BRSPACE Online Reorganizations
Advanced Reorganizations

Martin Frauendorfer, SAP Active Global Support
martin.frauendorfer@sap.com
April 2009
1. Overview
2. Advanced Reorganizations
Overview (1)

Reorganizations in general

- A reorganization in a database context is the recreation or cleanup of one or several segments.
- Several types of reorganization exist:
  - Index reorganization (e.g. via REBUILD or COALESCE), mainly used to reduce index fragmentation
  - Table reorganization with tools like:
    - BRSPACE (online, offline, export / import)
    - Oracle tools (DBMS_REDEFINITION, MOVE, EXP / IMP, EXPDP / IMPDP, …)
    - SAP functionality (ICNV, SE14, …)
  - Tablespace reorganization (reorganization of all tables in the tablespace and recreation of the tablespace)
- In this presentation we concentrate on the BRSPACE online reorganization of tables (which also includes tablespace reorganizations).
Overview (2)

In which situations can a table reorganization be useful?

- Reduction of table size
- Cleanup of table fragmentation
- Change of table storage parameters (e.g. INITRANS)
- Move of table to a different tablespace
- Reduction of chained and migrated rows
- Reduction of number of allocated extents
- Transition from LONG to LOB columns (Oracle >= 10g)
- Reduction of hot spots on disk level
- Activation of Transparent Data Encryption
- Change of table structure
Overview (3)

In which situations can a **tablespace** reorganization be useful?

- Activation of LMTS
- Activation of ASSM
- Cleanup of freespace block corruption
- Reduction of database size
- Reduction of backup size
- Reduction of filesystem size
- Reduction of fragmentation
- Reduction of number of datafiles
- Increase of number of datafiles
- Transition to new tablespace layout
- Other change to tablespace layout
- Optimization of filesystem layout
Overview (4)

How does a BRSPACE online reorganization work?

- Determination of CREATE commands for target table and indexes based on DBMS_METADATA.GETDDL
- Creation of target table with naming convention <source_table>#$
- Export of source table CBO statistics based on DBMS_STATS.EXPORT_TABLE_STATS
- Copy of source table data to target table based on DBMS_REDEFINITION.START_REDEF_TABLE. This package will implicitly create a materialized view log MLOG$_<source_table> to track all changes that are performed on the source table while the reorganization is running.
- Creation of target indexes with naming convention <source_index>#$
- Import of source table CBO statistics to target table based on DBMS_STATS.IMPORT_TABLE_STATS
- Finalizing of the online reorganization using DBMS_REDEFINITION.FINISH_REDEF_TABLE. In this step all changes in the materialized log are applied to the target table and the names of source and target table are exchanged.
- The source table is dropped.
- The „#$“ suffix of the target table indexes is removed.
What are the advantages of a BRSPACE Online Reorganization?

- It uses the advantages of DBMS_REDEFINITION:
  - The reorganization can be performed in parallel to production operation. This means the table is accessible and records can be inserted / updated / deleted.
  - If anything goes wrong during the reorganization the source table still exists unchanged and no restore / recovery is necessary.
  - Compared to an offline reorganization via export / import all data has to be copied only once and not twice.

- It provides possibilities for client and server side parallelism in order to speed up the reorganization if sufficient system resources exist and the reorganization should finish within a specific time window.

- It has already proven to be a reliable reorganization tool at many customers.

- It performs a lot of useful activities implicitly that would require a lot of manual efforts and impose the risk of errors if DBMS_REDEFINITION is used directly.
What are the restrictions of a BRSPACE Online Reorganization?

- It is only available as of Oracle 9i.
- It cannot handle tables with LONG or LONG RAW columns. As of 10g it is possible to convert them during an online reorganization on the fly to LOB columns (if SAP >= 6.40 is used).
- Temporarily twice the space is needed because both source and target table and indexes exist in parallel.
- If a table has no primary key constraint (or no unique index with NOT NULL columns) a ROWID based online reorganization is necessary that requires overhead (ROWID index, ROWID column).
- The setup of the materialized view is an overhead that can negatively impact the performance of the reorganization of many very small tables.
- Structure changes on the source table and its indexes are not possible during the reorganization.
How is a normal BRSPACE Online Reorganization started?

