

oracle重建索引的必要性

参考：<http://blog.csdn.net/leshami/article/details/23763963>

索引重建是一个争论不休被不断热烈讨论的议题。当然Oracle官方也有自己的观点，我们很多DBA也是遵循这一准则来重建索引，那就是Oracle建议对于索引深度超过4级以及已删除的索引条目至少占有现有索引条目总数的20% 这2种情形下需要重建索引。近来Oracle也提出了一些与之相反的观点，就是强烈建议不要定期重建索引。本文是参考了1525787.1并进行相应描述。

1、重建索引的理由

- a、Oracle的B树索引随着时间的推移变得不平衡(误解)
- b、索引碎片在不断增加
- c、索引不断增加，删除的空间没有重复使用
- d、索引 clustering factor （集群因子）不同步，可以通过重建修复(误解)

2、重建索引的本质

本质：重建索引在数据库内部是先执行删除操作，再执行插入操作。

3、反对重建索引的理由

- a. 大多数脚本都依赖 index_stats 动态表。此表使用以下命令填充：

```
analyze index ... validate structure;
```

尽管这是一种有效的索引检查方法，但是它在分析索引时会获取独占表锁。对于大型索引，其影响会是巨大的，因为在此期间不允许对表执行DML 操作。

虽然该方法可以在不锁表的情况下在线运行，但是可能要消耗额外的时间。

- b. 重建索引的直接结果是 REDO 活动可能会增加，总体系统负载也可能会提高。

插入/更新/删除操作会导致索引随着索引的分割和增长不断发展。

重建索引后，它将连接的更为紧凑；但是，随着对表不断执行 DML 操作，必须再次分割索引，直到索引达到平衡为止。

结果，重做活动增加，且索引分割更有可能对性能产生直接影响，因为我们需要将更多的 I/O、CPU 等用于索引重建。

经过一段时间后，索引可能会再次遇到“问题”，因此可能会再被标记为重建，从而陷入恶性循环。

因此，通常最好是让索引处于自然平衡和（或）至少要防止定期重建索引。

4、Oracle的最终建议

一般而言，极少需要重建 B 树索引，基本原因是 B 树索引很大程度上可以自我管理或自我平衡。

大多数索引都能保持平衡和完整，因为空闲的叶条目可以重复使用。

插入/更新和删除操作确实会导致索引块周围的可用空间形成碎片，但是一般来说这些碎片都会被正确的重用。

Clustering factor集群因子反映了给定的索引键值所对应的表中的数据排序情况。重建索引不会对集群因子产生影响，集群因子只能通过重组表的数据改变。

强烈建议不要定期重建索引，而应使用合适的诊断工具。

个人结论，如果重建索引的巨大工作量与之对应的是极小的收益，那就得不偿失。如果系统有可用空闲期，重建之前和之后的测量结果表明性能有提高，值得重建。

5、改良方法

通常是优先考虑index coalesce（索引合并），而不是重建索引。索引合并有如下优点：

- a、不需要占用近磁盘存储空间 2 倍的空间
- b、可以在线操作
- c、无需重建索引结构，而是尽快地合并索引叶块，这样可避免系统开销过大。

6、真正需要重建索引的情形

索引或索引分区因介质故障损坏

标记为UNUSABLE的索引需要重建

索引移动到新的表空间或需要改变某些存储参数

通过SQL*Loader加载数据到表分区后，需要重建索引分区

重建索引以启用键压缩

位图索引本质不同于B树索引，建议重建

7、相关参考

[Oracle 聚簇因子\(Clustering factor\)](#)

[Oracle 索引监控\(monitor index\)](#)

[Oracle 索引监控与外键索引](#)

[收集统计信息导致索引被监控](#)

[Oracle 监控索引的使用率](#)

[NULL 值与索引\(一\)](#)

[NULL 值与索引\(二\)](#)

[函数使得索引列失效](#)

[Oracle 索引质量分析](#)