

关于SQLServer中索引使用及维护简介 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/234/2021_2022__E5_85_B3_E4_BA_8ESQLS_c67_234028.htm

在应用系统中,尤其在联机事务处理系统中,对数据查询及处理速度已成为衡量应用系统成败的标准。而采用索引来加快数据处理速度也成为广大数据库用户所接受的优化方法。在良好的数据库设计基础上,能有效地使用索引是SQL Server取得高性能的基础,SQLServer采用基于代价的优化模型,它对每一个提交的有关表的查询,决定是否使用索引或用哪一个索引。因为查询执行的大部分开销是磁盘I/O,使用索引提高性能的一个主要目标是避免全表扫描,因为全表扫描需要从磁盘上读表的每一个数据页,如果有索引指向数据值,则查询只需读几次磁盘就可以了。所以如果建立了合理的索引,优化器就能利用索引加速数据的查询过程。但是,索引并不总是提高系统的性能,在增、删、改操作中索引的存在会增加一定的工作量,因此,在适当的地方增加适当的索引并从不合理的地方删除次优的索引,将有助于优化那些性能较差的SQLServer应用。实践表明,合理的索引设计是建立在对各种查询的分析和预测上的,只有正确地使索引与程序结合起来,才能产生最佳的优化方案。本文就SQLServer索引的性能问题进行了一些分析和实践。

一、聚簇索引(clustered indexes)的使用 聚簇索引是一种对磁盘上实际数据重新组织以按指定的一个或多个列的值排序。由于聚簇索引的索引页面指针指向数据页面,所以使用聚簇索引查找数据几乎总是比使用非聚簇索引快。每张表只能建一个聚簇索引,并且建聚簇索引需要至少相当该表120%的附加空间

，以存放该表的副本和索引中间页。建立聚簇索引的思想是：
1、大多数表都应该有聚簇索引或使用分区来降低对表尾页的竞争，在一个高事务的环境中，对最后一页的封锁严重影响系统的吞吐量。
2、在聚簇索引下，数据在物理上按顺序排在数据页上，重复值也排在一起，因而在那些包含范围检查(between、>=)或使用group by或order by的查询时，一旦找到具有范围中第一个键值的行，具有后续索引值的行保证物理上毗连在一起而不必进一步搜索，避免了大范围扫描，可以大大提高查询速度。
3、在一个频繁发生插入操作的表上建立聚簇索引时，不要建在具有单调上升值的列(如IDENTITY)上，否则会经常引起封锁冲突。
4、在聚簇索引中不要包含经常修改的列，因为码值修改后，数据行必须移动到新的位置。
5、选择聚簇索引应基于where子句和连接操作的类型。聚簇索引的侯选列是：
1、主键列,该列在where子句中使用并且插入是随机的。
2、按范围存取的列，如pri_order > 100 and pri_order < 300。
3、在group by或order by中使用的列。
4、不经常修改的列。
5、在连接操作中使用的列。

二、非聚簇索引(nonclustered indexes)的使用 SQLServer缺省情况下建立的索引是非聚簇索引，由于非聚簇索引不重新组织表中的数据，而是对每一行存储索引列值并用一个指针指向数据所在的页面。换句话说非聚簇索引具有在索引结构和数据本身之间的一个额外级。一个表如果没有聚簇索引时,可有250个非聚簇索引。每个非聚簇索引提供访问数据的不同排序顺序。在建立非聚簇索引时，要权衡索引对查询速度的加快与降低修改速度之间的利弊。另外，还要考虑这些问题：
1、索引需要使用多少空间。
2、合适的列是否稳定。
3、索引键

是如何选择的，扫描效果是否更佳。4、是否有许多重复值。对更新频繁的表来说，表上的非聚簇索引比聚簇索引和根本没有索引需要更多的额外开销。对移到新页的每一行而言，指向该数据的每个非聚簇索引的页级行也必须更新，有时可能还需要索引页的分理。从一个页面删除数据的进程也会有类似的开销，另外，删除进程还必须把数据移到页面上部，以保证数据的连续性。所以，建立非聚簇索引要非常慎重。非聚簇索引常被用在以下情况：1、某列常用于集合函数(如Sum,...)。2、某列常用于join,order by,group by。3、查寻出的数据不超过表中数据量的20%。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com