基于 Oracle 数据库的 SQL 语句优化

李展涛，曹英忠
（桂林理工大学 信息科学与工程学院，广西 桂林 541004）

摘 要：通过分析 Oracle 数据库执行 SQL 语句的过程，采用比较 SQL 语句优化之前和优化之后的执行时间和调用的数据块数量方法来判断优化效果，最后得到消耗时间少和调用数据块少的 SQL 语句。

关键词：Oracle 数据库；优化；优化器；索引

中图分类号：TP311.131 **文献标识码**：A **文章编号**：1674-7720(2011)21-0011-03

Optimization of SQL method based on Oracle database

Li Zhantao, Cao Yingzhong
(College of Information Science and Engineering, Guilin University of Technology, Guilin 541004, China)

Abstract: This paper analyzes the execution processes of SQL sentence in Oracle database, compares the execution time and number of data blocks called in before and after of optimization to determine the optimal effect. Finally, the paper gets the SQL statement that time consuming less and data block calling less.

Key words: Oracle database; optimization; optimizer; index

随着信息化技术在各行业的广泛应用，Oracle 数据库也越来越多地被使用到很多关键领域，成为国内高端数据库市场的主流产品和众多行业信息化系统的主要支柱。如何充分利用 Oracle 的各种功能来提高数据库的可用性，如何提高数据库的数据查询响应时间以及如何诊断数据库出现的问题已经成为不断提高 Oracle 应用水平和提高 Oracle 数据库应用系统性能的关键[1]。

1 SQL 查询过程及优化器

1.1 SQL 查询语句的执行过程[2]

查询优化最重要的就是对 SQL 语句进行优化。调整 SQL 对性能的改善要比调整其他方面明显得多。理解 SQL 语句的执行过程有助于更好地对其进行优化。SQL 语句在 Oracle 中是自动执行的，绝大多数用户不需要关心各个阶段的执行细节。但是，对执行的各个阶段的了解会有助于快速找到性能低下的 SQL 语句，帮助书写出更高效的 SQL 语句，进而解决问题。几乎所有的 SQL 语句都分为语法分析、执行、读取数据三大阶段进行处理[3]。SQL 查询语句的执行过程如图 1 所示。

1.2 Oracle 查询优化器[4]

SQL 是一种非过程化的语言，用户只需要发送取出数据的命令，对于数据的取出方式（如是通过索引还是全表扫描），则由数据库的优化器决定。Oracle 的优化器用来决定 SQL 访问数据的有效路径，使语句执行所需要的开销最小。在 Oracle 的发展过程中，一共开发过两类优化器：基于规则的优化器和基于成本的优化器。它们之间的不同之处主要在于取得代价的方法与衡量代价的大小不同。

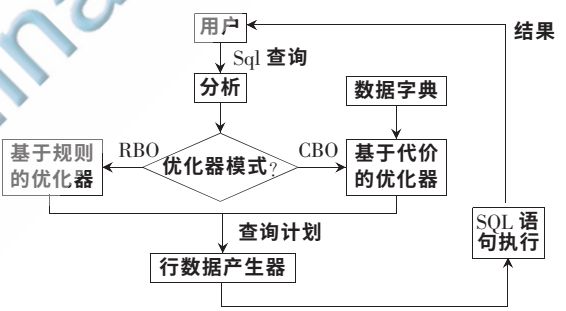

图 1 SQL 查询语句执行图

1.3 SQL 查询语句的执行计划

Oracle 要实现许多步骤才能完成 SQL 查询语句的执行，优化器将这些步骤组合在一起称为 SQL 查询语句的执行计划。从执行计划中可以看出数据库是如何执行查询语句的，判断出查询语句的执行是否高效，从而制定查询的优化方案。获取执行计划的方法有以下两种：(1)

用 Explain plan 命令对语句的执行过程的一些信息进行统计，Explain plan 用来显示优化器使用的执行计划而不实际运行查询；(2)用 Set Autotrace 动态查看每个 SQL 语句的执行计划，Autotrace 可以查看会话中每个 SQL 语句的执行计划。SQL 自动地进行 Explain plan 的工作，不用维护 plan table 表，因此使用非常方便。

2 系统优化措施

以具体的实例来说明系统优化问题以及调整方法。在某电子产品售后服务系统中，为加强对售后维修点备件使用情况的精确管理，库房发货人员对出库的每件备件粘贴一个唯一的一式两联条码，一联粘贴在发出的好备件上，另一联粘贴到从用户那里返回的坏备件上。发货业务和备件条码管理有关的 E-R 图如图 2 所示。

