



Platzersparnis in einer 10-TB-Datenbank durch Reorganisation und Einsatz der Advanced Compression Option

Susanne Jahr

DOAG-Konferenz Nürnberg 19.11.2014

Herrmann & Lenz Services GmbH

Herrmann & Lenz Solutions GmbH



- Erfolgreich seit 1996 am Markt
- Firmensitz: Burscheid (bei Leverkusen)
- Beratung, Schulung und Betrieb/Fernwartung rund um das Thema Oracle Datenbanken
- Schwerpunktthemen: Hochverfügbarkeit, Tuning, Migrationen und Troubleshooting / Monitoring
- Herrmann & Lenz Solutions GmbH
 - Produkte: Monitoring Module, Taskzone
 - Stand auf Ebene 2



Übersicht

- Problemstellung / Zielsetzung
- Konzept
- Umsetzung
- Fazit



Problemstellung / Zielsetzung

Ausgangssituation

- Auftragseingangs-Datenbank
- Betrieb im 3-Knoten-RAC mit ASM-Storage
- Physical Standby mit Data Guard
- Größe ca. 10 TB
- Großteil der Gesamt-Datenmenge: LOBs



Ziele

Platz- und Kostenersparnis

Reorganisation

Auslagerung
älter Daten

LOB-Segmente

Tabellen- und
Index-Segmente

Freigabe nicht
mehr benötigter
Platten



Konzept

Information Life Cycle Management

- Aufteilung der Daten in
 1. Aktuell
 2. Älter
 3. Archiv
- Eigene ASM-Diskgruppe für den Archiv-Speicher
- Storage-Ebenen "High Performance" und "Archive"
- "Aktuelle" Daten in "HP", Rest in "Archive"



Platzersparnis

- Komprimierung der LOB- und Datensegmente mit der ACO
- Gleichzeitige Reorganisation in die neuen Storage-Ebenen
- Teilweise In-Place-Reorganisation
- Sonderfall: "Altlasten-Tabelle"





Durchführung

Storage-Ebene Archive

- Tablespace für Archiv-Daten in der neuen ASM-Diskgruppe

```
CREATE TABLESPACE XXX_ARCHDATA DATAFILE  
SIZE 10G AUTOEXTEND ON NEXT 1G MAXSIZE  
1400G  
BLOCKSIZE 8K  
DEFAULT STORAGE (INITIAL 8M)
```



Fokus: LOBs

- LOBs: Großteil des Gesamt-Datenvolumens

TABLESPACE_NAME	FILE_MB	
XXX_CAD	7.170.000	→ LOB-Tablespace
XXX_DAT	598.720	
XXX_IDX	581.632	
XXX_PDAT	565.248	
XXX_PIDX	1.207.000	
Summe:	10.122.600	



LOB-Komprimierung (1)

- Plan:
 - Partitionsweise Komprimierung der LOB-Tabellen
 - Gleichzeitig Reorganisation in Tablespace XXX_ARCHDATA
- Kleines Problem:
 - LOBs liegen als BASIC-LOBs vor, die ACO erfordert jedoch SecureFiles
 - Nicht die gesamte Partition kann im neuen Tablespace gespeichert werden (abweichende Blockgröße: DB 16k, LOB-Tablespace 8k)



LOB-Komprimierung (2)

- Also:
 - Durchführung partitionsweise mit **dbms_redefinition**
 - in einem Schritt Umwandlung in SecureFiles
 - LOBs werden verschoben, Rest in-Place reorganisiert



dbms_redefinition (1)

- can_redef: Ist eine Partition / Tabelle bereit zur Redefinition?

```
EXEC dbms_redefinition.can_redef_table  
  (uname => 'PRODUKTIV',  
   tname => 'T299_PRINTOUT',  
   part_name => '&PART')
```



dbms_redefinition (2)

- Anlage Zwischentabelle

```
CREATE TABLE "PRODUKTIV"."I299_PRINTOUT"  
  ( "T299_PRINTOUT_ID" NUMBER(10,0) NOT NULL ENABLE,  
    ...  
  )  
  TABLESPACE XXX_PDAT STORAGE (INITIAL 8M)  
  LOB ("T299_PLOT") STORE AS SECUREFILE "T299_PLOT_SFLOB"  
    (TABLESPACE XXX_ARCHDATA STORAGE (INITIAL 8M) ENABLE  
    STORAGE IN ROW COMPRESS LOW RETENTION AUTO CACHE READS)  
/
```



dbms_redefinition (3)

- Versuch, die gesamte Partition in den XXX_ARCHDATA-Tablespace zu verschieben:

ORA-42012: error occurred while
completing the redefinition

ORA-14296: Table block size mismatch
in ALTER TABLE EXCHANGE
[SUB]PARTITION



dbms_redefinition (4)

