

通过使用索引提高 SQL SERVER 的性能

赵作辉^{1,2}, 卢伟清¹

(1.厦门大学信息科学与技术学院, 福建厦门, 361005; 2.徐州经贸高等职业学校, 江苏徐州, 221004)

摘要:以 SQL SERVER 2000 平台为例, 详细介绍了索引的实现原理, 探讨了如何通过表的索引来提高 SQL SERVER 的性能。

关键词:簇索引; 非簇索引; 索引调节; SQL SERVER

中图分类号:G254.36 文献标识码: A

索引是 SQL SERVER 使用的一种内部表结构, 它基于表中一个或多个列的值, 提供对表中行的快速存取。SQL SERVER 中的索引类似于一本书后的索引, 它能帮助你定位内容; 如果没有索引, SQL SERVER 要进行表格扫描, 读取表中的每一个记录才能找到所要的数据。如果设计合理, 索引可以提高查询的性能。但是, 为表中的每一列创建索引是不明智的。在创建索引的时候, 你要考虑一些时间、空间和性能的因素, 如创建索引要花时间, 每个创建的索引边同原先的数据表都需要空间来存储数据, 每次修改数据时索引都需要更新等。

1 索引的分类与结构

索引有两种类型: 簇索引、非簇索引。SQL SERVER 支持定义在表任意列(包括已计算的列)上的索引。如果表没有任何簇索引, 数据不以特定顺序存储, 这种结构叫做堆。

SQL SERVER 中的每个表在数据库文件中都分配了一个范围(8个连续的8kb页)。如果这个范围装满数据, 将给表分配另一个范围, 但是这些范围并非物理地连续地放在一块儿, 它们被分散在数据库文件中。因此, 存取堆中数据要比存取具有索引的表来得慢。

2 索引的实现原理

2.1 簇索引的工作原理

在簇索引中, 数据被物理地排序, 每个表允许创建一个簇索引, 所以你应在具有很高比例的唯一值和不经常被修改的属性上构建索引。

在簇索引中, 数据存储在 B- 树的中子层上, 表的数据页与以字母顺序存储在档案柜中的文件夹相似, 而数据行与存储在文件夹中文档相似。

当 SQL SERVER 使用簇索引查找获得根页需遵循以下步骤:

- 步骤 1: SQL SERVER 从 sysindexes 表中获得根页的地址, 查找值与根页中的关键值进行比较;
- 步骤 2: 找出小于或等于查找值的最大关键值的页;
- 步骤 3: 页指针进到索引的下一层;
- 步骤 4: 重复步骤 3 和步骤 4, 直至找到数据页;

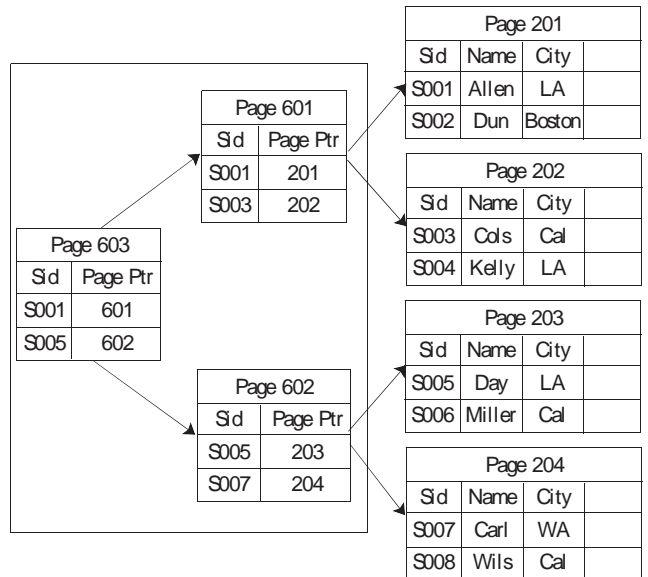


图 1 簇索引的存储结构

步骤 5: 在数据页上查找数据行, 直到找到查找值为止, 如果在数据页上找不到查找值, 则此查询无行返回。

考察图 1, 按 Sid 属性排序和存储在表中的 Student 表的行。

图 1 给出了出现在 Student 表上的簇索引。

为了查找任意记录, SQL SERVER 将从根页开始, 接着它沿着 B- 树移动, 将会在 B- 树的叶子页上找到数据值。例如, 如果使用上图显示的簇索引查找包含 Sid=S006 的行, 将要遵循以下步骤:

