

ORACLE 11GR2 COMPRESS 压缩技术介绍

BY SHOUG. 黄乾

SH'OUG

SHANGHAI ORACLE USERS GROUP

上海ORACLE用户组

How to Find SHOUG?



Google 上海oracle用户组

Web Images Maps Shopping More Search tools

About 5,960,000 results (0.36 seconds)

[上海Oracle用户组| SHOUG, 走近全系Oracle技术和数据库专家](#)
www.shoug.info/ Translate this page
SHOUG的全称是ShangHai Oracle Users Group, 中文为上海Oracle用户组。 SHOUG的成员仅仅局限于上海地区吗? 上海是国际化大都市, 我们将以上海为中心, ...
You visited this page on 5/20/13.

[Oracle 12c新特性-ORACLE数据库数据恢复、性能优化、故障诊断来...](#)
www.askmaclean.com/archives/.../oracle/oracle-12c Translate this page
Feb 26, 2013 - 《Oracle 12c新特性》-作者: Maclean Liu, 首发于Ask Maclean中文Oracle博客. ... 手机: 13764045638, ORA-ALLSTARS Exadata用户组QQ群:23549328 ... Database 12c进入release发布的倒计时, 可能在今年7月在上海举行 ...
You've visited this page 4 times. Last visit: 4/25/13

BASIC COMPRESS 说明	4
BASIC COMPRESS	4
BASIC COMPRESS 的影响	6
OLTP COMPRESS 说明	9
OLTP COMPRESS 说明	9
和 BASIC COMPRESS 区别说明	9
OLTP COMPRESS 的影响	9
OLTP COMPRESS 转换	10
新表开启 OLTP COMPRESS	10
已存在表开启 OLTP COMPRESS	10
OLTP COMPRESS 最佳实践	11
需要考虑的问题	11
测试过程	12
关于 Oracle Advanced Compression Advisor	20

BASIC COMPRESS 说明

basic compress 是 oracle 的基本组件，但是 oracle 又在 “advanced compression” 特别增加了 oltp compress 的功能，这 2 者之间有什么区别呢？oltp compress 又有些什么新的特点呢？首先要从 basic compress 来看 oracle compress 的特点

BASIC COMPRESS

compress 和普通存储数据方式有什么不同，最好的方式就是创建测试数据来看，并需要解答以下几个问题

- 1.如何触发 compress
- 2.对数据有何影响，包括如何 decompress
- 3.对性能有何影响

这里使用 dba_objects 作为数据源

测试表介绍

普通表基准 t1

```
SQL> create table t1
2 as
3 select * from dba_objects where rownum <= 50000;
```

测试表 t2，启用 basic compress

```
create table t2 compress basic
as
select * from dba_objects where rownum <= 50000;
```

测试表 t3.创建空表后插入数据

```
create table t3 compress basic
as
select * from dba_objects where rownum = 0;
insert into t3 select * from dba_objects where rownum <= 50000;
```

测试表 t4，产用 direct path insert 方式插入数据

```
create table t4 compress basic
as
```

```
select * from dba_objects where rownum = 0;

insert /*+ append */ into t4 select * from dba_objects where rownum <= 50000;
```

测试表 5 使用普通数据然后转换成 compress 表

```
create table t5
as
select * from dba_objects where rownum <= 50000;

alter table t5 compress basic;

alter table t5 move;
```

测试表分析结果

```
SQL> select table_name, blocks, pct_free, compression, compress_for
2 from dba_tables
3 where table_name in ('T1', 'T2', 'T3', 'T4', 'T5') and owner='HR';
```

TABLE_NAME	BLOCKS	PCT_FREE	COMPRESS	COMPRESS_FOR
T5	200	0	ENABLED	BASIC
T4	200	0	ENABLED	BASIC
T3	748	0	ENABLED	BASIC
T2	200	0	ENABLED	BASIC
T1	731	10	DISABLED	

