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

|  |
| --- |
| **关于tempdb的优化** |
| 缺省情况下，tempdb数据库是放置在master设备上，容量为2M，而临时数据库是活动最为平凡的数据库常常被用来排序、创建临时表、重格式化等操作，所以tempdb的优化应该受到特别的关注。  **第一步：将临时数据库与高速缓冲进行绑定。** 　　由于临时表的创建、使用，临时数据库会频繁地使用数据缓存，所以应为临时数据库创建高速缓存，从而可以使其常驻内存并有助于分散I/O： 　　1、创建命名高速缓存 　　　　sp\_cacheconfig “tempdb\_cache”,”10m”,”mixed” 　　2、重新启动server 　　3、捆绑临时数据库到tempdb\_cache高速缓存 　　　　sp\_bindcache “tempdb\_cache”, tempdb 　　4、若有大的I/O，配置内存池  **第二步：优化临时表**大多数临时表的使用是简单的，很少需要优化。但需要对临时表进行复杂的访问则、 　　应通过使用多个过程或批处理来把表的创建和索引分开。以下两种技术可以改善临时表的优化 　　1、在临时表上创建索引 　　　　1）临时表必须存在 　　　　2）统计页必须存在（即不能在空表上创建索引） 　　2、把对临时表的复杂的使用分散到多个批处理或过程中，以便为优化器提供信息 　　　　下面的这个过程需要进行优化： 　　　　　create proc base\_proc 　　　　　as 　　　　　select \* into #huge\_result from auths 　　　　　select \* from article, #huge\_result where article.author\_code= 　　　　　#huge\_result.author\_code and sex=”0”  　　使用两个过程可以得到更好的性能 　　1） 　　　　create proc base\_proc  　　　　as 　　　　select \* 　　　　into #huge\_result 　　　　from auths 　　　　exec select\_proc  　　2)  　　　　create proc select\_proc  　　　　as 　　　　select \* from article,#huge\_result 　　　　where article.author\_code=#huge\_result.author\_code and sex=”0”  　　说明：在同一个存储过程或批处理中，创建并使用一个表时，查询优化器无法决定这个表的大小。 |