



YES, WE CAN!

Oracle 索引优化思路-案例分享

Leonarding @ACOUG



About Me

- 刘盛 网名:leonarding
- 金融机构~数据架构师
- DATAGURU 版主
- ITPUB 版主
- QQ 40354446
- Email: leonarding@yeah.net
- 新浪微博: @itpub_leonarding
- 技术博客blog: www.leonarding.com



基于索引的SQL优化

- 选择索引的几点思考
- 函数索引之排序索引的场景应用
- 索引优化思路

场景介绍:

sino_loan 是一张大表 **60w**行

sino_org 是一张小表 **29**行

对这**2**个表进行关联查询

目的:

分页显示, 每三十行一页

第一页显示rownum **1~30**的数据

第二页显示rownum **31~60**的数据

第三页显示rownum **61~90**的数据

采集统计信息做分析

```
execute dbms_stats.gather_table_stats('sinojfs','sino_loan',CASCADE=>TRUE);  
execute dbms_stats.gather_table_stats('sinojfs','sino_org',CASCADE=>TRUE);
```

让Oracle了解表中数据分布结构，生成合理的执行计划

需要优化的SQL语句

```
select l.*,org.sorgname from sino_loan l, sino_org org where org.sorgcode =  
l.sorgcode and l.iid in  
(select t1.iid from  
    ( select t.iid, rownum rn from  
        ( select loan.iid from sino_loan loan where loan.istate = 0 order by  
            loan.dbillingdate desc, loan.iid desc )  
        t where rownum <= 500060 )  
    t1 where rn >= 500031);
```

SQL分解

- 1.先取大表sino_loan的字段istate=0的记录并进行降序排列
- 2.因为要进行每页30行的分页显示，我们随机取rownum =(500031 ~ 500060)
- 3.最后把符合要求的30行记录做关联org.sorgcode = l.sorgcode

选择索引的几点思考

1. 要不要添加索引

如果读取的数据占全表的99%，是否还要选择索引，为什么

2. 添加索引是用于读取or过滤

读取：TABLE ACCESS BY INDEX ROWID 二次I/O

过滤：INDEX RANGE SCAN 一次I/O

3. 添加什么类型的索引最佳

找到性能瓶颈，针对性的选择

定位瓶颈点

Id	Operation	Name	Rows	Bytes	TempSpc	Cost (%CPU)	Time
0	SELECT STATEMENT		500K	147M		22702 (1)	00:04:33
* 1	HASH JOIN		500K	147M		22702 (1)	00:04:33
2	TABLE ACCESS FULL	SINO_ORG	29	493		3 (0)	00:00:01
* 3	HASH JOIN RIGHT SEMI		500K	139M	11M	22696 (1)	00:04:33
4	VIEW	VW_NSO_1	500K	6348K		8458 (1)	00:01:42
* 5	VIEW		500K	8790K		8458 (1)	00:01:42
* 6	COUNT STOPKEY						
7	VIEW		598K	2920K		8458 (1)	00:01:42
* 8	SORT ORDER BY STOPKEY		598K	8762K	32M	8458 (1)	00:01:42
* 9	TABLE ACCESS FULL	SINO_LOAN	598K	8762K		5377 (1)	00:01:05
10	TABLE ACCESS FULL	SINO_LOAN	598K	159M		5377 (1)	00:01:05

谓词条件

```
Predicate Information (identified by operation id):
```

```
-----  
1 - access ("ORG"."SORGCODE"="L"."SORGCODE")  
3 - access ("L"."IID"="$nso_col_1")  
5 - filter ("RN">=500031)  
6 - filter (ROWNUM<=500060)  
8 - filter (ROWNUM<=500060)  
9 - filter ("LOAN"."ISTATE"=0)
```

- 1.全部是全表扫描过滤
- 2.瓶颈点一 过滤代价高，这里不是读取是过滤，与索引的属性相似
- 3.瓶颈点二 排序代价高，有没有一种索引在创建的时候就是排好序的

