

基于 Oracle 分布式数据库的查询优化

王 君, 祝永志, 魏榕晖, 李丙锋

(曲阜师范大学 计算机科学学院, 山东 日照 276826)

摘 要:随着信息和网络技术的发展, 分布式数据库得到了广泛的发展, 其中 Oracle 分布式数据库在企业管理中得到了极大的应用。为了能够更好地提高企业的竞争力, 对 Oracle 分布式数据库的查询进行优化成为必要。从介绍 Oracle 分布式数据库的两种基本结构出发, 重点介绍了几种不同的 Oracle 分布式数据库的查询优化方法及其 SQL 语句的实现, 并且通过试验对优化方法的性能进行了测试, 然后通过数据的对比体现出了查询优化前后的差别, 最后给出了可能影响查询优化性能的因素。

关键词: Oracle; 分布式; 数据库; 查询; 优化

中图分类号: TP311.133.1

文献标识码: A

文章编号: 1673-629X(2008)01-0157-04

Optimizing of Query Based on Oracle Distributed Database

WANG Jun, ZHU Yong-zhi, WEI Rong-hui, LI Bing-feng

(College of Computer Science, Qufu Normal University, Rizhao 276826, China)

Abstract: With the development of the information and Internet technology, the distributed database has made many public applications, in which Oracle distributional database obtained the enormous application in the business management. In order to get better enhance of the enterprise's competitive power, it is necessary to carry on the Oracle distributional database inquiry optimization. Mainly proposes some methods with the SQL sentences realization to the optimizing of query based on the introduction of the two kinds of structure about the Oracle distributed database, and carry on the test of the methods' performance through the experiment, then manifest the difference through the data contrast, and give some possibly factors of affecting the inquiry optimization performance.

Key words: Oracle; distributed; database; query; optimizing

0 引 言

分布式数据库是计算机网络与数据库的结合产物, 对比于集中式数据库来说具有高扩展性和可靠性, 能够实现资源共享, 提高数据的利用率, 并且能够实现分布性数据的远程控制, Oracle 分布式数据库则是在 Oracle 数据库本身的特点上将分布式数据库的优点融入^[1]。

1 Oracle 分布式数据库结构

Oracle 分布式数据库分为同构分布式数据库系统与异构分布式数据库系统两种。但两者都是采用了客户/服务器结构。Oracle 同构分布式数据库系统是由一台或多台计算机上的两个或多个 Oracle 数据库组成

的, 每台计算机可以同时访问多个数据库中的数据。Oracle 异构分布式数据库系统中则至少有一个数据库是非 Oracle 数据库, 因此当访问非 Oracle 数据库时就需要通过透明网关访问, 或者通过异构服务(HS)的 ODBC 代理 OLE DB 代理来进行访问。这两种分布式数据库系统的结构如图 1、图 2 所示。

当访问远程数据库时, 需要建立数据库链接。数据库链接是一个定义了两个数据库服务器间单向通信路径的指针。通过数据库链接你可以拥有一定的权限访问远程数据库的对象, 而省去到远程数据库注册新用户的麻烦。数据库链接的名称通常与该数据库的全局数据库名同名, 并且都是唯一的。全局数据库名由数据库名加数据库的网络域名构成, 其中网络的域名可以包含多个层次。例如: 某数据库的全局数据库名为 oracle1.rizhao.qufushifan.shandong, 则数据库链接的名称也为 oracle1.rizhao.qufushifan.shandong, 其中 oracle1 为数据库名, rizhao.qufushifan.shandong 为网络域名。通过这个数据库链接就可以访问远程数据库的模式对象, 例如:

收稿日期: 2007-03-23

基金项目: 山东省高等学校实验研究项目基金(2005-400)

作者简介: 王 君(1982-), 男, 山东青州人, 硕士研究生, 主要研究方向为网络与分布式数据库; 祝永志, 教授, 硕士生导师, 主要研究方向为网络与分布式系统。

