一 SQL TUNING

1 表的常用链接方式

NESTED LOOP: 对于被连接的数据子集较小的情况，嵌套循环连接是较好的选择。在嵌套循环中，外表驱动内表，外表返回的每一行都要在内表中检索找到它匹配的行，因此整个查询返回的结果集不能太大（大于 10000 不合适），要把返回子集较小的表作为外表（驱动表），而且在内表的连接字段上一定要有索引。

HASH JOIN : 哈希连接是大数据集连接时常用的方式，优化器使用两个表中较小的表，利用连接键在内存中建立散列表，然后扫描较大的表并探测散列表，找出与散列表匹配的行。

这种方式适用于较小的表完全可以放入内存的情况，这样成本就是访问两个表的成本之和。但是在表很大的情况下并不能完全放入内存，这时优化器将它分割成若干不同的分区，不能放入内存的部分就把该分区写入磁盘的临时段。

MERGE JOIN 通常情况下哈希连接的效果都比排序合并连接要好。然而如果行源已经被排过序，在执行排序合并连接时不需要再排序了，这时排序归并连接的性能会优于哈希连接。

2 CBO 和 rule 的区别

Oracle 的优化器有两种优化方式,即基于规则的优化方式(Rule-Based Optimization，简称为 RBO)和基于代价的优化方式(Cost-Based Optimization，简称为 CBO)，

RBO 方式：优化器在分析 SQL 语句时,所遵循的是 Oracle 内部预定的一些规则。

CBO 方式：它是看语句的代价(Cost),这里的代价主要指 Cpu 和内存。优化器在判断是否用这种方式时,主要参照的是表及索引的统计信息

优化模式包括 Rule、Choose、First rows、All rows 四种方式

Rule：基于规则的方式。

Choolse：默认的情况下 Oracle 用的便是这种方式。指的是当一个表或或索引有统计信息，则走 CBO 的方式，如果表或索引没统计信息，表又不是特别的小，而且相应的列有索引时，那么就走索引，走 RBO 的方式。

First Rows：它与 Choose 方式是类似的，所不同的是当一个表有统计信息时，它将是以最快的方式返回查询的最先的几行，从总体上减少了响应时间。

All Rows:也就是我们所说的 Cost 的方式，当一个表有统计信息时，它将以最快的方式返回表的所有的行，从总体上提高查询的吞吐量。没有统计信息则走 RBO 的方式。

设定选用哪种优化模式：

A 、 Instance 级 别 我 们 可 以 通 过 在 initSID.ora 文 件 中 设 定 OPTIMIZER_MODE=RULE/CHOOSE/FIRST_ROWS/ALL_ROWS 如果没设定 OPTIMIZER_MODE 参数则默认用的是 Choose 方式。

B、Sessions 级别通过 ALTER SESSION SET OPTIMIZER_MODE=RULE/CHOOSE/FIRST_ROWS/ALL_ROWS 来设定。

C、语句级别用 Hint（/*+ ... */）来设定

1、优化模式是 all_rows 的方式

2、表作过 analyze，有统计信息

3、表很小

3 不借助第三方工具，如何查看 sql 的执行计划

SET AUTOTRACE ON;
exlplain plan for sql
select * from table（dbms_xplan.diskplay）

4 如何定位重要(消耗资源多)的 SQL

1、statspack-- 在你库上业务最忙得时候抓 15 分钟的 report,看里面的 top sql
1 top 找到消耗资源多的 pid
2 确定是 oracle 的应用进程还是后台进程
3 根据 v$session,v$process,v$sqlarea 定位

或者通过以下方式：

1）查看值得怀疑的 SQL

```
selectsubstr(to_char(s.pct,'99.00'),2)||'%'load,
       s.executions executes,
       p.sql_text
 from(selectaddress,
              disk_reads,
              executions,
              pct,
              rank()over(orderbydisk_readsdesc) ranking
         from(selectaddress,
                     disk_reads,
                     executions,
                    100*ratio_to_report(disk_reads)over() pct
               fromsys.v_$sql
              wherecommand_type!=47)
       wheredisk_reads>50*executions) s,
       sys.v_$sqltext p
wheres.ranking<=5
  andp.address=s.address
orderby1, s.address, p.piece;
```

