什么时候oracle使用绑定变量性能反而更差 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_BB_80_E4_B9_88_E6_97_B6_E5_c102_143079.htm 当我在做培训时，在解释绑定变量的好处时，大家都比较容易理解。但是，对于并不是任何时候绑定变量都是最优的。这一点很多人不是和理解。下面就讨论一下在什么时候会出现绑定变量会使性能变差。扫描成本和OPTIMIZER_INDEX_COST_ADJ我们知道，在CBO模式下，Oracle会计算各个访问路径的代价，采用最小代价的访问路径作为语句的执行计划。而对于索引的访问代价的计算，需要根据一个系统参数OPTIMIZER_INDEX_COST_ADJ来转换为与全表扫描代价等价的一个值。这是什么意思呢？我们先稍微解释一下这个参数：OPTIMIZER_INDEX_COST_ADJ。它的值是一个百分比，默认是100，取值范围是1～10000。当估算索引扫描代价时，会将索引的原始代价值乘以这个百分比，将换算后的值作为与全表扫描代价比较的值。也就是说，当这个值为100时，计算出的索引扫描代价就是它的原始代价：COST_COM = COST_ORG * OPTIMIZER_INDEX_COST_ADJ/100看以下例子：SQL> create table T_PEEKING (a NUMBER, b char(1), c char(2000)). Table created. SQL>SQL> create index T_PEEKING_IDX1 on T_PEEKING(b). Index created. SQL> begin 2 for i in 1..1000 loop 3 insert into T_PEEKING values (i, A, i). 4 end loop. 5 6 insert into T_PEEKING values (1001, B, 1001). 7 insert into T_PEEKING values (1002, B, 1002). 8 insert into T_PEEKING values (1003, C, 1003). 9 10 commit. 11 end. 12 /

PL/SQL procedure successfully completed. 注意，我们给索引字段B插入的值中只有3个distinct值，记录数是1003，它的集的势很高（1003/3）=334。关于集的势的计算，可以参考我的另外一篇文档《关于集的势的计算》。SQL>SQL> analyze table T_PEEKING compute statistics for table for all indexes for all indexed columns. Table analyzed. SQL>我们看下索引扫描的代价是多少：SQL> show parameter OPTIMIZER_INDEX_COST_ADJ NAME TYPE VALUE---------- ---------- ----------optimizer_index_cost_adj integer 100 SQL> 0delete from plan_table. 0 rows 0deleted. SQL> SQL> explain plan for 0select /* index(a T_PEEKING_IDX1)*/ * from T_PEEKING a where b = :V. Explained. SQL> 0select lpad(, 2*(level-1))||operation|| ||options|| || 2 object_name|| ||decode(id, 0, Cost=||position) "Query 3 Plan_Table" 4 from plan_table 5 start with id = 0 6 connect by prior id = parent_id 7 . QueryPlan_Table---------- ----------SELECT STATEMENT Cost=113 TABLE ACCESS BY INDEX ROWID T_PEEKING INDEX RANGE SCAN T_PEEKING_IDX1 SQL>再看全表扫描的代价是多少：SQL> 0delete from plan_table. 3 rows 0deleted. SQL>SQL> explain plan for 0select /* full(a)*/ * from T_PEEKING a where b = :V. Explained. SQL>SQL> 0select lpad(, 2*(level-1))||operation|| ||options|| || 2 object_name|| ||decode(id, 0, Cost=||position) "Query 3 Plan_Table" 4 from plan_table 5 start with id = 0 6 connect by prior id = parent_id 7 .

QueryPlan_Table----------SELECT STATEMENT Cost=75 TABLE ACCESS FULL T_PEEKING SQL> 这时，我们可以计算得出让优化器使用索引（无提示强制）的OPTIMIZER_INDEX_COST_ADJ值应该，而大于66则会使用全表扫描：SQL> alter system set OPTIMIZER_INDEX_COST_ADJ=67. System altered. SQL>SQL> 0delete from plan_table. 2 rows 0deleted. SQL>SQL> explain plan for 0select * from T_PEEKING a where b = :V. Explained. SQL>SQL> 0select lpad(, 2*(level-1))||operation|| ||options|| || 2 object_name|| ||decode(id, 0, Cost=||position) "Query 3 Plan_Table" 4 from plan_table 5 start with id = 0 6 connect by prior id = parent_id.
QueryPlan_Table----------SELECT STATEMENT Cost=75 TABLE ACCESS FULL T_PEEKING SQL>SQL>SQL> alter system set OPTIMIZER_INDEX_COST_ADJ=66. System altered. SQL>SQL> 0delete from plan_table. 2 rows 0deleted. SQL>SQL> explain plan for 0select * from T_PEEKING a where b = :V. Explained. SQL>SQL> 0select lpad(, 2*(level-1))||operation|| ||options|| || 2 object_name|| ||decode(id, 0, Cost=||position) "Query 3 Plan_Table" 4 from plan_table 5 start with id = 0 6 connect by prior id = parent_id.
QueryPlan_Table----------SELECT STATEMENT Cost=75 TABLE ACCESS BY INDEX ROWID T_PEEKING INDEX RANGE

SCAN T_PEEKING_IDX1可以看出，在使用绑定变量时，参数OPTIMIZER_INDEX_COST_ADJ对于是否选择索引会有重要的影响。这里我们暂且不讨论索引扫描的原始成本是如何计算得出的。但是有一点很重要，在使用绑定变量时，计算出的成本是平均成本。在我们上面的例子中，字段B的值只有3个："A"、"B"、"C"，其中A最多，1003行中有1000行。因此，在索引上扫描值为A记录的成本为1000/1003 * 索引全扫描成本 ≈索引全扫描成本，我们看下它的成本是多少：SQL> alter system set OPTIMIZER_INDEX_COST_ADJ=100. System altered. SQL>SQL> 0delete from plan_table. 2 rows 0deleted. SQL>SQL> explain plan for 0select /* index(a T_PEEKING_IDX1)*/* from T_PEEKING a where b = A. Explained. SQL>SQL> 0select lpad(, 2*(level-1))||operation|| ||options|| || 2 object_name|| ||decode(id, 0, Cost=||position) "Query 3 Plan_Table" 4 from plan_table 5 start with id = 0 6 connect by prior id = parent_id. QueryPlan_Table----------SELECT STATEMENT Cost=336 TABLE ACCESS BY INDEX ROWID T_PEEKING INDEX RANGE SCAN T_PEEKING_IDX1可以看到，它的成本是336。因此索引的平均成本是(336 * 1003/1000) / 3 ≈ 113，也就是使用绑定变量使的成本。而扫描其它两个值"B"和"A"时代价就非常小。