

对系统性能优化的十点辨析

张建华 (浙江省文物考古研究所信息中心 浙江杭州 310014)

王群华 (浙江教育学院信息学院 310014)

摘要:系统性能尤如系统的生命,已越来越受到重视。本文根据作者的实践经验分别从十个方面对系统性能优化作了进一步的辨析,阐述了性能优化中一些技术和非技术方面的优化思想。

关键词:系统性能 性能优化

随着计算机技术的迅猛发展,以计算机网络和数据库技术为基础的各种信息化系统已成为当今政府机关及企事业单位运行的基础,而性能尤如系统的生命,已越来越受到重视。如何去优化系统性能,经常成为系统实现者们思考的问题,虽然目标只有一个,其途径却多种多样,如数据库优化、应用程序优化、内存优化、I/O 优化、网络优化等等,但系统性能优化是个复杂的问题,你在进行系统优化的过程中,“好心办坏事”的现象也时有发生。

本文从十个方面对系统性能优化作进一步的辨析,为广大系统实现者优化系统性能提供参考。

1 何时何处优化决定优化的成本与收益

这里涉及二个问题,即优化的时间点与着眼点。首先,对于时间点来说,在系统实现之初,实现者往往只关注业务流程的再设计和系统功能的实现,而忽略系统性能优化问题,随着数据的不断增长或用户数目的不断增加,系统的使用性能不断下降,此时如梦方醒,想起系统性能优化问题,但为时已晚,再去优化系统你将花费很大的代价。所以,系统优化这根弦就得从实现系统的第一步就绷紧。其次,对于优化的着眼点来说,应该按从大到小从枝到叶的原则进行,这样的优化次序才会让成本最低收益最大,而且在系统出现“瓶颈”时可顺利地对其扩展。为了优化系统性能,首先要从系统结构或软件框架入手,思考系统的最佳性能结构或框架,最后才到编程工具的控制选择及 SQL 语句优化之类。比如, C/S、B/S 混合结构的使用。C/S 结构可以实现数据的快速传递和安全存储,

但在远程数据管理方面存在不足; B/S 结构的系统易于集成和扩展,可以提供远程的数据服务与管理,但在数据的交互性、动态服务和图表显示方面尚有欠缺。因此,针对系统中的不同功能,应选择最合适的结构来实现。再如应用服务器的使用,对于大型系统应用来说,对以后的用户量要有一定的预见,一旦客户端的连接数量急剧上升,系统将如何应付? 应用服务器的使用将最大限度地减少与数据库服务器的连接数量,并可将相关的处理过程分担在多台服务器上完成,很好地解决网络拥堵问题,将极大地改善基于 Internet 的网络应用的性能。

2 逻辑库规范化程度越高越好吗

在设计数据库时尽量符合第三范式,以便最大限度地提高数据的一致性,减少数据冗余和不规则的更新,这几乎已是数据库设计的常识。但符合第三范式的数据库操作往往比第一范式、第二范式需要更多的表连接,会非常消耗 CPU 和磁盘 I/O 资源,从而降低了系统的性能。其实,某种程度上的非规范化设计反而可以改善系统的性能,也可以简化查询。非规范化过程可以根据性能方面的不同采用多种不同的方法进行,从实践经验看以下方法往往能提高性能:

(1) 如果规范化设计产生了四路或更多路的合并关系,则可以考虑在数据库实体中加入重复属性。

(2) 常用的计算字段(如总计等)可以考虑存储到数据库实体中。

(3) 重新定义实体以减少列数据或行数据的开支。相应的非规范化类型有:

① 把一个实体(表)分割成二个表(把所有的列分成 2 组)。这样就把频繁被访问的列同较少被访问的列分开了,如档案系统中将照片字段同其他字段分列到二个表中。

② 把一个实体(表)分割成二个表(把所有的行分成 2 组)。这种方法适用于那些将包含大量数据的实体(表),可以把频繁被访问的行同较少被访问的行分开。如应用中常要保留历史记录,如果数据行是作为子集被不同部门访问,那么采用这种方法很有益处。

