

Turn on the Turbo-Boost! (Speed up your LOB Migration)

Susanne Jahr

DOAG-Conference Nürnberg 16.11.2016



Herrmann & Lenz Services GmbH

Herrmann & Lenz Solutions GmbH

- Successful in the market since 1996
- Head Office: Burscheid (near Leverkusen)
- Consulting, training and operation/remote maintenance on the topic of the Oracle Database
- Focus: High availability, tuning, migrations and troubleshooting / monitoring
- Herrmann & Lenz Solutions GmbH
 - Products: Monitoring Module, Taskzone
 - Booth on level 2 (236)



Agenda

- Project
- Challenge: LOB migration
- Idea: HL-Tablepump (Where's my reinforcement, KITT?)
- Implementation (Turn on the Turbo-Boost!)
- Results
- Conclusions

Project



Situation

- Migration of an application (one schema) to a new database
- Target system: 6-node RAC 11gR2
- Data volume approximately 600GB, 400+ GB LOB data



Demands

- Migration window appr. 15 hours
- Defined maintenance window Friday evening / Saturday mid morning
- Availability of the database during the migration
- Oracle standard applications only (no 3rd-party-tools)

Challenge: LOB Migration



LOB Data

- Total data volume: appr. 600GB
- 410GB of the total volume are LOBs, distributed among two tables (400 / 10)
- Basic LOBs (CLOB)
- First tests with Oracle Data Pump Export / Import



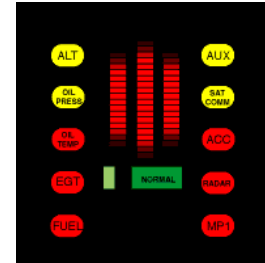
LOB Migration (I have a problem...)

- Data Pump Export:
ORA-01555: Snapshot too old
after 17 hours...
- Time windows has been exceeded already and the import hasn't even begun
 - That's not the way



LOB Migration 2

(You have all sorts of problems...)



- No parallelism when exporting / importing LOBs
- Each LOB row is transferred separately
- No usage of the array interface
- Speed up my LOBs – but how?



HL-Tablepump

(Where's my reinforcement, KITTT?)



What is the HL-Tablepump?

- Java-based application
- JDBC connects against the source and target databases
- LOB-transfer with INSERT / SELECT statements
- No dump file
- Array-Interface is used



HL-Tablepump - Requirements

- Current Java-Installation (JDK)
- Oracle-Client 10.2 or newer
- Accessibility of source and target system via Oracle Net (tnsnames.ora)

HL-Tablepump – Voraussetzungen (2)

- Theoretically possible: Running the application on any suitable client which fulfills the requirements
- In practice: Running the application on one of the participating servers, normally on the target system
 - More powerful resources (newer system)
 - Not yet in production – no impact on other applications and users



HL-Tablepump – Mode of Operation

- The use of arrays allows the transfer and writing of multiple LOB-rows per fetch (e.g. 100)
- Additionally possible: Parallelization of SELECT- and INSERT-processes

HL-Tablepump - Specifications

- Free choice of read-fetch-size (Improvement of read access and network transport)
- Free choice of write-batch-Size (Improvement of write operations)
- WHERE-clause can be specified per transport job (Horizontal distribution of data)

HL-Tablepump – Specifications (2)

- Flashback can be specified by SCN and Timestamp (Data consistency)
- Free choice of the number of worker threads (Parallelism of write processes)
- Direct-Path-Option can be specified
- NO-COMMIT-Option can be specified

Tablepump-Parameters (Excerpt)

Parameter	Value/ Function
-b, --batch-size	Batch-Size (Default: 100)
-c, --scn	SCN for Flashback-Query
-d, --direct-path-insert	Use Direct-Path-Insert (RDBMS >= 11.2)
-e, --threads	Thread-Pool
-f, --fetch-size	Fetch-Size (Default: 100)
-n, --source-table-name	Name of the source-table
-N, --target-table-name	Name of the target-table
-w, --where	SQL-WHERE-Clause



Implementation – Turn on the Turbo-Boost!

