如何监控Oracle索引的使用完全解析 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/205/2021_2022__E5_A6_82_E4_BD_95_E7_9B_91_E6_c102_205828.htm 研究发现，oracle数据库使用的索引不会超过总数的25%，或者不易他们期望被使用的方式使用。通过 监控数据库索引的使用，释放那些未被使用的索引，从而节省维护索引的开销，优化性能。 1、在oracle8i中，确定使用了那个索引的方法意味着要对存在语共享SQL区中的所有语句运行EXPLIAN PALN，然后查询计划表中的OPERATION列，从而识别有OBJECT_OWNER和OBJECT_NAME列所确定的那个索引上的索引访问。 下面是一个监控索引使用的脚本，这个脚本仅仅是一个样品，在某种条件下成立： 条件： 运行这个脚本的用户拥有权限解释所有的v$sqlarea中的sql，除了不是被SYS装载的。plan_table.remarks能够别用来决定与特权习惯的错误。对所有的共享池中SQL，参数OPTIMIZER_GOAL是一个常量，无视v$sqlarea.optimizer_mode。两次快照之间，统计资料被再次分析过。没有语句别截断。所有的对象都是局部的。所有被引用的表或视图或者是被运行脚本的用户所拥有，或者完全有资格的名字或同义词被使用。自从上次快照以来，没有不受"欢迎"的语句被冲洗出共享池(例如，在装载)。对于所有的语句，v$sqlarea.version_count = 1 (children)。 脚本: Code: [Copy to clipboard] set echo off Rem Drop and recreate PLAN_TABLE for EXPLAIN PLAN 0drop table plan_table. create table PLAN_TABLE (statement_id varchar2(30), timestamp date, remarks varchar2(80), operation varchar2(30), options

varchar2(255), object_node varchar2(128), object_owner varchar2(30), object_name varchar2(30), object_instance numeric, object_type varchar2(30), optimizer varchar2(255), search_columns number, id numeric, parent_id numeric, position numeric, cost numeric, cardinality numeric, bytes numeric, other_tag varchar2(255), partition_start varchar2(255), partition_stop varchar2(255), partition_id numeric, other long, distribution varchar2(30), cpu_cost numeric, io_cost numeric, temp_space numeric, access_predicates varchar2(4000), filter_predicates varchar2(4000)). Rem Drop and recreate SQLTEMP for taking a snapshot of the SQLAREA 0drop table sqltemp. create table sqltemp (ADDR VARCHAR2 (16), SQL_TEXT VARCHAR2 (2000), DISK_READS NUMBER, EXECUTIONS NUMBER, PARSE_CALLS NUMBER). set echo on Rem Create procedure to populate the plan_table by executing Rem explain plan...for sqltext dynamically create or replace procedure do_explain (addr IN varchar2, sqltext IN varchar2) as dummy varchar2 (1100). mycursor integer. ret integer. my_sqlerrm varchar2 (85). begin dummy:=EXPLAIN PLAN SET STATEMENT_ID= . dummy:=dummy||||addr|| || FOR ||sqltext. mycursor := dbms_sql.open_cursor. dbms_sql.parse(mycursor,dummy,dbms_sql.v7). ret := dbms_sql.execute(mycursor). dbms_sql.close_cursor(mycursor). commit. exception -- Insert errors into PLAN_TABLE... when others then my_sqlerrm := substr(sqlerrm,1,80). insert into plan_table(statement_id, remarks) values (addr,my_sqlerrm). --

close cursor if exception raised on EXPLAIN PLAN dbms_sql.close_cursor(mycursor). end. / Rem Start EXPLAINing all S/I/U/D statements in the shared pool declare -- exclude statements with v$sqlarea.parsing_schema_id = 0 (SYS) cursor c1 is 0select address, sql_text, DISK_READS, EXECUTIONS, PARSE_CALLS from v$sqlarea where command_type in (2,3,6,7) and parsing_schema_id != 0. cursor c2 is 0select addr, sql_text from sqltemp. addr2 varchar(16). sqltext v$sqlarea.sql_text%type. dreads v$sqlarea.disk_reads%type. execs v$sqlarea.executions%type. pcalls v$sqlarea.parse_calls%type. begin open c1. fetch c1 into addr2,sqltext, dreads,execs,pcalls. while (c1%found) loop insert into sqltemp values (addr2,sqltext,dreads,execs,pcalls). commit. fetch c1 into addr2, sqltext,dreads,execs,pcalls. end loop. close c1. open c2. fetch c2 into addr2, sqltext. while (c2%found) loop do_explain(addr2,sqltext). fetch c2 into addr2, sqltext. end loop. close c2. end. / Rem Generate a report of index usage based on the number of times Rem a SQL statement using that index was executed 0select p.owner, p.name, sum(s.executions) totexec from sqltemp s, (0select distinct statement_id stid, object_owner owner, object_name name from plan_table where operation = INDEX) p where s.addr = p.stid group by p.owner, p.name order by 2 desc. Rem Perform cleanup on exit (optional) 0delete from plan_table where statement_id in (0select addr from sqltemp). 0drop table sqltemp.关于这个脚本，有几个重要的地方需要注意，即它可能一起明显的开销，因此，应该在仔细地进行 权衡后才把它应用到繁忙的生产应用系统中区。 100Test 下载频道开通，各