

通过Oracle的缓冲区内部机制调整性能 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/142/2021_2022__E9_80_9A_E8_BF_87Orac_c102_142996.htm 在本篇Oracle高级教程里，我会探讨Oracle数据缓冲区的内部机制Oracle用这一内存来防止不必要的数据块从磁盘重读。理解Oracle数据缓冲区如何操作，是成功地运用它们调整数据库性能的关键。在Oracle 8i以前的版本里，当数据块被从磁盘送进数据缓冲区的时候，数据块会被自动地放置到最近使用过的数据列表的前部。但是，这种行为从Oracle 8i开始就变了：新数据缓冲区被放置在缓冲区链的中部。在调节数据库的时候，你的目标就是为数据缓冲区分配尽量多的内存，而不会导致数据库服务器在内存里分页。数据缓冲区每小时的命中率一旦低于90%，你就应该为数据块缓冲区增加缓冲区。数据块的存活时间在调入数据块之后，Oracle会不停地跟踪数据块的使用计数（touch count，也就是说，这个数据块被用户线程所访问的次数）。如果一个数据块被多次使用，它就被移动到最近使用过的数据列表的最前面，这样就能确保它会在内存里保存一段较长的时间。这种新的中点插入技术会确保最常使用的数据块被保留在最近使用过的数据列表的最前面，因为新的数据块只有在它们被重复使用的时候才会被移动到缓冲区链的最前面。总而言之，Oracle 8i数据缓冲池的管理要比先前的版本更加有效。通过将新的数据块插入缓冲区的中部，并根据访问活动（频率）调整缓冲区链，每个数据缓冲区就被分割成两个部分：热区（hot section），代表数据缓冲区的最近使用的一半；冷区（cold section），代表数据缓冲区的最早使用的

一半。只有那些被反复请求的数据块才会被移进每个缓冲池的热区，这就让每个数据缓冲区在缓冲常用数据块的时候效率更高。热区的大小要用下面的隐藏参数来配置：

`_db_percent_hot_default` `_db_percent_hot_keep`

`_db_percent_hot_recycle` Oracle公司作为官方没有推荐更改这些隐藏参数。只有懂得内部机制和希望调节其数据缓冲区行为的有经验人员才应该使用这些参数。找到热数据块 Oracle 8i保留着一个X\$BH内部查看表，用来显示数据缓冲池的相对性能。X\$BH查看表有下列数据列：Tim：两次使用之间的时间差，和`_db_aging_touch_time`参数相关。Tch：使用计数，它和被使用过`_db_aging_hot_criteria`次之后从冷区移入热区直接相关。由于Tch数据列用来追踪特定数据块的使用次数，所以你就能够编写一个字典查询来显示缓冲区里的热数据块使用计数大于10的数据块，就像下面这样：

```
SELECT obj object,
dbarfil file#, dbablk block#, tch touches
FROM x$bh
WHERE tch > 10
ORDER BY tch desc.
```

这项高级查询技术在用于追踪DEFAULT缓冲池里的对象时尤其有用。一旦定位了热数据块，你就能够把它们从DEFAULT缓冲池移动到KEEP缓冲池。完全缓冲数据库从Oracle 8i开始，随着64位寻址的出现，你就能够创建一个完全在数据缓冲区里缓冲的数据库。现今，任何数据库通常只用不到20G的内存就能够被完全缓冲，而更大的数据库仍然需要部分数据缓冲区。要利用完全数据缓冲的优势，就要记住从内存取回数据块和从磁盘取回数据块的时间差。对磁盘的访问时间是以毫秒或者说千分之一秒来计算的，而内存的速度是以纳秒或者说十亿分之一秒来计算的。因此内存的访问速度要快三个数量级，即通常要比磁盘的访问速度

块将近14,000倍。在完全缓冲Oracle数据库的时候，你需要仔细制定缓冲的规划，并在需要的时候增加db_block_buffers参数的值。在缓冲整个数据库的时候，多数据缓冲池就不再需要了，所以你可以在DEFAULT数据池里缓冲所有的数据块。

如果要计算已分配数据块的数量，就要用下面的命令：

```
0select sum(blocks)from dba_data_files.SUM(BLOCKS)-----  
2432600select sum(blocks)from  
dba_extents.SUM(BLOCKS)----- 321723
```

已使用数据块的真实数量要通过查看DBMS_ROWID以获得真实的数据库地址来确定，就像下面这样：

```
0select sum(blocks)from dba_data_files.
```

随着数据库的扩展，你必须不忘增加参数的值。在数据库的启动阶段，你需要调用一个脚本加载缓冲区，这一步很简单，就像执行0select count(*) from xxx.这样的命令，因为数据库里所有表格都会起作用。这就确保所有的数据块都被缓冲，从而大幅提高读取操作的性能。但是，写操作仍然会需要磁盘I/O。由于内存条的价格在持续下跌，现在只需要较小的或者不需要硬件投资就能够通过完全缓冲一个较小的数据库从而显著地提高性能。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