- 步骤 1: SQL SERVER 从 603 页的根页开始;
- 步骤 2: 它在页上查找小于或等于查找值的最大关键值。查找的结果是包含关于 Sid S005 指针的页;
- 步骤 3: 查找从 602 页继续;
- 步骤 4: Sid S005 在这里被找到, 查找在 203 页继续;

The Security Analysis on the JSP Application and the Solving Counter measures

YANG Ya-fei

ABSTRACT: This paper expounds the technical principle of JSP, analyzes on the reasons why the loophole in JSP environment occurs and the principles of network attack, and connecting with the practice, advances some countermeasures.

KEY WORDS: JSP; security loophole; network attack

步骤 5: 查找 203 页, 以找到所需的行。

2.2 非簇索引的工作原理

在非簇索引中, 行的物理顺序不同于索引的顺序; 非簇索引一般在用于连接和 WHERE 子句的列上创建, 且它的值可能被经常地修改; 当给出 CREATE INDEX 命令时, SQL SERVER 缺省地创建非簇索引, 每个表可多达 249 个非簇索引。

非簇索引的数据以随机顺序出现, 但逻辑顺序按其索引给出。数据行可能随机地分布于整个表。非簇索引树包含按照排序出现的索引键, 其索引的叶子层包含关于数据页的指针和数据叶子行号。

当 SQL SERVER 使用非簇索引查找值时, 它执行以下步骤:

- 步骤 1: SQL SERVER 从 sysindexes 表中获得根页的地址;
- 步骤 2: 查找值与根页中的关键值进行比较;
- 步骤 3: 找出小于或等于查找值的最大关键值的页;
- 步骤 4: 页指针进到索引的下一层;
- 步骤 5: 重复步骤 3 和步骤 4, 直到找到数据页;
- 步骤 6: 对于给出的值, 在叶子上查找行, 如果没有找到匹配的, 则此表没有包含匹配的行;

步骤 7: 如果找到匹配的行, 指针进到表中的数据页和行-ID 现在检索所请求的行。

非簇索引的存储结构见图 2。

图 2 显示了出现在 Student 表的 SId 属性上的非簇索引。为了查找任意值, SQL SERVER 将从根页开始, 沿着 B- 树查找直到到达一个叶子页。它将在叶子页中找到关于真正行的指针, 然后它使用这个指针访问表中的真正行。例如, 如果使用显示于图 2 中的非簇索引查找包含 SId = S006 的行, 应遵循以下步骤:

- 步骤 1: SQL SERVER 从 603 页的根页开始;
- 步骤 2: 在页中查找小于或等于要查找值的页上最大关键值, 即包含关于 SId S005 指针的页;
- 步骤 3: 查找从 602 页继续;

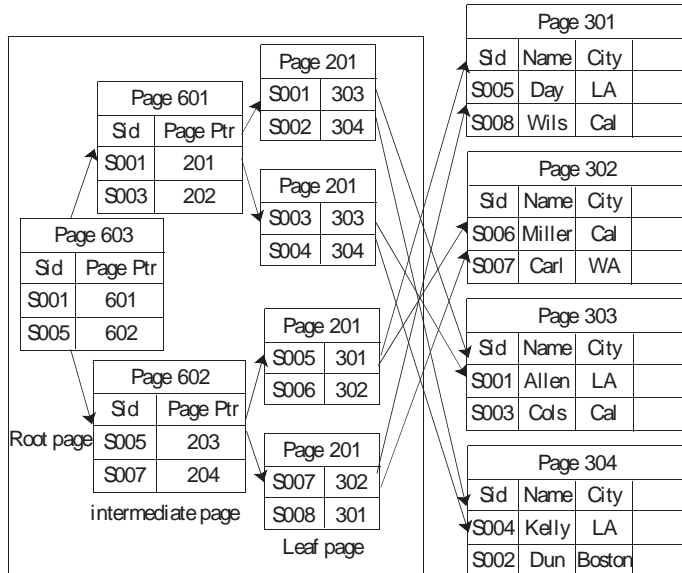


图 2 非簇索引的存储结构

步骤 4: SId S005 在这里被找到, 查找在 203 页继续;

