

# 数据库应用系统优化方法的研究

夏义全

(安徽建筑工业学院 电子与信息工程学院, 安徽 合肥 230601)

**摘 要:**数据库应用系统开发的首要任务是对需求的充分分析,预见性地考虑并解决影响系统高效率的瓶颈问题。针对数据库应用系统开发过程中如何提高系统高效率的瓶颈问题,分析了引起效率瓶颈的原因,阐述了提高服务器性能的一些方法和技术,并利用这些方法和技术来优化复杂数据库应用系统的性能,大大地提高数据库存取、查询等效率。就本地表和临时表的使用、二进制大对象 BLOB 的存取策略及数据库数据的整理和分类的必要性等给予了探讨。

**关键词:**数据冗余;存储过程;二进制大对象

**中图分类号:**TP311

**文献标识码:**A

**文章编号:**1673-629X(2008)07-0149-04

## Research on Optimization Method of Database Application System

XIA Yi-quan

(School of Electronics & Information Engineering, Anhui Institute of Architecture & Industry, Hefei 230601, China)

**Abstract:** The chief task in developing database application system is that the requirement of system should be analyzed, and the bottleneck is solved to improve the system efficiency. For the problem of how to promote system efficiency, analyzes the reasons of efficiency bottleneck, and expounds applications of some technology which can improve database server accessing efficiency. The performance of a complicated database system can be optimized by using these methods and technologies, and these methods can highly improve the efficiency of accessing and querying data. At the end, illustrate the following related questions such as application of local table and temporary table, accessing tactic of BLOB and necessity of assorting and classifying data in database.

**Key words:** data redundancy; stored procedure; BLOB

## 0 引言

在数据库应用系统的需求分析、设计、测试和试运行的整个开发环节中,常注意到功能的实现,而很难注意到性能的薄弱环节,即测试和试运行阶段中,由于测试方法的缺陷及较小的测试数据量,不能全面反映出系统的性能不足之处,系统投入实际运行一段时间后,系统的性能在不断降低,此时改善系统性能则需花费更多的人力和物力,而整个系统也就不可避免地成为了一个打补丁工程。文中就如何提高系统性能,对数据库设计和系统开发过程中的一些技术应用问题提出解决方法和策略。

## 1 提高数据库服务器效率的方法

### 1.1 反规范设计及技术的应用

数据库设计规范化的优点是减少了数据冗余、节

约了存储空间、相应逻辑和物理的 I/O 次数减少,同时加快了增、删、改的速度,但是数据库规范级别的增高会使关系数目增加,这样查询通常需要更多的连接操作,从而影响查询的速度。为了提高某些查询或应用的性能而有意破坏规范规则(称为反规范),常用的反规范技术有以下几种<sup>[1]</sup>:

(1)增加冗余列:有意使多个表中具有相同的列。常用来在查询时避免连接操作,但需要更多的磁盘空间,同时增加表维护的工作量。

(2)增加派生列:指表中增加的列是由来自其它表中的数据计算生成。它的作用是在查询时减少连接操作,避免使用集函数,但其也具有与冗余列同样的缺点。

(3)重新组表:当应用中需查看多表连接出来的数据,则把这几个表重组成一新表。这可减少连接,提高性能,但需更多的磁盘空间,同时也损失了数据概念上的独立性。

(4)分割表:有两种方式。①水平分割,按一列或多列数据值把数据行放到两个独立表中;②垂直分割,把主码和一些列放到一个表中,再把主码和另外列放

收稿日期:2007-10-06

基金项目:安徽省自然科学基金项目(2006KJ012C)

作者简介:夏义全(1959-),男,安徽合肥人,高级工程师,研究方向为数据库与信息系统。

到另一个表中。

反规范最大的问题是由于数据冗余可能出现数据的一致性和完整性问题<sup>[2]</sup>。但是只要细致分析实际应用的数据存取需求和实际需要的性能特点,采取一定的措施解决数据一致性和完整性问题,那么恰当地反规范策略将会大大提高系统的查询效率。

### 1.2 少用约束和规则

约束是库表中自动强制数据完整性的一种方式,它定义了列中允许值的规则。约束分为多种类型,其中 Check 约束是对放入列中的值进行限制,以使数据得以完整<sup>[1]</sup>。规则和 Check 约束的功能类似,也是用于对数据库表中列数值的检查限制。

创建表时可以对 Check 约束进行指定,而规则需要以单独的对象创建,再绑定到列上。一个列只能应用一个规则,但却可以应用多个 Check 约束。需要注意的是,约束对数据的有效性验证要比规则快。