通过测试可以发现，基表 T1 需要占用 731 blocks。T2 创建表时直接启用 basic compress 触发了 compress，占用 200blocks。T3 表虽然启用了 basic compress 但是 insert 还是不会触发 compress，占用了 748blocks。T4 通过 direct path insert 插入触发了 compress，占用 200blocks。T5 表根据文档描述经过了 move 操作后也触发了压缩占用 200blocks。

通过 pct free 值也可以发现，一旦表启用了 basic compress，oracle 会自动将 pct_free 设置为 0，这将会对 DML 操作带来影响。

通过通过 2 次不同的 insert 发现，尽管表已经启用了 basic compress 但是 compress 也是只有在 direct path insert 时才会触发。

并且根据文档描述，要将表转换成 basic compress 表需要对表重新定义，需要对表 move 操作。

BASIC COMPRESS 的影响

oracle 对 compress 的算法采用去除重复值的方法来进行压缩，每个重复值用一个符号（token）来代替，oracle 维护一个 symbol table 的结构，中间记录了每个 token 的值。

对 delete 操作对空间的影响

分别对 t1 和 t2 表执行 delete 操作

```
SQL> delete t1 where owner='SYS';
```

```
27501 rows deleted.
```

```
SQL> commit;
```

```
Commit complete.
```

```
SQL> delete t2 where owner='SYS';
```

```
27501 rows deleted.
```

```
SQL> commit;
```

```
Commit complete.
```

TABLE_NAME	BLOCKS	PCT_FREE	COMPRESS
COMPRESS_FOR			

T5	200	0	ENABLED	BASIC
T4	200	0	ENABLED	BASIC
T3	748	0	ENABLED	BASIC
T2	200	0	ENABLED	BASIC
T1	731	10	DISABLED	

通过 delete 操作发现，delete 和普通 DML 语句相同，并不会对 blocks 数量产生影响。

对 update 操作对空间的影响

由于 pct_free 的存在，需要测试 update 对压缩过后的表的的空间的影响。

```
SQL> update t1 set owner='HQ' where owner='SYS';

27501 rows updated.

SQL> commit;

Commit complete.

SQL> update t5 set owner='HQ' where owner='SYS';

27501 rows updated.

SQL> commit;

Commit complete.

SQL> select table_name,blocks, pct_free , compression, compress_for
2 from dba_tables
3 where table_name in ('T1','T2','T3','T4','T5') and owner='HR';

TABLE_NAME          BLOCKS  PCT_FREE COMPRESS
COMPRESS_FOR
```

T5	634	0	ENABLED	BASIC
T4	200	0	ENABLED	BASIC
T3	748	0	ENABLED	BASIC
T2	200	0	ENABLED	BASIC
T1	731	10	DISABLED	

通过测试可以发现，T5 表经过 update 后空间有大量增加，blocks 数量有大量增加，通过文档可以了解，oracle 对 basic compress 后的表 update 会首先将被修改的行的 token 展开（可以理解为 decompress）并且复制到另外一个 block 中，修改完成后也不会再次 compress 相关的数据，造成了 update 后 blocks 数量大量增加。

OLTP COMPRESS 说明

OLTP COMPRESS 说明

从 oracle 11g 开始引入了 Advanced Compression 选项，其中针对 table 的有 OLTP TABLE COMPRESS，使得 table 在压缩后更好的支持 DML 操作。

和 BASIC COMPRESS 区别说明

通过 advanced compress white paper 描述。OLTP Table Compression 的好处不仅仅是存储空间的节省，另一个重要的影响 Oracle 直接读压缩数据块的能力，因为不需要读 uncompress 的 block，在重复值越多的情况下，读 compress 会降低 I/O，从而提高性能，

