Are Oracle optimizer-hints wise in production code?

Asked 9 years, 9 months ago Modified 9 years, 9 months ago

Viewed 310 times

4

As I come up to speed on Oracle (having been a DB2-guy for the last couple decades), I see a A LOT of existing code that uses optimizer-hints in its queries.

From what I've read on various Oracle-focused web-sites, several Oracle "experts" advise AGAINST putting optimizer-hints in production code because:

- With every Oracle patch or upgrade, the hint will probably be wrong.
- With every DDL, the hint will probably be wrong.

One "expert" says:

> The reason to be wary of hinting is that by embedding hints in your SQL, you are overriding the optimizer and saying that you know more than it does – not just now, but every time in the future that your SQL will be run, irrespective of any other changes that may happen to your database. The likely consequence of this is that

> your SQL will possibly run sub-optimally now and almost certainly in the future.

(See http://allthingsoracle.com/a-beginners-guide-to-optimizer-hints/)

So, if optimizer-hints are commonly known to be "unwise", why are they so frequently used (... at least in the code I've seen)?

oracle-database **oracle11g**

Share

Improve this question

Follow

edited Mar 2, 2015 at 20:08

Jonathan Leffler

752k ● 145 ● 946 ● 1.3k

asked Mar 2, 2015 at 17:55

DaveSlash

55 ● 5

2 This really depends on type of the hint, Oracle version and use case. – ibre5041 Mar 2, 2015 at 18:01

3 As @ibre5041 says, it depends. What hints do you see frequently used? Something like FIRST_ROWS or ALL_ROWS would be OK in general, but something like INDEX(x y) not so good. – Tony Andrews Mar 2, 2015 at 18:14

2 Answers

Sorted by: Highest score (default)

To the extent that hints are common, it's generally because someone in the past prioritized fixing an acute issue rather than dealing with the underlying statistics problem. It's entirely possible that this prioritization was reasonable (particularly at the time) but it introduces technical debt that will probably have to be paid back.

When a system becomes unresponsive because a query plan changed and became massively less efficient, fixing the acute production issue generally takes precedence over identifying the root cause. Slapping a hint on a single query is generally both quicker and easier than figuring out the underlying issue or learning how to use the various tools Oracle provides to ensure query plan stability or to evolve plans over time.

Of course, having slapped a band-aid on a single query, if you don't then invest the time to understand why statistics sent the optimizer down the wrong path or why your approach to plan stability didn't work, it's likely that whatever statistics issue you have will cause other queries to perform poorly. It's very rare that misleading statistics would cause just one query in the system to perform poorly. Generally, that either leads to a viscous circle where more queries start performing badly leading to more hints being added or a virtuous circle where DBAs take a step back, work through what's going wrong to cause query performance to suffer, fix the underlying issue, and then remove the hints.

All that being said, there are a few hints that may be reasonably used relatively commonly depending on the sort of code you're using. If you've got a function that returns a `sys_refcursor` that is returned to a client application that you know is going to fetch the first few rows, display them to the user, and only ask for the next set of rows if they don't find what they're looking for, it makes sense to use a `FIRST_ROWS` hint pretty liberally because you know something the optimizer can't possibly know. You know that users are much more interested in the first few rows rather than the complete set of results. If you have lots of code that is using collections in SQL, you probably want to use a lot of `CARDINALITY` hints because otherwise Oracle has no idea how many elements the collection is likely to have.

Share Improve this answer

Follow

answered Mar 2, 2015 at 19:22

Justin Cave

231k ● 25 ● 377 ● 392

0

In my case, most of the hints I see in our production code are like these:

```
/*+APPEND*/
/*+ full(MP) parallel(MP, 16) */
/*+ PARALLEL(ME1, 16) FULL(ME1) DRIVING_SITE(ME1)
*/
```

Rarely (but every so often), I'll see an index-hint:

e.g.

```
/*+ index(MSL XCL01000) */
```

Thanks for your valuable input. I certainly understand a bit more the dangers and necessities of hints.

Share Improve this answer

Follow

answered Mar 5, 2015 at 0:45

DaveSlash

55 ● 5