

OCP033自制讲稿调整共享池pdf完整版 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/246/2021_2022_OCP033_E8_87_AA_E5_c102_246199.htm

1.共享池中的主要组件 /库缓存(library cache) 主要缓存共享SQL和PL/SQL相关信息 /数据词典缓存(data dictionary cache) 缓存Dictionary objects的信息，用于解析权限，表结构等 /UGA(User global are) 在shared server模式下如果没有配置大池(Larger_pool_size=0)时，UGA会占用shared pool的空间 SGA中除Shared Pool外还有：大池 如果使用RMAN备份 并行查询 共享服务器时应该配置 另外还有 data buffer cache(ppt第五章), redo log buffer(ppt第七章),由于 2. 调整shared pool size应该考虑什么 由于库缓存和数据词典缓存的miss引起的操作时间往往比SGA中其他组件的cache miss要多，因此在SGA中 首先应该考虑调整shared pool。而当调整shared pool时，则应该首先集中在库缓存上，因为库缓存的大小和数据词典缓存的大小没有单独的设置，而是oracle自动按照一定的算法在shared pool中分配给他们，而按照oracle的内存空间分配的算法，如果库缓存的命中率高的话，数据词典缓存的命中率也会较高的；库缓存的命中率高而数据词典缓存的命中率却很低的话几乎是很少的。如果共享池很小，这往往会消耗很多CPU资源并且容易引起竞争；但是如果你把共享池设到很大，也不一定是好事，那为什么呢？当你把共享池配得很大，是不是消耗了很多内存资源那，那么其他得组件能使用得内存资源是不是就少了，另外，在这个很大的大池中，缓冲的内容是不是很多啊，多了，在里面查找是不是就慢了。所以说共享池虽然不能很小，但是也不是越大

越好，还是要适当。 3. 共享池相关 /这个共享池是由初始化参数SHARED_POOL_SIZE来定义的，默认是8M,这个参数是动态的 /库缓存的内容 包含了Statement的文本, 分析后的Code, 以及相应的执行计划 /数据词典缓存主要是system中的数据词典表(对象的列定义信息，用户的权限信息等) 4. 库缓存 /用来缓存共享的SQL statements 和PL/SQL块，这些可以给所有的连接用户共享。 /由LRU(最小使用最先淘汰算法)来管理, 千万不要以为是FIFO(先进先出)算法管理的 /oracle server怎样知道你要执行的Statement等被缓存了呢？ --先通过一个hash算法将Statement text运算成一个hash数值 --然后通过这个hash值去找 5 几个重要的共享池相关的锁存器(latch) /latch也是一种锁，但是是一中由oracle server自行管理的用来保护共享的内存结构的低级锁(low lock)，和后面章节讲的LOCK是由区别的，latch不用我们来维护。 /share pool latch: 用来保护共享池中的内存分配 /library cache latch: 在shared pool中查找匹配的SQL等是起到保护作用，保护在查找的短暂过程中这个列表不会被改变 什么时候会改变，由于oracle server可以同时被很多用户连接使用，在这个过程中很由可能会由新的SQL进来，而把老的很少用的SQL给恰巧挤出去了，而我要找的就是这个很少用的SQL时那可能不就出问题了吗，这个latch能够保证在查找的这个过程中不会出现这个问题。 而在这些latches上如过由竞争的话，表示SQL、PL/SQL被重用的机率很小，可能是因为没有使用绑定变量或者设置的游标缓存的数量不够多。那么要减少这种竞争，我们又该怎么做呢？我们可以考虑什么原因引起的，然后用什么对策。 引起竞争的原因：没有共享的SQL(没有使用绑定变量), 共享SQL重新硬分析了（可

以查询V\$sqlarea中的parse_calls和excutions字段，如果某个SQL对应的parse_calls接近excutions，说明这条SQL经常被重新硬分析），库缓存的大小不够根据上面的这些因素，我们可以考虑调整下面几种情况来减少竞争的发生，理想情况是减少到0啦。优化您的SQL, 检查"热点块(hot blocks,同时访问的人多)", 调整CURSOR_SHARING参数的值，增大shared_pool_size 如果使用了需要大池的操作而没有配置大池，可以配置大池（比如共享模式，可以把UGA从shared pool移到large pool）

6 调整库缓存目标 /保持最小次数的分析，尽可能减少丢失，如何做：--确保用户能够共享statement --分配足够的空间保证statement尽可能的不被赶出去 经常要用的statement要保证他们能够呆在share pool里被找到 --避免invalidations(由于数据字典发生变法而引起的Statement失效叫invalidations,比如改变了数据结构，重新收集了统计等) /避免碎片，这样会浪费空间 --为大内存要求的对象(比如大于1万byte的SQL) 保留空间,由对应的参数设置 --栓住那些场用的大对象,最好是在instance一移动，马上就把他们keep到shared pool中 --尽可能的不要使用匿名PL/SQL块（把他们写成一些小的包中的存储过程或函数） --为shared server连接配置大池

7. 术语 V\$libaraycache /命名空间：SQL AREA, TABLE/PROCEDURE, BODY, TRIGGER 等 /gets: (parse)每当一条语句被分析一次时，该语句对应的名称空间的gets加1 /gethits：分析时在对应的名称空间找到已经存在时加1，其分析后的代码和执行计划在内存中找到了，不再执行硬分析，直接使用 /pins: (excution)每当一条语句执行一次时，该语句对应的名称空间的的pins加1 /reloads: (parse)因为找到的分析代

码版本已经过期或作废而被重新硬分析的次数 /invalidations: 因为数据词典发生变化, 该语句被标记成失效, 被迫重新做硬分析的次数 对于每一个命名空间, 在v@librarycache中会有一行与其相对应, 可以分析每一个命名空间的get命中率和reloads(丢失率), 这两个比例对监控优化library cache非常有用。 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