Is there a major performance gain by using stored procedures?

Asked 16 years ago Modified 1 year, 8 months ago Viewed 2k times

▲

7

▼

Is it better to use a stored procedure or doing it the old way with a connection string and all that good stuff? Our system has been running slow lately and our manager wants us to try to see if we can speed things up a little and we were thinking about changing some of the old database calls over to stored procedures. Is it worth it?

database performance logging stored-procedures

Share

Improve this question

Follow

edited Apr 17, 2023 at 9:43

dove

20.7k ● 14 ● 86 ● 110

asked Nov 28, 2008 at 13:49

Josh Mein

28.6k ● 15 ● 78 ● 88

1 "doing it the old way with a connection string" Using dynamic sql or stored procedures both require you to use a connection string! – fretje Jun 4, 2009 at 15:37

10 Answers

Sorted by: Highest score (default)

17

✓

The first thing to do is check the database has all the necessary indexes set up. Analyse where your code is slow, and examine the relevant SQL statements and indexes relating to them. See if you can rewrite the SQL statement to be more efficient. Check that you aren't recompiling an SQL (prepared) statement for every iteration in a loop instead of outside it once.

Moving an SQL statement into a stored procedure isn't going to help if it is grossly inefficient in implementation. However the database will know how to best optimise the SQL and it won't need to do it repeatedly. It can also make the client side code cleaner by turning a complex SQL statement into a simple procedure call.

Share Improve this answer Follow

answered Nov 28, 2008 at 13:52

JeeBee
17.5k ● 5 ● 52 ● 60

9

I would take a quick look at Stored Procedures are EVIL.

Share Improve this answer Follow

edited Nov 28, 2008 at 14:10

answered Nov 28, 2008 at 13:58

mattruma
16.7k ● 36 ● 108 ● 174

I wish I could add 10 points to this link. Describes exactly my thoughts on storing logic in the database layer.

– Brian C. Lane Nov 28, 2008 at 16:05

I do have to take exception to one of the author's points, however. It doesn't impact the final conclusion, but his statement that "there's nothing you can do in SQL you can't also do in app code" shows a common lack of understanding of set-based logic and its efficiencies. – GalacticCowboy Dec 8, 2008 at 14:03

6

So long as your calls are consistent the database will store the execution plan (MS SQL anyway). The strongest remaining reason for using stored procedures are for easy and sure security management.

If I were you I'd first be looking for adding **indices** where required. Also run a **profiling** tool to examine what is taking long and if that sql needs to changed, e.g. adding more Where clauses or restricting result set.

You should consider **caching** where you can.

Share Improve this answer Follow

edited Nov 28, 2008 at 14:04

answered Nov 28, 2008 at 13:53

dove
20.7k ● 14 ● 86 ● 110

Voted up for bolding the important keywords. Nice touch. :) – Jon Tackabury Dec 3, 2008 at 15:47

5

Stored procedures will not make things faster.

However, rearranging your logic will have a huge impact. The tidy, focused transactions that you design when thinking of stored procedures are hugely beneficial.

Also, stored procedures tend to use bind variables, where other programming languages sometimes rely on building SQL statements on-the-fly. A small, fixed set of SQL statements and bind variables is fast. Dynamic SQL statements are slow.

An application which is "running slow lately" does not need coding changes.

1. Measure. Measure. Measure. "slow" doesn't mean much when it comes to performance tuning. What is slow? Which exact transaction is slow? Which table is slow? Focus.
2. Control all change. All. What changed? OS patch? RDBMS change? Application change? Something changed to slow things down.
3. Check for constraints in scale. Is a table slowing down because 80% of the data is history that you use for reporting once a year?

Stored procedures are never the solution to performance problems until you can absolutely point to a specific block of code which is provably faster as a stored procedure.

Share Improve this answer

Follow

answered Nov 28, 2008 at 13:59

S.Lott

391k ● 82 ● 517 ● 788

The main reason why we are running slow is because we are in our busy season but it is running slower than it ever has before (even in previous busy seasons). Therefore, we are trying to do our best to speed it up a little. Thanks – Josh Mein Nov 28, 2008 at 14:01

"slower than before" is start, but it's pretty broad. The next step is to focus on a specific table, transaction, routine, procedure, job or task. Without focus, you can't prove that your change to a SP has a positive impact. – S.Lott Nov 28, 2008 at 14:15

We think it is our job table and its related tables because that is the one that is getting called the most this time of year so that is what we are going to focus on. – Josh Mein Nov 28, 2008 at 14:19

Slower than before - see the post on performance behaviour of queuing systems below. – ConcernedOfTunbridgeWells Nov 28, 2008 at 14:28

▲

4

▼

stored procedures can be really help if they avoid sending huge amounts of data and/or avoid doing roundtrips to the server,so they can be valuable if your application has one of these problems.

