



高效SQL
研发中心 曾庆典

GUANGZHOU KETENG INFORMATION TECHNOLOGY CO.,LTD.
PINGYUN RD,WEST HUANGPU AVE,GUANGZHOU

目录

第一章 Oracle Optimizer

第二章 SQL语句的执行步骤

第三章 诊断工具

第五章 本章

(无小节要点提示)

第一章 Oracle Optimizer

什么是Optimizer?



- 优化器(Optimizer) 是一组Oracle的内部例行程序，该例行程序用以决定SQL语句的最有效的访问路径。
- Oracle数据库会自动进行查询优化,这些操作对于用户来说是透明的
- 优化器模式通过初始化参数
- optimizer_mode设置

优化器的作用



- 评估表达式和条件
- 将语句转换为等价语句
- 决定如何访问数据
- 决定如何连接数据
- 决定最有效的访问路径

优化模式



在Oracle9i, 两种优化模式可以被选择:

- 基于规则的 Rule-based:
 - 使用一个分级系统
 - 语法(Syntax)驱动和字典(dictionary)驱动的
 - Oracle从早期版本提供RBO优化器, 至Oracle10g该优化器不再被支持
- 基于代价的 Cost-based:
 - 选择最低代价的路径
 - 统计(Statistics)驱动
 - 从Oracle7(1992)----->Oracle10g主要支持的优化器

设置优化模式



- 在实例级:

```
optimizer_mode =  
  {choose|rule|first_rows|first_rows  
  _n|  
  all_rows}
```

- 在会话级:

```
alter session set optimizer_mode =  
  {choose|rule|first_rows|first_rows  
  _n|  
  all_rows}
```

- 在语句级: 使用提示(hints)

```
/*+ FIRST_ROWS (x) */
```

What's New In Oracle10g?



- RBO不再被支持
- 缺省的优化器模式为all_rows

- Optimizer_mode

choose/rule 参数不再支持

all_rows/first_rows/first_rows_n 是Oracle10g支持的优化目标

Connected to:

Oracle Database 10g Enterprise Edition Release 10.1.0.2.0 - 64bit Production
With the Partitioning, OLAP and Data Mining options

SQL> show parameter optimizer_mode

NAME	TYPE	VALUE
optimizer_mode	string	ALL_ROWS

Why RBO is Dead?



```
SQL> select t1.object_name,t2.object_name
2 from big_table t1,big_table t2
3 where t1.object_id = t2.object_id
4 and t1.owner = 'WMSYS'
5 /
```

Execution Plan

```
-----
0  SELECT STATEMENT Optimizer=ALL_ROWS (Cost=619 Card=10989 Bytes=670329)
1  0  HASH JOIN (Cost=619 Card=10989 Bytes=670329)
2    1  TABLE ACCESS (FULL) OF 'BIG_TABLE' (TABLE) (Cost=309 Card=1000 Bytes=39000)
3    1  TABLE ACCESS (FULL) OF 'BIG_TABLE' (TABLE) (Cost=304 Card=99989 Bytes=2199758)
```

```
SQL> select /*+ rule */ t1.object_name,t2.object_name
2 from big_table t1,big_table t2
3 where t1.object_id = t2.object_id
4 and t1.owner = 'WMSYS'
5 /
```

Execution Plan

```
-----
0  SELECT STATEMENT Optimizer=HINT: RULE
1  0  TABLE ACCESS (BY INDEX ROWID) OF 'BIG_TABLE' (TABLE)
2    1  NESTED LOOPS
3      2  TABLE ACCESS (FULL) OF 'BIG_TABLE' (TABLE)
4      2  INDEX (RANGE SCAN) OF 'OBJECT_ID_IDX' (INDEX)
```

Continue...