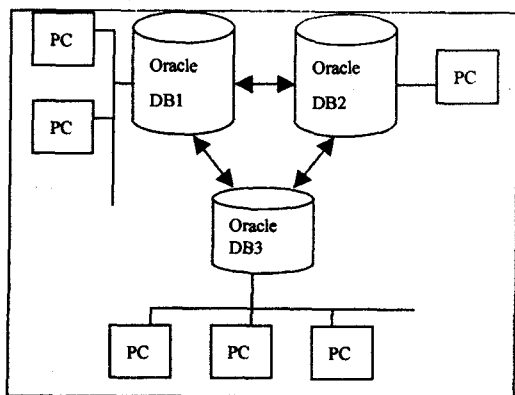


图 1 Oracle 同构分布式数据库系统

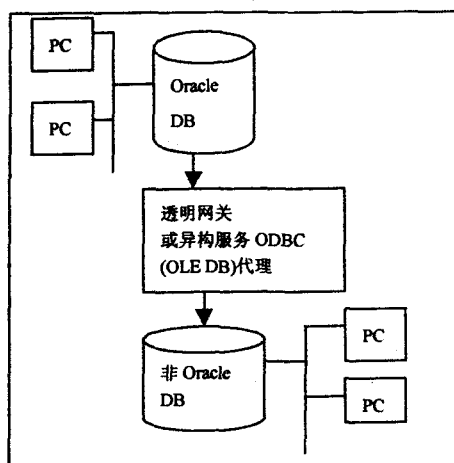


图 2 Oracle 异构分布式数据库系统

```
SELECT * FROM Scott. student@
```

```
oracle1.rizhao.qufushifan.shandong.
```

这样就可以通过数据库链接访问远程数据库的数据表 student^[2,3]。

2 Oracle 分布式数据库的查询优化

分布式数据库的数据查询涉及到多个数据库的远程查询,通常需要将数据查询分解为多个远程查询并将其送到远程节点,在远程节点执行查询后将结果返回给本地结点。

为了进行优化处理可以采用如下几种方法。

2.1 利用并行行中视图

优化分布式查询最好的方法是尽可能少地访问远程数据库并只检索所需的数据。当访问远程数据库的多个表时,可以通过改写该查询,使得仅查询该数据库一次,只将最少的数据传送到查询的执行站点,来改善查询性能。要重写查询,可以利用并行行中视图来实现。要访问一个远程数据库 abc.school,其中的多个表 student,class 中的数据例子:

```
SELECT stu.name, stu.no, class.no FROM
(SELECT name, no FROM student
```

```
@abc.school) stu, class;
```

其中括号里嵌入的 select 语句就是行中视图,stu 为视图别名。

再定义 Collocated (位于同一数据库中的两个或多个表)和 Inline view (在父 SELECT 语句中代替一个表的 SELECT 语句),那么 Collocated Inline view 就代表从单个数据库中的多个表中选取数据的一个行中视图,这样就减少了访问远程数据库的时间,改善了分布式查询的性能。同样,如果采用并行行中视图形成分布式查询,可以提高分布式查询的性能。

2.2 基于代价的优化

Oracle 基于代价的优化策略,可以透明地重写大多数分布查询,因此可以提高分布式查询的性能。例如要查询两个远程数据库中的表 remote a, remote b, 以及一个当地数据库中的表 local c 中的数据:

```
SELECT a.name, a.no, b.age, b.no, c.birthday, c.no
FROM remote a, remote b, local c
WHERE a.no=b.no
AND c.no=c.no
```

可以重写成:

```
SELECT r.name, r.no, r.age, r.no, c.birthday, c.no
FROM (SELECT a.age, a.no, b.birthday, b.no
FROM remote a, remote b
WHERE a.no=b.no) r, local c
WHERE c.no=r.no
```

这样在多个远程站点通过构造行中视图减少了查询数量,减少了网络传输代价。这种基于代价的优化策略可以被系统设置,当发布查询时,自动对查询进行优化,从而达到对用户的透明性。设置 Oracle 的优化器需要经过两个步骤:设置初始化参数和分析表。首先,修改初始化参数 OPTIMIZER_MODE,可以发布 ALTER SESSION 语句在会话层设置:

```
ALTER SESSION OPTIMIZER_MODE
=CHOOSE;
ALTER SESSION OPTIMIZER_MODE
=COST;
```

接着分析查询表,执行 ANALYZE 语句精确统计表中数据,为选择最佳的优化路径作准备。在执行 ANALYZE 语句时,Oracle 通常会为被分析的表加锁。例如,分析表 student 中的数据:

```
ANALYZE TABLE student COMPUTE STATISTICS;
```

通过上面两步基于代价的优化策略的优化器就设置完成了。需要注意的是在采集优化器的统计数据时,传统的方法是使用 ANALYZE 语句,但自从 Oracle9i 开始,Oracle 推荐使用 DBMS_STATS 包。因为 ANALYZE 语句只能串行地执行,而 DBMS_STATS