Share Improve this answer

Follow

answered Nov 28, 2008 at 15:01

Robert Gould

69.7k ● 61 ● 191 ● 275

2

After you finish your research you will realize there are two extreme views at opposite side of the spectrum. Historically the Java community has been against store procs due to the availability of frameworks such as hibernate, conversely the .NET community has used more stored procs and this legacy goes as far as the vb5/6 days. Put all this information in context and stay away from the extreme opinions on either side of the coin.

Speed should not be the primary factor to decide against or in favor of stored procs. You can achieve sp performace using inline SQL with hibernate and other frameworks. Consider maintenance and which other programs such as reports, scripts could use the same stored procs used by your application. If your scenario requires multiple consumers for the same SQL code, stored procedures are a good candidate, maintenance will be easier. If this is not the case, and you decide to use inline sql, consider externalizing it in config files to facilitate maintenance.

At the end of the day, what counts is what will make your particular scenario a success for your stakeholders.

Share Improve this answer Follow

answered Nov 28, 2008 at 15:51

The Landlord

31 ● 2

2

If your server is getting noticeably slower in your busy season it may be because of saturation rather than anything ineffecent in the database. Basic queuing theory tells us that a server gets hyperbolically slower as it approaches saturation.

The basic relationship is `1/(1-X)` where X is the proportion of load. This describes the average queue length or time to wait before being served. Therefore a server that is getting saturated will slow down very rapidly when the load spikes.

A server that is 25% loaded will have an average service time of 1.333K for some constant K (loosely, K is the time for the machine to perform one transaction). A server that is 50% loaded will have an average service time of 2K and a server that is 90% loaded will have an average service time of 10K. *Given that the slowdowns are hyperbolic in nature, it often doesn't take a large change in overall load to produce a significant degradation in response time.*

Obviously this is somewhat simplistic as the server will be processing multiple requests concurrently (there are more elaborate queuing models for this situation), but the broad principle still applies.

So, if your server is experiencing transient loads that are saturating it, you will experience patches of noticeable slow-down. Note that these slow-downs need only be in one bottlenecked area of the system to slow the whole process down. If you are only experiencing this now in a busy season there is a possibility that your server has simply hit a constraint on a resource, rather than being particularly slow or inefficient.

Note that this possibility is not antithetical to the possibility of inefficiencies in the code. You may find that the way to ease the bottleneck is to tune some of your queries.

In order to tell if the system is bottlenecked, start gathering profiling information. If you can find resources with a large number of waits, this should give you a good starting point.

The final possibility is that you need to upgrade your server. If there are no major inefficiencies in the code (this might well be the case if profiling doesn't indicate any disproportionately large bottlenecks) you may simply need bigger hardware. I have no idea what your volumes are, but don't discount the possibility that you may have outgrown your server.

Share Improve this answer

Follow

edited Dec 8, 2008 at 13:31

answered Nov 28, 2008 at 14:43

ConcernedOfTunbridgeWells

66.5k ● 15 ● 148 ● 198

0

Yes, stored procs is a step forward towards acheiving good performance. The main reason is that stored procedures can be pre-compiled and their execution plan cached.

You however need to first analyse where your performance bottlenecks are really - so that you approach this exercise in a structured way.

As it has been suggested in one of the responses, try analyse using a profiler tool where the problem is - e.g do you need to create indexes...

Cheers

Share Improve this answer Follow

answered Nov 28, 2008 at 14:03

Julius A
39.5k ● 26 ● 78 ● 97

Most modern RDBMSes will pre-compile and cache execution plans for prepared statements anyway.
– Jon Skeet Nov 28, 2008 at 14:07

0

Like all of the above posts suggest, you first want to clean up your SQL statements, have appropriate indexes. caching can be tricky, I cant comment unless I have more detail on what you are trying to accomplish.

But one thing about sprocs, make sure you dont let it generate **dynamic SQL statements** because for one, it will be pointless and it can be subjected to **SQL Injection** attacks...this has happened in one of the projects I looked into.

I would recommend sprocs for updates mainly, and then select statements. good luck :)

Share Improve this answer

Follow

answered Dec 3, 2008 at 15:28

LVS

-1

You can never say in advance. You must do it and measure the difference because in 9 out of 10 cases, the bottleneck is not where you think.

If you use a stored procedure, you don't have to transmit the data. DBs are usually slow at executing [EDIT]complex[/EDIT] stored procedures [EDIT]with loops, higher math, etc[/EDIT]. So it really depends on how much work you would need to do, how slow your network is, how fast the DB executes this particular code, etc.

Share Improve this answer

Follow

edited Nov 28, 2008 at 15:12

answered Nov 28, 2008 at 13:54

Aaron Digulla

328k ● 110 ● 622 ● 834