CBO:

call	count	cpu	elapsed	disk	query	current	rows
Parse	1	0.01	0.00	0	0	0	0
Execute	1	0.00	0.00	0	0	0	0
Fetch	1652	0.94	0.94	2636	4304	0	24763
total	1654	0.95	0.95	2636	4304	0	24763

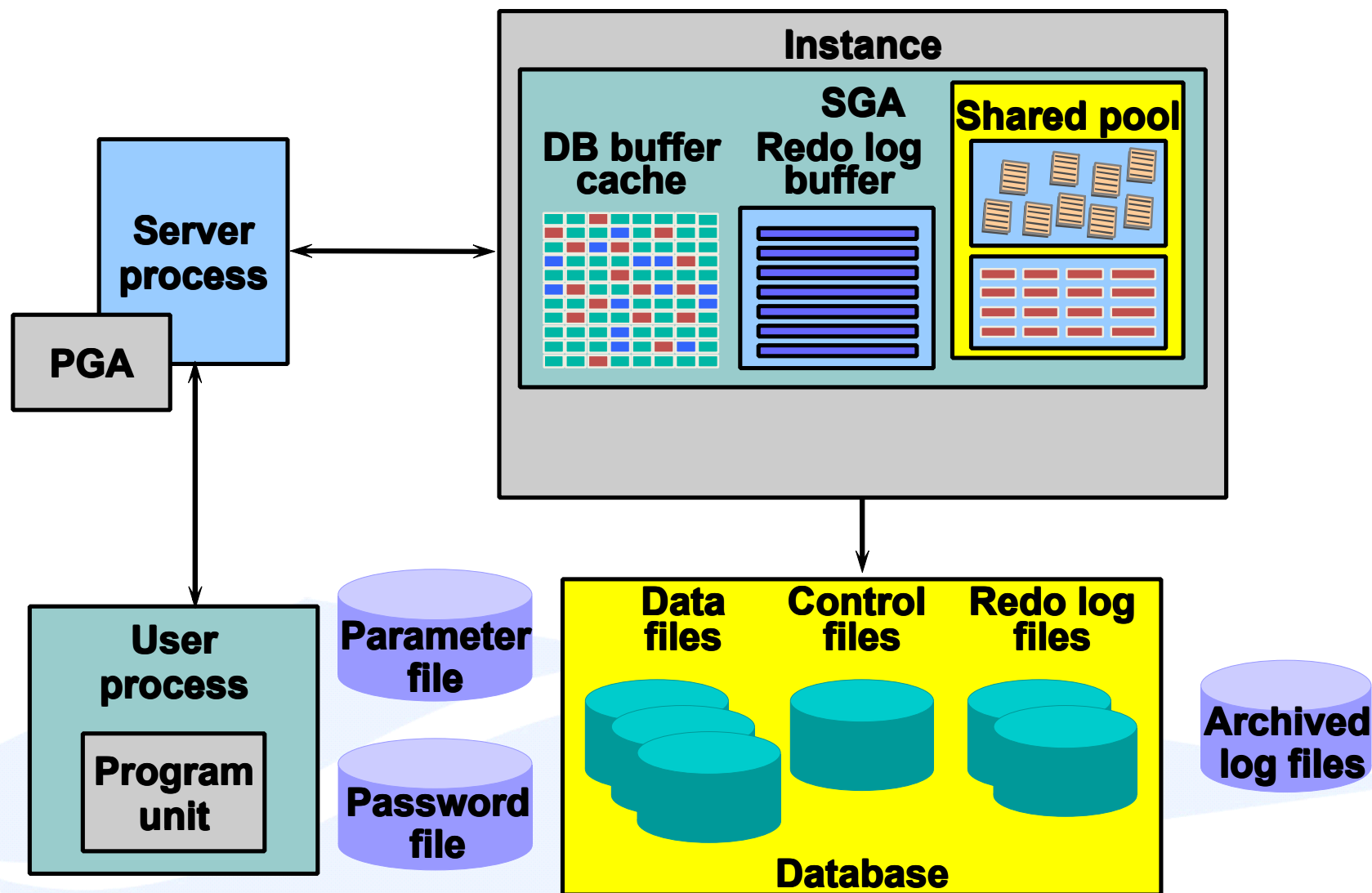
RBO:

call	count	cpu	elapsed	disk	query	current	rows
Parse	1	0.01	0.00	0	0	0	0
Execute	1	0.00	0.00	0	0	0	0
Fetch	1652	57.69	56.26	1544	1202607	0	24763
total	1654	57.70	56.27	1544	1202607	0	24763

