

Oracle 之虚拟索引

1.1 BLOG 文档结构图

Oracle 之虚拟索引
1.1 BLOG 文档结构图
1.2 前言部分
1.2.1 导读和注意事项
1.3 虚拟索引(Virtual Index)
1.3.1 虚拟索引类型及特点
1.4 实验部分
1.4.1 个人示例
About Me

1.2 前言部分

1.2.1 导读和注意事项

各位技术爱好者，看完本文后，你可以掌握如下的技能，也可以学到一些其它你所不知道的知识，~O(∩_∩)O~：

① Oracle 虚拟索引的使用

Tips：

① 本文在 ITpub (<http://blog.itpub.net/26736162>)、博客园 (<http://www.cnblogs.com/lhrbest>) 和微信公众号 (xiaomaimiaolhr) 有同步更新

② 文章中用到的所有代码，相关软件，相关资料请前往小麦苗的云盘下载 (<http://blog.itpub.net/26736162/viewspace-1624453/>)

③ 若文章代码格式有错乱，推荐使用搜狗、360 或 QQ 浏览器，也可以下载 pdf 格式的文档来查看，pdf 文档下载地址：<http://blog.itpub.net/26736162/viewspace-1624453/>，另外 itpub 格式显示有问题，可以去博客园地址

阅读

④ 本篇 BLOG 中命令的输出部分需要特别关注的地方我都用灰色背景和粉红色字体来表示，比如下边的例子中，

thread 1 的最大归档日志号为 33，thread 2 的最大归档日志号为 43 是需要特别关注的地方；而命令一般使用黄色背景和红色字体标注；对代码或代码输出部分的注释一般采用蓝色字体表示。

```
List of Archived Logs in backup set 11
Thrd Seq      Low SCN      Low Time      Next SCN      Next Time
-----
1      32          1621589      2015-05-29 11:09:52 1625242      2015-05-29 11:15:48
1      33          1625242      2015-05-29 11:15:48 1625293      2015-05-29 11:15:58
2      42          1613951      2015-05-29 10:41:18 1625245      2015-05-29 11:15:49
2      43          1625245      2015-05-29 11:15:49 1625253      2015-05-29 11:15:53

[ZHLHRDB1:root]:/>lsvg -o
T_XDESK_APP1_vg
rootvg
[ZHLHRDB1:root]:/>
00:27:22 SQL> alter tablespace idxtbs read write;

====> 2097152*512/1024/1024/1024=1G
```

本文如有错误或不完善的地方请大家多多指正，ITPUB 留言或 QQ 皆可，您的批评指正是我写作的最大动力。

1.3 虚拟索引(Virtual Index)

传统的性能优化和调整工作，大都是在系统上线之后，由运维团队进行的。当系统数据量积累到一定程度之后，原有一些隐藏的问题就不断出现。所以，在大数据量、应急场景下进行 SQL 调优，往往是运维团队经常遇到的问题。添加索引是我们经常使用的性能优化手段。在遇到问题的时候，试一试添加索引，看看能不能改变执行计划，是我们分析和解决问题的过程手段。但是对于大数据表情况下，快速的创建索引是比较困难的事情。这个时候，我们可以利用 Oracle 的 virtual index 技术。

在数据库优化中，索引的重要性不言而喻。但是，在性能调整过程中，一个索引是否能被查询用到，在索引创建之前是无法确定的，而创建索引是一个代价比较高的操作，尤其是数据量较大的时候。

虚拟索引(Virtual Index)是定义在数据字典中的伪索引，但没有相关的索引段。虚拟索引的目的是模拟索引的存在而不用真实的创建一个完整索引。这允许开发者创建虚拟索引来查看相关执行计划而不用等到真实创建完索引

才能查看索引对执行计划的影响，并且不会增加存储空间的使用。如果我们观察到优化器生成了一个昂贵的执行计划并且 SQL 调整指导建议我们对某些的某列创建索引，但在生产数据库环境中创建索引与测试并不总是可以操作。我们需要确保创建的索引将不会对数据库中的其它查询产生负面影响，因此可以使用虚拟索引。

虚拟索引不是物理存在的，它并不会创建实际的索引段，只是在数据字典中加了一个索引的记录，使得优化器能够意识到一个索引的存在，从而判断是否使用该索引作为访问路径。当然，实际上最终查询的访问路径是不会使用该虚拟索引的。所以，虚拟索引的用处就是用来判断一个索引对于 SQL 的执行计划的影响，尤其是对整个数据库的影响，从而判断是否需要创建物理索引。可以将这个索引删掉并重建常规索引。

虚拟索引与不可见索引的不同之处在于不可见索引是有与之相关的存储的，只是优化器不能选择它们。而虚拟索引没有与之相关的存储空间。由于这个原因，虚拟索引也被称为无段索引。