步骤 5: 查找 203 页以找到指向真正行的指针, 203 页是最后或索引的叶子页;

步骤 6: 查找移向表的 203 页, 以找到真正的行。

3 通过索引的合理设计提高查询性能

索引的主要作用是提高存取数据页的速度。服务器并非对所需的数据扫描每一页, 而是扫描索引, 获得存储单元的地址, 并直接地存取信息。索引也是一种用于实施数据唯一性的机制, 创建唯一索引将使列出唯一的值。通常创建在连接、WHERE 子句、ORDER BY 子句和 GROUP BY 子句中用到的列上, 加速了表之间的连接, 提高了查询的速度。

建立索引时要考虑以下因素: 索引可用来实施行的唯一性, 索引对数据大多数是唯一的列很有用; 索引对具有大量重复数据的列没有大用处; 当你修改索引列的数据时, 相关索引会被自动更新, 你需要时间和资源来维护索引; 你不应该创建不被经常性使用的索引; 簇索引应在非簇索引之前被创建, 簇索引改变了行的顺序, 如果非簇索引在簇索引之前被创建, 那么它需要被重新构造; 典型的非簇索引在外键上创建。

建立索引是数据库设计时重要考虑的因素, 在性能测试的过程, 我们可以通过 SQL SERVER 提供的索引调节向导 (Index Tuning Wizard), 它为我们选择和创建可能最好的关于数据库的一组索引提供帮助。同时使用此向导的主要优点是: 你不必具有数据库复杂性、表之间关系以及工作负载和服务器工作的全面知识。

需要向 Index Tuning Wizard 指出将在数据库上执行的查询的类型, 向导将跟着分析要在数据库执行的查询, 并建议为提高性能而需要创建的索引。在数据库中被重复执行的查询的集合被称为工作负载, 一个工作负载可能是一批 SQL 语句, 也可以是作为文件 SQL SERVER 跟踪。如果不存在工作负载, 你可以使用 SQL Profiler 创建工作负载。

在确定了工作负载后, 你可以运行 Index Tuning Wizard, 向导将建议一组可能是最好的索引, 为了提高数据库性能, 这些索引是需要的。

以下是 Index Tuning Wizard 的使用: 对给定的工作负载, 为数据库建议一组可能是最好的索引组合; 建议推荐的关于索引、表中查询的分布、工作负载中的查询性能等的影响都可被分析; 对于少量的问题查询, 将推荐调整数据库的方法; 它将指出高级选项, 如可以被定制的磁盘空间约束。

向导提供的建议可立即执行, 或者被保存以后由用户手工执行, 或以后通过创建执行 SQL 脚本的 SQL SERVER 去实现。

通过索引的合理设计提高查询性能, 当然在关于表的索引方面, 提高 SQL SERVER 的性能还涉及运行过程, 索引文件和数据文件都可能产生数据碎片。我们可以运用 SQL SERVER 提供的 DBCC SHOWCONTIG 和 DBCC DBREINDEX 工具来提高查询性能。

参考文献

[1] 萨师焯, 王珊. 数据库系统概论 [M]. 北京: 高等教育出版社, 2004: 136- 151.

(责任编辑: 戚米莎)

第一作者简介: 赵作辉, 男, 1972 年 2 月生, 1996 年毕业于淮海工学院, 现为厦门大学信息科学与技术学院 2003 级在职硕士研究生, 讲师, 徐州经贸高等职业学校信息技术系, 江苏省徐州市, 221004.

Improve the Performance of the SQL SERVER through the Use of Index Table

ZHAO Zuo-hui, LU Wei-qing

ABSTRACT: Taking the platform SQL SERVER 2000 as an example, this paper introduces in detail the principle of realizing the index, and probes into how to improve the performance of the SQL SERVER through the use of the index table.

KEY WORDS: cluster index; non-cluster index; index adjustment; SQL SERVER