较佳的办法是以规范化的设计为出发点,然后出于特定的原因有选择地非规范化某些表。当然,这里需要使用数据库管理系统的一些机制(如触发器)来自动维护数据库中数据的一致性与完整性。

3 SQL 语句越简洁越好吗

SQL 语句不是越简洁越好,可读性与性能往往不一致,有时要以增加语句的复杂度来换取性能。常用的方法有:UNION、临时表、存储过程。如以下语句:

```
select * from Idrecord where id in ('0', '1')
```

最好写成:

```
select * from Idrecord where id = '0' UNION  
select * from Idrecord where id = '1'
```

原因是 where 子句中的 in 在逻辑上相当于 or,语法分析器分将 in('0','1') 转化为 id='0' or id='1' 来执行,无法使用索引来优化性能。

如果有几个简单的数据库查询语句,即使它们之间没有关系,也可以把它们整合到一个查询中,虽然程序的可读性大大降低了,但减少了系统 I/O 操作次数,从而性能得以提高。

4 索引也有副作用

建立索引是提高查询速度的最有效的手段,在数据库设计中可以根据需要针对基本表上最有可能查询的属性建立索引。一个基本表可以针对不同的属性建立多个索引,但是索引也有副作用:

(1) 表上的索引过多会使数据库系统在进行 insert、update、delete 操作时花费较多的时间来维护索引,从而影响它们的性能。

(2) 对一个经常被更新的列建立索引也会严重影响性能。

(3) 对一个小型表进行索引也不划算,因为对于小表来说表扫描往往更快而且费用低。

(4) 在不同值少的列上建立索引不但不会提高查询效率,反而会严重降低更新速度。如在雇员表的“性别”列上(只有“男”或“女”两个值)就不必建立索引,因为它基本上都要扫描至少一半的记录。

同样,聚簇索引的使用可以在查询语句中使用 order by 子句、查询结果为大量一定范围内值的列、需要频繁进行 update、delete 语句操作时,能明显提高操作的效率。但是如果在需要频繁进行 insert 操作的表的一个自然增长的列上建立聚簇索引,每一个 insert 插入新行都必须加在表的最后一页,而 insert 操作时要锁定当前的数据页,这样极易引起表的锁定争用甚至发生死锁,大大降低了 insert 的性能。针对这种情况,可对一个非自然增长的列建立聚簇索引,使插入的数据随机分散在表内的任意地方,以最小化多个插入进程间的页锁争用发生的可能性。

总之,合理的索引设计要建立在对各种查询的分析和预测之上。

5 多条件并列时把范围最小的条件放在前面还是后面好

对于这一点,就要看数据库管理系统对 where 子句的解析顺序了。Oracle 采用自下而上的顺序解析 where 子句,因此表之间的连接必须写在其他 where 条件之前,那些可以过滤掉最大数量记录的条件必须写在 where 子句的末尾。而 SQL SERVER 却与 Oracle 刚好相反。

6 连接池中的连接个数越多越好吗

应用服务器作为系统基础,对运行效率有很大影响,应用服务器连接池中的连接个数决定了数据库并发访问的个数,那么连接池中的连接是不是越多性能改善就越好呢?这显然不是的。试想,当只有 20 个线程在运行的时候,数据库连接池中即使有 40 个连接也是无法全部使用的,由此可知单纯调整数据库连接池的数目并不能确实有效,而应该将线程数调整到与连接池最大数目相等,在线程数与数据库连接池中的连接数基本一致的情况下才能最大程度地提高性能。

