

35Tb Umziehen mit iXTTS

IncrementalCrossPlatformTransportTablespaces

Abstrakt

Incremental Crossplatform Transport Tablespaces ist eine Oracle-Funktion

Die Aufgabe war, eine Datenbank von einer Solaris-Plattform zu einer Exadata-Plattform zu verschieben, natürlich mit minimaler Ausfallzeit. Die beiden Plattformen haben unterschiedliche Endian und die Datenbank ist 35+ Tb. Die gewählte Lösung ist auf Transportable Tablespaces basiert, mit RMAN der die Endian-Konvertierung ausführt, gleichzeitig mit inkrementellen Backup und Updates der Tablespaces. Allerdings fehlt hier das Übertragen der Benutzer metadata (views, procedures usw), und die müssen ja mit.

Die Präsentation handelt über die durchgeführten Schritte, mit dem "Wie" und "Warum" für jeden einzelnen Schritt, sowie das gelegentliche "Was ist hier los?!"

Über mich:

Michael A. G. Möller

Auch mit Namen Msquare (M-hoch-2 M²)

Die letzten 50 Jahre mit IT beschäftigt
(Ich habe früh angefangen)

20 Jahre Oracle Erfahrung,
davon 8 Jahre bei Oracle

Oracle Certified Master
OakTable Mitglied

Obwohl ich
viel lieber ein
Steampunk
Tesla-Spulen
Kanonier auf
einem Dampf-
Luftschiff sein
möchte ...



Inhalt

- Erst ein bisschen Hintergrund über das Problem und Transportable Tablespace Überblick
- Dann Beschreibung des XTTS Verlaufes und den Anpassungen
- Schliesslich das Ergebnis - nicht das Übliche

Das Objekt (35Tb Datenbank)

Quelle: Solaris, Einzel Instanz, 12.1.0.2, Dateisystem

Grösse: 35Tb. Davon 85% (~30Tb) im gleichen "TTS set" mit 3000 Dateien, 90 TS

Zwischen-hüpf-speicher: Gemeinsame NFS/ZSF Disk (Rechner in verschiedenen VLAN)

Ziel: Exadata, RAC, ASM

NB: Solaris hat entgegengesetztes Endian zur Exadata.

NB2: Minimierung von Ausfallzeiten: Erst eine grobe Kope, dann ein inkrementelles Update, dann Ausfall für die letzte Übertragung

DB Kopie Möglichkeiten

Export/import.	Nicht inkrementell.
Kopi & Konvertieren	Keine Endian Konv. oder Inkrem.
TTS in neue DB	Benützer Metadata Fehlt
DataGuard	Ungleiches Endian
Golden Gate	Erst exp/imp, dann einholen. Grosses produkt

Basis TTS Durchgang

Gleichartige Datenbanken (Version, Plattform usw.)

Das TTS muss eigenständig sein

Tablespaces in ReadOnly setzen

Dateien kopieren und ein spezielles expdp ausführen

 RMAN hat ein bestimmtes Kommando, das beides in einem Schritt erledigt

Am Ziel (wenn Datei Kopi fertig ist) ein spezielles impdp ausführen

 Giebt es nicht in RMAN

Tablespace in beiden Datenbanken zurück zu ReadWrite setzen.

Das Werkzeug (iXTTS)