函数索引之排序索引

语法:

create index 索引名 on 表名 (函数【字段】.....) 所属表空间
命令:

```
create index idx_sino_loan_c1 on sino_loan(istate,dbillingdate desc,iid desc)  
tablespace sinojfs_idx;
```

sino_loan大表三个字段istate, dbillingdate, iid上创建索引

索引目的解析

istate字段: 使用索引单字段过滤比全表扫描多字段效率高, 扫描块数少

dbillingdate & iid字段: 添加排序索引, 省略了排序过程, 节约时间。

排序索引特性: 先进行“键值”排序, 再保存。抽取出来的数据本身就是排好序的, 无需在临时段中进行排序操作, 从而提高效率。

添加索引后的执行计划

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	329	16264 (1)	00:03:16
1	NESTED LOOPS		1	329	16264 (1)	00:03:16
2	NESTED LOOPS		1	312	16263 (1)	00:03:16
3	VIEW	VW_NSO_1	500K	5860K	13969 (1)	00:02:48
4	HASH UNIQUE		1	11M		
* 5	VIEW		500K	11M	13969 (1)	00:02:48
* 6	COUNT STOPKEY					
7	VIEW		3301K	37M	13969 (1)	00:02:48
* 8	INDEX RANGE SCAN	IDX_SINO_LOAN_C1	3301K	88M	13969 (1)	00:02:48
9	TABLE ACCESS BY USER ROWID	SINO_LOAN	1	300	1 (0)	00:00:01
10	TABLE ACCESS BY INDEX ROWID	SINO_ORG	1	17	1 (0)	00:00:01
* 11	INDEX UNIQUE SCAN	PK_ORG	1		0 (0)	00:00:01

声明 sino_loan 表现有数据330w行， sino_org 表现有数据29行

添加索引后的执行计划

谓词条件

```
Predicate Information (identified by operation id):  
-----
```

```
5 - filter("RN">=500031)  
6 - filter(ROWNUM<=500060)  
8 - access("LOAN"."ISTATE"=0)  
11 - access("ORG"."SORGCODE"="L"."SORGCODE")
```

从六个谓词过滤降低到了四个

优化成果

小结：优化成果

- (1) 执行时间：从6秒降低到1秒
- (2) consistent gets：从48327降低到 2144
- (3) Cost：从58259降低到16264
- (4) 消除了全表扫描和排序耗时过程

索引优化思路

- 1 拿到SQL获取真需求
- 2 大数据扫描巧用索引缩小结果集
- 3 定位瓶颈，找出门路
- 4 了解结构，针对应用
- 5 精巧设计，SQL飞起来

篇外话

原始版SQL

```
select * from  
( select t.*, rownum rn from  
  ( select loan.*, org.sorgname from sino_loan loan,sino_org org where  
    loan.istate = 0 and loan.sorgcode = org.sorgcode order by loan.dbillingdate,loan.iid  
    desc ) t  
  where rownum <= 500060 )  
  where rn >= 500031;
```

优化版SQL

```
select l.*,org.sorgname from sino_loan l, sino_org org where org.sorgcode =  
l.sorgcode and l.iid in  
(select t1.iid from  
  ( select t.iid, rownum rn from  
    ( select loan.iid from sino_loan loan where loan.istate = 0 order by  
      loan.dbillingdate desc, loan.iid desc )  
    t where rownum <= 500060 )  
  t1 where rn >= 500031);
```

思维逻辑误区

- 1.SQL的创作常常是以人的思维逻辑来导向的人怎么想->SQL就怎么写
- 2.两种路线
 - (1)关联后产生59w行，然后选出30行
 - (2)优先从59w行选出30行，然后在关联不同路线，成本有着天壤之别
- 3.条条大道通罗马，实现同一个目标，可多思考多选择

Q & A