

Pass-thru SQL performance vs Stored Proc

Source:

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.programming/2005-01/2441.html>

From: Tim Mavers (*webview_at_hotmail.com*)

Date: 01/13/05

Date: Thu, 13 Jan 2005 08:49:51 -0600

I know this has been talked about before, but I couldn't find any specific answers to my question searching Google Groups.

What I am curious about is if SQL-Server (2000) does any caching of pass-thru SQL code? In other words, inside my C# code I execute straight TQL (e.g. "select * from users") vs. calling a stored procedure to do the same thing (this is a very simple example), does SQL-Server cache the former? If so, would stored procedures perform better?

This all arises because I am working on a project where it seems like we are going to have to create a lot of dynamic SQL based on user selects (mainly searching). A user can select one to more fields to search on which we have to 'AND' to our query. This is somewhat difficult to do in stored procs as you have to account for all the possible search criteria and add them all as parameters. This is where straight in-line (or pass-thru) SQL helps.

I thought I read somewhere that SQL-Server will cache in-line SQL calls (compile it as it would a stored proc) so you wouldn't have as much as a performance penalty. Is this true? It seems that if we are passing dynamically built SQL, how can it cache it? Would SQL only cache calls with the exact same signature? If so, how does it do this with a stored proc where param values are always different?

And finally, should I be concerned about performance issues when using inline sql? Right now all our code is in stored procs.

Thanks!