

Re: Poor performance when executing stored procedure

Source:

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.programming/2004-07/5831.html>

From: Bill Cheng (*billchng_at_online.microsoft.com*)

Date: 07/26/04

Date: Mon, 26 Jul 2004 13:00:07 GMT

Hi Daniel,

Generally I would write stored procedures to do only one job. If SQL Server has to re-compile the SP, it takes shorter time to compile a smaller stored procedure.

I am not sure if that answers your concern, as I am not very sure about your real concerns. In the first post, you mention poor performance of a stored procedure. However, in this post, the issue seems different. If I could know your real concerns, it is better.

Bill Cheng

Microsoft Online Partner Support

Get Secure! – www.microsoft.com/security

This posting is provided “as is” with no warranties and confers no rights.

| Thread-Topic: Poor performance when executing stored procedure
| thread-index: AcRv7DhNh64oaMiUTFiHHz16/NR1Ww==
| X-WBNR-Posting-Host: 153.95.95.93
| From: "=?Utf-8?B?RGFuaWVsIFdhbHplbmJhY2g=?=" <daniel.walzenbach.NOSPAM@freudenberg.de>
| References: <6E4C2C57-00A5-4338-A64C-D1212950A089@microsoft.com>
<40FF8C92.E3D756B9@toomuchspamalready.nl>
| Subject: Re: Poor performance when executing stored procedure
| Date: Thu, 22 Jul 2004 06:03:02 -0700
| Lines: 60
| Message-ID: <8C902CB9-B19D-4C37-8FCD-D3D4004CB0D3@microsoft.com>
| MIME-Version: 1.0
| Content-Type: text/plain;
| charset="Utf-8"
| Content-Transfer-Encoding: 8bit
| X-Newsreader: Microsoft CDO for Windows 2000
| Content-Class: urn:content-classes:message
| Importance: normal
| Priority: normal

microsoft.public.sqlserver.programming: Re: Poor performance when executing stored procedure

| X-MimeOLE: Produced By Microsoft MimeOLE V6.00.3790.0
| Newsgroups: microsoft.public.sqlserver.programming
| NNTP-Posting-Host: TK2MSFTNGXA03.phx.gbl 10.40.2.157
| Path: cpmsftngxa06.phx.gbl!TK2MSFTNGXA01.phx.gbl!TK2MSFTNGXA03.phx.gbl
| Xref: cpmsftngxa06.phx.gbl microsoft.public.sqlserver.programming:459736
| X-Tomcat-NG: microsoft.public.sqlserver.programming

| Hello,

| Thank you all for your help. Unfortunately I have to confess that I tried to simplify the problem and misdirected you all. As a matter of fact I have a couple of sp which are all responsible for a business object. E.g. there are sps like spCustomer, spOrders, spRequest whereas each handles everything for one business object. Those sp therefore tend to grow really big. By now I understand (or at least this is how I could explain myself this behavior) that the sql optimizer tries to optimize the whole sp instead of only the exact part of the sp I want to execute (there might be a part in the sp to create a customer while another part selects customer details or searches for orders belonging to one customer). Additionally which is even worse those sps can execute each other. I now assume that the sql optimizer tried to compile lets say spCustomer, was busy analysing a whole bunch of stuff which wasn't necessary for the query to perform, then executed spOrders which is executed in spCustomers and then got stuck in a loop.

| By now I assume that I have to split my sps in smaller batches of sql which only do what they are supposed to do. Whereas spOrders allowed me to create orders, update or search them I should now create a spOrderCreate, a spOrderUpdate and a spOrderSearch. Could you recommend this approach? I got used to the other since it was kind of "how it is done here" and I just accepted it without scrutinizing. What would be best practices on how to write sps more efficiently? I assume I just got one: Do only what really is necessary in one sp, am I right?

| Best regards and thanks again for your help!

| Daniel

| "Gert-Jan Strik" wrote:

| > If there is no blocking going on, then this query (with your
| > description) should return instantaneous.
| >
| > 1. If the query
| >
| > select count(*) from sometable (NOLOCK) where id=@SomeID
| >
| > returns immediately, then you might be looking at a locking problem.
| >

Re: Poor performance when executing stored procedure

microsoft.public.sqlserver.programming: Re: Poor performance when executing stored procedure

|> 2. If the query
|>
|> select count(*) from sometable where id=@SomeID OPTION(MAXDOP 1)
|>
|> returns immediately, you might have hit the parallelism bug
|>
|> 3. the last interesting one would be
|>
|> select count(*) from sometable (INDEX=0) where id=@SomeID
|>
|>
|> By the way: does your table by any chance have a nonclustered index, but
|> no clustered index? Just checking...
|>
|> Hope this helps,
|> Gert-Jan
|>
|>
|>
|> Daniel Walzenbach wrote:
|>>
|>> Hi,
|>>
|>> I have a stored procedure which simply does a
|>>
|>> Select count(*) from sometable where ID=@SomeID
|>>
|>> whereas @SomeID is a parameter of this sp. Since the table is tiny (5
rows) and only has a couple of columns execution should be pretty fast.
Oddly though it takes the sp almost 2-3 sec. return the result. Does
anybody know what could be the reason for this behaviour? It seems that
query analyser waits a bit before executing the sp. I understand that the
optimizer has to compile the sp first, put it into its memory and than is
able to execute to sp. Is there any possibility that sql server got
confused and wants to compile the sp before every execution?
|>> Do you have any idea what else could be the reason for this strange
behaviour?
|>>
|>> Thank you! Best regards
|>>
|>> Daniel
|>
|> --
|> (Please reply only to the newsgroup)
|>
|