

Zünde den Turbo-Boost – LOB-Migration beschleunigt

Susanne Jahr
Herrmann & Lenz Services GmbH, Burscheid

Schlüsselworte

Migrationen, LOBs, Data Pump Export / Import, SE 1, SE 2, HL-Tablepump, Securefile LOBs

Projektziel

Für die Migration einer Anwendung auf einer Oracle-Datenbank bei einem Kundenprojekt musste u.a. ein Schema mit mehreren LOB-Tabellen übertragen werden. Quell-System war ein RAC mit Oracle 10gR2, Zielsystem ebenfalls ein RAC mit Oracle 11gR2, jeweils SuSE Linux. Die Gesamt-Datenmenge betrug ca. 600 GB, davon über 400GB LOBs, im Wesentlichen auf BLOB-Spalten in zwei Tabellen (eine mit 400GB, eine mit ca. 10 GB LOB-Daten) verteilt.

Anforderungen

Der Migrationstermin war durch eins von vier jährlichen Wartungsfenstern an einem Wochenende festgelegt und konnte nicht verschoben werden. Da die Datenbank auch am Wochenende grundsätzlich verfügbar sein musste, gab es die Anforderung, ein Zeitfenster von Freitag Abend, 18:30h – Samstag Vormittag ca. 10:00h für die Gesamt-Migrationszeit einzuhalten. Der Kunde forderte die Durchführung der Migration mit Oracle-Standard-Mitteln.

Testphase Oracle Data Pump

Zunächst wurden Tests mit Oracle Data Pump Export / Import durchgeführt. Schnell stellte sich jedoch heraus, dass hiermit das geforderte Zeitfenster nicht eingehalten werden konnte: Der Data Pump Export brach nach ca. 17 Stunden mit einem ORA-01555: Snapshot too old ab. Da an diesem Punkt das vorgegebene Zeitfenster bereits zum Großteil aufgebraucht war, mussten weitere Lösungsansätze her. Aufgrund der Anforderung, nur Oracle-eigene Tools einzusetzen, schied eine Lösung mit Replikations-Tools wie Shareplex oder Dbvisit Replicate aus.

Problem: Übertragung von LOB-Datensätzen beim Data Pump

Eigentlich sollte das verfügbare Zeitfenster für die Gesamt-Datenmenge ausreichend sein. Problematisch ist der hohe Anteil an LOBs.

Bei der Übertragung von LOBs mit Oracle Data Pump

- kann keine Parallelisierung verwendet werden
- wird jeder LOB-Locator einzeln / sequentiell gelesen

Vorschlag: HL Tablepump

Bei der HL-Tablepump handelt es sich um ein in Java geschriebenes Programm, das über JDBC Connects an die Quell- und die Zieldatenbank durchführt und unter Verwendung des Array-Interfaces mittels SELECT und INSERT die LOB-Daten der im Aufruf spezifizierten Tabelle direkt und ohne Zwischenspeicherung in einer Dump-Datei auf das Zielsystem überträgt.

Spezifikationen:

- Lese-Fetch-Size frei wählbar (Optimierung Lesezugriffe und den Transport übers Netzwerk)
- Schreib-Batch-Size frei wählbar (Optimierung Schreibvorgänge)
- WHERE-Klausel pro Transportjob spezifizierbar (horizontale Datenaufteilung)
- Flashback via SCN und Timestamp spezifizierbar (Datenkonsistenz)
- Anzahl Worker-Threads frei wählbar (Parallelisierung der Schreib-Prozesse)

- Direct-Path-Option spezifizierbar
- NO-COMMIT-Option spezifizierbar

Durch die Verwendung von Arrays ist es möglich, z.B. 100 LOB-Datensätzen in einem Fetch zu übertragen bzw. zu schreiben, während bei Oracle Data Pump immer nur ein einziger LOB-Datensatz gleichzeitig exportiert bzw. importiert werden kann. Dadurch sowie durch die mögliche Parallelisierung der SELECT- und INSERT-Prozesse auf der Quell- und Zielseite können signifikante Performance-Gewinne gegenüber Oracle Data Pump Export / Import erzielt werden.

Voraussetzungen zur Ausführung:

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

Parameter der Tablepump (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

Die HL-Tablepump kann prinzipiell auf einem beliebigen Rechner ausgeführt werden. Die beste Performance wird erzielt, wenn das Programm auf dem Zielsystem gestartet wird:

- Weniger Netzwerk-Verkehr
- Nutzung deutlich leistungsfähigerer Hardware-Ressourcen gegenüber einem PC

Vorbereitungen

In der Quelldatenbank ist für eine ausreichend große `undo_retention` sowie einen hinreichend dimensionierten Undo-Tablespace zu sorgen.

