

A Complete Understanding of RMAN Compression (Doc ID 563427.1)

In this Document

[Goal](#)

[Solution](#)

A Complete Understanding of RMAN Compression

[1. Null Compression](#)

[2. Unused Block](#)

[3. Binary Compression](#)

[4. Various Compression Types:](#)

[5. Undo Block Compression/Optimization](#)

[References](#)

APPLIES TO:

Oracle Database - Enterprise Edition - Version 8.1.7.4 and later

Information in this document applies to any platform.

Checked for relevance on 08-Feb-2016

GOAL

How does RMAN compression works ?

Goal of this document is to give you a Complete Understanding of

1. Null compression
2. Unused block compression
3. Binary compression

SOLUTION

A Complete Understanding of RMAN Compression

By default RMAN has three types of compression :

1. Null compression
2. Unused block compression
3. Binary compression

Till Oracle Version 10.1 only 'Null' compression is done by default but from Oracle Version 10.2 'Null' and 'unused block' compression is done. The types of compression are done automatically, no special command is required.

'Null' and 'unused block' compression are filtering which blocks are send to the backups. 'Binary' compression is an additional compression on the blocks send to the backup.

1. Null Compression

When backing up datafiles into backup sets, RMAN does not back up the contents of data blocks that have never been allocated. In previous releases, this behavior was referred to as NULL compression. This means RMAN will never backup the blocks that have never been used. RMAN, through Oracle version 9i and forward has performed null compression.

Example : You have a tablespace having one datafile of size 100MB and out of 100MB only 50 MB is used. Then RMAN will backup only 50MB.

NOTE: Null Compression also applies to Standard Edition and SE ONE

2. Unused Block

From Oracle version 10.2 forward, RMAN skips the blocks that do not currently contain data and this is called 'Unused Block Compression'. RMAN now creates more compact backups of datafiles, by skipping datafile blocks that are not currently used to store data. Skipping unused data blocks where possible enables RMAN to back up datafiles using less space, and can make I/O more efficient. In previous releases, RMAN only supported NULL compression, which skipped space in datafiles that had never been allocated. No extra action is required on the part of the DBA to use this feature.

Example : You have a tablespace having one datafile of size 100MB and out of 100MB, 50MB is used by the user tables. Then user dropped a table belonging to that tablespace which was of 25MB, with the new unused block compression on 25MB of the files is backed up. In this example if null compression is used then it would have backed up 50MB because null compression will consider the blocks that are formatted/ever used.

Unused Block Compression is done, if all of the following conditions apply:

- + The COMPATIBLE initialization parameter is set to 10.2
- + There are currently no guaranteed restore points defined for the database
- + The datafile is locally managed
- + The datafile is being backed up to a backup set as part of a full backup or a level 0 incremental backup
- + The backup set is being created on DISK
or
backup is done to TAPE using "OSB" (Oracle Secure Backup)!

Note:

Unused block compression is NOT used if backup done to tape using a THIRD PARTY BACKUP SOFTWARE !
Unused block compression is ONLY used in Enterprise Edition.

3. Binary Compression

Binary Compression can be done by specifying "**AS COMPRESSED**" clause in backup command, this compression is called as binary compression.

RMAN can apply a binary compression algorithm as it writes data to backup sets. This compression is similar to the compression provided by many tape vendors when backing up data to tape. But we cannot give exact percentage of compression. This binary compression algorithm can greatly reduce the space required for disk backup storage. It is typically 2x to 4x, and greater for text-intensive databases.

The command to take the compressed backup :

```
RMAN> backup as compressed backupset database;
```

- + There is some CPU overhead associated with compressing backup sets. If the database being backed up is running at or near its maximum load, you may find the overhead from using AS COMPRESSED BACKUPSET unacceptable. In most other circumstances, compressing backupsets saves enough disk space to be worth the CPU overhead.
- + There is no special command to restore database from the compressed backupsets, the restore command will be the same as with uncompressed backups.
- + The restore from the compressed backupset will take more time than uncompressed backupsets.

In addition to the existing binary compression of backup in oracle 10G, RMAN 11G executable offers a wider range of compression levels with the Advanced Compression Option (ACO). The default compression algorithm setting is BASIC and does not require the Advanced Compression Option.

If, however, you have enabled the Oracle Database 11g Release 2 Advanced Compression Option, you can choose from the following compression levels:

LOW - Least impact on backup throughput and suited for environments where CPU resources are the limiting factor.

MEDIUM - Recommended for most environments. Good combination of compression ratios and speed

HIGH - Best suited for backups over slower networks where the limiting factor is network speed

NOTE: Only BASIC compression is allowed in Standard Edition

Compression can be used for backupset of datafile, archive log and controlfiles. For example:

```

RMAN> backup as compressed backupset archivelog all;
RMAN> backup as compressed backupset database;
RMAN> backup as compressed backupset current controlfile;

```

RMAN compress the backupset contents before writing to disk. No extra decompression steps are required during recovery for rman compressed backup;

To configure the compression algorithm:

```

RMAN> CONFIGURE COMPRESSION ALGORITHM '<alg_name>';

```

4. Various Compression Types:

For various compression types you can refer to V\$RMAN_COMPRESSION_ALGORITHM

```

SQL> select ALGORITHM_NAME, ALGORITHM_DESCRIPTION, ALGORITHM_COMPATIBILITY from
V$RMAN_COMPRESSION_ALGORITHM ;

```

ALGORITHM_NAME	ALGORITHM_DESCRIPTION	ALGORITHM_COMPATIB
BZIP2	good compression ratio	9.2.0.0.0
BASIC	good compression ratio	9.2.0.0.0
LOW	maximum possible compression speed	11.2.0.0.0
ZLIB	balance between speed and compression ratio	11.0.0.0.0
MEDIUM	balance between speed and compression ratio	11.0.0.0.0
HIGH	maximum possible compression ratio	11.2.0.0.0

6 rows selected.

5. Undo Block Compression/Optimization

Starting 11g, RMAN performs undo block optimization. In backup undo optimization, RMAN excludes undo not needed for recovery of a backup, that is, for transactions that have been committed. Undo optimization is only possible if:

- This is a backup set backup
- Full or incremental level 0
- Not a validate
- Backup piece version is 11.0 or above
- User has not disabled undo optimization with:
_undo_block_compression = FALSE
- Backup is going to DISK or OSB tape
- No Guaranteed Restore Point (This check is enabled from 11.2 onwards)

REFERENCES

Didn't find what you are looking for?