- **Table reorganization:**
  
  \[
  \text{brspace} \ -f \ \text{tbreorg} \ -t \ <\text{table\_name}>
  \]

- **Tablespace reorganization:**
  
  \[
  \text{brspace} \ -f \ \text{tbreorg} \ -s \ <\text{source\_tsp}> \ -t \ "*" \ -n \ <\text{target\_tsp}>
  \]

- **Additional useful options exist, e.g.:**
  
  \[
  \text{-p} \ <\text{degree}>
  \]
  
  † BRSPACE parallelism degree

  \[
  \text{-e} \ <\text{degree}>
  \]
  
  † PX parallelism degree

  \[
  \text{-l} \ <\text{category}>
  \]
  
  † Initial extent size category

  \[
  \text{-i} \ <\text{target\_ind\_tsp}>
  \]
  
  † Definition of target index tablespace (if different from target table tablespace)

- **For more detailed information see note 646681 and the BRSPACE online documentation.**
Advanced Reorganization Scenarios Overview

Example for advanced reorganization scenarios:

- **Optimization of data clustering**
  - Changing the sort order of the table records
  - Transition to an Index Organized Table (IOT)

- **Partitioning**
  - Activation of partitioning for a formerly unpartitioned table
  - Repartitioning of an already partitioned table

- **Table structure changes**
  - Additional columns
  - Modified columns (e.g. datatype and default value change)
  - Removed columns
Scenario 1: Optimization of Data Clustering (1)

Motivation

- In many cases good table data clustering in central tables is a key factor for good performance.
- This factor is often ignored and instead optimizations are only performed on index, CBO or ABAP side.
- Typical candidates for a better clustering:
  - RESB: Sorting by MATNR
  - DFKKOP: Sorting by GPART / VKONT / VTREF
  - VDBEKI / VDBEPI: Sorting by RANL
- If e.g. 100 records for one DFKKOP business partner (GPART) exist and they are scattered across the whole table, in the worst case 100 table blocks have to be read to retrieve the records.
- If the table is sorted by GPART and there are e.g. 10 records in each block, only about 10 blocks need to be read.
- This can easily improve the total access performance by factor 5.
- With BRSPACE the sorting can be done online.
Warning

n Only do this if you understand clearly how the critical and typical table accesses look like!

n With the wrong data clustering you can cause critical performance problems!

n It is a good idea to monitor the access predicates and the shared cursor cache data of the different table accesses for a longer time before deciding to implement a different data clustering.
Scenario 1: Optimization of Data Clustering (2)

Approach 1: Sorting table records by an index

- Directly supported by BRSPACE reorganization option "-sortind <index_name>"

- Advantages:
  - No table structure change

- Disadvantages:
  - Only a temporary solution because future DML operations can reduce the clustering more and more. So it may be necessary to perform the sorting on a regular basis.
Scenario 1: Optimization of Data Clustering (3)

Approach 2: Transition to an Index Organized Table (IOT)

- No automatic support by BRSPACE, manual adaptations needed
- Steps:
  - Start the BRSPACE online reorganization with the additional option "-d first".
  - BRSPACE will stop after it had created the CREATE commands for the target table and indexes in $SAPDATA_HOME/sapreorg/<reorg_subdir>/ddl.sql
  - Now the CREATE TABLE command can be adapted manually:
    - Remove ") PCTFREE"
    - Add instead ", CONSTRAINT "<table_name>~<iotindex_suffix>" PRIMARY KEY (<iotindex_columns>) ORGANIZATION INDEX"
    - Remove "PCTUSED <percent>"
    - Remove CREATE command for secondary index if used as IOT primary key index.
  - Afterwards the BRSPACE reorganization can be continued with "c"
Scenario 1: Optimization of Data Clustering (3)

Approach 2: Transition to an Index Organized Table (IOT)

- **Advantages:**
  - Permanently guaranteed sort order
  - Table and IOT index combined in one segment

- **Disadvantages:**
  - Errors due to manual adaptations possible
  - Typical IOT disadvantages:
    - Larger secondary indexes
    - Overhead of secondary index accesses
    - Additional maintenance tasks like logical ROWID recalculation
    - Table fragmentation can increase because of its index design
    - Primary key constraint required
    - Only supported by ABAP DDIC if IOT is based on SAP primary index

† Mainly useful for IOTs on primary index on tables with a small number of secondary indexes
Motivation

- **Partitioning can be useful for several reasons, e.g.:**
  - **Optimization of data clustering:** Similar like the previous approaches also partitioning can be used to optimize data clustering
    - Partitioning based on clustering column
  - **Archiving / deletion support:** Partitioning can also be used to make sure that the table can be easily cleaned after archiving or deletion activities using DROP PARTITION or MERGE PARTITION
    - Partitioning based on archiving criteria
  - **Administration support:** Large segments that are difficult to administer (e.g. consistency check, statistics creation, reorganization) can be split into smaller partitions that can be administered individually
    - E.g. hash partitioning based on any column
Scenario 2: Partitioning (2)

Procedure

- No automatic support by BRSPACE, manual adaptations needed
- Steps as described for the IOT approach
- PARTITION clauses have to be added to the CREATE TABLE and CREATE INDEX commands as required