7 视图对性能的影响

视图是关系数据库系统提供给用户以多种角度观察数据库中数据的重要机制,利用视图可以大大简化用户的工作,但是从查询优化的角度来看,视图则不利于查询效率的提高。如果查询处理器直接对视图进行操作的话,查询优化器所以生成的执行计划的唯一选择就是先执行视图定义,再将视图的查询结果作为一个临时表参与查询的其余处理。这种处理方式增加了系统 I/O 操作次数,在绝大多数情况下效率低下。

8 临时表也有副作用

临时表虽然占用磁盘空间,但当需要反复访问同一张或几张表时,可以把表的经常用到的字段作为一个子集排序后放到临时表中,由于其数据量远少于原表,反而减少了 I/O 操作的数量。但是,在主表修改频繁的情况下,临时表不能反映主表的修改,容易丢失数据,并且在分布式系统中临时表的使用还会带来多个查询进程同步的问题,因此要慎用。

9 充分使用优化工具

在可能的条件下,应充分重视各类工具在系统性能优化中的作用,工具可以增加系统的性能或者找出系统性能的瓶颈。如可以借助于性能测试工具 Rational Quantify 找出系统中的性能瓶颈;使用关系数据库管理系统中的优化器来优化 SQL 语句,以提高访问数据的效率;使用 Memory Optimizer 优化计算机内存,让它运行速度加快 200%,并预防 90% 的 windows 崩溃;使用 WinRam Turbo 给予内存、文件、磁盘以最优化设置;使用 Quarterdeck MagnaRAM 工具在内存中开辟一个独立的高速缓存区,并将最频繁调用的数据写入内存高速缓存区而不是写入硬盘以提高系统 I/O 操作性能。等等。

10 性能优化不是一劳永逸的!

影响系统性能的因素多种多样,随着系统运行环境的变化,系统的性能也会发生变化,系统性能的优化是一个不断摸索、总结的过程,故而性能优化不是一劳永逸的,一旦系统有了变化,就要采取相应的手段重新

优化。

以上十点是我们在对系统进行性能优化时的一些经验与感受,借此文加以总结,希望对大家有所帮助。

参考文献

- 1 邹腊梅等,内存管理与系统优化,《河北理工学院学报》,2003,25(3):63-66.
- 2 范孝良、国秀丽,企业信息化系统应注重数据库的性能优化,《上海第二工业大学学报》,2006,23(1):53-56.
- 3 张青锋,基于 C/S 模式的企业 MIS 系统中性能优化的实现,《周口师范学院学报》,2004,21(5):86-88.
- 4 于大伟、田地,基于 WEB 信息系统的优化管理及架构调整,《长春工程学院学报(自然科学版)》,2006,7(1):54-56.
- 5 杨劲,数据库管理系统在医院信息系统设计 and 应用中的优化,《北京生物医学工程》,2005,24(2):131-133.
- 6 时代, B/S 结构系统的优化问题探讨,《国土资源信息化》,2004(1):41-45.
- 7 李增强等, Microsoft SQL Server 2000 数据库系统设计优化,《计算机系统应用》,2006(10):88-89.34.
- 8 陈士川, SQL Server 数据库应用系统性能优化技术,《电脑知识与技术》,2004(5):83-86.
- 9 曾玲等, SQL Server 应用系统性能优化的有效途径,《计算机与数字工程》,2005,33(9):147-149.
- 10 董玉杰、李小军, 关系数据库系统的查询优化策略,《计算机与现代化》,2005(8):72-74.
- 11 陈国旗, Web 的财务查询系统的性能优化与实现,《中国计量学院学报》,2006,17(3):233-237.
- 12 孙萧寒、蒋平江, 关系型数据库系统查询优化分析,《甘肃科技》,2005,21(3):54-55.72.
- 13 郑丽莉, 分析与优化基于网络的数据库系统,《中国科技信息》,2005(13):65.66.
- 14 魏先民, 电子商务的系统优化研究,《商场现代化》,2006(4):81.
- 15 萨师煊、王珊, 数据库系统概论[M], 北京:高等教育出版社,2000.