QPI - Query Performance Insights

Last updated by | Vitor Tomaz | Feb 24, 2023 at 3:28 AM PST

Contents

Issue

This is a "How To" TSG or rather a pointer to the QPI - Query Performance Insights library on GitHub. The library is created and maintained by Jovan Popovic and other members from PG. Query Performance Insights (QPI) is a collection of useful scripts that enable you to find what is happening with your SQL Server 2016+ or Azure SQL Database (Single or Managed Instance). This is a set of helper views, functions, and procedures that wrap Query Store and Dynamic Management Objects.

Why do I need this kind of library?

SQL Server/Azure SQL Database provides a lot of views that we can use to analyze query performance (dynamic management object and Query Store views). However, sometimes it is hard to see what is happening in the database engine. There are DMVs and Query Store, but when we need to get the answer to the simple questions such as "Are there some queries that are currently blocked", "Did query performance change after I added an index?" or "How many IOPS do we use?", we need to dig into Query Store schema, think about every single report, or search for some useful query. Also, for most of the views, you would need to read several articles to understand how to interpret the results.

This is the reason why <the QPI author> have collected the most useful queries that find information from underlying system views and wrapped them in a set of useful views.

Example content on QPI

Getting information about your workload:

```
select * from qpi.db_queries;        -- All database queries recorded in Query Store

select * from qpi.queries;           -- Currently running queries
select * from qpi.bre;               -- Active Backup/Restore requests
select * from qpi.blocked_queries;   -- Information about the currently blocked queries
select * from qpi.query_locks;       -- Information about the locks that the queries are holding

select * from qpi.cpu_usage;         --  Information about CPU usage.
select * from qpi.memory;            --  Information about memory usage.
```

Getting the information about the system performance:

```
select * from qpi.sys_info;          --  Get CPU & memory
select * from qpi.volumes;           --  Get info about used and available space on the storage volumes

exec qpi.snapshot_file_stats;        --  Take the file statistics baseline
<run some query or workload>
select * from qpi.file_stats;        --  Get the file stats

exec qpi.snapshot_wait_stats;        --  Take the wait statistics baseline
<run some query or workload>
select * from qpi.wait_stats;        --  Get the wait stats

exec qpi.snapshot_perf_counters;     -- Take the performance counter baseline (required for some perf counters
<run some query or workload>
select * from qpi.perf_counters;     -- Get the perf counter values
```

Public Doc Reference

- QPI - Query Performance Insights
- Paul Randal:
  - Wait statistics library
  - Wait statistics - tell me where it hurts
  - How to examine IO subsystem latencies
- Erin Stellato: What Virtual Filestats Do, and Do Not, Tell You About I/O Latency
- Aaron Bertrand: Determine system memory
- Dimitri Furman and SqlCat team blog posts
- Tim Ford and Louis Davidson: Performance tunning with DMV
- Ajith Krishnan: Interpreting the counter values from sys.dm_os_performance_counters

How good have you found this content?