SQL SERVER Performance Tuning Notes

http://msbiskills.com/

What is a Plan Cache & Why it is important?

Let’s go through some basic notes about Plan Cache -

1. **Well SQL Server creates execution plans for queries and stores them in a location of memory called Plan cache. Plan cache uses some part of the memory which is allocated to Buffer Pool.**

2. **Prior to SQL 2005 it was called Procedure Cache.**

3. **Now the question is that why we are saving these execution plans- See for SQL Server it expensive to go through all the steps that are required to generate execution plans each time the query is submitted. While SQL Server can do all this in less than a millisecond, depending on the query it can take seconds or even minutes to create an execution plan, so SQL Server will keep plans in Plan cache for further usage.**

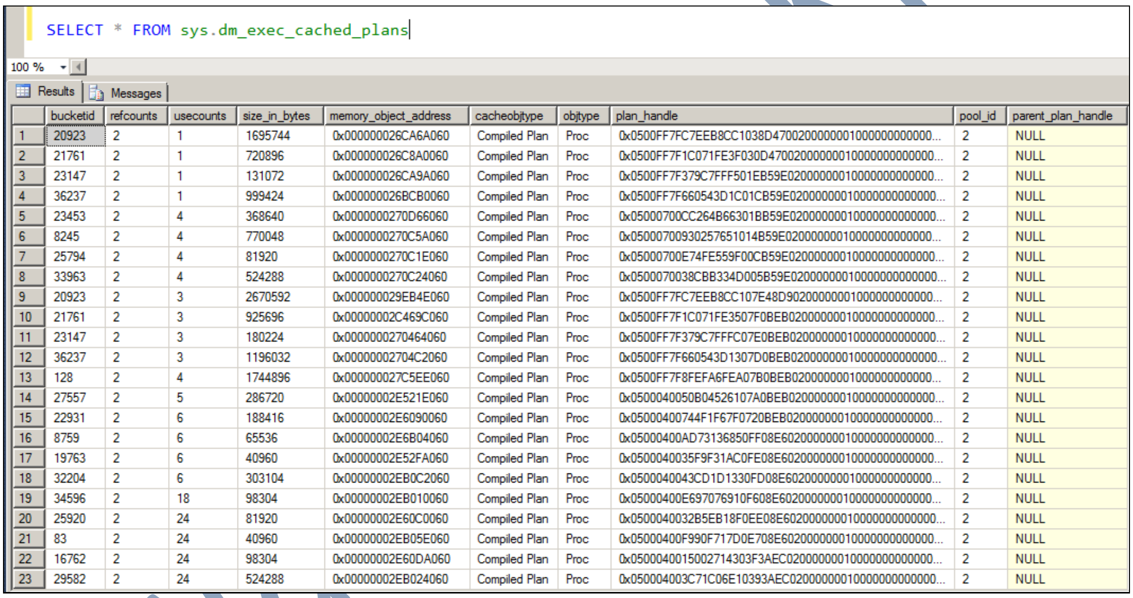
4. **Execution plans are stored in 4 different memory areas.**
   a. **CACHESTORE_OBJCP – Stored Procedures, Triggers and Functions**
   b. **CACHESTORE_SQLCP – Used for Ad-hoc SQL queries**
   c. **CACHESTORE_PHDR – Used for View, Constraints and defaults.**
   d. **CACHESTORE_XPROC – Used for Extended Stored Procedures.**

5. **Each plan cache store contains a hash table with full of buckets.**

6. **Plans are stored in each bucket based on simple hash algorithm.**
    a. **The algorithm is (Object_id * Database_Id) mod (Hash Table Size)**

7. **Whenever you get a query to execute SQL Server check the plan cache if the execution plans exists in the plan cache. It checks this using Hash Key + Cache Key**

8. **How to check whether query execution plan exists in the plan cache or not**
    a. `SELECT * FROM sys.dm_exec_cached_plans`



```
SELECT * FROM sys.dm_exec_cached_plans
```