规则和约束的使用也会占用系统的开销。不必要的规则和约束是可以略去的,即使是必要的规则和约束,也可以通过客户端的代码对要输入数据库中的数据进行完善的检查来完成。所以,在设计阶段就应考虑只使用非常必要的约束和规则。

### 1.3 使用索引

据微软公司统计:即使对于一个面向频繁插入和更新操作的应用程序来说,数据库读取次数一般也超过数据库写入次数的 5~10 倍。而索引(Index)可加快读取数据库表中的数据,如一较大的表,在加索引后,通常要花几分钟或十几分钟来完成的查询只要几秒或十几秒。因此有理由对需要频繁查询的表增加索引<sup>[3]</sup>。

索引采用 B 树结构,数据库对 B 树按搜索关键字排序,可在搜索关键字的任何子词条集合上无须对整个表进行扫描,就可找到所需数据。索引分为簇索引和非簇索引,没有簇索引的表是按堆结构存储数据,所有的数据均添加在表尾部。建立了簇索引的表,其数据在物理上会按簇索引键的顺序存储,一个表只允许有一个簇索引,因此根据 B 树结构,可以理解添加任何一种索引均能提高按索引列查询的速度,但会降低插入、更新、删除操作的性能,尤其是当填充因子(Fill Factor)较大时,向一个已满的索引页添加某个新行时,数据库把大约一半的行移到新页中以便为新行腾出空间,这种重组称为页拆分,它会降低性能并使表中的数据存储产生碎片。故对索引较多的表进行频繁的插入、更新、删除操作,建表和索引时应设置较小的填充因子,以便在各数据页中留较多的自由空间,减少页分割及重组的工作。

索引的不足之处是需要占用一个可观的额外空间。SQL Server 中建一个聚簇索引,要占用大约 1.2 倍于数据大小的空间;尽管索引降低了添加、删除和更新行的速度,但多数情况下,它所带来的数据检索速度的优势大大超过它的不足之处。

### 1.4 使用存储过程和触发器

存储过程和触发器(特殊的存储过程)是 SQL 语句和流程控制语句的预编译集合。存储过程可包含程序流、逻辑及对数据库的查询、修改、删除等操作,允许用户声明变量、有条件执行以及利用数据库服务器的其它强大编程功能来实现用户的复杂的业务流程。它可接受参数、输出参数、返回单个或多个结果集以及返回值。编译好的存储过程作为数据库的一个对象可由应用程序通过调用来执行。

使用存储过程从提高效率方面来讲它确实具有很多优点。例如:对于大数据集合的操作,当需要在数据库服务器和客户端之间进行经常性的数据交互时,如果不使用存储过程,一来易使程序进入长等甚至死机,二来在网络环境下会造成网络阻塞。此种情况下,当对数据处理的业务流程和规则进行充分的分析后,可以编写出存储过程,间或可以使用临时表等,当客户端调用时只需把需要的参数值直接传给存储过程或临时表,存储过程就在服务器上执行。这样可以大大减少网络环境下的数据流量,同时客户端的编程也将非常简单,而且存储过程在创建时即在服务器上进行过编译,所以执行起来比单个 SQL 语句要快。由于数据库服务器的硬件环境可以改善,那么这将很大程度上充分发挥服务器的性能,提高应用系统的效能。此外,大中型数据库服务器都有系统存储过程,可以从自己的存储过程内引用其它存储过程或系统存储过程。

### 1.5 优化 SQL 语句

尽管 DBMS 的基于关系代数的优化器包括了很复杂的优化技术,对于同一 SQL 语句可以考虑很多种不同的执行计划,从中选出最优方案。写出高效率的 SQL 语句是很重要的,因为不同的 SQL 语句效率差异很大,而且 SQL 语句看起来很容易理解,使用 SQL 语句来实现应用功能也并不难,可以写出多个语句来实现同一功能,但是往往很少考虑 SQL 语句的效率,这需要对 SQL 语言的深刻理解。下面是 SQL 示例语句。

语句 1:

```
SELECT DeptNo, DeptName  
FROM TabDept WHERE DeptNo NOT IN  
(SELECT DeptNo FROM TabEmp);
```

语句 2:

```
SELECT DeptNo, DeptName
FROM TabDept WHERE NOT EXISTS
(SELECT DeptNo
FROM TabEmp
WHERE TabDept. DeptNo = TabEmp. DeptNo);
```

