|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | | |

|  |  |  |
| --- | --- | --- |
|  | **Index & Performance** |  |
|  | | |
|  | 在应用系统中，尤其在联机事务处理系统中，对数据检索及处理速度已成为衡量应用系统成败标准。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。  　　索引的使用效果不仅仅依赖于SQL Server的优化策略，在相当程度上也依赖于应用程序的设。怎样正确地使用索引，不能一概而论，究竟是让索引满足程序设计的需要，或是程序设计遵循已建立的索引，这两者是相符相承的。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。  　　建立索引的目地是为了优化检索速度，如果检索所需要的时间过长，便有理由怀疑是否索引不存在或者优化器没有使用索引。尚若是索引不存在，那么就要取决于用户是否愿意用空间来换取时间，使用索引来解决检索速度慢的问题。如果优化器未使用表上已有的索引，那么要分析为什么，关于这一点将在后一点篇幅来说明如果update的效率很低，很可能是由于表上有太多的索引需要维护，从而浪费了时间。  **优化器怎样使用索引**   Table scan  　　如果表上没有任何索引，那么检索将采用Table Scan方式进行，其所用时间主要依赖于表的大小。  　　例如：  - dbcc checktable 测出表占76923页 -系统每秒读取50页 -76923页/50页/秒=1538秒 （大约25分钟）  　　如果系统有比较大的cathe，某些数据可能由于以前已被读到内存，那么读取数据时间可能会低于估计的时间。一般情况下，Tablescan检索是由于表上没有ClusteredIndex或者优化器认为，表中将有20%的数据做为结果追回。    使用索引（条件为指定值）  索引中包含指定记录的值及地址，SQL Server不必做全表扫描。  　　例： select \* from title where title\_id="mc3021"  当优化器认为读取索引页I/O加读取数据页I/O比做Table Scan效果更好时，检索将使用索引。    使用索引（条件为某范围内值）  例：  select \* from titles  where title\_id >"BU1032"  and title-id <"mc3032"  如果数据是排序的（有Clustered索引），索引将被用来限制数据的扫描范围。   使用索引避免检索排序所需要的时间。  例：  select \* from titles  order by title\_id  对Clustered索引来说，如果索引顺序与Server顺序一致，那么上面的查寻不需要重新排列返回结果。但是，若数据存储本身是升序排列，而查寻要求降序排列，那么索引对加快查寻没有任何作用。  　　对于Non-Clustered索引，优化器将判断查寻Non\_Clustered索引页，找到满足条件的数据进行排序是否比Table Scan更快，优化器将找出最佳结果。从以上几例可以看出，并非在表上建立了Clustered或on-Clustered索引之后，就一定会被使用，优化器是否使用索引取决于数据的查寻命令，SQL Server将从几个检索方案中选择最佳的一个。  在什么样的条件下才选择Clustered索引呢？  　　选择什么样的索引基于用户对数据的检索条件，这些条件体现于where从句和join表达式。如果你的应用与以下情况相符，你可以考虑选择Clustered索引。   * 主键时常作为where子句的条件 * 某一列经常以这样的格式出现在where表达式中(x<=column <="y)" * 某一列非常频繁地被访问 * 某列被用作order by或group by * 某列很少被改写 * 某列常出现在join中。   **Non-Clustered常被用在以下情况：**   * 某列常用于Aggregate函数（如Sum,....） * 某列常用于join,order by,group by。 * 查寻检索出的数据不超过表中数据量的20%。   **索引覆盖**  　　检索覆盖是指Non\_Clustered索引项中包含查寻所需要的全部信息。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Non-Clustered索引中包含结果数据，那么它的检索速度将快于Clustered索引。覆盖索引的缺点：由于索引项比较多，要占用比较大的空间。而且update操作会引起索引值改变。  SQL Server对索引的限制   * 每个表上最多仅能有一个Clustered索引。 * 如果表上有一个Clustered索引，最多还能有249 Non-Clustered索引。 * 当没有Clustered索引时，则可有250个Non-Clustered索引。 * 索引最多建立在256个列上。 * 当索引被创建时，SQL Server需要120%的附加空间。   **索引维护**  　　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。上面讲到，某些不合适的索引影响到SQL Server的性能，这时需要用户自己来维护索引。一种方法是删除老的索引，重新建新的索引。另外一种方法是保持索引统计有效（使用命令update statistics），在以下情况下需要重新索引。   * 使用数据模式发生了较大变化。 * 某段时间内有极大量的数据插入。 * SQL Server排序改变。 * dbcc发现索引错误。   　　当重建Clustered索引时，这张表的所有Non-Clustered索引将被重建。  **维护索引统计表：**  　　数据库拥有者必须用命令维护统计表。 update statistics table\_name [index\_name]  **索引优化调整**  　　用这条命令可以改善创建索引的性能,减少建索引所用的时间。 sp\_configure "extent i/o buffers",nnnn带来的影响是增加了extent i/o buffers大小，在SQL Server使用内存不变情况下，减少了procedure和data cathe，而且同一时刻仅有一个用户能用到extent buffer。 |  |
|  |  |  |