2）查看消耗内存多的 sql

```
selectb.username ,a. buffer_gets ,a.executions,
a.disk_reads/decode(a.executions,0,1,a.executions),a.sql_text SQL
fromv$sqlarea a,dba_users b
wherea.parsing_user_id = b.user_id
 anda.disk_reads >10000
order bydisk_readsdesc;
```

3）查看逻辑读多的 SQL

```
select*
 from(selectbuffer_gets, sql_text
        fromv$sqlarea
       wherebuffer_gets>500000
       orderbybuffer_getsdesc)
whererownum<=30;
```

4）查看执行次数多的 SQL

```
selectsql_text, executions
 from(selectsql_text, executionsfromv$sqlareaorderbyexecutionsdesc)
whererownum<81;
```

5）查看读硬盘多的 SQL

```
selectsql_text, disk_reads
 from(selectsql_text, disk_readsfromv$sqlareaorderbydisk_readsdesc)
whererownum<21;
```

6）查看排序多的 SQL

```
selectsql_text, sorts
 from(selectsql_text, sortsfromv$sqlareaorderbysortsdesc)
whererownum<21;
```

7）分析的次数太多，执行的次数太少，要用绑变量的方法来写 sql

```
setpagesize600;
setlinesize120;
selectsubstr(sql_text,1,80) "sql",count(*),sum(executions) "totexecs"
 fromv$sqlarea
whereexecutions<5
groupbysubstr(sql_text,1,80)
havingcount(*)>30
orderby2;
```

8）游标的观察

```
setpages300;
selectsum(a.value), b.name
 fromv$sesstat a, v$statname b
wherea.statistic#=b.statistic#
  andb.name='opened cursors current'
groupbyb.name;

selectcount(0)fromv$open_cursor;
```

```
selectuser_name, sql_text,count(0)
 fromv$open_cursor
groupbyuser_name, sql_text
havingcount(0)>30;
```

9 查看当前用户&username 执行的 SQL

```
selectsql_text
 fromv$sqltext_with_newlines
where(hash_value, address)in
       (selectsql_hash_value, sql_address
        fromv$session
       whereusername='&username')
orderbyaddress, piece;
```

网友答案：select sql_text

```
from v$sql
where disk_reads > 1000 or (executions > 0 and buffer_gets/executions > 30000);
```

5 如何跟踪某个 session 的 sql

1） oracle 自带的 sql trace 程序可以跟踪本地 session

sys: alter system set sql_trace = true;对所有会话跟踪

schema: alter session set sql_trace = true； 对某个 session 会话跟踪 sql 语句

用 tkprof 来格式化跟踪文件输出

tkprof 输出内容包括 1 sql 语句 2 统计信息 3explain table 执行计划

2 ）基于 DBMS_MONITOR 包来跟踪会话，这种情况一般是

通过该程序包可以跟踪从客户机到中间层、再到后端数据库的任何用户会话，从而可以较为容易的标识创建大量工作量的特定用户。

会话：基于会话 ID 和序列号 DBMS_MONITOR.session_trace_enable(sid,serial#,true);

DBMS_MONITOR.session_trace_enable(sid,serial#,false);

客户端标识符：允许跨越多个会话设置跟踪，基于登录 ID 指定终端用户。

客户端设置的参数:DBMS_SESSION.SET_IDENTIFIER 过程设置该值

运行语句:dbms_monitor.client_id_trace_enable('identifier',true,false);

dbms_monitor.client_id_trace_disable('identifier');

实例：基于实例名指定给定的实例

dbms_monitor.database_trace_enable(instance_name=>'orcl');

服务名：指定一组相关的应用程序，使用 DBMS_SERVICE.CREATE_SERVICE 过程设置该值,

dbms_monitor.serv_mod_act_trace_enable(service_name=>'orcl1',module_name=>'salary_update',action_name=>'insert_item');

dbms_monitor.serv_mod_act_trace_disable(service_name=>'orcl1',module_name=>'salary_update',action_name=>'insert_item');