压缩率的高低取决于重复数据的多少。

OLTP COMPRESS 的影响

根据之前的描述特点，OLTP COMPRESS 不会对 read 造成影响，但是由于 write 操作需要额外的步骤，所以对 write 操作会有一定影响，不过 Oracle 批处理的 compress 要优于每次写操作时进行压缩。当一个 block 初始化时会保持 uncompress 状态，直到数据接近 block 控制阈值 (PCTFREE)，当某个事务导致数据达到这个 threshold，block 里的所有数据都会被 compressed。随后，又更多的数据被添加进来，再次接近阈值，再次被压缩，直到整个 block 达到最高的 compression。所以批量的写操作性能受影响较小。

OLTP COMPRESS 转换

本部分主要列出了如何将普通表转换为 OLTP COMPRESS

新表开启 OLTP COMPRESS

新表开启 OLTP COMPRESS

新建的表如果需要开启 OLTP COMPRESS 只需要在建表语句后增加“COMPRESS FOR OLTP”即可

已存在表开启 OLTP COMPRESS

alter table 方式

使用 alter table COMPRESS FOR OLTP 方式打开 OLTP COMPRESS, 此种方式会使得后续数据

在线重定义

如果使用在线重定义 **DBMS_REDEFINITION** 可以不仅为将来数据而且可以为现有数据压缩。

使用在线重定义可以在操作时保持表始终可以访问交易并且可以设置并行来提高性能

注意：当使用在线重定义后，分区表的 global index 需要重新 rebuild

alter table move compress for oltp

此种方式可以不仅为将来数据而且可以为现有数据压缩。

但是此种方式会造成表只可读但是 DML 语句会被 block。

注意：此种方式完成后需要对所有相关表进行 rebuild 操作。

OLTP COMPRESS 最佳实践

需要考虑的问题

详细的测试

在考虑使用 OLTP COMPRESS 之前需要详尽的测试，需要尽可能的复制生产环境

这样可以提供最真实性能数据和对生产环境的模拟。

压缩比

数据重复率越高，压缩比越高，更高的压缩率意味着更少的空间占用，更快的备份和恢复速度，

测试时需要针对不同的数据重复率来测试不同的压缩率，

CPU 使用率

使用压缩总会使得 CPU 的开销增加。

需要测试 CPU 在不同的压缩情况下，不同的操作情况下增加的幅度，例如在 DML 情况下 CPU 使用率的情况。

Oracle Advanced Compression Advisor

可以使用 Oracle Advanced Compression Advisor 来大致计算可以通过 OLTP COMPRESS 节约多少空间。

限制对象

OLTP COMPRESS 无法针对列数量超过 255 的表做压缩，也同时不支持 LONG 类型

LOB 的压缩

LOB 建议使用 securefile 作为存储对象，可以使用 securefile 的压缩特性

测试过程

准备测试用表

```
SQL> drop table test purge;

Table dropped.

SQL> create table test as select * from dba_objects;

Table created.

SQL> drop table test_oltp purge;

drop table test_oltp purge
          *
ERROR at line 1:
ORA-00942: table or view does not exist

SQL> create table test_oltp compress for oltp as select * from
dba_objects;

Table created.
```

至此有 2 张表 test 和 test_oltp 其中 test 为普通表，test_oltp 为 oltp compress 压缩表

```
SQL>select
owner,table_name,blocks,num_rows,COMPRESSION,COMPRESS_FOR from
dba_tables where owner='HR' and table_name in
('TEST','TEST_OLTP')
```

OWNER	TABLE_NAME	BLOCKS	NUM_ROWS	COMPRESS	COMPRESS_FOR
HR	TEST_OLTP	366	76355	ENABLED	OLTP
HR	TEST	1113	76354	DISABLED	

测试全表扫描时间

通过测试可以发现 `oltp compress` 可以显著增加全表扫描性能，更少的 `block` 读取可以提高性能（测试中需要 `flush buffer_cache`）

```
03:09:17 SQL> select count(*) from hr.test;

COUNT(*)
-----
       76354

Elapsed: 00:00:00.41

03:09:22 SQL> select count(*) from hr.test_oltp;

COUNT(*)
-----
       76355

Elapsed: 00:00:00.34
```