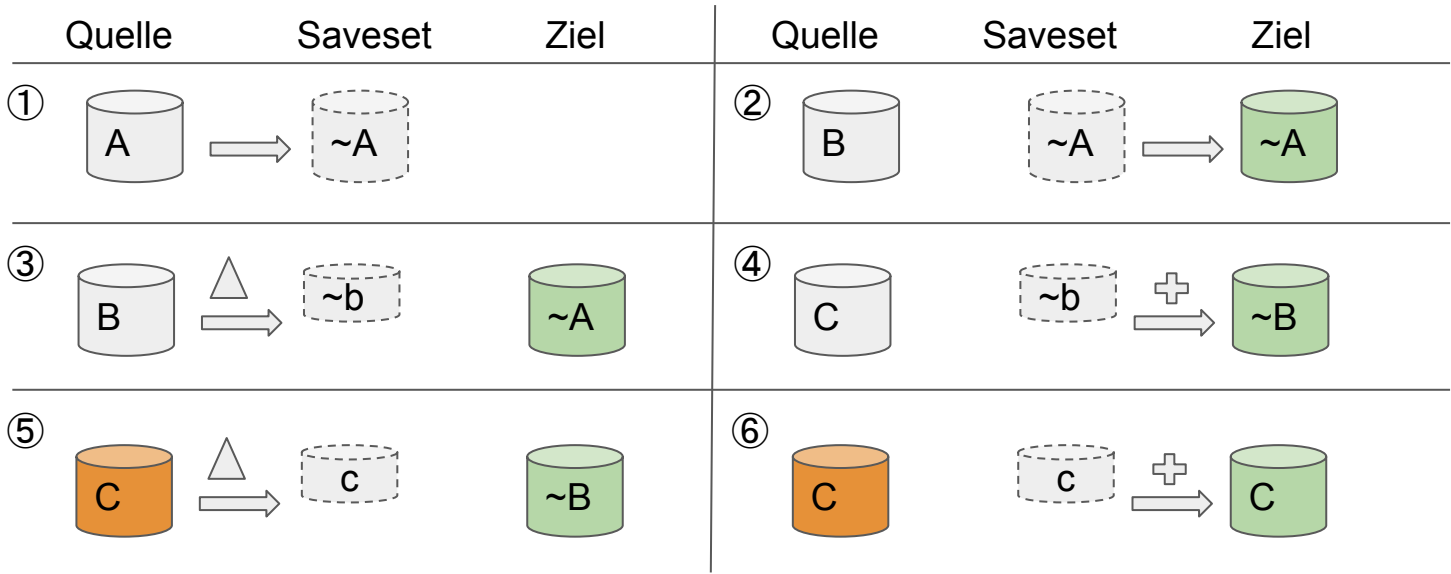
Incremental Cross Platform Transportable Tablespaces

MOS note **2005729.1** & 1389592.1

Basis TTS

- + RMAN kopiert und legt ins ASM
- + RMAN macht die Endian Änderung
- + RMAN benützt Updatable Inkrementelles Backup
- + RMAN erledigt das expdp und kopiert die DMP Datei

Schönes Diagramm vom Updateable Incremental process



Aber das sind nur die Daten Tabellen (Tablespaces)

- Benutzer Nicht-Datenstrukturen (Views, Prozeduren ...)
 - TTS überträgt die grants/triggers von Daten Objekten
- System Metadaten (Schedules, Audits, ...)
 - Einige sollten ausgeschlossen werden

Dazu dass "Catch 22" Problem

- (X)TTS erwartet das Ziel Benutzer existieren
- exp/imp von einem Schema erwartet das die Daten Tabellen existieren
 - Übrigens, die Daten werden eben nicht exp/imp'et

Script erforderlich

- Der XTTS RMAN Verlauf hat keine Metadaten von dem Quellendatei-backupset-Zieldatei Verhältnis - nichts in der Kontrolldatei verzeichnet
- Jedes Kommando muss die vollen Dateinamen angeben
 - Bei dieser Datenbank ungefähr 3000 Dateien und 100'e Savesets
- **xttdriver.pl** von Oracle, erledigt das für ein vollständiges Tabellen-Set.
 - Die grosse TTS hat 30TB, 2500 Dateien, 30.000 Segmente, 10 Tabellen
- Man **könnte** Dateien übertragen (anstelle Tabespaces) und mit "ein bisschen pipelining" parallel steuern und weniger Zwischenspeicher benutzen

