

优化数据库大幅度提高Oracle的性能 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/143/2021\\_2022\\_\\_E4\\_BC\\_98\\_E5\\_8C\\_96\\_E6\\_95\\_B0\\_E6\\_c102\\_143685.htm](https://www.100test.com/kao_ti2020/143/2021_2022__E4_BC_98_E5_8C_96_E6_95_B0_E6_c102_143685.htm) 几个简单的步骤大幅度提高Oracle性能--我优化数据库的三板斧。数据库优化的讨论可以说是一个永恒的主题。资深的Oracle优化人员通常会要求提出性能问题的人对数据库做一个statspack，贴出数据库配置等等。还有的人认为要抓出执行最慢的语句来进行优化。但实际情况是，提出疑问的人很可能根本不懂执行计划，更不要说statspack了。而我认为，数据库优化，应该首先从大的方面考虑：网络、服务器硬件配置、操作系统配置、Oracle服务器配置、数据结构组织、然后才是具体的调整。实际上网络、硬件等往往无法决定更换，应用程序一般也无法修改，因此应该着重从数据库配置、数据结构上来下手，首先让数据库有一个良好的配置，然后再考虑具体优化某些过慢的语句。我在给我的用户系统进行优化的过程中，总结了一些基本的，简单易行的办法来优化数据库，算是我的三板斧，呵呵。不过请注意，这些不一定普遍使用，甚至有的会有副作用，但是对OLTP系统、基于成本的数据库往往行之有效，不妨试试。（注：附件是Burleson写的用来报告数据库性能等信息的脚本，本文用到）

一．设置合适的SGA 常常有人抱怨服务器硬件很好，但是Oracle就是很慢。很可能是内存分配不合理造成的。(1)假设内存有512M，这通常是小型应用。建议Oracle的SGA大约240M，其中：共享池（SHARED\_POOL\_SIZE）可以设置60M到80M，根据实际的用户数、查询等来定。数据块缓冲区可以大致分

配120M-150M，8i下需要设置DB\_BLOCK\_BUFFERS

，DB\_BLOCK\_BUFFER\*DB\_BLOCK\_SIZE等于数据块缓冲区大小。9i下的数据缓冲区可以用db\_cache\_size来直接分配。

(2)假设内存有1G，Oracle的SGA可以考虑分配500M：共享池分配100M到150M，数据缓冲区分配300M到400M。(3)内存2G，SGA可以考虑分配1.2G，共享池300M到500M，剩下的给数据块缓冲区。(4)内存2G以上：共享池300M到500M就足够啦，再多也没有太大帮助；(Biti\_rainy有专述)数据缓冲区是尽可能的大，但是一定要注意两个问题：一是要给操作系统和其他应用留够内存，二是对于32位的操作系统，Oracle的SGA有1.75G的限制。有的32位操作系统上可以突破这个限制，方法还请看Biti的大作吧。

## 二．分析表和索引，更改优化模式

Oracle默认优化模式是CHOOSE，在这种情况下，如果表没有经过分析，经常导致查询使用全表扫描，而不使用索引。这通常导致磁盘I/O太多，而导致查询很慢。如果没有使用执行计划稳定性，则应该把表和索引都分析一下，这样可能直接会使查询速度大幅提升。分析表命令可以用ANALYZE TABLE 分析索引可以用ANALYZE INDEX命令。对于少于100万的表，可以考虑分析整个表，对于很大的表，可以按百分比来分析，但是百分比不能过低，否则生成的统计信息可能不准确。可以通过DBA\_TABLES的LAST\_ANALYZED列来查看表是否经过分析或分析时间，索引可以通过DBA\_INDEXES的LAST\_ANALYZED列。下面通过例子来说明分析前后的速度对比。（表CASE\_GA\_AJZLZ大约有35万数据，有主键）首先在SQLPLUS中打开自动查询执行计划功能。（第一次要执行\RDBMS\ADMIN\utlxplan.sql来创建PLAN\_TABLE这个表）

```

SQL> SET AUTOTRACE ON SQL>SET TIMING ON 通过SET
AUTOTRACE ON 来查看语句的执行计划，通过SET TIMING
ON 来查看语句运行时间。 SQL> 0select count(*) from
CASE_GA_AJZLZ. COUNT(*) ----- 346639 已用时间: 00:
00: 21.38 Execution Plan 0 SELECT STATEMENT
Optimizer=CHOOSE 1 0 SORT (AGGREGATE) 2 1 TABLE
ACCESS (FULL) OF CASE_GA_AJZLZ ..... 请注
意上面分析中的TABLE ACCESS(FULL)，这说明该语句执行
了全表扫描。而且查询使用了21.38秒。这时表还没有经过分
析。下面我们来对该表进行分析： SQL> analyze table
CASE_GA_AJZLZ compute statistics. 表已分析。 已用时间: 00:
05: 357.63。 然后再来查询： SQL> 0select count(*) from
CASE_GA_AJZLZ. COUNT(*) ----- 346639 已用时间: 00:
00: 00.71 Execution Plan 0 SELECT STATEMENT
Optimizer=FIRST_ROWS (Cost=351 Card=1) 1 0 SORT
(AGGREGATE) 2 1 INDEX (FAST FULL SCAN) OF PK_AJZLZ
(UNIQUE) (Cost=351 Card=346351) ..... 请
注意，这次时间仅仅用了0.71秒！这要归功于INDEX(FAST
FULL SCAN)。通过分析表，查询使用了PK_AJZLZ索引，磁
盘I/O大幅减少，速度也大幅提升！下面的实用语句可以用来
生成分析某个用户的所有表和索引，假设用户是GAXZUSR：
SQL> set pagesize 0 SQL> spool d:\analyze_tables.sql. SQL> 0select
analyze table ||owner||.||table_name|| compute statistics. from
dba_tables where owner=GAXZUSR. SQL> spool off SQL> spool
spool d:\analyze_indexes.sql. SQL> 0select analyze index
||owner||.||index_name|| compute statistics. from dba_indexes where

```

```
owner=GAXZUSR. SQL> spool off SQL> @d:\analyze_tables.sql  
SQL> @d:\analyze_indexes.sql
```

解释：上面的语句生成了两个sql文件，分别分析全部的GAXZUSR的表和索引。如果需要按照百分比来分析表，可以修改一下脚本。通过上面的步骤，我们就完成了对表和索引的分析，可以测试一下速度的改进啦。建议定期运行上面的语句，尤其是数据经过大量更新。当然，也可以通过dbms\_stats来分析表和索引，更方便一些。但是我仍然习惯上面的方法，因为成功与否会直接提示出来。

100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)