Find Top Consumers

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

Contents

Issue

This is a "How to" TSG for identifying the top resource-consuming queries in a database.

Investigation / Analysis

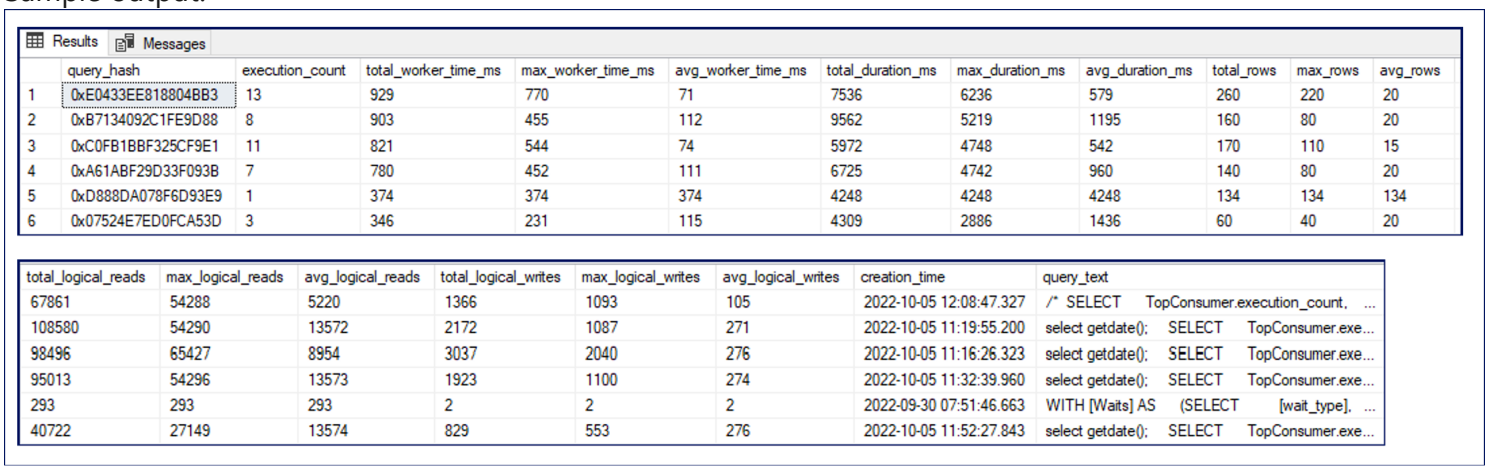
Identify the Top consumers

Find the Top consumers on the database. Change the query below to sort the data according to resource type:

- by worker time consumption for checking CPU
- by duration to check the longest running queries
- by logical reads to check the read IO consumption
- by logical writes to check the (logical) write IO consumption

```
SELECT TOP(20)
    qs.query_hash,
    sum(qs.execution_count) AS 'execution_count',

    sum(qs.total_worker_time)/1000 AS total_worker_time_ms,
    sum(qs.max_worker_time)/1000 AS max_worker_time_ms,
    (sum(qs.total_worker_time) / sum(qs.execution_count)) / 1000  AS avg_worker_time_ms,

    sum(qs.total_elapsed_time)/1000 AS total_duration_ms,
    sum(qs.max_elapsed_time)/1000 AS max_duration_ms,
    (sum(qs.total_elapsed_time) / sum(qs.execution_count)) / 1000 AS avg_duration_ms,

    sum(qs.total_rows) AS total_rows,
    sum(qs.max_rows) AS max_rows,
    sum(qs.total_rows) / sum(qs.execution_count) AS avg_rows,

    sum(qs.total_logical_reads) AS total_logical_reads,
    sum(qs.max_logical_reads) AS max_logical_reads,
    sum(qs.total_logical_reads) / sum(qs.execution_count) AS avg_logical_reads,

    sum(qs.total_logical_writes) AS total_logical_writes,
    sum(qs.max_logical_writes) AS max_logical_writes,
    sum(qs.total_logical_writes) / sum(qs.execution_count) AS avg_logical_writes,

    min(qs.creation_time) AS creation_time,
    min(t.text) AS [query_text]

FROM sys.dm_exec_query_stats AS qs --WITH (NOLOCK)
CROSS APPLY sys.dm_exec_sql_text(sql_handle) AS t
GROUP BY qs.query_hash

ORDER BY sum(qs.total_worker_time) DESC  -- uncomment this line for top CPU
--ORDER BY sum(qs.total_elapsed_time) DESC  -- uncomment this line for top Duration
--ORDER BY sum(qs.total_logical_reads) DESC  -- uncomment this line for top Read I/O
--ORDER BY sum(qs.total_logical_writes) DESC  -- uncomment this line for top Write I/O
```