XTTS Verlauf und Anpassungen

Stufe 0 - Allgemeine Vorbereitung

RMAN disk channels als Standard

Andere RMAN backups anhalten - Bandbreite und wenn SBT benützt wird

Gemeinsame erreichbar Speicher, Verzeichnisse

`/mountpoint/XTTS/runid/SCRIPT ../DUMP ../TMPDIR ../BKPSET` - 100+Tb - benützt

Selbstverständlich: Verbindung zur Datenbank (`ORACLE_SID ...`)

Stufe 0 - Vorbereitung der Quelle

x tt.properties:

```
tablespaces=NEW_DATA_I_256K,NEW_DATA_T, ...
```

```
platformid=2 (Der RMAN kommando braucht den genauen tekst von V$TRANSPORTABLE_PLATFORM)
```

```
backupformat=/temp-nfs-disk
```

```
storageondest=+DATAC/<DBname>/datafile (fehlt beschreibung in Vermerkung)
```

```
parallel=n (weglassen, =1 und =n geben verschiedene ablauf/ergebnisse)
```

TMPDIR zeigt auf directory mit `export TMPDIR=/temp-nfs-disk/...`

Die xtt-package im aktuellen Verzeichnis (und eine zusätzliche Kopi anderswo :-))

Die expdp Stufe braucht ein Verzeichnis

```
Create Directory XTTS_DIR for '/temp-nfs-disk/...' ;
```

Stufe 0 - Vorbereitung des Ziels

Erstelle neue Datenbank

"Identisch" zur Quelle - Version, Compatible, Character set, Timezone file

TMPDIR zeigt auf directory mit `export TMPDIR=/temp...`

(Nehme Kopi von xtt-package und die neue xtt.properties von Quelle)

Gleiche Verzeichnisstruktur wie bei der gemeinsamen disk (unterschiedlicher mount Punkt)

Stufe 1 - Initial Tablespace

Quelle

```
$ORACLE_HOME/perl/bin/perl xttdriver.pl --backup
```

Implizite Übertragung (aufgrund gemeinsamer disk) von /temp-nfs-disk/*.bkp
und TMPDIR tsbkupmap.txt, xttnewdatafiles.txt, incrbackups.txt

Ziel

```
$ORACLE_HOME/perl/bin/perl xttdriver.pl --restore
```

Zufügen `--debug 2` gibt mehr info, aber verlangsamt die Vorberarbeitung

Was tut Stufe 1 - Erste Übertragung

Quelle

```
backup for transport allow inconsistent incremental level 0  
datafile 7,8,9,... format '/temp-nfs-disk/%N_%f_%U.bkp';
```

Ziel (Für jedes Backupset, enthält möglicherweise mehrere Dateien)

```
restore from platform 'Solaris[tm] OE (64-bit)' foreign  
datafile 7 format '+DATAC/<orig_name>' from backupset  
'/temp-nfs-disk/<TSname>_7_0nqnq8e4_1_1.bkp';
```

Stufe 2 - Inkrementelles Einholen (beliebig wiederholen)

Quelle

```
$ORACLE_HOME/perl/bin/perl xttdriver.pl --bkpinc
```

Implizite Übertragung (gemeinsame Disk) von /temp-nfs-disk/*
und TMPDIR xttplan.txt, tsbkupmap.txt, incrbackups.txt

Ziel

```
$ORACLE_HOME/perl/bin/perl xttdriver.pl --recover
```

Quelle (nochmals)

```
$ORACLE_HOME/perl/bin/perl xttdriver.pl --s
```

Was tut Stufe 2 - Inkrementelles Einholen (beliebig wiederholen)

Quelle (für jede Tabelle)

```
set nocfau;
```

```
Backup for Transport Allow Inconsistent Incremental From SCN  
1342222 Tablespace 'USRTAB' format '/temp-nfs-disk/%U';
```

Ziel (für jedes Backupset, enthält möglicherweise mehrere Dateien)

```
Recover From platform 'Solaris[tm] OE (64-bit)' Foreign  
Datafilecopy '+DATAC/<origname>', '+DATAC/<origname>' From  
backupset '/temp-nfs-disk/0sqnqdr4_1_1';
```