Oracle 文档中并没有提到虚拟索引的创建语法，实际上就是普通索引语法后面加一个 NOSEGMENT 关键字即可，B*TREE INDEX 和 BITMAP INDEX 都可以。

必须设置隐含参数 "_USE_NOSEGMENT_INDEXES"=TRUE (默认为 FALSE) 后 CBO 优化器模式才能使用虚拟索引，RBO 优化器模式无法使用虚拟索引。

--创建虚拟索引，首先要将 _use_nosegment_indexes 的隐含参数设置为 true

```
SQL> alter session set "_use_nosegment_indexes"=true;
```

--虚拟索引的创建语法比较简单，实际上就是普通索引语法后面加一个 nosegment 关键字

```
SQL> create index ix_t_id on t(object_id) nosegment;
```

索引已创建。

Oracle Virtual Index 是一个研究工具，是我们在投产环境上继续 SQL 优化方案研究时候的不错工具。它既满足了让我们创建索引，看执行计划效果的需求。同时也不会消耗很多的索引 build 资源。

查找系统中已经存在的虚拟索引：

```
SELECT INDEX_OWNER, INDEX_NAME
FROM DBA_IND_COLUMNS
WHERE INDEX_NAME NOT LIKE 'BIN$%'
MINUS
SELECT OWNER, INDEX_NAME
```

```
FROM DBA_INDEXES;
```

1.3.1 虚拟索引类型及特点

虚拟索引支持 B-TREE 索引和 BIT 位图索引，在 CBO 模式下 ORACLE 优化器会考虑虚拟索引，但是在 RBO 模式下需要添加 hint 才行。

同样的数据量，若使用 nosegment 虚拟索引使用的时间很短。

1. 虚拟索引无法执行 alter index 选项

```
SQL> alter index IX_T_ID rebuild;
alter index IX_T_ID rebuild*
第 1 行出现错误:
ORA-08114: 无法变更假索引
```

2. 使用回收站特性的时候，虚拟索引必须显示 drop，才能创建同名的索引。

```
SQL> create index ind_status on t(status);
索引已创建。
SQL> drop table t;
表已删除。
SQL> flashback table t to before drop;
闪回完成。
SQL> select table_name,index_name,status from user_indexes where table_name='T';
TABLE_NAME          INDEX_NAME          STATUS
-----
T                   BIN$7jAF1UG6blzgQAB/AQAPyw==$0 VALID
SQL> create index ind_object_id on t(object_id);
索引已创建。
SQL> create index inds_status on t(status);create index inds_status on t(status)
*
第 1 行出现错误:
ORA-01408: 此列列表已索引
```

3. 不能创建和虚拟索引同名的实际索引；
4. 可以创建和虚拟索引包含相同列但不同名的实际索引；
5. 虚拟索引分析并且有效，但是数据字典里查不到结果。

1.4 实验部分

1.4.1 个人示例

```
SYS@lhrdb> select * from v$version;
BANNER
-----
Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
PL/SQL Release 11.2.0.4.0 - Production
CORE 11.2.0.4.0 Production
TNS for IBM/AIX RISC System/6000: Version 11.2.0.4.0 - Production
```

```
NLSRTL Version 11.2.0.4.0 - Production
SYS@lhrdb> CREATE TABLE T_VI_20160818_01_LHR AS SELECT * FROM DBA_OBJECTS;
Table created.
```

虚拟索引的创建语法比较简单，实际上就是普通索引语法后面加一个 `nosegment` 关键字

```
SYS@lhrdb> CREATE INDEX IX_VI01_ID ON T_VI_20160818_01_LHR(OBJECT_ID) NOSEGMENT;
Index created.
```

从数据字段中是无法找到这个索引的。

```
SYS@lhrdb> SELECT INDEX_NAME,STATUS FROM DBA_INDEXES WHERE TABLE_NAME='T_VI_20160818_01_LHR';
no rows selected
SYS@lhrdb> COL OBJECT_NAME FORMAT A10
SYS@lhrdb> SELECT D.OWNER,D.OBJECT_NAME,D.OBJECT_TYPE FROM DBA_OBJECTS D WHERE
D.OBJECT_NAME='IX_VI01_ID';
OWNER                                OBJECT_NAME OBJECT_TYPE
-----
SYS                                IX_VI01_ID INDEX
SYS@lhrdb> SELECT TO_CHAR(DBMS_METADATA.GET_DDL('INDEX','IX_VI01_ID')) FROM DUAL;
TO_CHAR(DBMS_METADATA.GET_DDL('INDEX','IX_VI01_ID'))
-----
CREATE INDEX "SYS"."IX_VI01_ID" ON "SYS"."T_VI_20160818_01_LHR" ("OBJECT_ID")
PCTFREE 10 INITRANS 2 MAXTRANS 255 NOSEGMENT
```