模块名：开发人员在其应用程序代码中使用 DBMS_APPLICATION_INFO.SET_MODULE 过程设置该值

操作名：开发人员在其应用程序中使用 DBMS_APPLICATION_INFO.SET_ACTION 过程设置该值 trcsess 提取跟踪文

件，可以基于会话 ID 或者模块名称，具体参考 trcsess 帮助 usage
这个很常用
3 ） DBMS_system 包也可以跟踪系统中其他 session 的 sql 执行语句
dbms_system.set_sql_trace_in_session(sid,serial#,true);
dbms_system.set_sql_trace(true)跟踪本地 session
4 ）使用 10046 事件来实现
运用 10046 进程 sql 跟踪
全局或者 session 范围
level 1
level 4 : 1+bind variable
level 8 : 1+wait events
level 12 : 1 +4+8
tkprof 的时候 4,8,12 的信息被忽略
全局设定，修改 initial 文件
event='10046 trace name context forever,level12'
alter system set events
session 设定
alter session set events='10046 trace name context forever,level 8';
alter session set events='10046 trace name context off';
以上 4 中 session 跟踪方法，通过 dbms_monitor,和 dbms_system 可以用来跟踪其他指定 session 的 sql，其他两个方法都是本地
session 或者 system 级别的跟踪

6 sql 调整最需要关注的是什么

调整的目的就是为了消耗最小的资源来完成功能，通过查看执行计划和各种统计信息来分辨调整后的 sql 对资源的耗费情况，来找出一个成本最小的 sql 语句

7 谈谈对索引的认知

索是某个表中一列或若干列值的集合和相应的指向表中物理标识这些值的数据页的逻辑指针清单，索引上的值是严格有序的
对 dml 的影响：
i 索引通常能提高 select/update/delete 的性能,会降低 insert 的速度,

8 使用索引一定能提供查询性能吗

一定会提高，并且随着你查询的数据量增加体现的更加明显。

9 绑定变量的含义和优缺点

绑定变量是相对文本变量来讲的,所谓文本变量是指在 SQL 直接书写查询条件，这样的 SQL 在不同条件下需要反复解析,绑定变量是指使用变量来代替直接书写条件，查询 bind value 在运行时传递，然后绑定执行。优点是减少硬解

析,降低 CPU 的争用,节省 shared_pool ;缺点是不能使用 histogram,sql 优化比较困难
绑定变量是 Oracle 解决硬解析的首要利器，能解决 OLTP 系统中 library cache 的过度耗用以提高性能。首先其实质是变量，有些类似于我们经常使用的替代变量，替代变量使用&占位符，只不过绑定变量使用:
优点：可以在 library cache 中共享游标，避免硬解析以及与之相关的额外开销
在大批量数据操作时将呈数量级来减少闩锁的使用，避免闩锁的竞争
缺点：绑定变量被使用时，查询优化器会忽略其具体值，因此其预估的准确性远不如使用字面量值真实，尤其是在表存在数据倾斜(表上的数据非均匀分布)的列上会提供错误的执行计划。从而使得非高效的执行计划被使用。

10 如何固定执行计划

使用 RBO 优化器
ALTER SESSION SET OPTIMIZER_MODE=RULE/CHOOSE/FIRST_ROWS/ALL_ROWS
加提示/*+hints*/
锁定统计信息
dbms_stats
采用存储概念
网上答案：
query_rewrite_enabled = true
star_transformation_enabled = true
optimizer_features_enable = 9.2.0

11 pga 和 sga 的管理方式，临时表空间的作用

8I：pga 手动管理，sga 手动管理
9i：pga 自动管理，sga 半自动管理
10g：pga 自动管理，sga 自动管理，但互不相关
11g;pga 和 sga 统一使用 assm 管理
临时表空间用来管理数据库排序操作以及用于存储临时表、中间排序结果等临时对象，当 ORACLE 里需要用到 SORT 的时候，并且当 PGA 中 sort_area_size 大小不够时，将会把数据放入临时表空间里进行排序
Oracle 8i 中 sort_area_size/sort_area_retained_size 决定了排序所需要的内存

如果排序操作不能在 sort_area_size 中完成,就会用到 temp 表空间

Oracle 9i 中如果 workarea_size_policy=auto 时,

排序在 pga 内进行,通常 pga_aggregate_target 的 1/20 可以用来进行 disk sort;

如果 workarea_size_policy=manual 时,排序需要的内存由 sort_area_size 决定

在执行 order by/group by/distinct/union/create index/index rebuild/minus 等操作时,

如果在 pga 或 sort_area_size 中不能完成,排序将在临时表空间进行(disk sort),

临时表空间主要作用就是完成系统中的 disk sort.

