# Things You Can Find in the Plan Cache



#### ABOUT ME

- > Name: Matan Yungman
- > Role: Technical Evangelist at Madeira
- > Website: www.madeirasql.com
- > Mail: matan@madeira.co.il
- > Twitter: @MatanYungman
- > Podcast: <u>www.sqlserverradio.co.il</u>

www.sqlserverradio.com



 $\triangleright$ 





- > What is the Plan Cache
- > Ways to explore it
  - And what we can find while exploring
- > How to successfully leverage it
- > Querying it to identify problematic queries
- > Querying Execution Plan XML







#### THE PROCESS IS EXPENSIVE

- > Query Execution Time
- ≻ CPU
- > Memory

# The Solution: Save Plan for Future Reuse



#### Our Mission: Maximize Reuse While Fixing The Cases Where Reuse Is Not Good

#### THE PLAN CACHE

- > Built from 4 cache stores
  - CACHESTORE\_OBJCP Object Plans
  - CACHESTORE\_SQLCP Ad hoc and Prepared Plans
  - CACHESTORE\_PHDR Bound Trees
  - CACHESTORE\_XPROC Extended Procedures
- Steals its memory from the buffer pool
- > Each store is basically a hash table
- > Age-out algorithm



## **PIVING IN - VIEWS**

- > sys.dm\_exec\_cached\_plans
  - One row per batch
  - Use counts
  - Object type, Cache object type
  - Plan\_handle
- > sys.dm\_exec\_query\_stats
  - One row per statement in a batch
  - Execution count, Plan generation count
  - IO, Duration
  - Plan\_handle, Sql\_handle
  - Query\_hash, Query\_plan\_hash





# **DIVING IN - FUNCTIONS**



- > sys.dm\_exec\_sql\_text
  - Pass plan\_handle/sql\_handle to get batch text
  - Use offsets to get query level text
- > sys.dm\_exec\_query\_plan
  - Use to get the query plan
  - Returns XML plan of all the statements in a batch
- > sys.dm\_exec\_text\_query\_plan
  - Use to get the plan at the statement level
  - Returns nvarchar(max). Cast it to XML



#### CACHE LOOKUP

- > Objects (Stored Procedures, Functions, Triggers)
  - (ObjectID \* DB\_ID) % (Hash Table Size)
- > Ad-hoc and prepared statements
  - ObjectID = Query Text Hash
  - Exact text match is needed
- > Also for reuse to be possible:
  - Set Options
  - User ID (Schema ID)
  - DB\_ID
  - And More...



## **COMMON PROBLEMS: AD-HOC**

- > Lots of plans with use count of 1
- > Not necessarily user queries
  - ADO.Net, EXEC(), System Queries
- > How to identify:
  - Search for ad-hoc plans with use count = 1



## COMMON PROBLEMS: AP-HOC

- > Optimization options:
  - Optimize for ad hoc workloads
  - Switch to parameterized methods
  - Turn on Forced Parameterization
    - Try to test it's appropriate for your system
    - Can also be used selectively through plan guides



**RECOMMENDED QUERY #1** SELECT SUM(CAST(size in bytes BIGINT))/1024.0/1024.0 AS SizeInMB, COUNT(\*) AS PlanCount FROM sys.dm exec cached plans WHERE objtype = 'Adhoc' AND cacheobjtype = 'Compiled Plan' AND usecounts = 1



# **COMMON APPLICATION PROBLEMS**

#### > Nhibernate

- String parameter lengths are supplied according to each specific execution
- Solution: Nhibernate "prepare\_sql" parameter
- Linq To SQL & Entity Framework
  - Same as Nhibernate
  - Solution: Upgrade to Dot Net 4.0



# **COMMON APPLICATION PROBLEMS**

#### > Simple ADO.Net queries

- Act as Ad-hoc queries
- Optimization options:
  - Optimize for ad-hoc workloads option
  - ADO.Net parameterized queries (supply length for strings)
  - Forced parameterization



## QUEBY\_HASH & QUEBY\_PLAN\_HASH

- Can help identify potentially reusable queries
- Recommended query #2:

SELECT query\_hash,COUNT(\*) AS PlanCount FROM sys.dm\_exec\_query\_stats GROUP BY query\_hash HAVING COUNT(\*) > 1



## **AUTOMATIC PLAN EVICTION**

- > Age-out algorithm based on:
  - Plan generation cost
  - Usage
- > On memory pressure, all plan costs are decreased by 1
- Plans with cost = 0 can be evicted
- > Ad-hoc plans always enter with cost = 0
  - Cost incremented by 1 on each reuse
- For other plans
  - Cost is reset to the original value on reuse
- Ad-hoc max cost = 16
- Prepared max cost = 256
- No max cost for stored procedures

#### Madeira SQL Server Services

#### MANUAL PLAN EVICTION

- > DBCC FREEPROCCACHE
  - Clears all plans
- > Can also be more specific
  - plan\_handle / sql\_handle / pool\_name
- > DBCC FREESYSTEMCACHE
  - Specific cache store: 'SQL Plans' / 'Object Plans'
- > Undocumented
  - DBCC FLUSHPROCINDB Specific database



#### SEARCHING FOR PROBLEMATIC QUERIES

- > We can answer questions like:
  - Which queries are most IO/CPU/Time intensive?
    - Absolute numbers or per run
  - Which database is the most resource intensive?
  - What is the value a plan was optimized for?
  - Which queries/objects are frequently recompiled?
  - And much more..



#### THE RECOMPILE DILEMMA



- > Plan is optimized for first execution's parameters
- > Local variable values not known to the optimizer
- > Possible solution: Recompile
  - Stored procedure level:
    - Create Procedure..With Recompile
    - Exec..With Recompile
    - sp\_recompile
  - Statement level:
    - Option (Recompile)

> As we saw, it comes with a performance penalty



#### But there's a problem

#### Recompile Ruins the Statistics We Rely On When Querying the Plan Cache!

#### THE RECOMPILE DILEMMA

- > Create Procedure..With Recompile means:
  - "Generate new plan on each run and don't cache it"
- > Exec..With Recompile means:
  - "Generate new plan for this run and don't cache it"
- > Option (Rcompile) means:
  - Generate new plan on each run
  - Set execution\_count to 1

– Increase plan\_generation\_num by 1
Madeira
SQL Server Services

#### **OTHER SOLUTIONS**

- > Local variable problem?
  - Pass the values as parameters from the outside
  - Use Dynamic-SQL (preferably sp\_executesql)
- > Parameter Sniffing problem?
  - Multiple procedures for different cardinalities
  - Optimize for (@param = value)
  - Optimize for unknown / parameter masking
  - Plan freezing
    - sp\_create\_plan\_guide\_from\_handle
  - Option (use plan)
    - Can also be done with a plan guide



## **QUERYING EXECUTION PLAN XML**

- Can answer questions like:
  - Which plans suffer from missing indexes?
  - Which plans suffer from Implicit\_Conversion?
  - Which plans use a Clustered Index Scan?
  - Which plans have inaccurate row count estimation?
  - And more..



#### SUMMARY

- > Reuse is almost always a good thing
- > Try to prevent Plan Cache bloat
- > Remember the effects of Recompile
- > The Plan Cache is your workload Get to know it
- > Scripts:

– madeirasql.com/things-you-can-find-sqlbits



#### **Questions?**

#### Thank you !