Stufe 3 - Endgültiges Einholen (Ausfallzeit für Benutzer)

Quelle

```
Alter Tablespace <xyz> Read Only ;  
$ORACLE_HOME/perl/bin/perl xttdriver.pl --bkpexport
```

Implizite Übertragung (gemeinsame disk) von /temp-nfs-disk/*
und TMPDIR xttplan.txt, tsbkupmap.txt, incrbackups.txt

Stufe 3^M - Meta und Non-segment Data (läuft parallel)

Quelle

expdp

Full=Yes

Exclude=Database_Export/Tablesapce,Table,Index

Exclude=~~Database_Export/Directory~~,User:"IN('PERFSTAT')"

(+ dumpfile, directory, ...)

Stufe 3 - Finale, fortgesetzt (immer noch Ausfallzeit für Benutzer)

Ziel

```
$ORACLE_HOME/perl/bin/perl xttdriver.pl --resincrdmp
```

Stufe 3^M: impdp nur die user/roles (Nächste Folie)

Durchgang/Edit das generierte impdp-script und Ausführen von

```
. xttplugin
```

Die beteiligten Tablespaces auf Read Write stellen

```
Alter Tablespace <xyz> Read Write ;
```

Stufe 3^M - User meta data (Immer noch Ausfallzeit für Benutzer)

Ziel

Import der users (vor der XTTS Stufe 3)

```
impdp
```

```
  Include=USER,ROLE,PROFILE,DIRECTORY,PASSWORD_VERIFY...  
  (+ dumpfile, directory, ...)
```

Import der übrigen User/System Meta Data (nach XTTS Stufe 3)

```
impdp
```

```
  (+ dumpfile, directory, ...)
```

Plus eine Kleinigkeit mit default tablespaces

Was tut Stufe 3 - Endgültiges Einholen (Ausfallzeit d. Benutzer)

Quelle

```
Alter Tablespace <xyz> Read Only ;
```

```
Backup for Transport Incremental from scn 1346666 Tablespace  
<list of TS> Format '/temp-nfs-disk/%U' DataPump Format  
'/temp-nfs-disk/%U';
```

Es passiert der gleiche inkrementale datafile backup, zusätzlich wird ein dumpfile in einem backup-saveset generiert. Der dumpfile entsteht durch so etwas wie

```
expdp transport_tablespace=<list of TS> dumpfile=$OH/dbs...
```

Was tut Stufe 3 - Endgültiges Einholen (Ausfallzeit d. Benutzer)

Ziel

Für jedes Backupset und jede Datei (erstellt eine separate cmd Datei für jeden Backupset)

```
recover from platform 'Solaris[tm] OE (64-bit)' Foreign DataFileCopy  
'+DATAC/<origname>' from backupset '/temp-nfs-disk/0vqnqefq_1_1';
```

Der dump wird heraus geholt

```
restore from platform 'Solaris[tm] OE (64-bit)' dump file  
'impdp78972_340.dmp' datapump destination '<TMPDIR>' from backupset  
'/temp-nfs-disk/10qnqeg2_1_1';
```

xttpugin **script *nach* edit**

```
impdp tablespaces=<list> dumpfile= directory= logfile=  
Alter Tablespace <xyz> Read Write ;
```

Letzte Stufe - Durchgang, Kontrolle und Öffnung

Durchgang der Logdateien - bekannte Fehler (durch Testläufe) aussortieren

Kleine manuelle Korrekturen von speziellen "DBA" und maschinenspezifischen Elemente wie Verzeichnisse.

Maschinen Einzelheiten wie Directories.