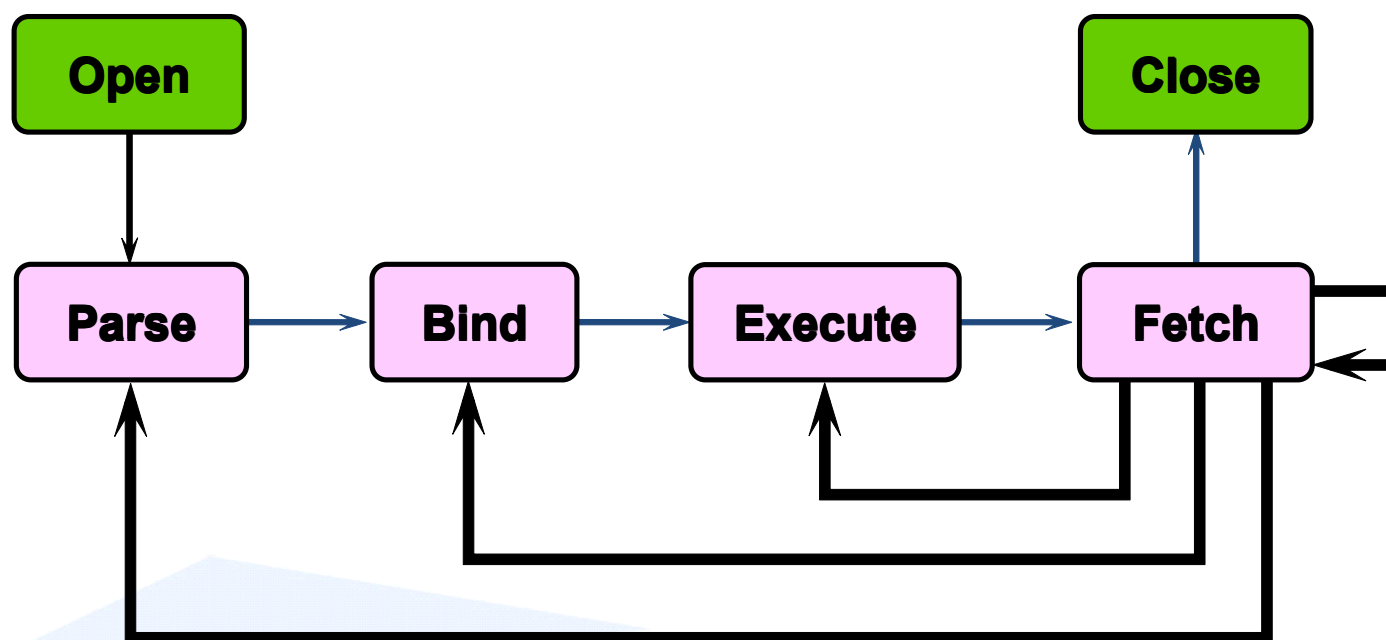
(无小节要点提示)

第二章 SQL语句的执行步骤

SQL语句的执行步骤



SQL 语句处理过程



SQL语句处理过程



– Parse

- Searches for the statement in the shared pool
- Checks syntax
- Checks semantics and privileges
- Merges view definitions and subqueries
- Determines execution plan

SQL语句处理过程



— Bind

- Scans the statement for bind variables
- Assigns (or reassigns) a value

SQL语句处理过程



– Execute

- Applies the execution plan
- Performs necessary I/O and sorts for data manipulation language (DML) statements

– Fetch

- Retrieves rows for a query
- Sorts for queries when needed
- Uses an array fetch mechanism