Prerequisites

- Provision of a sufficient undo retention / undo tablespace in the source database
- Creation of empty target tables without indexes / constraints in the target database
- NOLOGGING-mode for the target tables and LOB columns
- Use of a previously identified `current_scn` for consistency (Tablepump-parameter `-c`)

Implementation

- Start of the tablepump on two RAC nodes in the target environment with 4 threads each
- Distribution among the processes with a WHERE-clause on the key columns
- Logging of the operations carried out by the tablepump to STDOUT --> Log file `nohup.out`



Program Call

```
java -Xms128m -Xmx512m -cp .:commons-cli-  
1.1.jar:log4j-1.2.15.jar:  
ojdbc5.jar:HLTablePump.jar  
de.hlservices.oracle.TablePump -s  
jdbc:oracle:oci:@SOURCE.world -T  
jdbc:oracle:oci:@TARGET.world -e 4 -d -c  
17759713352 -u system -p <system_pwd> -U system  
-P <system_pwd> -o <schema_source> -n  
<table_source> -O <schema_target> -N  
<table_target> -w "id < 1035000"
```



```
[INFO ] [main] [11:20:19,045]: [Start...]  
[INFO ] [main] [11:20:19,055]: [Fetch-Size : 500]  
[INFO ] [main] [11:20:19,055]: [Batch-Size : 100]  
[INFO ] [main] [11:20:19,055]: [Buffer-Clobs: true]  
[INFO ] [main] [11:20:19,055]: [Threads : 4]  
[INFO ] [main] [11:20:19,055]: [Where : id < 1035000]  
[WARN ] [main] [11:20:19,055]: [SCN set to de.hlservices.oracle.TablePumpDescriptor@6d0d1642 !!!]  
[INFO ] [main] [11:20:19,343]: [connected to url: jdbc:oracle:oci:@SOURCE - Username: system]  
...  
[INFO ] [Thread-1] [11:20:20,524]: [sendBatch...]  
[INFO ] [Thread-4] [11:20:20,524]: [get next array...]  
[INFO ] [Thread-1] [11:20:21,087]: [commit...]  
[INFO ] [Thread-1] [11:20:21,093]: [sendBatch + commit finished. duration=0 sec.]  
[INFO ] [Thread-4] [11:20:21,677]: [current array: record-count=100 / summary: record-count=200, duration=1 sec.]  
...
```

Quellsystem

Zielsystem





Results, Subsequent Work

- Transfer duration of the bigger LOB table ~ 9 hours
- Manual index creation afterwards
- Transfer of all other (non-LOB) tables of the schema with datapump export / -import; consistent with the same `flashback_scn` as the tablepump transfer
- Reset of the NOLOGGING-parameters
- Documentation with the tablepump and datapump log files
- Random Check of rows and LOBs from source and target environment with `diff` and WinMerge, respectively



Remarks

- Use of the tablepump sped up the migration significantly, not to say it made the migration possible within the defined time window in the first place
- The tablepump has since been used in other projects, too, with even significantly larger amounts of LOB data (> 2 TB)
- In general: better use securefile LOBs than Basic LOBs (Performance!)
- Starting with Oracle 12: Transformation of basic LOBs into securefile LOBs during the datapump import
 - Parameter: `TRANSFORM=LOB_STORAGE:SECUREFILE`



Questions & Contact

- Mail: susanne.jahr@hl-services.de
- Web: www.hl-services.de
- Blog: blog.hl-services.de
- Twitter: @soesa69
- Live: Exhibition, booth 236

Level 2 (yellow), 2. booth on the left from the escalator



Thank you for your attention!

- Please do also attend our further presentations:
 - **Nachhaltiges Monitoring von Oracle-Datenbanken**
 - Sascha Westermann (Co-Referent), heute 15:00h, Konferenzraum EG
 - **Alles ist eine Aufgabe - mit Struktur erfolgreich sein**
 - Peter Bekiesch, 17.11.2016 9:00h, Saal Prag