使用虚拟索引，首先要将 `_use_nosegment_indexes` 的隐含参数设置为 `true`

```
SYS@lhrdb> ALTER SESSION SET "_USE_NOSEGMENT_INDEXES"=TRUE;
Session altered.
SYS@lhrdb> SHOW PARAMETER optimizer_mode
NAME                                TYPE                                VALUE
-----
optimizer_mode                      string                             ALL_ROWS
SYS@lhrdb> SET AUTOTRACE TRACEONLY
SYS@lhrdb> SET LINE 9999
SYS@lhrdb> SELECT * FROM T_VI_20160818_01_LHR WHERE OBJECT_ID=1;
no rows selected
Execution Plan
-----
Plan hash value: 3209519479
```

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		14	2898	5 (0)	00:00:01
1	TABLE ACCESS BY INDEX ROWID	T_VI_20160818_01_LHR	14	2898	5 (0)	00:00:01
* 2	INDEX RANGE SCAN	IX_VI01_ID	312		1 (0)	00:00:01

Predicate Information (identified by operation id):

```
-----
2 - access("OBJECT_ID"=1)
Note
-----
- dynamic sampling used for this statement (level=2)
Statistics
```

```
-----
0 recursive calls
0 db block gets
1249 consistent gets
0 physical reads
0 redo size
1343 bytes sent via SQL*Net to client
509 bytes received via SQL*Net from client
1 SQL*Net roundtrips to/from client
0 sorts (memory)
0 sorts (disk)
0 rows processed
SYS@lhrdb>
```

以下看的是真实执行计划，显然是用不到索引。

```
SYS@lhrdb> SET AUTOTRACE OFF
SYS@lhrdb> ALTER SESSION SET STATISTICS_LEVEL=ALL;
Session altered.
SYS@lhrdb> SELECT * FROM T_VI_20160818_01_LHR WHERE OBJECT_ID=1;
no rows selected
SYS@lhrdb> SELECT SQL_ID,CHILD_NUMBER,SQL_TEXT FROM V$SQL WHERE SQL_TEXT LIKE '%SELECT * FROM
T_VI_20160818_01_LHR WHERE OBJECT_ID=1%';
SQL_ID          CHILD_NUMBER  SQL_TEXT
-----
d5v59m8vyyz7d    0  SELECT * FROM T_VI_20160818_01_LHR WHERE OBJECT_ID=1
SYS@lhrdb> SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY_CURSOR('d5v59m8vyyz7d',0,'ALLSTATS LAST'));
PLAN_TABLE_OUTPUT
-----
SQL_ID d5v59m8vyyz7d, child number 0
-----
SELECT * FROM T_VI_20160818_01_LHR WHERE OBJECT_ID=1
Plan hash value: 847945500

| Id | Operation          | Name                | Starts | E-Rows | A-Rows |   A-Time   | Buffers |
|----|-----|-----|-----|-----|-----|-----|-----|
|  0 | SELECT STATEMENT    |                     |      1 |        |        | 00:00:00.01 |    1249 |
|*  1 | TABLE ACCESS FULL | T_VI_20160818_01_LHR |      1 |      14 |        | 00:00:00.01 |    1249 |

Predicate Information (identified by operation id):
-----
   1 - filter("OBJECT_ID"=1)
Note
-----
   - dynamic sampling used for this statement (level=2)
22 rows selected.
```

查找系统中已经存在的虚拟索引：

```
SYS@lhrdb> SELECT INDEX_OWNER, INDEX_NAME
2 FROM DBA_IND_COLUMNS
3 WHERE INDEX_NAME NOT LIKE 'BIN$%'
4 MINUS
5 SELECT OWNER, INDEX_NAME
6 FROM DBA_INDEXES;
```

INDEX_OWNER	INDEX_NAME
SYS	IX_VI01_ID

About Me

- 本文作者：小麦苗，只专注于数据库的技术，更注重技术的运用
- 本文在 ITpub (<http://blog.itpub.net/26736162>)、博客园(<http://www.cnblogs.com/lhrbest>)和个人微信公众号 (xiaomaimiao_lhr) 上有同步更新
- QQ 群：230161599 微信群：私聊
- 本文 itpub 地址： <http://blog.itpub.net/26736162/viewspace-2123687/> 本文博客园地址：

<http://www.cnblogs.com/lhrbest/articles/5784950.html>

- 本文 pdf 版：<http://yunpan.cn/cdEQedhCs2kFz>（提取码：ed9b）
 - 小麦苗分享的其它资料：<http://blog.itpub.net/26736162/viewspace-1624453/>
 - 联系我请加 QQ 好友(642808185)，注明添加缘由
 - 于 2016-08-18 15:00~ 2016-08-18 19:00 在中行完成
 - 【版权所有，文章允许转载，但须以链接方式注明源地址，否则追究法律责任】
-

长按识别二维码或微信客户端扫描下边的二维码来关注小麦苗的微信公众号：xiaomaimiaolhr，学习最实用的数据库技术。



小麦苗