Sample output:



Results | Messages

	query_hash	execution_count	total_worker_time_ms	max_worker_time_ms	avg_worker_time_ms	total_duration_ms	max_duration_ms	avg_duration_ms	total_rows	max_rows	avg_rows
1	0xE0433EE818804BB3	13	929	770	71	7536	6236	579	260	220	20
2	0xB7134092C1FE9D88	8	903	455	112	9562	5219	1195	160	80	20
3	0xC0FB1BBF325CF9E1	11	821	544	74	5972	4748	542	170	110	15
4	0xA61ABF29D33F093B	7	780	452	111	6725	4742	960	140	80	20
5	0xD888DA078F6D93E9	1	374	374	374	4248	4248	4248	134	134	134
6	0x07524E7ED0FCA53D	3	346	231	115	4309	2886	1436	60	40	20

total_logical_reads	max_logical_reads	avg_logical_reads	total_logical_writes	max_logical_writes	avg_logical_writes	creation_time	query_text
67861	54288	5220	1366	1093	105	2022-10-05 12:08:47.327	/* SELECT TopConsumer.execution_count, ...
108580	54290	13572	2172	1087	271	2022-10-05 11:19:55.200	select getdate(); SELECT TopConsumer.exe...
98496	65427	8954	3037	2040	276	2022-10-05 11:16:26.323	select getdate(); SELECT TopConsumer.exe...
95013	54296	13573	1923	1100	274	2022-10-05 11:32:39.960	select getdate(); SELECT TopConsumer.exe...
293	293	293	2	2	2	2022-09-30 07:51:46.663	WITH [Waits] AS (SELECT [wait_type], ...
40722	27149	13574	829	553	276	2022-10-05 11:52:27.843	select getdate(); SELECT TopConsumer.exe...

Retrieve the corresponding execution plan

If you simply need the ***estimated execution plan*** of a query, you can get it with the following query - set the query_hash to a value that you had retrieved with the query from above:

```
SELECT TOP 20 qs.query_hash, t.text, qp.query_plan
FROM sys.dm_exec_query_stats AS qs --WITH (NOLOCK)
CROSS APPLY sys.dm_exec_sql_text(sql_handle) as t
CROSS APPLY sys.dm_exec_query_plan(plan_handle) as qp
WHERE qs.query_hash = 0xE0433EE818804BB3
```

Sample output:

Results | Messages

	query_hash	text	query_plan
1	0xE0433EE818804BB3	/* SELECT TopConsumer.execution_count, TopConsumer.avg_worker_time_ms as AVG_C...	<ShowPlanXML xmlns="http://schemas.microsoft.com/sqlserver/2004/07/showplan...
2	0xE0433EE818804BB3	/* SELECT TopConsumer.execution_count, TopConsumer.avg_worker_time_ms as AVG_C...	<ShowPlanXML xmlns="http://schemas.microsoft.com/sqlserver/2004/07/showplan...
3	0xE0433EE818804BB3	/* SELECT TopConsumer.execution_count, TopConsumer.avg_worker_time_ms as AVG_C...	<ShowPlanXML xmlns="http://schemas.microsoft.com/sqlserver/2004/07/showplan...

If you need the ***actual execution plan*** of a query, refer to article How to Capture the Actual Execution Plan. This will be helpful if you are investigating a plan regression and you suspect the values from the estimated execution plan to be incorrect.

Public Doc Reference

- sys.dm_exec_query_stats (Transact-SQL)

Classification

Root cause path -

Workload performance/User-issue/error/Throttling errors/resource limit

How good have you found this content?