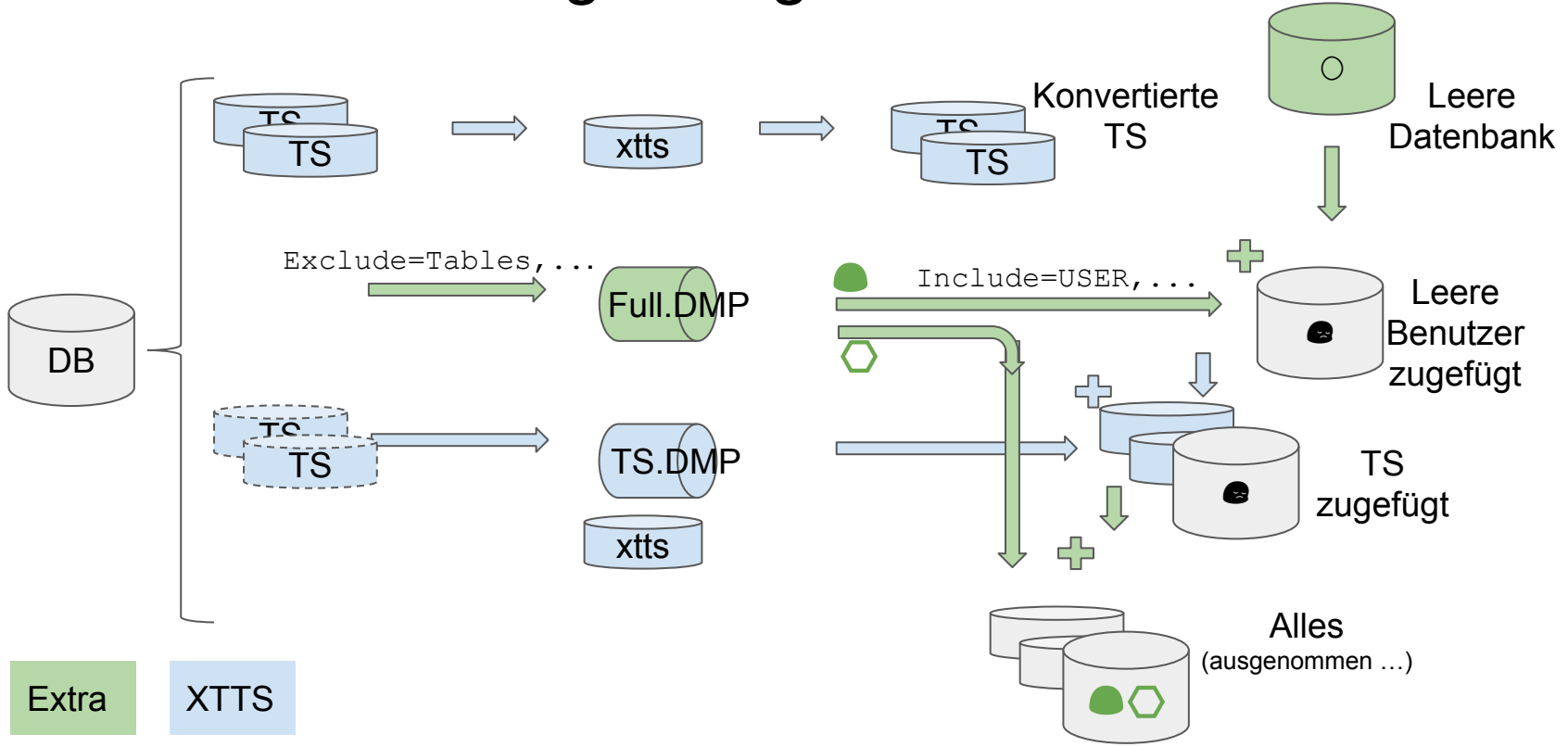
Alle übliche Aktivitäten wenn eine Datenbank verschoben wird, wie TNSNAMES Änderung