Advantages:
- Permanently guaranteed clustering and reduced segment sizes

Disadvantages:
- Errors due to manual adaptations possible
- Often continuous partitioning maintenance necessary
Scenario 3: Structure Changes (1)

Motivation

- Structure changes like added or modified columns often cause trouble on large tables:
  - Long conversion times because each record has to be modified individually
  - Sometimes even longer conversion times because of ASSM problems
  - Danger of significant amount of chained and migrated rows if blocks don't contain sufficient freespace for additional data
  - SAP standard tools like SE11 / SE14 / ICNV don't provide a possibility for parallelization

- A conversion based on a BRSPACE online reorganization provides the following advantages:
  - Shorter runtime due to parallelism possible
  - No danger of chained and migrated rows as target table is created from scratch
  - Conversion can be done fully online
Scenario 3: Structure Changes (2)

Procedure

- No automatic support by BRSPACE, manual adaptations needed
- Utilization of the undocumented BRSPACE column mapping feature
- Attention: Errors in the manual procedure can result in a wrong table structure and a restore may be required!
- Functionality available as of BRSPACE 7.00 (40) and 7.10 (20)
- BRSPACE profile with parameter _reorg_col_map for column mapping information has to be created
- BRSPACE has to be started with this profile ("-p <profile>"), the "-d first" option and the "-UCM" switch in order to activate the column mapping feature.
- When BRSPACE stops, the CREATE TABLE command has to be adapted as required (additional columns, removed columns, changed columns).
- Before doing the final switch BRSPACE will stop a second time. This allows you to control the time of the switch (e.g. during upgrade downtime).
Scenario 3: Structure Changes (2)

Procedure

- No automatic support by BRSPACE, manual adaptations needed
- Utilization of the undocumented BRSPACE column mapping feature
- Attention: Errors in the manual procedure can result in a wrong table structure and a restore may be required!
- Functionality available as of BRSPACE 7.00 (40) and 7.10 (20)
- BRSPACE profile with parameter `_reorg_col_map` for column mapping information has to be created
- BRSPACE has to be started with this profile (`"-p <profile>"`), the `-d first` option and the `-UCM` switch in order to activate the column mapping feature.
- When BRSPACE stops, the CREATE TABLE command has to be adapted as required (additional columns, removed columns, changed columns).
- Before doing the final switch BRSPACE will stop a second time. This allows you to control the time of the switch (e.g. during upgrade downtime).
Scenario 3: Structure Changes (3)

Case Study

n During a SAP upgrade the following changes are necessary for table DBERCHZ4:
   n Add column GGVERTRAG (VARCHAR2(18), DEFAULT ' ')
   n Add column OLD_BELNR (VARCHAR2(12), DEFAULT ' ')
   n Increase length of column LAUFNR from 2 to 5 characters, fill up with '0' on left side

n The incremental conversion (ICNV) of DBERCHZ4 took days and weeks because the table contained 4 billion records.

n If it is possible to use BRSPACE we could do the conversion also online and we could use advanced possibilities like parallelization.

n Therefore it was decided to use BRSPACE for the conversion.
Scenario 3: Structure Changes (4)

Case Study (2) - Procedure

- Implementation of BRSPACE 7.00 (40)
- Definition of column mapping parameter in BRSPACE profile:

  `_reorg_col_map = "
  MANDT                      MANDT,
  BELNR                      BELNR,
  ...                        ...
  CONCAT('000', LAUFNR)      LAUFNR,
  ...                        ...
  "

- Start of BRSPACE:

  `brspace -p <profile> -f tbreorg -t DBERCHZ4 -e 12 -d first -n <target_tablespace> -l 2 -UCM`

  -e 12: Parallel execution with degree 12
  -d first: Stop after DDL commands are generated
  -l 2: Define initial extent with 64K
  -UCM: Use column mapping parameter (_reorg_col_map)```
Scenario 3: Structure Changes (5)

Case Study (3) - Procedure

- Wait until BRSPACE stops and modify the generated DDL commands in ddl.sql:
  - Add: "GGVERTRAG" VARCHAR2(18) DEFAULT ' ' NOT NULL ENABLE,
  - Add: "OLD_BELNR" VARCHAR2(12) DEFAULT ' ' NOT NULL ENABLE,
  - Change LAUFNR:
    - Old: "LAUFNR" VARCHAR2 (2) DEFAULT '00' NOT NULL ENABLE,
    - New: "LAUFNR" VARCHAR2 (5) DEFAULT '00000' NOT NULL ENABLE,

- Continue BRSPACE run

- With the ",-UCM" option BRSPACE stops before the final switch happens

- During the upgrade downtime BRSPACE was continued and the structure switch was executed.

- The total conversion runtime was 11:05 hours.
Thank you!