queries, their execution plans and runtime statistics. This information is available through the Query Store catalogue views. At first glance, this is not something radically new because there was DMO's providing similar information in the previous versions of Microsoft SQL Server. However, the Dynamic Management Views retrieve information from procedure cache, where only the last execution plans are stored. Plans also get evicted from the cache when the server restarts or due to memory pressure. Given that the Query Store retains multiple execution plans per query, it can help diagnose performance problems that are related to plan changes.

Since the Query Store keeps history of execution plans and runtime statistics throughout query executions, there are many possibilities for query performance analysis. The Query Store can be used to:

- Determine the number of times a query was executed;
- Identify top queries by execution time, memory consumption, etc.;
- Easily find a plan performance regression;
- Audit the history of query plans for a given query;

- Analyse the resource (CPU, I/O, and Memory) usage for a particular database.

The Query Store further by opens up the opportunity to direct the query processor to use one of the stored execution plans for a query. This is referred to as plan forcing. Data changes all the time, so the Query Optimizer might generate a different plan for the same query. Sometimes the new plan is less efficient than some of the previous plans. Plan forcing can resolve a query performance regression caused by a plan change.

The Query Store can be accessed also through Microsoft SQL Server Management Studio. In the structure of the database with enabled Query Store there is a section named Query Store. This section contains several reports, presenting information from the Query Store:

- Regressed Queries;
- Overall Resource Consumption;
- Top Resource Consuming Queries;
- Tracked Queries.

An example of the Top Resource Consuming Queries Report is presented in Figure 2.

As a conclusion, the Query Store has the following advantages:

- The Query Store stores the information permanently on disk;
- The Query Store is integrated in Microsoft SQL Server and Management Studio;
- It stores data about all executed queries;
- It stores data about query plans changes;
- It provides user interface with graphical representation of collected data;
- The query execution plan can be forced in order to resolve a query performance regression;

6. Comparison of the tools

Examining different instruments of gathering information about the performance of queries could be defined the following characteristics:

- Permanent storage of data;