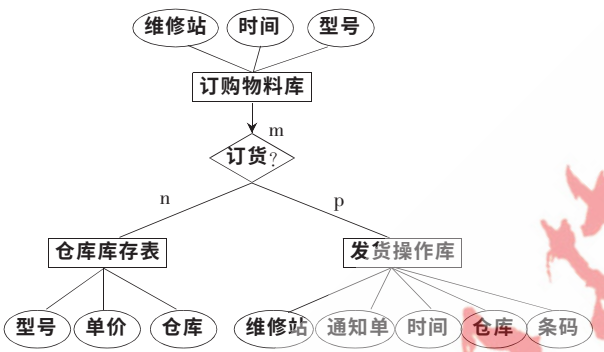

图 2 发货业务 E-R 图

这个系统中有一个查询出库信息详单的视图，该视图在系统运行初期的查询速度较快，但随着时间推移，数据量增加，其中有些表的数据量已达 20 万行以上，导致该视图的查询速度明显变慢，而由于资金等各方面的原因，短期内很难从硬件方面对系统进行升级。因此决定在其运行的 Oracle 9i 平台上进行优化。在进行优化前，该视图的查询时间为 1′07″左右，运行的硬件环境为：P42.66、1GB 内存、240 GB 普通 IDE 硬盘。在 SQL*Plus 中优化前的运行时间如图 3 所示。

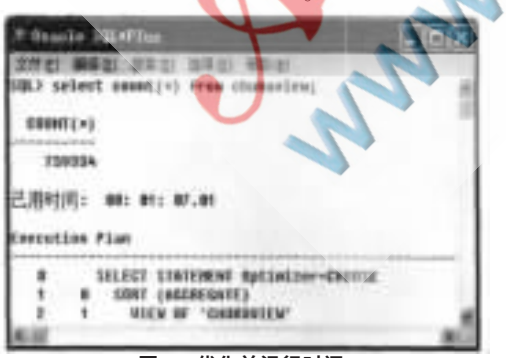

图 3 优化前运行时间

2.1 优化 SQL 语句

2.1.1 分析 SQL 语句的执行计划

T1、T2、T3 都是大表，且在 T1 表上一个组合索引：T1 (C1,C2)，注意 C1 列为索引的引导列。对于查询：Select T1.C4 from T1,T2,T3 where T2.C4=6 and T1.C1=T2.C1 and T1.C2=T3.C2 and T3.C3=7，跟踪该查询的执行计划如图 4 所示。

图 4 查询的执行计划

分析图 4 查询计划，找出各个表之间的关联关系，从而得到执行计划中哪个表为驱动表。在执行计划中，需要知道哪个操作是先执行的，哪个操作是后执行的，这对于判断哪个表为驱动表有用处。

执行计划的第 3 列，即字母部分，每列值的左面有空格作为缩进字符。在该列值左边的空格越多，说明该列值的缩进越多，该列值也越靠右。如图 4 的执行计划所示：第一列值为 6 的行的缩进最多，即该行最靠右；第一列值为 4、5 的行的缩进一样，其靠右的程度也一样，但是第一列值为 4 的行比第一列值为 5 的行靠上；在上下关系方面，只对连续的、缩进一致的行有效。对于 NESTED LOOPS 部分，最右、最上的操作是 TABLE ACCESS(FULL)OF 'T2'，所以这一操作先执行，该操作对应的 T2 表为第一个驱动表(外部表)，T1 表即为内部表。T2 与 T1 表做嵌套循环后生成了新的 row source，对该 row source 进行排序后，与 T3 表对应的排了序的 row source(应用了 T3.C3=7 限制条件)进行 MERGEJOIN 连接操作。所以由此可以得出如下事实：T2 表先与 T1 表做嵌套循环，然后将生成的 row source 与 T3 表做排序合并连接。通过分析上面的执行计划，不能认为 T3 表一定在 T1、T2 表之后才被读取，事实上，T2 表有可能与 T3 表同时被读入内存，因为将表中的数据读入内存的操作可能为并行的。

事实上许多操作可能为交叉进行，因为 Oracle 读取数据时，如果就是需要一行数据也是将该行所在的整个数据块读入内存，而且还有可能为多块读。看执行计划时，其关键不是看哪个操作先执行，哪个操作后执行，而关键是看表之间连接的顺序 (如需知道哪个为驱动表，这需要从操作的顺序进行判断)、使用了何种类型的关联及具体的存取路径(如判断是否利用了索引)，在从执行计划中判断出哪个表为驱动表后，根据掌握的知识判断该表作为驱动表。在这个例子中，T2 为驱动表，表的连接顺序为 (T2->T1)->T3，查询的过程中也使用到了 T1 表中的索引，因此，Oracle 优化器对其进行的优化效果是比较好的。如果分析了执行计划发现不合适，就要