则可以根据指定的并行程度提供并行查询,因此可有效提高查询效率。DBMS_STATS 中提供了新的过程调用,它为启用表监控提供了快捷方式。例如访问表 abc.student,为了让优化器挑选出最佳方案,可以采用如下语句:

```
BEGIN
  DBMS_STATS.GATHER_TABLE_STATS ('abc', 'stu-
  dent');
END;
```

因为基于代价的优化策略对用户是透明的,所以不能改善多个分布式查询的方案,并且当分布式查询中包含合计、子查询或者复杂 SQL 时,此优化策略不起作用,这就需要使用提示来改善分布式查询的功能^[4]。

2.3 使用提示

使用提示可以扩展基于代价的优化策略的能力,进一步对查询语句进行优化,当你了解分布式数据库的结构、分布式查询的过程、网络通信、负载、统计数据等数据库环境的相关知识后,你可以指定提示来指导基于代价的优化策略。

2.3.1 NO_MERGE 提示

NO_MERGE 提示能防止在查询中将行中视图融合到一个潜在的非并列 SQL 语句。该提示嵌入在 SELECT 语句中,或者作为参数出现在带有行中视图的 SELECT 语句的开头,或者出现在行中视图的查询块中。例如:

在开头中作为参数:

```
SELECT /* + NO_MERGE (v) */
  student.no, v.avg (age)
FROM student, (SELECT no, avg (age) FROM class) v
WHERE student.no = v.no;
```

在查询块中作为参数:

```
SELECT student.no, avg (age)
FROM student,
  (SELECT /* + NO_MERGE */no, avg (age) FROM class)
```

v

```
WHERE student.no = v.no;
```

2.3.2 DRIVING_SITE 提示

DRIVING_SITE 提示用于指定远程查询的站点,这样查询可以在远程查询,然后将数据返回给本地站点,从而节省了数据来回传输的代价。例如:

```
SELECT /* + DRIVING_SITE (student) */ * FROM
class, student@abc.
  School WHERE class.no = student.no;
```

2.4 数据复制

对于远程数据库的查询,根据分布式数据库的结

构可以看出,本地数据库访问远程数据库的通常方法是用数据库链接来进行远程的结构化查询,但这样的代价相当高。通过设计物化视图可以降低代价,并且提高性能,因此 Oracle 的复制技术为此提供了办法。

对于那些琐碎而且经常需要查询的远程数据复制到本地数据库后就不用每次查询都进行远程链接,只需查看本地数据库,这样可以节省大量的网络代价。通常复制远程数据可以采用多种方法,应用 copy 命令最简单,直接从远程数据库拷贝数据,例如:

```
COPY FROM student@abc.school
CREATE stu
SELECT * FROM student;
```

从远程数据库 abc 中的表 student 复制数据创建本地数据表 stu;这样就无需再访问远程数据库,这种复制方式适用于那些不经常更新的数据。对于数据库中随时更新的数据,为了达到访问数据的同步性,应该采用快照复制的方法。Oracle 分布式选项 (Oracle Distributed Option) 提供了快照复制的方法,快照可以随时更新,达到数据修改同步。

此外,还可以通过 OEM 中的复制管理器 (Replication Manager) 采用图形化方式复制^[5]。

2.5 SQL 语句的优化

不管采用哪种优化方法都要通过 SQL 语句进行查询,因此 SQL 语句的优化不可缺少。远程查询 SQL 语句优化需要注意下面几点:(1) 尽量避免全表扫描;(2) 使用基于索引优化的 SQL 语句;(3) 使用 SQL HINT;(4) 通过服务器上的 SQL 共享减少分析调用^[6]。

2.6 批量请求

因为远程数据库的访问不可避免地要用到网络,因此网络代价的减少同样可以提高远程查询的性能。减少网络代价的一个常用方法就是批量请求,即当查询远程数据库的多行数据时,采用批量阵列的操作执行相同的查询比一次读取多行数据会节省许多代价。自从 Oracle 8i 引入 BULK COLLECT 特性后阵列操作在远程数据查询中得到了越来越多的应用。

