

Zünde den Turbo-Boost! (LOB-Migration beschleunigt)

Susanne Jahr

DOAG-Konferenz Nürnberg 16.11.2016



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 (236)



Übersicht

- Projektziel
- Herausforderung LOB-Migration
- Idee: HL-Tablepump (Ich brauch' Verstärkung, KITT)
- Durchführung (Zünde den Turbo-Boost!)
- Ergebnisse

Projekt

Situation

- Migration einer Anwendung (ein Schema) auf neue Datenbank
- Zielsystem: 6-Knoten RAC 11gR2
- Datenvolumen ca. 600GB, davon ca. 400GB LOB-Daten



Anforderungen

- Migrationsfenster ca. 15 Stunden
- Definiertes Wartungsfenster Freitag Abend / Samstag Vormittag
- Verfügbarkeit der Datenbank während der Migration
- Nur Oracle-eigene Mechanismen (keine Drittanbieter-Tools)



Herausforderung LOB-Migration



LOB-Daten

- Datenmenge gesamt: ca. 600GB
- Davon ca. 410GB verteilt auf zwei Tabellen (400 / 10)
- Basic LOBs (CLOB)
- Zunächst Tests mit Oracle Data Pump Export / Import



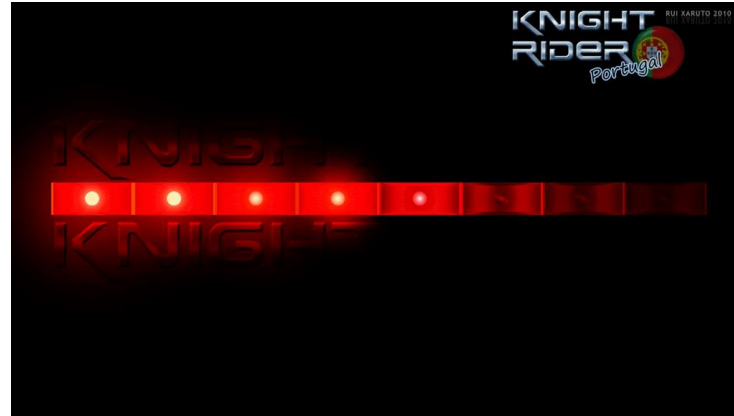
LOB-Migration (Ich habe ein Problem...)

- Data Pump Export: nach 17 Stunden
ORA-01555: Snapshot too old
- Zeitfenster schon vor Beginn des Imports überschritten
 - So geht's nicht



LOB-Migration 2 (Du hast viele Probleme...)

- Keine Parallelisierung bei LOB-Export / Import
- Jeder LOB-Datensatz wird einzeln übertragen
- Keine Nutzung des Array-Interfaces
- LOBs beschleunigen – aber wie?



HL-Tablepump
(Ich brauch‘ Verstärkung, KITT!)



Was ist die HL-Tablepump?

- Java-basiertes Programm
- JDBC-Connects gegen Quell- und Zieldatenbank
- LOB-Übertragung mit INSERT / SELECT-Statements
- Keine Dump-Datei
- Nutzung des Array-Interfaces



HL-Tablepump - Voraussetzungen

- Aktuelle Java-Installation (JDK)
- Oracle-Client mindestens der Version 10.2
- Erreichbarkeit von Quell- und Zielsystem über Oracle Net (tnsnames.ora)



HL-Tablepump – Voraussetzungen (2)

- Theoretisch möglich: Ausführung auf einem beliebigen Client, der die Voraussetzungen erfüllt
- Praxis: Ausführung auf einem der beteiligten Server, i.A. auf dem Zielsystem
 - Leistungsfähigere Ressourcen, da neueres System
 - noch nicht produktiv, also kein Impact auf andere Anwendungen / User



HL-Tablepump - Funktionsweise

- Verwendung von Arrays ermöglicht die Übertragung / das Schreiben von multiplen LOB-Datensätzen pro Fetch (z.B. 100)
- Zusätzlich möglich: Parallelisierung von SELECT- und INSERT-Prozessen



HL-Tablepump - Spezifikationen

- Lese-Fetch-Size frei wählbar (Optimierung Lesezugriffe und Transport übers Netzwerk)
- Schreib-Batch-Size frei wählbar (Optimierung Schreibvorgänge)
- WHERE-Klausel pro Transportjob spezifizierbar (horizontale Datenaufteilung)



HL-Tablepump – Spezifikationen (2)

- Flashback via SCN und Timestamp spezifizierbar (Datenkonsistenz)
- Anzahl Worker-Threads frei wählbar (Parallelisierung der Schreib-Prozesse)
- Direct-Path-Option spezifizierbar
- NO-COMMIT-Option spezifizierbar



Tablepump-Parameter (Auszug)

Parameter	Bedeutung / Funktion
-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



Durchführung – Zünde den Turbo-Boost!



Vorbereitungen

- Hinreichend große Undo-Retention / Undo-Tablespace in der Quell-Datenbank
- Anlage der leeren Tabellen ohne Indizes / Constraints auf der Ziel-Datenbank
- NOLOGGING-Modus für Zieltabellen und LOB-Spalten
- Übertragung zu einer zuvor ermittelten `current_scn` konsistent (Tablepump-Parameter `-c`)



Durchführung

- Start der Tablepump mit zwei Prozessen zu je 4 Threads auf zwei der RAC-Nodes in der Zielumgebung
- Aufteilung auf die Prozesse über WHERE-Klausel der Spalten
- Protokollierung der durchgeführten Tablepump-Aktionen nach STDOUT → Log-Datei `nohup.out`



Aufruf

```
java -Xms128m -Xmx512m -cp .:commons-cli-  
1.1.jar:log4j-1.2.15.jar:  
ojdbc5.jar:HLTablePump.jar  
de.hl.services.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





Ergebnis, Nacharbeiten

- Übertragung der größeren LOB-Tabelle in insgesamt ca. 9 Stunden
- Anlage der Indizes manuell im Anschluss
- Übertragung der restlichen Tabellen des Schemas mit Datapump Export / -Import zur selben `flashback_scn` wie die Tablepump
- Zurücksetzen der NOLOGGING-Einstellungen
- Dokumentation durch Log-Dateien
- Stichprobenartige Vergleiche von Datensätzen inkl. LOBs mit `diff` bzw. WinMerge



Bemerkungen

- Migration durch Einsatz der Tablepump signifikant beschleunigt bzw. überhaupt erst innerhalb des vorgegebenen Zeitfensters ermöglicht
- Einsatz auch bei anderen Kundenprojekten mit teilweise noch deutlich größeren LOB-Mengen (> 2 TB)
- Allgemein: Besser Securefile LOBs als Basic LOBs verwenden (Performance!)
- Ab Oracle 12: Umwandlung von Basic LOBs in Securefile Lobs während des Datapump-Imports
 - Parameter: `TRANSFORM=LOB_STORAGE:SECUREFILE`



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 236
Ebene 2 (gelb), 2. Stand links von der Rolltreppe



Vielen Dank für Ihre Aufmerksamkeit!

- Besuchen Sie auch unsere weiteren Vorträge:
 - **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