对 SQL 语句进行更改，或用 Oracle 提供的提示(Hints)使优化器可以选择正确的驱动表，以更为合理的顺序进行表的连接。

2.1.2 使用提示(Hints)干预执行计划[5]

基于成本的优化器智能化程度很高，绝大多数情况下它能对 SQL 进行合理地优化，减轻了 DBA 的负担。但有时受到一些因素的影响，优化器也会选择很差的执行计划，使某个语句的执行变得奇慢无比。此时就需要 DBA 进行人为的干预，告诉优化器使用所指定的存取路径或连接类型生成执行计划，从而使语句高效地运行。例如，如果认为对于一个特定的语句，执行全表扫描要比执行索引扫描更有效，则就可以指示优化器使用全表扫描。在 Oracle 中，是通过为语句添加 Hints(提示)来实现干预优化器优化的目的。Hints 是 Oracle 提供的一种机制，用来告诉优化器按照技术人员告诉它的方式生成执行计划：

(1)使用的优化器的类型。

(2)基于代价的优化器的优化目标，是 all_rows 还是 first_rows。

(3)表的访问路径，是全表扫描，还是索引扫描，还是直接利用 rowid。

(4)表之间的连接类型。

(5)表之间的连接顺序。

(6)语句的并行程度。

Hints 只应用在其所在 SQL 语句块上，对其他 SQL 语句或语句的其他部分没有影响。除了“RULE”提示外，一旦使用别的提示，语句就会自动地改为使用 CBO 优化器，此时如果数据字典中没有统计数据，就会使用缺省的统计数据。所以如果使用 CBO 或 Hints 提示，则最好对表和索引进行定期的分析。

对于表的访问，可以使用两种 Hints：FULL 和 ROWID。FULL 提示告诉 Oracle 使用全表扫描的方式访问指定表。例如：

```
SELECT/*+FULL(EMPLOYEE)*/
FROM EMPLOYEE WHERE EMP_NO=9527;
```

索引 Hints 告诉 Oracle 使用基于索引的扫描方式，不必说明具体的索引名称。例如：

```
SELECT/*+INDEX(LODGING)*/LODGING
FROM LODGING
WHERE MANAGER='BILL GATES';
```

使用 Hints 对 Oracle 优化器缺省的执行路径进行手工修改是一个很有技巧性的工作，一般建议只针对特定的、少数的 SQL 进行 Hints 的优化。绝大多数情况下，只要 SQL 书写规范，Oracle 查询优化器的工作情况是相当理想的。Hints 提示虽然能带来一些方便，但是不能滥用，因为这种方法过于复杂，缺乏必要的通用性和应变能力，同时增加了维护上的代价。

调整 SQL 语句后的查询时间图 5 所示。

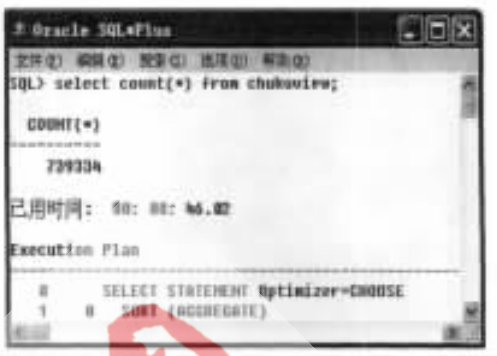

图 5 调整 SQL 语句后的查询时间

目前数据库规模越来越大，数据量呈指数级上升，使数据库的性能越来越重要。Oracle 数据库内部结构复杂，影响系统性能因素较多，但在系统硬件不变的情况下，SQL 语句的优化是性能得以提高的根本。但优化并不能一劳永逸，随着表结构的改变和数据量的增加，优化也必须实时调整。

参考文献

[1] 卞荣兵.基于 ORACLE 数据库性能优化的研究[J].应用技术，2002(9)：36-38.

[2] 钟小权.Oracle 数据库的 SQL 语句优化[J].计算机与现代化，2011(3).124-126.

[3] 谷小秋，李德昌.索引调整优化 oracle 9i 工作性能的研究[J].计算机工程与应用，2005，26：174-176.

[4] 路川.Oracle 10g 宝典[M].北京：电子工业出版社，2009.

[5] 仇道霞.Oracle 数据库性能调整优化 [J]. 山东轻工业学报，2010，24(3).52-54.

(收稿日期：2011-07-06)

作者简介：

李展涛，男，1985 年生，研究生，主要研究方向：数据库技术。

曹英忠，女，1986 年生，研究生，主要研究方向：云计算技术。