Öffne den Listener/Service

Noch ein paar Kleinigkeiten

- Die MOS note ist für 12c, aber hat einige Reste von der 11g Version
 - Obwohl einige zusätzliche Kommandos in der 11g Version detailliert sind
- `xtt.properties` beschreibt (veraltete) 11g Namen
 - Ignorieren/weglassen funktioniert.
- `xttdriver.pl` verknüpft neue Dateinamen am Ziel
"+diskpath/tablespace"
- RMAN platziert die vorübergehende DMP Datei in `$ORACLE_HOME/dbs`
 - (? Das gleiche wie CONFIGURE SNAPSHOT CONTROLFILE NAME Stelle)
- Die ASM Namen erscheinen als ein alias, an der richtigen Stelle mit Name
 - Beispiel `+DATAC/<name>` und `+DATAC/<database>/DATAFILE/<name>.xxx.yyyy`

Schöne Zeichnung vom ganzen Prozess



Ergebnisse

Der gute Plan

- Erst mit eigener kleinen Datenbank den process prüfen
- Dann mit test kopi.
- Wahrscheinliche Fehler beheben, wiederholen
- Dokumentation
- Generalprobe
- Der grosse Tag

Projekt : Timeline & Deadlines (Stand Anfang März)

- Mein Engagement begann Ende November.
- Zugang, Speicherplatz, Netzwerk, Exadata Umbau bis Mitte Dezember
 - In der Zwischenzeit mit kleiner Test-datenbank xttdriver Funktionalität und "Internals" überprüft.
 - Einen halb-herzigen TestRun während der Weihnachtsferien durchgeführt
- 1. Versuch (5 Tage zum kopieren) erwartete Fertigstellung Mitte Januar.
[Diese Präsentation eingereicht](#)
- Bug und neue Versuche bis Mitte Februar (jeder Versuch *d-a-u-e-r-t* !)
 - Das M² Verfahren parallel entwickelt, und eingesetzt - an jedes xttdriver aufgetauchte Problem angepasst.
- Der Produktionslauf musste vor dem 1. April laufen, ein kompletter problemfreier Lauf davor, und Anwendung einer Testversion wurde "im letzten Monat" erwartet.
Immer noch keinen einzigen "normalen" Lauf durchgeführt. (Zwei Teilerfolge)
- Oracle Support

Unangenehme Überraschungen

- `xttdriver.pl` Abbruch bei >100 Dateien im TTS
- `xttdriver.pl` verliert (!?!) Dateien in ASM
 - Gleiche Namen in verschiedenen paths - Verschiedene gross/klein
- `impdp` Abbruch unter Import ORA-600
- `rman` Abbruch bei 2Gb dmp (kleinerer TTS, hatte mehrere Segmente) wahrscheinlich `expdp`
- Hauptspeicher Mangel in `rman` unter sehr langem Kommando (Doc ID 559449.1)
- Inkremental Restore Abbruch `rman` (Bug:22761995)
- ...

3-12204045	11:52 AM
RMAN ORA-7445 and ORA-600 on Recover incremental datafile ★	>
👤 MICHAEL.MOLLER, Development Working	
3-12057197	11:08 AM
xttdriver.pl fails on incremental recover ★	>
👤 MICHAEL.MOLLER, Review Update	
3-12201782	1:27 AM
RMAN crash ORA600 on saving a DUMPFILE ★	>
👤 MICHAEL.MOLLER, Review Update	
3-1216865	Yesterday 2:17 PM
RMAN Recover Foreign Datafilecopy ends without doing anything ★	>
👤 MICHAEL.MOLLER, Work In Progress	
3-1216237	Yesterday 2:02 PM
xttsdriver.pl fails with ORA-600 [klaprs_11] or ORA-600 [klaprs_75] ★	>
👤 MICHAEL.MOLLER, Review Update	
3-12202603	Yesterday 2:00 PM
Spin of 3-12190335651 ★	>
👤 MICHAEL.MOLLER, Close Requested	
3-12190335	Yesterday 1:58 PM
Full Import error - RADM_FPTM ★	>
👤 MICHAEL.MOLLER, Close Requested	
3-1214757	Yesterday 10:17 AM

