Adhoc workloads

Last updated by | Vitor Tomaz | Feb 24, 2023 at 3:32 AM PST

Contents

Issue

In some cases you might see a huge plan cache when troubleshooting performance related issues. Note that the plan cache resides in memory, that is shared with other memory clerks. Like so, managing the plan cache is extremely important. Adhoc queries can in some cases contribute to a bloated plan cache. Note that execution plan Adhoc queries will generally be used only once. In other words, the execution plan will not be reused.

Note that is normal that you have Adhoc queries on a database - they are associated with some quick checks done by application support professionals, DBA`s, etc. For example: a DBA checks the structure of a table and records in order to come up with a good index, or a Application support professional does a SELECT count(*) on a table to see how many records he has on a log table, or he searches for a specific entry on a log table.

What would not be considered normal is an application that relies heavily on AdHoc queries. These queries can come in different ways:

- stored procedures that uses dynamic sql, generating different queries depending on the input values
- an application that builds different queries, depending on the input values given by the user

Investigation/Analysis

You can use the query below to check the size of each Memory Clerk. Look for CACHESTORE_SQLCP - AdHoc query plans.

```
 SELECT TOP(10) mc.[type] AS [Memory Clerk Type], CAST((SUM(mc.pages kb)/1024.0) AS DECIMAL (15,2)) AS [Memory Usage (MB)] FROM sys.dm_os_memory_clerks AS mc WITH (NOLOCK) GROUP BY mc.[type] DER BY SUM(mc.pages_kb) DESC OPTION (RECOMPILE);
```

To check for single used Adhoc query plans (returns the text and size on cache)

```
 SELECT TOP(50) DB NAME(t.[dbid]) AS [Database Name], t.[text] AS [Query Text], cp.objtype AS [Object Type], cp.cacheobjtype AS [Cache Object Type], cp.size in bytes/1024 AS [Plan Size in KB] FROM sys.dm exec cached plans AS cp WITH (NOLOCK) CROSS APPLY sys.dm exec sql text(plan handle) AS t WHERE cp.cacheobjtype = N'Compiled Plan' AND cp.objtype IN (N'Adhoc', N'Prepared') AND cp.usecounts = 1 ORDER BY cp.size_in_bytes DESC, DB_NAME(t.[dbid]) OPTION (RECOMPILE);
```

Mitigation

The customer can enable the OPTIMIZE_FOR_AD_HOC_WORKLOADS setting to help reduce the plan cache usage. In short what this setting will do is to reduce the size of the plan for AdHoc queries, by keeping only a small compiled plan stub when the query is executed for the first time (on a second execution, it will store a full

plan).
But note that this setting is not the optimal scenario (even if small, a plan is stored on the plan cache). The goal must always be to avoid AdHoc queries and use stored procedures, functions, etc.

How good have you found this content?