 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，SQL Server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SQL Server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SQL Server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、lt.=、gt.=)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 lt. 200。 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 SQL Server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%。三、覆盖索引(covering indexes)的使用 覆盖索引是指那些索引项中包含查寻所需要的全部信息的非聚簇索引，这种索引之所以比较快也正是因为索引页中包含了查寻所必须的数据,不需去访问数据页。如果非聚簇索引中包含结果数据,那么它的查询速度将快于聚簇索引。 但是由于覆盖索引的索引项比较多,要占用比较大的空间。而且0update操作会引起索引值改变。所以如果潜在的覆盖查询并不常用或不太关键，则覆盖索引的增加反而会降低性能。 四、索引的选择技术 p_detail是住房公积金管理系统中记录个人明细的表，有890000行，观察在不同索引下的查询运行效果，测试在C/S环境下进行，客户机是IBM PII350(内存64M),服务器是DEC Alpha1000A(内存128M),数据库为SYBASE11.0.3。1、 0select count(*) from p_detail where op_datelt.’19991231’and pri_surplus1gt.’19990101’and pay_month between‘199908’and’199912’不建任何索引查询1 1分15秒 查询2 1分7秒 在op_date上建非聚簇索引查询1 57秒 查询2 57秒 在op_date上建聚簇索引查询1 lt.1秒 在op_date、pay_month、pri_surplus1上建索引查询1 lt.1秒 从以上查询效果分析，索引的有无，建立方式的不同将会导致不同的查询效果，选择什么样的索引基于用户对数据的查询条件,这些条

件体现于where从句和join表达式中。一般来说建立索引的思路是：(1)主键时常作为where子句的条件，应在表的主键列上建立聚簇索引，尤其当经常用它作为连接的时候。(2)有大量重复值且经常有范围查询和排序、分组发生的列，或者非常频繁地被访问的列，可考虑建立聚簇索引。(3)经常同时存取多列，且每列都含有重复值可考虑建立复合索引来覆盖一个或一组查询，并把查询引用最频繁的列作为前导列，如果可能尽量使关键查询形成覆盖查询。(4)如果知道索引键的所有值都是唯一的，那么确保把索引定义成唯一索引。(5)在一个经常做插入操作的表上建索引时，使用fillfactor(填充因子)来减少页分裂，同时提高并发度降低死锁的发生。如果在只读表上建索引，则可以把fillfactor置为100。(6)在选择索引键时，设法选择那些采用小数据类型的列作为键以使每个索引页能够容纳尽可能多的索引键和指针，通过这种方式，可使一个查询必须遍历的索引页面降到最小。此外，尽可能地使用整数为键值，因为它能够提供比任何数据类型都快的访问速度。 五、索引的维护 上面讲到,某些不合适的索引影响到SQL Server的性能,随着应用系统的运行,数据不断地发生变化,当数据变化达到某一个程度时将会影响到索引的使用。这时需要用户自己来维护索引。索引的维护包括：1、重建索引 随着数据行的插入、删除和数据页的分裂，有些索引页可能只包含几页数据，另外应用在执行大块I/O的时候，重建非聚簇索引可以降低分片，维护大块I/O的效率。重建索引实际上是重新组织B-树空间。在下面情况下需要重建索引：(1)数据和使用模式大幅度变化。(2)排序的顺序发生改变。(3)要进行大量插入操作或已经完成。(4)使用大块I/O的查询的磁

盘读次数比预料的要多。(5)由于大量数据修改，使得数据页和索引页没有充分使用而导致空间的使用超出估算。(6)dbcc检查出索引有问题。当重建聚簇索引时，这张表的所有非聚簇索引将被重建。2、索引统计信息的更新 当在一个包含数据的表上创建索引的时候，SQL Server会创建分布数据页来存放有关索引的两种统计信息：分布表和密度表。优化器利用这个页来判断该索引对某个特定查询是否有用。但这个统计信息并不动态地重新计算。这意味着，当表的数据改变之后，统计信息有可能是过时的，从而影响优化器追求最有工作的目标。因此，在下面情况下应该运行0update statistics命令：(1)数据行的插入和删除修改了数据的分布。(2)对用truncate table删除数据的表上增加数据行。(3)修改索引列的值。六、结束语 实践表明，不恰当的索引不但于事无补，反而会降低系统的执行性能。因为大量的索引在插入、修改和删除操作时比没有索引花费更多的系统时间。例如下面情况下建立的索引是不恰当的：1、在查询中很少或从不引用的列不会受益于索引，因为索引很少或从来不必搜索基于这些列的行。2、只有两个或三个值的列，如男性和女性(是或否)，从不会从索引中得到好处。另外，鉴于索引加快了查询速度，但减慢了数据更新速度的特点。可通过在一个段上建表，而在另一个段上建其非聚簇索引，而这两段分别在单独的物理设备上来改善操作性能。2009年上半年全国计算机等级考试参考答案请进入计算机考试论坛 2009年全国计算机等级考试报名信息汇总 2009年NCRE考试有新变化 2009年全国计算机等级考试大纲 2009年上半年全国计算机二级考试试题及答案 2009年上半年全国计算机等级考试试题答案汇总 100Test 下载频道