Stored Procedure Execution Plan - Data Manipulation

Asked 15 years, 10 months ago Modified 13 years, 6 months ago

Viewed 2k times

I have a stored proc that processes a large amount of data (about 5m rows in this example). The performance varies wildly. I've had the process running in as little as 15 minutes and seen it run for as long as 4 hours.

For maintenance, and in order to verify that the logic and processing is correct, we have the SP broken up into sections:

1. `TRUNCATE` and populate a work table (indexed) we can verify later with automated testing tools.
2. Join several tables together (including some of these work tables) to product another work table

Repeat 1 and/or 2 until a final output is produced.

My concern is that this is a single SP and so gets an execution plan when it is first run (even `WITH RECOMPILE`). But at that time, the work tables (permanent tables in a Work schema) are empty.

I am concerned that, regardless of the indexing scheme, the execution plan will be poor.

I am considering breaking up the SP and calling separate SPs from within it so that they could take advantage of a re-evaluated execution plan after the data in the work tables is built. I have also seen reference to using EXEC to run dynamic SQL which, obviously might get a `RECOMPILE` also.

I'm still trying to get `SHOWPLAN` permissions, so I'm flying quite blind.

sql sql-server stored-procedures sql-execution-plan

Share

Improve this question

Follow

edited Jun 8, 2011 at 16:28

Dustin Laine

38.5k ● 10 ● 89 ● 125

asked Feb 6, 2009 at 3:36

Cade Roux

89.6k ● 40 ● 184 ● 266

5m rows isn't big. Crikey, in DB2/z, we have configuration tables bigger than that :-) – paxdiablo Feb 6, 2009 at 3:58

5m is just large enough that it seems like the execution plans might be starting to really matter. We have some daily feeds of over 25m rows. – Cade Roux Feb 6, 2009 at 4:06

7 Answers

Sorted by: Highest score (default)

2

Are you able to determine whether there are any locking problems? Are you running the SP in sufficiently small transactions?

Breaking it up into subprocedures should have no benefit.

Somebody should be concerned about your productivity, working without basic optimization resources. That suggests there may be other possible unseen issues as well.

Share Improve this answer

Follow

answered Feb 6, 2009 at 9:17

dkretz

37.6k ● 13 ● 83 ● 140

They are telling me I might get SHOWPLAN week after next ;-) – Cade Roux Feb 6, 2009 at 15:48

1

Grab the free copy of "Dissecting Execution Plan" in the link below and maybe you can pick up a tip or two from it that will give you some idea of what's really going on under the hood of your SP.

http://dbalink.wordpress.com/2008/08/08/dissecting-sql-server-execution-plans-free-ebook/

Share Improve this answer

Follow

answered Feb 6, 2009 at 4:48

MarlonRibunal

4,087 ● 3 ● 33 ● 37

1

Are you sure that the variability you're seeing is caused by "bad" execution plans? This may be a cause, but there may be a number of other reasons:

- "other" load on the db machine
- when using different data, there may be "easy" and "hard" data
- issues with having to allocate more memory/file storage
- ...

Have you tried running the SP with the same data a few times?

Also, in order to figure out what is causing the runtime/variability, I'd try to do some detailed measuring to pin the problem down to a specific section of the code. (Easiest way would be to insert some log calls at various points in the sp). Then try to explain why that section is slow (other than "5M rows ;-)) and figure out a way to make that faster.

For now, I think there are a few questions to answer before going down the "splitting up the sp" route.

Share Improve this answer

Follow

answered Feb 6, 2009 at 6:30

Thorsten

13.2k ● 18 ● 64 ● 79

It ran for 11 hours last night before I killed it. – Cade Roux Feb 6, 2009 at 15:47

It sounds like the way you wrote your SP you can always restart it without too much trouble. So instead of doing "the whole thing", just do the first step (and comment out the rest) and see how long that takes, then steps one and two and so on. That way you should be able to find the slow part.
– Thorsten Feb 6, 2009 at 21:51

1

You're right it is quite difficult for you to get a clear picture of what is happening behind the scenes until you can get the "actual" execution plans from several executions of your overall process.

One point to consider perhaps. Are your work tables physical of temporary tables? If they are physical you will get a performance gain by inserting new data into a new table without an index (i.e. a heap) which you can then build an index on after all the data has been inserted.

Also, what is the purpose of your process. It sounds like you are moving quite a bit of data around, in which case you may wish to consider the use of partitioning. You can switch in and out data to your main table with relative ease.

Hope what I have detailed is clear but please feel free to pose further questions.

Cheers, John

Share Improve this answer

Follow

answered Feb 6, 2009 at 9:11

John Sansom

41.8k ● 9 ● 73 ● 87

1

In several cases I've seen this level of diversity of execution times / query plans comes down to statistics. I would recommend some tests running update stats against the tables you are using just before the process is run. This will both force a re-evaluation of the execution plan by SQL and, I suspect, give you more consistent results. Additionally you may do well to see if the differences in execution time correlate with re-indexing jobs by your dbas. Perhaps you could also gather some index health statistics before each run.

If not, as other answerers have suggested, you are more likely suffering from locking and/or contention issues.

Good luck with it.

Share Improve this answer

Follow

answered Feb 8, 2009 at 22:35

Timbo

4,533 ● 2 ● 28 ● 29

0

The only thing I can think that an execution plan would do wrong when there's no data is err on the side of using a table scan instead of an index, since table scans are super fast when the whole table will fit into memory. Are there other negatives you're actually observing or are

sure are happening because there's no data when an execution plan is created?

You can force usage of indexes in your query...

Seems to me like you might be going down the wrong path.

Share Improve this answer

Follow

answered Feb 6, 2009 at 4:40

ChrisN

3,413 ● 26 ● 33

0

Is this an infeed or outfeed of some sort or are you creating a report? If it is a feed, I would suggest that you change the process to use SSIS which should be able to move 5 million records very fast.

Share Improve this answer

Follow

answered Feb 6, 2009 at 18:02

HLGEM

96.4k ● 15 ● 119 ● 189

It's a backend business process, but yes, I have been pushing for some work (appropriate) to be performed in the ETL, to no avail. – Cade Roux Feb 6, 2009 at 19:50