	bucketid	refcounts	usecounts	size_in_bytes	memory_object_address	cacheobjtype	objtype	plan_handle	pool_id	parent_plan_handle
1	20923	2	1	1695744	0x000000026CA6A060	Compiled Plan	Proc	0x0500FF7FC7EEB8CC1038D470020000000100000000000...	2	NULL
2	21761	2	1	720896	0x000000026C8A0060	Compiled Plan	Proc	0x0500FF7F1C071FE3F030D4700200000001000000000000...	2	NULL
3	23147	2	1	131072	0x000000026CA9A060	Compiled Plan	Proc	0x0500FF7F379C7FFF501EB59E020000000100000000000000...	2	NULL
4	36237	2	1	999424	0x000000026BCB0060	Compiled Plan	Proc	0x0500FF7F660543D1C01CB59E02000000010000000000000...	2	NULL
5	23453	2	4	368640	0x000000270D66060	Compiled Plan	Proc	0x05000700CC264B66301BB59E02000000010000000000000...	2	NULL
6	8245	2	4	770048	0x0000000270C5A060	Compiled Plan	Proc	0x050007009302576510 14B59E02000000010000000000000...	2	NULL
7	25794	2	4	81920	0x0000000270C1E060	Compiled Plan	Proc	0x05000700E74FE559F00CB59E02000000010000000000000...	2	NULL
8	33963	2	4	524288	0x0000000270C24060	Compiled Plan	Proc	0x0500070038CBB334D005B59E02000000010000000000000...	2	NULL
9	20923	2	3	2670592	0x000000029EB4E060	Compiled Plan	Proc	0x0500FF7FC7EEB8CC107E48D9020000000100000000000...	2	NULL
10	21761	2	3	925696	0x00000002C469C060	Compiled Plan	Proc	0x0500FF7F1C071FE3507F0BEB02000000010000000000000...	2	NULL
11	23147	2	3	180224	0x000000270464060	Compiled Plan	Proc	0x0500FF7F379C7FFFC07E0BEB02000000010000000000000...	2	NULL
12	36237	2	3	1196032	0x00000002704C2060	Compiled Plan	Proc	0x0500FF7F660543D1307D0BEB02000000010000000000000...	2	NULL
13	128	2	4	1744896	0x000000027C5EE060	Compiled Plan	Proc	0x0500FF7F8FEFA6FEA07B0BEB0200000001000000000000...	2	NULL
14	27557	2	5	286720	0x00000002E521E060	Compiled Plan	Proc	0x050004050B04526107A0BEB02000000010000000000000...	2	NULL
15	22931	2	6	188416	0x00000002E6090060	Compiled Plan	Proc	0x05000400744F1F67F0720BEB0200000001000000000000...	2	NULL
16	8759	2	6	65536	0x00000002E6B04060	Compiled Plan	Proc	0x05000400AD73136850FF08E60200000001000000000000...	2	NULL
17	19763	2	6	40960	0x00000002E52FA060	Compiled Plan	Proc	0x050004035F9F31AC0FE08E60200000001000000000000...	2	NULL
18	32204	2	6	303104	0x00000002EB0C2060	Compiled Plan	Proc	0x050004043CD1D1330FD08E6020000000100000000000...	2	NULL
19	34596	2	18	98304	0x00000002EB010060	Compiled Plan	Proc	0x05000400E697076910F608E6020000000100000000000...	2	NULL
20	25920	2	24	81920	0x00000002E60C0060	Compiled Plan	Proc	0x050004032B5EB18F0EE08E60200000001000000000000...	2	NULL
21	83	2	24	40960	0x00000002EB05E060	Compiled Plan	Proc	0x05000400F990F717D0E708E60200000001000000000000...	2	NULL
22	16762	2	24	98304	0x00000002E60DA060	Compiled Plan	Proc	0x050004001500271430 3F3AEC0200000001000000000000...	2	NULL
23	29582	2	24	524288	0x00000002EB024060	Compiled Plan	Proc	0x050004003C71C06E10393AEC0200000001000000000000...	2	NULL

9. **It is one of the great strengths of SQL Server, since it reduces the expense of creating plans.**

10. **Please note that SQL Server does not keep execution plans in memory forever.**

11. **Compilation means Plan creation. You will get a new plan.**

12. Recompilation means existing plan cannot be used any more so a new one is made.

13. Recompilations happens in below cases- (Not all)

   a. If there are any schema related changes
   b. sp_recompile on a table or view
   c. Changing the set options with in the execution of the query
   d. Changing/updating/Deleting Index
   e. Updating Statistics manually or via automated means
   f. Stale Statistics deleted – 20% of rows + 500 new rows added.
   g. Manual recompile
   h. With Recompile Option in a stored procedure. 2 types of recompile options are available – Statement level (Recompile query hint @ statement level) and procedure level (With Recompile).

14. Let's say you don't want to recompile. There are some ways to achieve this.

   a. Trivial Plan
   b. If you have used OPTION hint KEEPFIXED Plan
   c. Automatic statistics updates have been disabled

15. How to clear Plan Cache

```
DBCC FREEPROCCACHE
```

   a. Removes all elements from the plan cache, removes a specific plan from the plan cache by specifying a plan handle or SQL handle, or removes all cache entries associated with a specified resource pool.

```
USE AdventureWorks2012;
GO
SELECT * FROM Person.Address;
GO
SELECT plan_handle, st.text
FROM sys.dm_exec_cached_plans
CROSS APPLY sys.dm_exec_sql_text(plan_handle) AS st
WHERE text LIKE N'SELECT * FROM Person.Address%';
GO
```