(无小节要点提示)

第三章 诊断工具

诊断工具概述



- STATSPACK
- EXPLAIN PLAN
- SQL trace 和 TKPROF
- SQL*Plus autotrace 特性
- Oracle SQL Analyze

在STATSPACK中关于SQL的报告

STATSPACK 收集了下列关于SQL的信息:

- SQL ordered by gets
- SQL ordered by reads
- SQL ordered by executions
- SQL ordered by parse calls

EXPLAIN PLAN



- 不需要打开跟踪就可以使用
- 需要 PLAN_TABLE 表:

```
SQL> @$ORACLE_HOME/rdbms/admin/utlxplan
```

- 创建解释表:

```
SQL> Explain plan for  
2 select last_name from hr.employees;
```

- 查询 plan_table 表显示执行计划:
 - 直接查询 PLAN_TABLE
 - 使用脚本 utlxpls.sql (隐藏了并行查询的信息)
 - 使用脚本 utlxplp.sql (显示并行查询信息)

The EXPLAIN PLAN Command



→ **EXPLAIN PLAN** →
[**SET STATEMENT_ID**
= '*text*']
→
[**INTO *your plan table***]
→
→ **FOR *statement*** →

显示执行计划



```
SQL> select  id
2      ,      lpad(' ',2*level)||operation||
3             decode(id, 0,' Cost = '||position)
4             ||' '||options
5             ||' '||object_name  as "Query Plan"
6 from      plan_table
7 where     statement_id = 'demo01'
8 connect by prior id = parent_id start with id=0;
```

ID Query Plan

```
-----
0 SELECT STATEMENT Cost =
1 TABLE ACCESS BY INDEX ROWID PRODUCTS
2 AND-EQUAL
3 INDEX RANGE SCAN PRODUCTS_PROD_CAT_IX
4 INDEX RANGE SCAN PRODUCTS_PROD_SUBCAT_IX
```

将执行计划保留在内存中

- 使用这个特性, Oracle服务器将保留一个cache 在内存中的SQL语句的执行计划.
- 当SQL语句在库高速缓存中过期时,相应的cache执行计划也被移走.
- 使用这个特性的主要益处在于能够更好的诊断查询的性能.

察看内存中执行计划的视图

- 动态性能视图V\$SQL_PLAN可以被用来察看cached游标的执行计划.
- 这个视图包含了PLAN_TABLE的所有列 (除了LEVEL列), 另外还新增了7个列.
- 在PLAN_TABLE出现的列在V\$SQL_PLAN中也有相同的值
- 在V\$SQL中也新增了列 .

在 V\$SQL 视图中新增加列

用于支持保留在内存中的执行计划

- 新列 PLAN_HASH_VALUE 已经被增加到 V\$SQL 视图中.
- 这个列是构建于相应执行计划的一个散列 (hash) 值.
- 这个列可被用来象比较 SQL 文本的 HASH_VALUE 列一样比较游标的计划.

查询V\$SQL_PLAN



```
SELECT id
      , lpad (' ', depth) || operation operation
      , options , object_name , optimizer , cost
FROM V$SQL_PLAN
WHERE hash_value = 912244748
      AND address      = '67D419DC'
START WITH id = 0
CONNECT BY
      (      prior id          = parent_id
        AND prior hash_value    = hash_value
        AND prior child_number = child_number
      )
ORDER SIBLINGS BY id, position;
```

ID	OPERATION	OPTIONS	OBJECT_NAME	OPT	COST
0	SELECT STATEMENT			CHO	
1	TABLE ACCESS	BY INDEX ROWID	PRODUCTS		
2	AND-EQUAL				
3	INDEX	RANGE SCAN	PRODUCTS_PROD_CAT_IX		
4	INDEX	RANGE SCAN	PRODUCTS_PROD_SUBCAT_I		

SQL*Plus AUTOTRACE



- 创建 PLAN_TABLE 表
- 创建和授予 Plustrace 角色

```
SQL> @$ORACLE_HOME/sqlplus/admin/plustrce.sql  
SQL> grant plustrace to scott;
```

```
set autotrace [ off | on | traceonly ]  
               [ explain | statistics ]
```

使用 SQL Trace 和 TKPROF

为了使用 SQL trace 和 TKPROF:

- 设置初始化参数.
- **Alter session set SQL_Trace = true**
- 运行应用.
- **Alter session set SQL_Trace = false**
- 使用 TKPROF 格式化显示跟踪文件的内容.
- 解释输出的结果.