测试索引创建时间

经过测试 `oltp_compress` 可以提高 `b-tree` 索引创建速度。

```
03:16:25 SQL> create index ind_test on test(object_id);

Index created.

Elapsed: 00:00:05.66

03:17:51 SQL> create index ind_test_oltp on test_oltp(object_id);

Index created.

Elapsed: 00:00:02.18
```

测试索引读取时间

2 者索引用时相差无几

```
03:23:33 SQL> select count(object_id) from test_oltp;

COUNT (OBJECT_ID)
-----
                76355

Elapsed: 00:00:00.11

03:23:39 SQL> select count(object_id) from test;

COUNT (OBJECT_ID)
-----
                76354

Elapsed: 00:00:00.10
```

测试 DML 语句的时间

去除索引后测试 2 张表的 DML 语句的时间差异

update 语句测试后发现 update compress 后的表会比非 compress 表慢

```
03:31:44 SQL> update hr.test set owner='PUBLIC' where
owner='PUBLIC2';

28027 rows updated.

Commit complete.

Elapsed: 00:00:01.18

03:32:06 SQL> commit;

Commit complete.
```

```
Elapsed: 00:00:00.00

03:32:12 SQL> update hr.test_oltp set owner='PUBLIC' where
owner='PUBLIC2';

28027 rows updated.

Commit complete.

Elapsed: 00:00:04.66

03:32:18 SQL> commit;

Commit complete.

Elapsed: 00:00:00.01

03:32:24 SQL> update hr.test set owner='PUBLIC2' where
owner='PUBLIC';

28027 rows updated.

Commit complete.

Elapsed: 00:00:02.55

03:32:42 SQL> update hr.test_oltp set owner='PUBLIC2' where
owner='PUBLIC';

28027 rows updated.

Commit complete.

Elapsed: 00:00:08.86
```

insert 语句从多次测试来看 compress 后较 compress 前耗时较长

```
03:41:23 SQL> insert into hr.test_oltp select * from dba_objects;

76355 rows created.

Commit complete.
```

```
Elapsed: 00:00:04.98

03:41:34 SQL> insert into hr.test select * from dba_objects;

76355 rows created.

Commit complete.

Elapsed: 00:00:02.38
```

普通表测试转换为 oltp compress

alter table move 方式

```
03:48:21 SQL> alter table test move compress for oltp;

Table altered.

Elapsed: 00:00:04.35
```

但是此过程会 block 其余 session 的对表的 DML 操作

在线重定义方式

创建 pk 约束

```
03:58:54 SQL> alter table hr.test add constraint pk_test primary
key (object_id);

Table altered.

Elapsed: 00:00:00.83
```

在线重定义前检查

```
03:59:04 SQL> BEGIN

03:59:13 2
DBMS_REDEFINITION.CAN_REDEF_TABLE('hr','test',DBMS_REDEFINITION.C
ONS_USE_PK);
```



```
03:59:13 3 END;

03:59:13 4 /

PL/SQL procedure successfully completed.
```

