

# Microsoft SQL Server 2000 数据库系统设计优化

李增强 管文强 王海勇 (大众日报社信息技术部 济南 250014)

**摘要:** Microsoft SQL Server 2000 是微软在 Windows 平台上开发的数据库, 尽管 Microsoft SQL Server 2000 的效率很高, 但在具体的使用中还需要不断优化数据库系统, 使得 Microsoft SQL Server 2000 的潜力得到全部发挥。数据库系统在设计及开发阶段进行数据库性能优化的成本最低、收益最大。本文就 Microsoft SQL Server 2000 数据库系统在设计方面的优化进行了讨论。

**关键词:** Microsoft SQL Server 2000 数据库 设计优化

Microsoft SQL Server 作为微软在 Windows 平台上开发的数据库, 最早脱胎于 Sybase 的数据库系统(微软购买了 Sybase 公司的技术, 推出了 Microsoft SQL Server 6.0), 并在原有功能的基础上与 Windows 进行了优异的融合。推出了新版本——Microsoft SQL Server 2000, 虽然在开放性和安全性方面 Microsoft SQL Server 2000 无法同 Oracle 等大型数据库相比(Microsoft SQL Server 只能在 Windows 上运行, 且没有获得任何安全证书)<sup>[1]</sup>。但 Microsoft SQL Server 2000 推出后在数据库管理系统市场获得了比较大的市场占有率, 虽然其原因是多方面的, 但是除了其性能优势和技术的先进性外, 可优化性和技术的可获得性方面的优势是绝对不能忽视的重要原因<sup>[1]</sup>。

尽管 Microsoft SQL Server 2000 的效率很高, 但在具体的使用中还需要不断优化数据库系统, 使得 Microsoft SQL Server 2000 的潜力得到全部发挥, 由于微软公司不仅仅是数据库系统供应商, 也是操作系统供应商, 所以在两个系统的兼容和优化上, 有着其他企业不可比拟的优势。数据库内部优化的一些原则可以最大化发挥 Microsoft SQL Server 2000 的可优化性。

一个数据库系统的生命周期可以分成: 设计、开发和成品三个阶段。在设计及开发阶段进行数据库性能优化的成本最低、收益最大<sup>[2]</sup>。

## 1 逻辑数据库设计优化<sup>[2]</sup>

数据库的逻辑设计, 包括表与表之间的关系, 这是优化关系数据库的核心。一个好的逻辑数据库设计可以为优化数据库和应用程序打下良好的基础。标准化

的数据库逻辑设计包括用大量的、有相互关系的窄表来代替很多列的宽数据表。使用标准化数据库的优点如下:

- 由于表窄, 可使排序和建立索引更为迅速。
- 由于多表, 则可以使用多簇的索引。
- 索引更窄, 更紧凑。
- 每个表中可以包含少量的索引, 提升执行 INSERT、UPDATE 和 DELETE 语句的速度。
- 更少的空值和更少的多余值, 增加了数据库的紧凑性。

由于数据库的规范化设计减少了数据冗余, 也减少了用于存储数据的页, 提高应用程序的效率并减少因数据不一致引起错误的可能性。但表关系也许需要通过复杂的合并来处理。这样会降低系统的性能。某种程度上非规范化可以改善系统的性能, 非规范化过程可以根据性能方面不同的考虑用多种不同的方法进行, 以下方法经实践验证往往能提高性能。

(1) 如果规范化设计产生了许多 4 路或更多路合并关系, 就可以考虑在数据库实体(表)中加入重复属性(列)。

(2) 常用的计算字段(如总计、最大值等)可以考虑存储到数据库实体中。把用户经常需要在查询和报表中用到的, 在表的记录量很大时, 有必要把计划总数作为一个独立的字段加入到表中。可采用触发器以在客户端保持数据的一致性。

(3) 重新定义实体以减少外部属性数据或行数据的开支。相应的非规范化类型是:

- ① 把一个实体(表)分割成二个表(把所有的属

性分成 2 组)。这样就使频繁被访问的数据同较少被访问的数据分开了。这种方法要求在每个表中复制首要关键字。这样产生的设计有利于并行处理,并将产生列数较少的表。

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)。在应用中常要保留历史记录,但是历史记录很少用到。因此可以把频繁被访问的数据同较少被访问的历史数据分开。而且如果数据行是作为子集被逻辑工作组(部分、销售分区、地理区域等)访问,那么这种方法也是很有好处的。

较佳的策略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。无论哪种设计,都应利用 SQL Server 下列功能自动维护数据库的完整性。