启用或禁用 SQL Trace



- 在实例级:

SQL_TRACE = {TRUE|FALSE}

- 在会话级:

```
SQL> alter session set SQL_TRACE = {true|false};
```

```
SQL> execute DBMS_SESSION.SET_SQL_TRACE  
2          ({true|false});
```

```
SQL> execute DBMS_SYSTEM.SET_SQL_TRACE_IN_SESSION  
2          (session_id, serial_id, {true|false});
```

- 其它方法:

```
Dbms_system.set_ev  
Event 10046
```

格式化跟踪文件



```
OS> tkprof tracefile outputfile [options]
```

- TKPROF command examples:

```
OS> tkprof
```

```
OS> tkprof ora_901.trc run1.txt
```

```
OS> tkprof ora_901.trc run2.txt sys=no  
sort=execpu print=3
```

TKPROF 命令的选项



- **`SORT = option`**
- **`PRINT = n`**
- **`EXPLAIN = username/password`**
- **`INSERT = filename`**
- **`SYS = NO`**
- **`RECORD = filename`**

TKPROF 提供的统计信息

- Count: 执行调用的次数
- CPU: CPU 使用的毫秒数
- Elapsed: 总的持续时间
- Disk: 物理读数
- Query: 一致性读(consistent read)的逻辑读数
- Current: 当前模式(current mode)的逻辑读数
- Rows: 处理的行数

TKPROF 的其他输出



- TKPROF 也输出下面的这些信息：
 - Recursive SQL statements
 - Library cache misses
 - Parsing user ID
 - Execution plan
 - Optimizer mode or hint

TKPROF 输出例子: No Index

```
...
select cust_first_name, cust_last_name, cust_city, cust_state_province
from customers
where cust_last_name = 'Smith'
```

call	count	cpu	elapsed	disk	query	current	rows
Parse	3	801.14	800.00	0	0	1	0
Execute	3	0.00	0.00	0	0	0	0
Fetch	21	1802.59	9510.00	48	4835	10	234
total	27	2603.73	10310.00	48	4835	11	234

Misses in library cache during parse: 3
Optimizer goal: CHOOSE
Parsing user id: 44

Rows	Row Source Operation
78	TABLE ACCESS FULL CUSTOMERS

```
...
```

TKPROF 输出例子: Unique Index

```
select cust_first_name, cust_last_name, cust_city, cust_state_province
from customers
where cust_last_name = 'Smith'
```

call	count	cpu	elapsed	disk	query	current	rows
Parse	1	0.00	0.00	0	0	0	0
Execute	1	0.00	0.00	0	0	0	0
Fetch	7	100.14	1500.00	2	87	0	78
total	9	100.14	1500.00	2	87	0	78

Misses in library cache during parse: 1

Optimizer goal: CHOOSE

Parsing user id: 44

Rows	Row Source Operation
78	TABLE ACCESS BY INDEX ROWID CUSTOMERS
78	INDEX RANGE SCAN (object id 32172)

一些TKPROF 信息的误区

- Read consistency trap
- Schema trap
- Time trap
- Trigger trap

谢谢聆听
Thanks



广州科腾信息技术有限公司

广州市天河区珠江新城华夏路10号11楼1105室505-80488 100 100 51088

地址：广州市天河区珠江新城华夏路10号11楼1105室505-80488 100 100 51088

邮编：510660 510660 510660 510660 510660

网址：www.kiten.com.cn 服务热线：400 0121 222