创建中间表

```
03:59:14 SQL> CREATE TABLE "HR"."TEST_OLTP_TEMP"
04:01:33 2      ( "OWNER" VARCHAR2(30),
04:01:33 3          "OBJECT_NAME" VARCHAR2(128),
04:01:33 4          "SUBOBJECT_NAME" VARCHAR2(30),
04:01:33 5          "OBJECT_ID" NUMBER,
04:01:33 6          "DATA_OBJECT_ID" NUMBER,
04:01:33 7          "OBJECT_TYPE" VARCHAR2(19),
04:01:33 8          "CREATED" DATE,
04:01:33 9          "LAST_DDL_TIME" DATE,
04:01:33 10         "TIMESTAMP" VARCHAR2(19),
04:01:33 11         "STATUS" VARCHAR2(7),
04:01:33 12         "TEMPORARY" VARCHAR2(1),
04:01:33 13         "GENERATED" VARCHAR2(1),
04:01:33 14         "SECONDARY" VARCHAR2(1),
04:01:33 15         "NAMESPACE" NUMBER,
04:01:33 16         "EDITION_NAME" VARCHAR2(30)
04:01:33 17     ) SEGMENT CREATION IMMEDIATE
04:01:33 18     PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
04:01:33 19     COMPRESS FOR OLTP LOGGING
04:01:33 20     STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1
04:01:33 21     MAXEXTENTS 2147483645
04:01:33 22     PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
04:01:33 23     BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT
04:01:33 24     CELL_FLASH_CACHE DEFAULT)
04:01:33 25     TABLESPACE "USERS";
```

```
Table created.
```

```
Elapsed: 00:00:00.16
```

开始在线重定义

```
04:01:35 SQL> BEGIN
```

```
04:02:47 2 DBMS_REDEFINITION.START_REDEF_TABLE('hr',  
'TEST','TEST_OLTP_TEMP',
```

```
04:02:47 3 '*',
```

```
04:02:47 4 dbms_redefinition.cons_use_pk);
```

```
04:02:47 5 END;
```

```
04:02:47 6 /
```

```
PL/SQL procedure successfully completed.
```

```
Commit complete.
```

```
Elapsed: 00:00:03.49
```

在线重定义创建关联对象

```
04:02:51 SQL> DECLARE
```

```
04:03:46 2 num_errors PLS_INTEGER;
```

```
04:03:46 3 BEGIN
```

```
04:03:46 4 DBMS_REDEFINITION.COPY_TABLE_DEPENDENTS('hr',  
'TEST','TEST_OLTP_TEMP',
```

```
04:03:46 5 DBMS_REDEFINITION.CONS_ORIG_PARAMS, TRUE, TRUE,  
TRUE, TRUE, num_errors);
```

```
04:03:46 6 END;
```

```
04:03:47 7 /
```

```
PL/SQL procedure successfully completed.
```

```
Commit complete.
```

```
Elapsed: 00:00:09.47
```

查看是否有错误

```
04:03:56 SQL> select object_name, base_table_name, ddl_txt from
DBA_REDEFINITION_ERRORS;

no rows selected

Elapsed: 00:00:00.04
```

再次 sync 重定义表

```
04:04:11 SQL> BEGIN

04:04:44 2 DBMS_REDEFINITION.SYNC_INTERIM_TABLE('hr',
'TEST','TEST_OLTP_TEMP');

04:04:44 3 END;

04:04:44 4 /

PL/SQL procedure successfully completed.

Commit complete.

Elapsed: 00:00:00.10
```

完成在线重定义

```
04:04:45 SQL> BEGIN

04:05:09 2 DBMS_REDEFINITION.FINISH_REDEF_TABLE('hr',
'TEST','TEST_OLTP_TEMP');

04:05:09 3 END;

04:05:09 4 /

PL/SQL procedure successfully completed.

Commit complete.

Elapsed: 00:00:04.91

04:05:15 SQL> select COMPRESSION,compress_for from dba_tables
where table_name='TEST' and owner='HR';

COMPRESS COMPRESS_FOR
```

```

-----
ENABLED  OLTP

Elapsed: 00:00:00.27

```

关于 Oracle Advanced Compression Advisor

使用 advisor

可以使用 Oracle Advanced Compression Advisor 来大致计算可以通过 OLTP COMPRESS 节约多少空间。

11gR2 已经包含了 Advisor 如果其他版本需要至以下链接处下载

<http://www.oracle.com/technetwork/database/options/compression/compression-advisor-095705.html>