- Redefinition der gewünschten Partition

```
EXEC dbms_redefinition.start_redef_table  
  (uname => 'PRODUKTIV',  
   orig_table => 'T299_PRINTOUT',  
   int_table => 'I299_PRINTOUT',  
   part_name => '&PART')
```

```
EXEC dbms_redefinition.finish_redef_table  
  (uname => 'PRODUKTIV',  
   orig_table => 'T299_PRINTOUT',  
   int_table => 'I299_PRINTOUT',  
   part_name => '&PART')
```



OLTP-Komprimierung

- Komprimierung großer Segmente ohne LOBs
- Anschließender ggfs. Rebuild der "UNUSABLE" Indizes

```
ALTER TABLE t038_option MOVE PARTITION P0001 COMPRESS FOR OLTP STORAGE  
(INITIAL 8M NEXT 8M);
```

```
BEGIN
```

```
  FOR r IN (SELECT index_name, ip.partition_name FROM user_indexes i join  
            user_ind_partitions ip using (index_name) WHERE i.table_name =  
            'T038_OPTION' and ip.status = 'UNUSABLE')
```

```
  LOOP
```

```
    EXECUTE IMMEDIATE 'ALTER INDEX "' || r.index_name || '" REBUILD  
PARTITION "' || r.partition_name || '" STORAGE (INITIAL 8M NEXT 8M)';
```

```
  END LOOP;
```

```
END;
```

```
/
```



Spezialfall: "Altlasten-Tabelle"

- Große, nicht partitionierte Tabelle
- Historische Daten über alle Aufträge inkl. Dokumente (meist pdf)
- Bisher keinerlei organisierte Bereinigung, daher Wachstum von ca. 1 GB (2007) auf ca. 856 GB (2014)

Bereinigung "Altlasten-Tabelle"

- Nach Absprache mit Fachabteilung Löschung nicht mehr benötigter Datensätze
- Einsatz von **dbms_parallel_execute**
 - Definition von Datenbereichen anhand der Row-IDs
 - Parallele Ausführung des gewünschten Statements per Job Scheduler
- Anschließende Migration des LOB-Segments mit **dbms_redefiniition**

dbms_parallel_execute (1)

```
DECLARE
del_t038 VARCHAR2(30) := 'del_t038';
l_sql_stmt VARCHAR2(32767);
l_try NUMBER;
l_status NUMBER;

BEGIN
-- Create the TASK
DBMS_PARALLEL_EXECUTE.create_task (task_name =>
    del_t038);
```



dbms_parallel_execute (2)

```
-- Chunk the table by the ROWID
DBMS_PARALLEL_EXECUTE.create_chunks_by_rowid(task_name =>
    del_t038,
    table_owner => 'PRODUKTIV',
    table_name => 'T038_OPTION',
    by_row => TRUE,
    chunk_size => 100000);

-- DML to be execute in parallel
l_sql_stmt := 'DELETE FROM produktiv.t038_option WHERE ...
    AND rowid BETWEEN :start_id AND :end_id';
```



dbms_parallel_execute (3)

```
-- Run the task
DBMS_PARALLEL_EXECUTE.run_task(task_
    name => del_t038,
    sql_stmt => l_sql_stmt,
    language_flag => DBMS_SQL.NATIVE,
    parallel_level => 10);
END;
```



Reorganisation "Altlasten-Tabelle"

- Ganze Tabelle wird reorganisiert (keine Partitionen)
- Viel DML zwischen **start_redef** und **finish_redef**
- Daher zwischendurch Synchronisation der Original- mit der Interim-Tabelle

```
EXEC dbms_redefinition.sync_interim_table  
('PRODUKTIV', 'T469_IMP_DOC_BIN', 'I469');
```



Migration "Altlasten-Tabelle"

- Vor dem **finish redefinition**: Kopie der abhängigen Objekte

```
DECLARE
    v_ne PLS_INTEGER;
BEGIN
    dbms_redefinition.copy_table_dependents('PRODUKTIV',
    'T469_IMP_DOC_BIN', 'I469', num_errors => v_ne,
    ignore_errors => TRUE);
    dbms_output.put_line('Errors: ' || v_ne);
END;
/
```



Fazit (bis jetzt...)

Ergebnis

- Reduzierung des in der Datenbank benötigten Platzes um ca. 3,8 TB
- Verschiebung in Archive-Storage ca. 2,7 TB
LOB-Segmente
- To be continued ...

Fragen & Kontakt

- Mail: susanne.jahr@hl-services.de
- Web: www.hl-services.de
- Blog: blog.hl-services.de
- Twitter: @soesa69
- Live: Hier in der Ausstellung Stand 212
Ebene 2 (gelb) direkt gegenüber der Rolltreppe

