.Net Rocks!
Visual Studio 2012
Launch Road Trip

Optimizing Procedural Code

Kimberly L. Tripp
Kimberly@SQLskills.com
Kimberly L. Tripp

Consultant/Trainer/Speaker/Writer
President/Founder, SY Solutions, Inc.
- e-mail: Kimberly@SQLskills.com
- blog: http://www.SQLskills.com/blogs/Kimberly
- Twitter: @KimberlyLTripp

Author/Instructor for SQL Server Immersion Events
Instructor for multiple rotations of both the SQL MCM & SharePoint MCM
Author/Manager of SQL Server 2005 Launch Content, co-author/Manager for SQL Server 2008 Jumpstart Content, Author/Speaker at TechEd, SQL Connections, SQLPASS, ITForum, Conference Co-chair for SQL Connections
Author of several SQL Server Whitepapers on MSDN/TechNet: Partitioning, Snapshot Isolation, Manageability, SQLCLR for DBAs
Author/Presenter for more than 25 online webcasts on MSDN and TechNet (two series and other individual webcasts)
Co-author MSPress Title: SQL Server 2008 Internals, the SQL Server MVP Project (1 & 2), and SQL Server 2000 High Availability
Presenter/Technical Manager for SQL Server HOL DVDs

I love this stuff... feel free to ask questions!

©SQLskills.com
http://www.SQLskills.com
SQLskills

- Team of world-renowned SQL Server experts
  - Paul S. Randal @PaulRandal
  - Jonathan Kehayias @SQLPoolBoy
  - Joseph Sack @JosephSack
  - Glenn Berry @GlennAlanBerry
  - Erin Stellato @ErinStellato
- Instructor-led training ⇒ Immersion Events
- Online training ⇒ Pluralsight (http://pluralsight.com/)
- Consulting: health checks, hardware, performance
- Remote DBA
- Become a SQLskills Insider
Overview

- Back to the basics...
- Never say never, never say always...
  - It depends!
- Problem area to highlight
  - Parameter sniffing
    - What is it?
    - When/why does it become problematic?
  - Recompilation in procedures
    - What to recompile?
    - How to effectively use statement-level recompilation
Processing Stored Procedures

Creation

Parsing

Resolution

Resolution*

Optimization

Compilation

“sniffing” runtime params

A procedure's plan is NOT saved to disk; only metadata is saved at procedure creation. Use `sys.procedures`, `sp_procs`, functions and views to see metadata.

Compiled plan placed in unified plan cache. Re-used on subsequent executions parameter sniffing PROBLEMS.

Execution

(use when a plan does not already exist in cache)
Procedure Caching
Isn’t That the Point?

- Reusing plans can be good
  - When different parameters don’t change the optimal plan, then saving and reusing is excellent!
  - SQL Server saves time in compilation
- Reusing plans can be VERY bad
  - When different parameters wildly change the size of the result set and the optimal plans vary, then reusing the plan can be horribly bad
  - If indexes are added to base tables, existing plans may not leverage them
    - Don’t worry, there’s a simple solution here:
      - EXEC sp_recompile <tablename>
Recompilation Issues

RECOMPIATION = OPTIMIZATION
OPTIMIZATION = RECOMPIATION

- When do you want to recompile?
- What options do you have for recompilation – and at what granularity?
- How do you know you need to recompile?
- Do you want to recompile the entire procedure or only part of it?
- Can you test it?
Options for Recompilation

- CREATE ... WITH RECOMPILE
- EXECUTE ... WITH RECOMPILE
- sp_recompile objname

Statement-level reccompilation

- The 2000+ ways (still have benefits)
  - Dynamic string execution (statement’s plan not stored with procedure’s plan)
  - Modularized code (reduced sniffing problems with conditional logic/branching)

Better way to change compilation/optimization

- 2005+: OPTION(RECOMPILE)
- 2005+: OPTION (OPTIMIZE FOR (@variable_name = constant, ...))
- 2008+: OPTION (OPTIMIZE FOR UNKNOWN)
Statement-Level Recompilation

In 2005 and 2008: “inline” reccompilation for statements

- **OPTION (RECOMPILE)**
  - Excellent when parameters cause the execution plan to widely vary
  - Bad because EVERY execution will recompile – but only for the statements

- **OPTION (OPTIMIZE FOR**
  - (`@variable = literal, ...`) )
  - Excellent when large majority of executions generate the same optimization time
  - You don’t care that the minority may run slower with a less than optimal plan?

- **2008 only: OPTION (OPTIMIZE FOR UNKNOWN)**
  - Use the “all density” (average) instead of the histogram
Stored Procedure Resources

- Plan Caching in SQL Server 2008

- Batch Compilation, Recompilation, and Plan Caching Issues in SQL Server 2005

- PSS SQL Server Engine Blog

- KB 243586: Troubleshooting Stored Procedure Recompilation

- KB 308737: How to identify the cause of recompilation in an SP:Recompile event (SQL Server 2000 SP2+ has 6 subclass values, SQL Server 2005 has 11 subclass values and SQL Server 2008 RTM has 14 subclass values)

- KB 263889: Description of SQL Server blocking caused by compile locks
Plan Cache Pollution Resources

Blog posts:
- Plan cache and optimizing for adhoc workloads
- Plan cache, adhoc workloads and clearing the single-use plan cache bloat
- Clearing the cache - are there other options?
- Statement execution and why you should use stored procedures

Category: Optimizing Procedural Code

MSDN Article: How Data Access Code Affects Database Performance, Bob Beauchemin