11gR2 的 package 为 *DBMS_COMPRESSION*

```

PROCEDURE GET_COMPRESSION_RATIO

```

Argument Name	Type	In/Out	Default?
SCRATCHTBSNAME	VARCHAR2	IN	
OWNNAME	VARCHAR2	IN	
TABNAME	VARCHAR2	IN	
PARTNAME	VARCHAR2	IN	
COMPTYPE	NUMBER	IN	
BLKCNT_CMP	BINARY_INTEGER	OUT	
BLKCNT_UNCMP	BINARY_INTEGER	OUT	
ROW_CMP	BINARY_INTEGER	OUT	
ROW_UNCMP	BINARY_INTEGER	OUT	
CMP_RATIO	NUMBER	OUT	
COMPTYPE_STR	VARCHAR2	OUT	

SUBSET_NUMROWS	NUMBER	IN	DEFAULT
FUNCTION GET_COMPRESSION_TYPE RETURNS NUMBER			
Argument Name	Type	In/Out	Default?

OWNNAME	VARCHAR2	IN	
TABNAME	VARCHAR2	IN	
ROW_ID	ROWID	IN	
PROCEDURE INCREMENTAL_COMPRESS			
Argument Name	Type	In/Out	Default?

OWNNAME	VARCHAR2 (30)	IN	
TABNAME	VARCHAR2 (128)	IN	
PARTNAME	VARCHAR2 (30)	IN	
COLNAME	VARCHAR2	IN	
DUMP_ON	NUMBER	IN	DEFAULT
AUTOCOMPRESS_ON	NUMBER	IN	DEFAULT
WHERE_CLAUSE	VARCHAR2	IN	DEFAULT

测试 advisor

运行 advisor 如下:

```

SQL> set serveroutput on

SQL> declare

  2  v_blkcnt_cmp    pls_integer;

  3  v_blkcnt_uncmp pls_integer;

  4  v_row_cmp      pls_integer;

  5  v_row_uncmp    pls_integer;

  6  v_cmp_ratio    number;

  7  v_comptype_str varchar2(60);

  8  begin

  9  dbms_compression.get_compression_ratio(

```

```

10  scratchtbsname => upper('&ScratchTBS'),
11  ownname        => upper('&ownername'),
12  tablename      => upper('&TableName'),
13  partname       => NULL,
14  comptype       => dbms_compression.COMP_FOR_OLTP,
15  blkcnt_cmp     => v_blkcnt_cmp,
16  blkcnt_uncmp   => v_blkcnt_uncmp,
17  row_cmp        => v_row_cmp,
18  row_uncmp      => v_row_uncmp,
19  cmp_ratio      => v_cmp_ratio,
20  comptype_str   => v_comptype_str);

21  dbms_output.put_line('Estimated Compression Ratio:
'||to_char(v_cmp_ratio));

22  dbms_output.put_line('Blocks used by compressed sample:
'||to_char(v_blkcnt_cmp));

23  dbms_output.put_line('Blocks used by uncompressed sample:
'||to_char(v_blkcnt_uncmp));

24  end;

25  /

Enter value for scratchtbs: EXAMPLE

old 10: scratchtbsname => upper('&ScratchTBS'),
new 10: scratchtbsname => upper('EXAMPLE'),

Enter value for ownername: HR

old 11: ownname        => upper('&ownername'),
new 11: ownname        => upper('HR'),

Enter value for tablename: EMPLOYEES

old 12: tablename      => upper('&TableName'),
new 12: tablename      => upper('EMPLOYEES'),

Estimated Compression Ratio: 2

Blocks used by compressed sample: 1

Blocks used by uncompressed sample: 2

```

作者个人简介



黄乾， Oracle ACS 高级服务顾问黄乾 qian.huang@oracle.com

黄乾 是华东 Oracle ACS 的知名技术专家，拥有近 10 年 Oracle 数据库领域从业经验，目前在甲骨文公司中国上海工作，所服务的行业包括电信运营商、通讯设备制造商、银行、金融业等。

主要客户包括：

工商银行、外汇交易中、上海银行、浙江移动、浙江电信、上海电信、上海移动、江苏移动等。