12 存在表 T(a,b,c,d),要根据字段 c 排序后取第 21—30 条记录显示，请给出 sql

1.select * from (select c.*,rownum as rn from (select * from t order by c desc) c) where rn between 21 and 30

二 数据库基本概念类

1:pctused and pctfree 表示什么含义有什么作用

pctused 与 pctfree 控制数据块是否出现在 freelist 中,

pctfree 控制数据块中保留用于 update 的空间,当数据块中的 free space 小于 pctfree 设置的空间时,

该数据块从 freelist 中去掉,当块由于 dml 操作 free space 大于 pct_used 设置的空间时,该数据库块将

被添加在 freelist 链表中。

2:简单描述 table / segment / extent / block 之间的关系

table 创建时,默认创建了一个 data segment,

每个 data segment 含有 min extents 指定的 extents 数,

每个 extent 据据表空间的存储参数分配一定数量的 blocks

3:描述 tablespace 和 datafile 之间的关系

一个 tablespace 可以有一个或多个 datafile,每个 datafile 只能在一个 tablespace 内,

table 中的数据,通过 hash 算法分布在 tablespace 中的各个 datafile 中,

tablespace 是逻辑上的概念,datafile 则在物理上储存了数据库的种种对象。

4:本地管理表空间和字典管理表空间的特点，ASSM 有什么特点

本地管理表空间(Locally Managed Tablespace 简称 LMT)

8i 以后出现的一种新的表空间的管理模式，通过位图来管理表空间的空间使用。

字典管理表空间(Dictionary-Managed Tablespace 简称 DMT)

8i 以前包括以后都还可以使用的一种表空间管理模式，通过数据字典管理表空间的空间使用。

动段空间管理(ASSM)，

它首次出现在 Oracle920 里有了 ASSM，链接列表 freelist 被位图所取代，它是一个二进制的数组，

能够迅速有效地管理存储扩展和剩余区块(free block)，因此能够改善分段存储本质，

ASSM 表空间上创建的段还有另外一个称呼叫 Bitmap Managed Segments(BMB 段)。

5:回滚段的作用是什么

事务回滚：当事务修改表中数据的时候，该数据修改前的值(即前影像)会存放在回滚段中，

当用户回滚事务(ROLLBACK)时，ORACLE 将会利用回滚段中的数据前影像来将修改的数据恢复到原来的值。

事务恢复：当事务正在处理的时候，例程失败，回滚段的信息保存在 undo 表空间中，

ORACLE 将在下次打开数据库时利用回滚来恢复未提交的数据。

读一致性：当一个会话正在修改数据时，其他的会话将看不到该会话未提交的修改。

当一个语句正在执行时，该语句将看不到从该语句开始执行后的未提交的修改(语句级读一致性)

当 ORACLE 执行 Select 语句时，ORACLE 依照当前的系统改变号(SYSTEM CHANGE NUMBER-SCN)

来保证任何前于当前 SCN 的未提交的改变不被该语句处理。可以想象：当

一个长时间的查询正在执行时，

若其他会话改变了该查询要查询的某个数据块，ORACLE 将利用回滚段的数据前影像来构造一个读一致性视图。

6:日志的作用是什么

记录数据库事务,最大限度地保证数据的一致性与安全性

重做日志文件：含对数据库所做的更改记录，这样万一出现故障可以启用数据恢复,一个数据库至少需要两个重做日志文件

归档日志文件：是重做日志文件的脱机副本，这些副本可能对于从介质失败中进行恢复很必要。

7:SGA 主要有那些部分，主要作用是什么

SGA：db_cache/shared_pool/large_pool/java_pool

db_cache:

数据库缓存(Block Buffer)对于 Oracle 数据库的运转和性能起着非常关键的作用，

它占据 Oracle 数据库 SGA(系统共享内存区)的主要部分。Oracle 数据库通过使用 LRU

算法，将最近访问的数据块存放到缓存中，从而优化对磁盘数据的访问.