Beide betroffenen Tabellen wurden auf der Zieldatenbank leer sowie ohne Indizes angelegt und in den NOLOGGING-Modus versetzt, und zwar sowohl die Tabelle selbst als auch die LOB-Spalten:

```
ALTER TABLE target_schema.target_table
MODIFY LOB (DATA)
(NOLOGGING) ;
```

```
ALTER TABLE target_schema.target_table NOLOGGING;
```

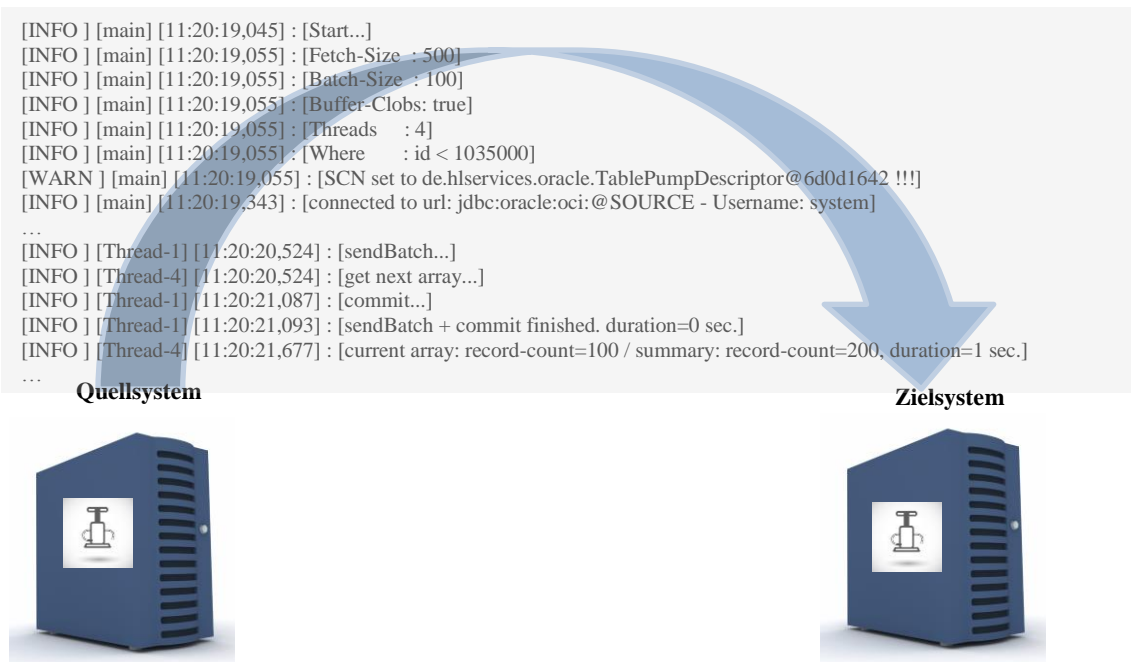
Die Übertragung war zu einer zuvor ermittelten `current_scn` konsistent (Tablepump-Parameter `-c`).

Durchführung

Die HL-Tablepump wurde mit 6 Threads und zwei Prozessen gestartet, wobei die Aufteilung auf die beiden Prozesse durch eine WHERE-Klausel der Schlüsselspalten erfolgte. Die ausgeführten Aktionen der Tablepump werden nach STDOUT protokolliert, können also beim Programmstart mit nohup in der Datei nohup.out nachvollzogen werden.

Aufruf:

```
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 6 -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"
```



Ergebnis, Nacharbeiten

Die Dauer der Übertragung der größeren LOB-Tabelle betrug während eines Tests ca. 9 Stunden (11:20 – 17:24 der erste, 11:21- 20:25 der zweite Prozess). Die übrigen Tabellen des zu migrierenden Schemas wurden mit Data Pump Export / Import zur selben flashback_scn wie die in der Tablepump verwendeten übertragen.

Die NOLOGGING-Einstellungen wurden nach der Übertragung wieder rückgängig gemacht. Die Indizes der Tabellen wurden anschließend manuell angelegt. Aufgrund der NOLOGGING-Operationen ist so bald wie möglich ein Full Backup der Zieldatenbank anzufertigen.

Auf Anforderung des Kunden wurden einzelne LOB-Datensätze zwischen Quell- und Zielsystem stichprobenartig verglichen (diff bzw. WinMerge).

Bemerkungen

Der Einsatz der HL-Tablepump verringerte die Gesamtlaufzeit der Migration signifikant bzw. ermöglichte erst die Durchführung im vorgegebenen Zeitfenster. Auch in anderen Kundenprojekten wurde die Tablepump bereits auch für noch deutlich größere LOB-Mengen (teilweise > 2 TB) erfolgreich eingesetzt.

Allgemein ist der Einsatz von Securefile-LOBs im Gegensatz zu Basic LOBs empfehlenswert, da die Performance bei der Übertragung per Data-Pump-Import bei Verwendung von Securefile-LOBs deutlich besser ist. Mit dem Parameter TRANSFORM=LOB_STORAGE:SECUREFILE ist es seit Oracle 12c möglich, während des Data-Pump-Imports die LOBs der importierten Tabellen von BASIC in SECUREFILE umzuwandeln.

Kontaktadresse

Susanne Jahr
Herrmann & Lenz Services GmbH
Am Ziegelfeld 28
51399 Burscheid

Telefon: +49 2174-30710-14
Fax: +49 2174-30710-22
E-Mail: susanne.jahr@hl-services.de

Internet: www.hl-services.de, blog.hl-services.de