Das Rad neu erfinden

- XTTS Begrenzung:
 - Zu wenig Fortschritt-Information ohne debug
 - Nur monolithische Schritte - Pipeline nicht möglich
- Eignes script erfunden
 - Jede Datei ihr eigenes saveset (um es einfach zu halten)
 - Gab Logfiles und Neustart möglichkeit
 - Umgehen der frühen Fehler, unter anderem mit
 - Rman das expdp ausführt (dumpfile clause)
 - lange Kommando Zeilen mit 1000s Namen (ausser der letzte impdp)
 - Keine ASM aliases
- NichtGeprüft:
 - Quellen Datenbank erweitert mit Dateien zwischen Start und letztem delta

M² script

- Ähnliche Struktur-Stufen : fast die gleichen `rman` Kommandos
- Gleiche Nichtaufnahme von User/system meta data – Gleiche Lösung.
- PLSQL (keine Erfahrung in Perl, zu wenig Linux shell), 6 Stufen mit je einem script
 - ~1000 Zeilen, hauptsächlich `UTL_File()` Zeug.
 - Abhängig von Logdateien (oder `V$RMAN_OUTPUT`) für einige Verknüpfungen
- Halb manuell – 5 to 15 erzeugte scripts pro Stufe:
Damit ist Pipelining , Parallelisierung und Neustart möglich

Oracle-2-Oracle

- Das Oracle ACS Team schlug ihre eigene proprietäre Methode vor, um die Frist einzuhalten
 - aud kann nur von ACS Personen durchgeführt werden
- Es besteht aus einer Mischung von exp/imp und GoldenGate
 - Grosse Skript-komplexe verwendet massive Parallelisierung

Ergebnisse mit der xttdriver Methode - abgebrochen

- Alle (?) bugs wurden korrigiert oder mit workarounds von Oracle gelöst
 - Der exp/imp workaround noch offen: DUMPFILKlausul vermeiden, manuell durchführen
- Immer noch zu wenige Informationen und keine Möglichkeit zum Neustart.
 - Parallelität teilweise einstellbar

Verlauf mit xttdriver wurde abgebrochen (und anstelle die M² Methode angewendet)

- Keine Zeit um alternativen modus mit SQLNet mit direkter Kopi zu testen
- Nicht bestätigt ob es die gleiche exp/imp Probleme hatte (wahrscheinlich, da gleiches Kommando)

Ergebnisse mit der M² Methode - Enigermassen

- Der XTTS Teil des Processes hat "im Prinzip" funktioniert
- Fast alle Fehler umgegangen
Mit extra Übertragung der impdp verschwundenen Tabellen
- Unbekannt - Zeitaufwand und byte-Grösse der Delta (kein Produktions Test)
- Unbekannt - Zeitaufwand/Parallelisierung der downtime Stufe (kein normaler Testablauf)
- Aber: 😊 Fertige umgezogene Datenbank für Benutzertest geliefert 😊

Ergebnisse mit der O-2-O Methode

CAVEAT: Keinen direkten Kenntnisse von der ACS Gruppen Arbeit

- Ungefähr die gleiche Mänge bugs mit export, Dateitypen, Objektanzahl
- Hauptkopierung brauchte 3 mal mehr als angenommen (6 anstelle 2 Tage)
- Hatten auch einen fertigen Datenbank Umzug geschafft

Wo bleibt die Fanfare ?

"The operation succeeded, but the patient died"

Der Kunde des Kunden entschliesst sich das die Deadline zu knapp war, um sicher zu sein, dass eine Exadata Version der Datenbank fertig und geprüft geliefert werden könnte.

Projekt auf Herbst verschoben.

Schlussatz

*XTTS hat funktioniert (mit M^2 script),
aber es waren viele Probleme
mit dieser Datenbank (Grösse, Anzahl Objekte).*

Fragen und Antworten - jemand?