语句1和语句2实现同样的功能,但在不同数据量、不同系统的优化器下的执行计划可能完全不同,性能也可能有巨大差异。对于ORACLE来说,当子查询中的select关键字后的字段DeptNo有not null约束或者有这样暗示时语句1效率较好,另外如果主查询中表TabDept大,子查询中的表TabEmp小,也应当使用语句1;如果主查询表TabDept中记录少,子查询表TabEmp中记录多,并有索引,则语句2性能较好。

再如,在SQL Server中pubs数据库authors表中有两个作者的名字是Ringer。如果从这个表中取出所有的名字,名字Ringer将会显示两次。在特定情况下,一个表中的某个字段有重复的值,当希望每个值只被选取一次,可用关键字DISTINCT来做到这一点,但强制服务器返回互不相同的值却要增加系统的运行开销。运用ORDER BY子句同样会也消耗服务器的资源。因此,不是必需的时候可以不使用关键字DISTINCT和ORDER BY等子句。

## 2 相关问题的探讨

### 2.1 本地表和临时表的合理使用

在整个数据库应用系统中,按数据处理所在的机器把数据处理方式分为:(1)客户端数据处理;(2)应用层服务器数据处理;(3)数据库服务器数据处理。

在整个数据库应用系统的网络中,按流动数据信息的类型可分为:(1)指令信息;(2)状态信息;(3)数据结果信息。

指令信息和状态信息往往是比较小的,而数据结果信息很多时候却是大数据集合,这时在网上传输就会占用较多的网络资源。特别是大数据集合在客户端和服务端之间的频繁交互,会使客户机长等,还会造成网络阻塞<sup>[4]</sup>。为避免这种情况,可详细分析业务流程,对系统数据进行分类,合理规划客户端、应用服务器、数据库服务器的数据处理,尽量让网络上流动的是状态信息、指令信息和必要的结果集。为达到这样的目的,可使用建立本地表和临时表。

本地表就是为了某种需要在客户端或服务端上设立的数据库表。例如,某项处理对象是大数据集合,在处理过程中客户端需要向数据库服务器频繁提交中间结果,这时可以考虑把数据库服务器端的所有相关信息一次性存储到本地表,然后再对该表进行相关处理,

最后把整个结果集存回数据库服务器中。

临时表是在服务器上或在本机中为了某种需要设立的数据库表,当任务完成后就从服务器或本地机上删除。例如上文1.4节中,使用存储过程和触发器就是把相关的处理过程放在服务器端,可使用临时表暂放中间结果,其目的也是让网络上的流动信息尽量是指令或状态信息。

总之,适当地使用本地表和临时表可以减少网络信息流量,提高整个系统的性能。

### 2.2 二进制大对象BLOB的存取策略

在数据库应用系统中,对图形、图像、声音等对象的处理是很常见的,如地理信息系统GIS中的图像数据的存取,此类对象的特点是数据量大。对此类大对象存取的策略通常是:

(1)数据库的表中存放的是把大对象作为文件所存放的目录位置;(2)直接把大对象存放在数据库的表中。

对于第一种策略,由于大对象是在数据库的外部,虽便于存取,但不便于管理;对于第二种策略,虽便于管理,但不便于存取。对于大对象存放在数据库表中的情况,存取时用通常使用的编程方法是不能适用的。

一般大型数据库都支持大对象类型,SQL Server中能存储每行大到2G的二进制对象(BLOB),有image、text和ntext三种数据类型。下面是BCB写的存取BLOB的示例程序。

BLOB字段的读取:

```
TBlobStream * TemplateStream;
char * TempPlatePtr;
TemplateStream = new TBlobStream((TBlobField *) Blob-
Query->FieldByName("SearchTem"), bmReadWrite);
TemplatePtr = new char[TemplateStream->Size];
TemplateStream->Read(TemplatePtr, TemplateStream->
Size);
```

BLOB字段的写入:

```
TBlobStream * TemplateStream;
char * TempPlatePtr;
TemplateStream = new TBlobStream((TBlobField *) Blob-
Query->FieldByName("SearchTem"), bmReadWrite);
TemplatePtr = new char[TemplateStream->Size];
TemplateStream->Write(TemplatePtr, TemplateStream->
Size);
```

由于系统资源总是有限的,对于较大的数据对象,如果一次读(存)取大量数据,可能会引起客户机长等或死机,或者服务器性能大大下降,也会引起网络的阻塞,这对系统效率产生不利的影响。因此,读写处理时,可以将图像、声音等大对象数据进行分段读写,并

