

解析Oracle数据库内在索引和参数的调整 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/253/2021_2022__E8_A7_A3_E6_9E_90Orac_c102_253979.htm 虽然系统级性能调优不是结构化查询语言（SQL）性能调优的最好解决方案，但是，当你不能改变某些SQL语句时（例如，提供商提供的某些包中的SQL语句），使用SQL处理的常规模式来调节系统性能是极其有帮助的。在日常工作经验的基础上，Oracle数据库管理员改变某些优化参数来适应库缓冲区中SQL类型的变化是很常见的事。一些较为常见的变更：1、针对Oracle参数的更改。对optimizer_mode，optimizer_index_cost_adj和optimizer_index_caching的改变能对SQL执行计划产生巨大影响。2、针对统计参数的更改。使用dbms_stats包导入特定的统计参数（针对当然处理模式作了调整的）可对SQL的执行速度产生巨大影响。3、使用自动化查询重新写入。使用Oracle实体化视图能够预先聚集、预先汇总数据，从而减少运行时刻表连接的数量。对于更新比较少的数据库，也可以通过预先连接表来提高处理速度。一些对性能调优最重要的Oracle优化参数如下：* optimizer_mode（优化模式） - 在Oracle 9i中，有许多优化模式，都是由参数optimizer_mode的值决定的。这个参数的取值范围是rule, choose, all_rows, first_rows, first_rows_1, first_rows_10和first_rows_100. 我们以定义“最好的”执行计划作为开始点。在任何给定的时间，库缓冲区中的所有SQL语句都需要有“最好的”执行计划（当然，由于在任何给定的时间里处理需求可能不同，所以这个最优执行计划可能会经常发生变化）。什么是“最好的”执

行计划？是返回结果最快的执行计划还是使用最少计算资源的执行计划？很明显，答案依赖于你的数据库的处理过程，Oracle提供了两种优化模式，允许你选择你认为的“最好的”执行计划：1、`optimizer_mode=first_rows`——相对全表扫描访问，这个优化模式更注重索引访问。当你想要一个查询以最快的速度返回结果行时，即使它的逻辑输入输出总量比全表扫描高，也要使用这个模式在线访问系统一般都使用这个模式，因为终端用户想要尽快地看到第一页查询结果。2、`optimizer_mode=all_rows`——这个优化模式更注重全表扫描（特别是并发全表扫描），因为在这种情况下服务器资源的开销最小。这个模式一般被用于批处理进程和数据仓库中，它们的目标都是使服务器消耗的资源最小化。3、`optimizer_mode=first_rows_n`——从Oracle 9i开始，又有一种新的优化模式针对某些返回小结果集的查询进行优化。其取值范围是`first_rows_1`、`first_rows_10`和`first_rows_100`，使用这些参数值可以确保Oracle能够优化这类SQL。虽然参数`optimizer_mode`控制了“基于代价的优化”的总体行为，还有其他Oracle参数也会对“基于代价的优化”产生相当大的影响。Oracle提供了一些重要的参数来控制“基于代价的优化”做出的选择：1、`optimizer_index_cost_adj`——这个参数可用来调整“基于代价的优化”相对于全表扫描访问而言，更加倾向于索引访问的程度。这个值越小，“基于代价的优化”就越有可能使用一个可用的索引。2、`optimizer_index_caching`——这个参数告诉Oracle你的索引在内存的数据缓冲区中的可能性有多大。对这个参数的设置将会影响到“基于代价的优化”做出的对一个表连接（嵌套循环

) 使用索引还是使用全表扫描选择。 3、
db_file_multiblock_read_count——当把这个值设置得比较大时（使用更大的服务器），“基于代价的优化”识别出分散的（多块）读操作的代价或许比识别顺序读操作的代价更小一些。这就使得“基于代价的优化”更加倾向于全表扫描。但是从Oracle 9.2版本开始，情况不再是这样了。当计算系统统计表时，它包含了“多块读操作记数”（MBRC），这个数字决定了全表扫描的成本。Oracle 10g则更进一步，加入了一些“系统默认值”，这些默认值是非常不合适的。对于Oracle 9.2版本而言，请注意Metalink上的149560.1。 1、
parallel_automatic_tuning——当该参数设置为“开启”时，对于含有许多CPU的Oracle服务器，全表扫描并发执行。因为并发全表扫描的速度可以非常快，所以“基于代价的优化”对于索引访问开销很大，因此更加倾向于使用全表扫描。 2、
hash_area_size（假如不使用pga_aggregate_target的话）——这个参数设置“基于代价的优化”相对于使用嵌套循环和排序合并表连接来说，更倾向于使用哈希连接的程度。 3
、sort_area_size（只当不使用参数pga_aggregate_target时）——这个参数影响了“基于代价的优化”做出的执行索引访问还是执行对结果集的排序的决定。这个参数值越高，则在内存中执行排序（比使用临时表空间快上千倍）的可能性就越大，同时“基于代价的优化”相对于使用预先排序好的索引检索，更倾向于使用直接排序。 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