shared_pool:

共享池的大小对于 Oracle 性能来说都是很重要的。

共享池中保存数据字典高速缓冲和完全解析或编译的的 PL/SQL 块和 SQL 语句及控制结构

large_pool:

使用 MTS 配置时，因为要在 SGA 中分配 UGA 来保持用户的会话，就是用

Large_pool 来保持这个会话内存

使用 RMAN 做备份的时候，要使用 Large_pool 这个内存结构来做磁盘 I/O 缓存器

java_pool:

为 java procedure 预备的内存区域,如果没有使用 java proc,java_pool 不是必须的

8 oracle 系统进程主要有哪些，作用是什么

数据写进程(dbwr)：负责将更改的数据从数据库缓冲区高速缓存写入数据文件

日志写进程(lgwr)：将重做日志缓冲区中的更改写入在线重做日志文件

系统监控(smon) ：检查数据库的一致性如有必要还会在数据库打开时启动数据库的恢复

进程监控(pmon) ：负责在一个 Oracle 进程失败时清理资源

检查点进程(chpt)：负责在每当缓冲区高速缓存中的更改永久地记录在数据库中时,更新控制文件和数据文件中的数据库状态信息。

归档进程(arcn) ：在每次日志切换时把已满的日志组进行备份或归档

作业调度器(cjq) :负责将调度与执行系统中已定义好的 job,完成一些预定义的工作.

恢复进程(reco) :保证分布式事务的一致性,在分布式事务中,要么同时 commit,要么同时 rollback;

三 备份恢复类

1:备份如何分类

逻辑备份：exp/imp
物理备份：
RMAN 备份
full backup/incremental backup(累积/差异)
热备份:alter tablespace begin/end backup;
冷备份:脱机备份(database shutdown)

2:归档是什么含义

关于归档日志：Oracle 要将填满的在线日志文件组归档时,则要建立归档日志(archived redo log)。
其对数据库备份和恢复有下列用处：
数据库后备以及在线和归档日志文件，在操作系统和磁盘故障中可保证全部提交的事物可被恢复。
在数据库打开和正常系统使用下，如果归档日志是永久保存，在线后备可以进行和使用。

数据库可运行在两种不同方式下：
NOARCHIVELOG 方式或 ARCHIVELOG 方式数据库在 NOARCHIVELOG 方式下使用时，不能进行在线日志的归档,如果数据库在 ARCHIVELOG 方式下运行，可实施在线日志的归档。

3:如果一个表在 2004-08-04 10:30:00 被 drop，在有完善的归档和备份的情况下，如何恢复?

手工拷贝回所有备份的数据文件
startup mount;
sql> alter database recover automatic until time '2004-08-04:10:30:00';
alter database open resetlogs;

4:rman 是什么,有何特点?

RMAN(Recovery Manager)是 DBA 的一个重要工具，用于备份、还原和恢复 oracle 数据库, RMAN 可以用来备份和恢复数据库文件、归档日志、控制文件、系统参数文件,也可以用来执行完全或不完全的数据库恢复。RMAN 有三种不同的用户接口：
COMMAND LINE 方式、GUI 方式(集成在 OEM 中的备份管理器)、API 方式(用于集成到第三方的备份软件中)。
具有如下特点：
1)功能类似物理备份，但比物理备份强大 N 倍;
2)可以压缩空块;
3)可以在块水平上实现增量;
4)可以把备份的输出打包成备份集，也可以按固定大小分割备份集;
5)备份与恢复的过程可以自动管理;
6)可以使用脚本(存在 Recovery catalog 中)
7)可以做坏块监测