2.7 排序合并连接或者 Hash 连接

在分布式查询中,对于分布在不同节点上的表进行嵌套连接查询时速度会非常慢,这是由于 Oracle 会将所有远程数据传到本地进行处理的结果。所以对于这类查询最好采用排序合并连接或者 Hash 连接^[7]。例如通过本地表 stu 访问远程数据库 abc 的表 Student 时的嵌套连接:

```
SELECT count (*) FROM stu
WHERE exists
```

```
(SELECT * FROM student@abc
WHERE stu.no=student. no);
```

可改写成排序合并连接:

```
SELECT count (*) FROM stu
WHERE stu.no in
(SELECT no FROM student@abc);
```

或者改写成带有提示的 Hash 连接:

```
SELECT /* + use_hash (stu, student) */ count (*)
FROM stu WHERE exists
(SELECT * FROM student@abc
WHERE stu.no=student. no);
```

3 性能测试

从上述查询优化方法中,可以看出在 Oracle 分布式数据库查询中通常可以将多种查询优化方法混合使用,从下面的程序段中可以看出综合使用上述查询优化方法前后性能的差别:

```
SELECT * FROM emp
WHERE exists
(SELECT * FROM dept@aaa WHERE emp.deptno=dept.
```

deptno);

已用时间: 00: 00: 55.07

Execution Plan

```
-----
0  SELECT STATEMENT Optimizer=CHOOSE
1  0 FILTER
2  1 TABLE ACCESS (FULL) OF 'EMP'
3  1 REMOTE *
3  SERIAL_FROM_REMOTE
```

Statistics

```
-----
7 recursive calls
6 db block gets
642 consistent gets
0 physical reads
7 redo size
```

502201 bytes sent via SQL * Net to client

78972 bytes received via SQL * Net from client

601 SQL * Net roundtrips to/from client

```
4 sorts (memory)
0 sorts (disk)
```

8999 rows processed

```
SELECT /* + use_hash (emp, dept) */ * FROM emp WHERE
exists
```

```
(SELECT * FROM dept@aaa WHERE emp.deptno=dept. dept-
no);
```

已用时间: 00: 00: 55.04

Execution Plan

```
-----
0  SELECT STATEMENT Optimizer=CHOOSE (Cost =
```

11 Card=1833 Bytes=84318)

```
1 0 HASH JOIN (SEMI) (Cost=11 Card=1833 Bytes=
84318)
```

```
2 1 TABLE ACCESS (FULL) OF 'EMP' (Cost=7 Card=
1833 Bytes=60489)
```

```
3 1 REMOTE * (Cost=1 Card=41 Bytes=533)
```

```
3 SERIAL_FROM_REMOTE
```

Statistics

```
-----
0 recursive calls
3 db block gets
45 consistent gets
0 physical reads
0 redo size
```

502153 bytes sent via SQL * Net to client

78972 bytes received via SQL * Net from client

601 SQL * Net roundtrips to/from client

```
0 sorts (memory)
```

```
0 sorts (disk)
```

8999 rows processed

详细数据对比如下所示:

性能测试标准	优化前	优化后
耗时间	00:55:07	00: 55:04
recursive calls	7	0
db block gets	6	3
consistent gets	642	45
physical reads	0	0
sorts (memory)	4	0
sorts (disk)	0	0

当然,在分布式查询中,还存在着多种因素影响优化性能,例如:Oracle 数据库存在的缺陷,网络性能的好坏等因素都有可能影响到查询优化的结果。因此在 Oracle 分布式数据库查询过程中,应该考虑各个方面的因素,综合各种优化方法,来找出最好的查询方案。

参考文献:

- [1] 吉斌武,黄 锐. 分布式数据库中的查询优化[J]. 桂林航天工业高等专科学校学报,2006(3):29-31.
- [2] Ingram G. Oracle 性能优化[M]. 张建明,英 字译. 北京:清华大学出版社,2003.
- [3] Ozdu T M, Valduniez P. 分布式数据库系统原理[M]. 北京:清华大学出版社,2002.
- [4] 李 华,赵建平. 分布式数据库数据查询的优化处理方法[J]. 长春理工大学学报,2005(4):85-70.
- [5] 赖万钦. Oracle 复制技术在分布式信息系统中的同步应用[J]. 计算机时代,2007(3):37-39.
- [6] 李风云,王守强. 基于分布式数据库的 SQL 语句性能优化[J]. 电脑知识与技术,2005,15:3-4.
- [7] 胡欣杰. Oracle91 数据库管理员指南[M]. 北京:北京希望电子出版社,2002.