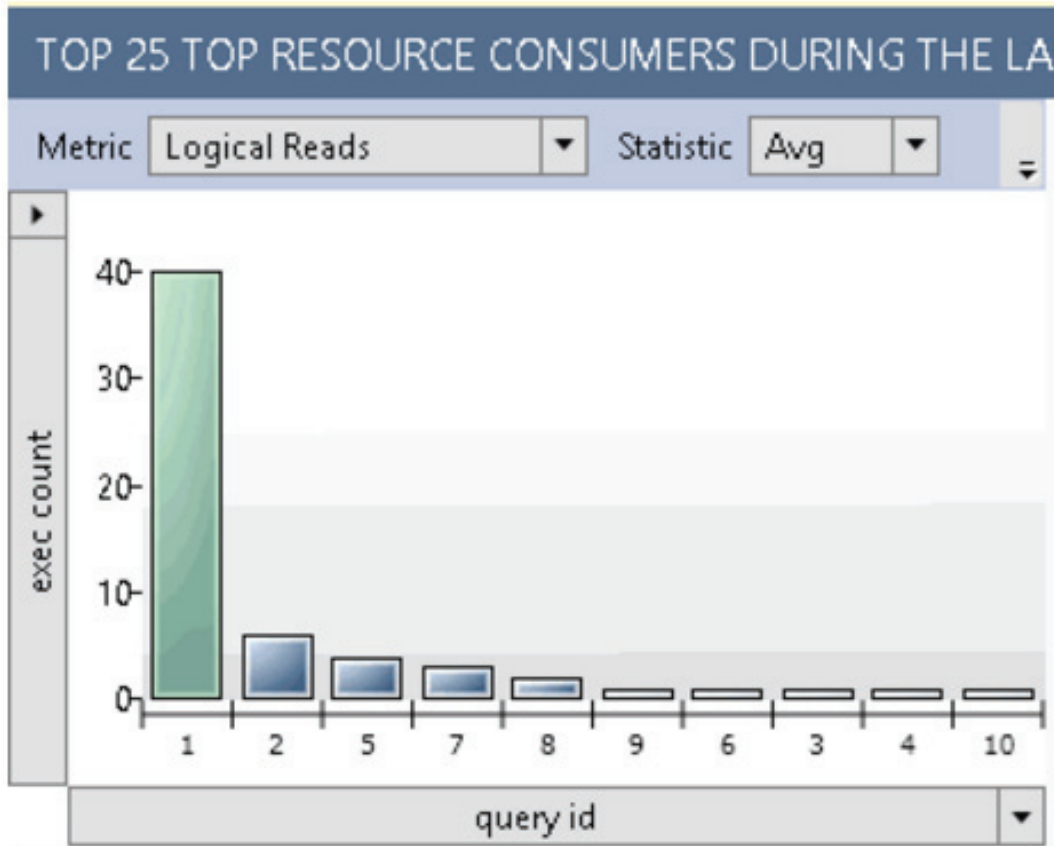


Fig. 2. An example of the Top Resource Consuming Queries Report

- Integration with SQL Server Management Studio;
- Collecting data about all queries;
- Collecting data about query execution plan changes;
- Graphical representation of collected data;
- Ability for query performance tuning. Using these indicators, the different instruments for collecting information about query performance in Microsoft SQL Server, which were presented above, were compared in the following table.

Table 1. Comparison of the tools for collecting information about query performance in Microsoft SQL Server

Characteristics	DMO's	Extended Events	Data Collector	Query Store
Permanent storage	no	yes	yes	yes
Integration with Management Studio	no	no	yes	yes
Collecting data about all queries	no	no	no	yes
Collecting data about plan changes	no	no	no	yes
Graphical representation	no	no	yes	yes
Ability for query performance tuning	no	no	no	yes

Conclusion

Over the years new tools for gathering information about query performance have been added in different versions of the Microsoft SQL Server. At the same time, some the tools available until then have been improved. Naturally, newer instruments offer more and better opportunities not only for collecting and analysing information but also provide for a direct impact on query execution. Each of the aforementioned tools would be useful for an occasional observation of a query. Query Store is the tool recommended for the comprehensive and thorough analysis of the performance of queries which provides for the adjustment of their performance. It is definitely superior to other tools on all indicators.

References

- Cunningham, C., 2014. Query Store - A New SQL Query Tuning Feature. [Online] Last accessed at <http://www.sqlpass.org/summit/2014/Sessions/Details.aspx?sid=7217>
- Delaney, K., 2013. Microsoft SQL Server 2012 Internals. s.l.:Microsoft Press.
- Fritchey, G., 2013. SQL Server 2012 Query Performance Tuning. 3rd ed. s.l.:Apress.
- Laar, E. v. d., 2015. The SQL Server 2016 Query Store: Overview and Architecture. [Online] Available at: <https://www.simple-talk.com/content/article.aspx?article=2308>
- McGehee, B., 2008. SQL Server 2008: Performance Data Collector. [Online] Last accessed at: <https://www.simple-talk.com/sql/learn-sql-server/sql-server-2008-performance-data-collector/>
- Microsoft Corporation, 2014. Data Collection. [Online] Last accessed at: <http://msdn.microsoft.com/en-us/library/bb677179.aspx>
- Microsoft Corporation, 2015. Monitoring Performance By Using the Query Store. [Online] Last accessed at: <https://msdn.microsoft.com/en-US/library/dn817826.aspx>
- Microsoft Corporation, n.d. Extended Events. [Online] Last accessed at: <http://technet.microsoft.com/en-us/library/bb630282.aspx>
- Petrovic, M., 2014. SQL Server performance monitoring with Data Collector. [Online] Last accessed at: <http://www.sqlshack.com/sql-server-performance-monitoring-data-collector/>
- Radoev, M., 2014. Using Information from Extended Events in Microsoft SQL Server for Identifying Problematic Queries. Sofia, 4th International Conference on Application of Information and Communication Technology and Statistics in Economy and Education (ICAICTSEE – 2014).
- Radoev, M., 2014. Using the Data Collector in Microsoft SQL Server for the query performance analysis. Sofia, 4th International Conference on Application of Information and Communication Technology and Statistics in Economy and Education (ICAICTSEE – 2014).
- Radoev, M., 2015. New Opportunities for a Query Performance Analysis in MS SQL Server 2016 using the Query Store. Sofia, 5th International Conference on Application of Information and Communication Technology and Statistics in Economy and Education (ICAICTSEE – 2015).
- Sunil Agarwal, B. B. K. E. J. T. K. C. B. P., 2009. Troubleshooting Performance Problems in SQL Server 2008. SQL Server Technical Article, March.
- Vieira, R., 2009. Professional Microsoft SQL Server 2008 Programming. s.l.:John Wiley & Sons.