5:standby 的特点

备用数据库(standby database)：ORACLE 推出的一种高可用性(HIGH AVAILABLE)数据库方案，
在主节点与备用节点间通过日志同步来保证数据的同步，备用节点作为主节点的备份
可以实现快速切换与灾难性恢复,从 920 开始，还开始支持物理与逻辑备用服务器。
Oracle 9i 中的三种数据保护模式分别是：
1)、MAXIMIZE PROTECTION ：最大数据保护与无数据分歧，LGWR 将同时传送到备用节点，
在主节点事务确认之前，备用节点也必须完全收到日志数据。如果网络不好，引起 LGWR 不能传送数据，将引起严重的性能问题，导致主节点 DOWN 机。
2)、MAXIMIZE AVAILABILITY ：无数据丢失模式，允许数据分歧，允许异步传送。
正常情况下运行在最大保护模式，在主节点与备用节点的网络断开或连接不正常时，自动切换到最大性能模式，

主节点的操作还是可以继续的。在网络不好的情况下有较大的性能影响。

3)、MAXIMIZE PERFORMANCE：这种模式应当可以说是从 8i 继承过来的备用服务器模式，异步传送，无数据同步检查，可能丢失数据，但是能获得主节点的最大性能。9i 在配置 DATA GUARD 的时候默认就是 MAXIMIZE PERFORMANCE

6:对于一个要求恢复时间比较短的系统(数据库 50G,每天归档 5G)，你如何设计备份策略
rman/每月一号 level 0 每周末/周三 level 1 其它每天 level 2

四调优

1 对于一个存在系统性能的系统，说出你的诊断处理思路

1 做 statspack 收集系统相关信息
了解系统大致情况/确定是否存在参数设置不合适的地方/查看 top 5 event/查看 top sql 等
2 查 v$system_event/v$session_event/v$session_wait
从 v$system_event 开始,确定需要什么资源(db file sequential read)等
深入研究 v$session_event,确定等待事件涉及的会话
从 v$session_wait 确定详细的资源争用情况(p1-p3 的值:file_id/block_id/blocks 等)
3 通过 vsql/vsqltext/v$sqlarea 表确定 disk_reads、(buffer_gets/executions)值较大的 SQL

2:列举几种诊断 IO、CPU、性能状况的方法

top/vmstat
statspack
sql_trace/tkprof
查 v$system_event/v$session_event/v$session_wait
查 v$sqlarea(disk_reads 或 buffer_gets/executions 较大的 SQL)

3:对 statspack 有何认识

StapSpack 是 Oracle 公司提供的一个收集数据库运行性能指标的软件包，该软件包从 8i 起，在 9i、10g 都有显著的增强。该软件包的辅助表(存储相关参数与收集的性能指标的表)由最初的 25 个增长到 43 个。收集级别参数由原来的 3 个(0、5、10)增加到 5 个(0、5、 6、7、10)通过分析收集的性能指标，数据库管理员可以详细地了解数据库目前的运行情况，对数据库实例、等待事件、SQL 等进行优化调整。利用 statspack 收集的 snapshot,可以统计制作数据库的各种性能指标的统计趋势图表。

4

:如果系统现在需要在一个很大的表上创建一个索引，你会考虑那些因素，如何做以尽量减小对应用的影响
在系统比较空闲时;nologging 选项(如果有 dataguard 则不可以使用 nologging)大的 sort_ared_size 或 pga_aggregate_target 较大

5:对 raid1+0 和 raid5 有何认识

RAID 10(或称 RAID 1+0)与 RAID 0+1 不同，它是用硬盘驱动器先组成 RAID 1 阵列，然后在 RAID 1 阵列之间再组成 RAID 0 阵列。RAID 10 模式同 RAID 0+1 模式一样具有良好的数据传输性能，但却比 RAID 0+1 具有更高的可靠性。RAID 10 阵列的实际容量为 M×n/2，磁盘利用率为 50%。RAID 10 也需要至少 4 个硬盘驱动器构成，因而价格昂贵。 RAID 10 的可靠性同 RAID 1 一样，但由于 RAID 10 硬盘驱动器之间有数据分割，因而数据传输性能优良。RAID 5 与 RAID 3 很相似，不同之处在于 RAID 5 的奇偶校验信息也同数据一样被分割保存到所有的硬盘驱动器，而不是写入一个指定的硬盘驱动器，从而消除了单个奇偶校验硬盘驱动器的瓶颈问题。RAID 5 磁盘阵列的性能比 RAID 3 有所提高，但仍然需要至少 3 块硬盘驱动器。其实际容量为 M×(n-1)，磁盘利用率为(n-1)/n 。