- ① 利用 CHECK 约束,保证字段的有效性。
- ② 利用 DEFAULT 和 NOT NULL 约束,保证输出必要字段值。
- ③ 利用 PRIMARY KEY 和 UNIQUE 约束,保证字段值的唯一性。
- ④ 利用 FOREIGN KEY 约束,保证记录的参照完整性。
- ⑤ 使用 IDENTITY 字段,高效生成唯一行的标识符。
- ⑥ 利用 TIMESTAMP 字段确保在多用户更新间进行高效并发检查。
- ⑦ 用户定义的数据类型确保到定义在数据库内的一致性。

这些服务器强制的规则有助于避免因应用程序本身未完全强制完整性规则而引起数据错误,尽可能高效地强制数据完整性。

## 2 索引设计优化

改变 SQL Server 性能的方法很多,但提高速度最快的方法就是索引。优化程序利用索引可以避免表扫描,并减少因查询而造成的 I/O 开销。建议采用如下的方法设计索引:

- 将更新尽可能多的行查询写入单个语句内,而不要使用多个查询更新相同的行。仅使用一个语句,就可以利用优化的索引维护。
- 使用索引优化向导分析查询并获得索引建议。

- 对聚集索引使用整型键,另外,在唯一列非空列或 IDENTITY 列上创建聚集索引可以获得性能收益。

- 比较窄的索引具有比较高的效率。对于比较窄的索引来说,每页上能存放较多的索引行,而且索引的级别也较少。所以,缓存中能放置更多的索引页,这样也减少了 I/O 操作。

- SQL Server 优化器能分析大量的索引和合并可能性。所以与较少的宽索引相比,较多的窄索引能向优化器提供更多的选择。但是不要保留不必要的索引,因为它们将增加存储和维护的开支。对于复合索引、组合索引或多列索引,SQL Server 优化器只保留最重要的列的分布统计信息。这样,索引的第 1 列应该有很大的选择性。

- 表上的索引过多会影响 UPDATE、INSERT 和 DELETE 的性能,因为所有的索引都必须做相应的调整。另外,所有的分页操作都被记录在日志中,这也会增加 I/O 操作。

- 对一个经常被更新的列建立索引,会严重影响性能。

- 由于存储开支和 I/O 操作方面的原因,较小的自组索引比较大的索引性能更好一些。但它的缺点是要维护自组的列。

- 尽量分析出每一个重要查询的使用频度,这样可以找出使用最多的索引,然后可以先对这些索引进行适当的优化。

- 查询中的 WHERE 子句中的任何列都很可能是个索引列,因为优化器重点处理这个子句。

- 对小于 1 个范围的小型表进行索引是不划算的,因为对于小表来说表扫描往往更快而且费用低。

- 与“ORDER BY”或“GROUP BY”一起使用的列一般适于做分簇索引。如果“ORDER BY”命令中用到的列上有分簇索引,那么就不会再生成 1 个工作表了,因为行已经排序了。“GROUPBY”命令则一定产生 1 个工作表。

- 分簇索引不应该构造在经常变化的列上,因为这会引起整行的移动。在实现大型交易处理系统时,尤其要注意这一点,因为这些系统中数据往往是频繁变化的。

- 在查询经常用到的所有列上创建非聚集索引。

(下转第 34 页)

### 3 优化查询语句

SQL Server 查询消耗大量资源,如何优化查询和提高查询性能,建议从下面几个方向考虑。

- 在 SQL Server 中,系统数据库 tempdb 会根据需要而自动扩展,使用查询可以利用大量的临时空间。在执行一个查询时,可用一个 WHERE 语句来限制必须处理的行数,除非完全需要,否则应避免在一个表中无限制地读取并处理所有记录行。

- 将一个大的查询拆成多步执行查询。

- 添加更多内存。

- 在有多个处理器的计算机上运行 SQL Server。多个处理器使 SQL Server 得以利用并行查询。

- 如果查询使用游标,则确定如果使用效率更高的游标类型(如快速只进游标)或单纯查询能否更有效地编写游标查询。

- 如果应用程序使用循环,可考虑在查询内放入循环。

- 不要对同一查询内的单个表使用多个别名以模拟索引交叉。

- 利用 query governor 配置选项和设置。可以使用 query governor 配置选项阻止执行长时间运行的查询。从而防止消耗系统资源。

- 避免执行 Left join 和使用 NULL 值<sup>[3]</sup>

#### 参考文献

- 1 宋益众,Microsoft SQL Server 2000 数据库管理系统性能研究[J],电脑知识与技术,2005.18,69-72。
- 2 尹萍,SQL Server 数据库性能优化[J],计算机应用与软件,Vol 22, No 3, Mar 2005, 51-54。
- 3 邹承俊、唐晏,如何提高基于 SQL Server 应用程序的性能[J],电脑知识与技术,2005.9,8-9。
- 4 潘瑞芳,主流数据库性能比较及数据库构架分析[J],浙江传媒学院学报,2005 年第 1 期,62-64。