要考虑每次读写合理的数据量,这样的处理会较好。

### 2.3 对数据库数据的整理和分类

随着时间不断推延,数据库应用系统中某些表的数据也在很快地累积,有时数据量还是比较庞大的。比如一个中等城市供电公司每月用户的抄表数据就有几万条甚至几十万条,虽然采用分布式数据库系统,通过分散处理再集中汇总的方式来解决数据量暴增的问题,但将大大增加各种费用和管理工作量<sup>[5]</sup>;如采取集中管理方式而又没有相应的措施,那将会使系统的负担越来越大,效率越来越低,最终会拖垮整个系统。

针对这种情况,需要在数据库设计时就要预见到,否则将会给后续的开发环节带来很大的麻烦。具体解决时,还要对具体的实际情况进行分析。就上面问题,在集中管理方式下可采用如下的解决方案:由于抄收表的数据量会很大,增长也很快,根据实际情况把相应的数据分类后建相应的表用于存放数据。例如在对每户电表抄表后,用户需要交费,可以把这些数据分为两类:欠费数据信息——存放在欠费表中;交费数据信息——存放在交费历史信息表中,这样可以使交费处理的时候查询量大大减小。由于交费历史信息表中的数据信息还是很庞大的,更进一步采取的措施是按月、季或年为时间单位建立新表,把交费历史信息表中相应的月、季或年的数据放到相应的表中,并编制具有灵活的条件组合查询或模糊条件查询功能的客户端程

序,这样就可以大大提高数据的查询效率。基于这样的考虑,可以在客户端加入数据整理和分类的功能菜单。

### 3 结束语

数据库应用系统的开发首要的任务是对于需求的充分分析,在此基础上预见性地考虑并解决系统高效率的瓶颈问题就成为关键。文中介绍的一些提高系统性能的方法和技术不但适用于中小的系统,更是适用于大型的数据库应用系统。这些方法的综合应用可以解决各种复杂系统的性能问题,能大大地提高数据库存取、查询等效率。

#### 参考文献:

- [1] 萨师煊,王 珊.数据库系统概论[M].北京:高等教育出版社,2003.
- [2] 陶 勇,丁维明.数据库中规范化与反规范化设计的比较与分析[J].计算机技术与发展,2006,16(4):107-109.
- [3] 邓 曦,卢正鼎,张 巍,等.多数据库系统查询优化算法的研究[J].小型微型计算机系统,2004,25(3):451-454.
- [4] 孔 哲,孟丽容.数据库连接策略优化方法[J].山东大学学报:工学版,2003,33(6):652-657.
- [5] 贾 焰,王治英,韩伟红,等.分布式数据库技术[M].北京:国防工业出版社,2003.

(上接第 145 页)

- [4] 蔡 杰,熊齐邦.DDoS 攻击下的 IP 追踪技术[J].计算机技术与发展,2007,17(3):159-162.
- [5] Savage S, Wetherall D, Karlin A, et al. Practical Network Support for IP Traceback[C]//In Proceedings of ACM SIGCOMM. [s.l.]:[s.n.],2000.
- [6] Song D, Perrig A. Advanced and Authenticated Marking

Schemes for IP Traceback[C]//In Proceedings of ACM INFOCOM. [s.l.]:[s.n.],2001.

- [7] Oudot L. Fighting Internet Worms with honeypots[EB/OL]. 2003. <http://www.securityfocus.com/infocus/1740>.
- [8] 诸葛建伟.蜜罐及蜜网技术简介[EB/OL]. 2004. <http://www.honeynet.org.cn/reports/蜜罐及蜜网技术简介>.

(上接第 148 页)

设都提供了很好的参考,具有较高的实用性和一定的社会价值。

#### 参考文献:

- [1] 宋金玲,肖 寒,盛业华. GIS 在数字校园建设中的应用[J].北京测绘,2002,4(3):10-12.
- [2] 孟令奎,史文中,张鹏林.网络地理信息系统原理与技术[M].北京:科学出版社,2005.
- [3] 杨祖虎. Arc IMS 初级教程[M]. 北京:ArcInfo 中国技术咨

询与培训中心,2001.

- [4] 黄丙湖,闫国年,张亦含,等.基于 ArcIMS 的环保 WebGIS 的设计与实现[J].南京师范大学学报:工程技术版,2004,4(2):59-61.
- [5] 杨晨毅,刘吉平.基于 SDE 的 GIS 空间和属性数据在 RDBMS 中的集成[J].计算机仿真,2003,20(11):110-112.
- [6] 窦长娥,刘仁义,刘 南.基于 ArcIMS 的旅游地理信息系统设计与实现[J].计算机应用研究,2006(2):160-165.