Migrasi database Oracle yang besar ke lingkungan yang heterogen AWS

AWS Panduan Preskriptif

AWS Panduan Preskriptif: Migrasi database Oracle yang besar ke lingkungan yang heterogen AWS

Table of Contents

Migrasi database Oracle yang besar ke lingkungan lintas platform AWS

Incheol Roh, Amazon Web Services ()AWS

Maret 2021 (riwayat dokumen)

Saat memigrasikan database Oracle yang lebih besar dari 100 TB dari lokasi ke Amazon Web Services (AWS) Cloud, waktu henti migrasi merupakan faktor serius. Secara khusus, jika sistem adalah sistem mission-critical, seperti perencanaan sumber daya perusahaan global (ERP) atau sistem eksekusi manufaktur (MES), Anda ingin mengurangi downtime sebanyak mungkin untuk kelangsungan bisnis.

Ada banyak cara untuk memigrasikan database Oracle antar sistem yang memiliki format endian berbeda, seperti yang disebutkan dalam Strategi untuk Migrasi Database Oracle ke. AWS Salah satu pendekatan ini adalah Oracle cross-platform transportable tablespaces (XTTS), yang dapat Anda gunakan untuk mengurangi waktu henti migrasi dari hari ke jam.

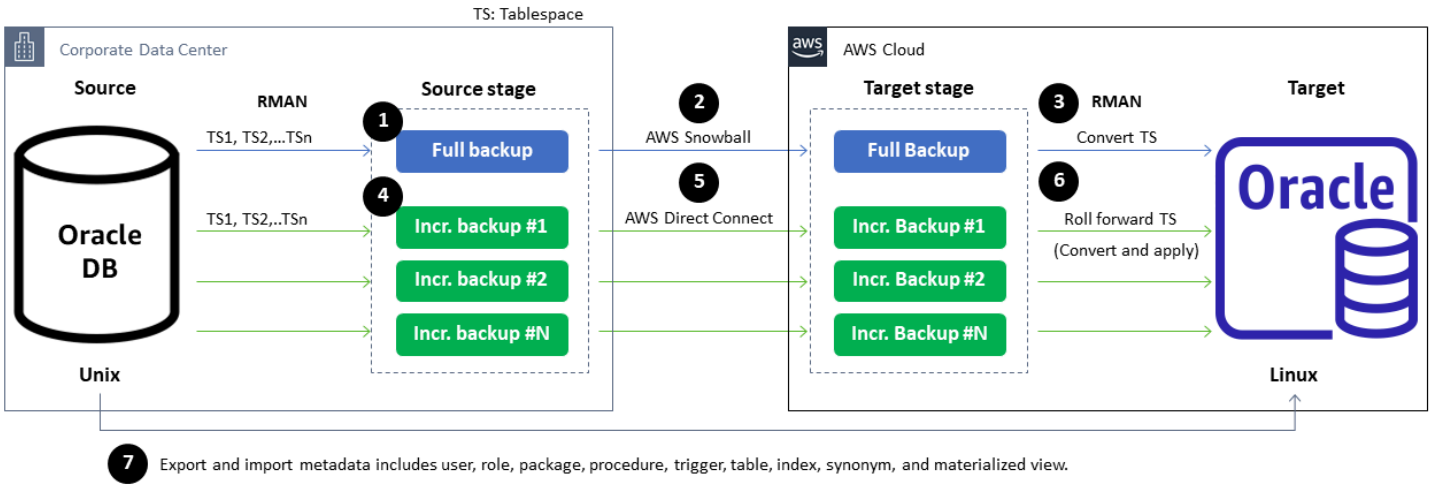
Menggabungkan Oracle XTTS dengan Oracle Recovery Manager (RMAN) incremental backup dapat secara signifikan mengurangi jumlah downtime yang diperlukan untuk memindahkan data antar platform yang menjalankan format endian yang berbeda.

Panduan ini memperkenalkan cara menggunakan AWS Snowball Edge, AWS Direct Connect, dan Amazon FSx untuk Lustre dengan Oracle XTTS dengan cadangan tambahan RMAN. Tujuan dari pendekatan ini adalah untuk meminimalkan downtime migrasi di lingkungan yang memiliki kumpulan data yang sangat besar.

Gambaran umum

Ini adalah proses konseptual migrasi database Oracle untuk AWS menggunakan Oracle XTTS dan RMAN incremental backup dengan Snowball Edge,, dan FSx for Lustre. Direct Connect

Diagram berikut menunjukkan langkah-langkah migrasi tingkat tinggi untuk database Oracle di berbagai format endian.



1. Buat cadangan penuh dari semua ruang meja.
2. Gunakan Snowball Edge untuk memindahkan cadangan dari tahap sumber ke tahap target.
3. Mengkonversi tablespaces ke database target.
4. Buat cadangan tambahan.
5. Gunakan Direct Connect untuk mentransfer cadangan tambahan dari tahap sumber ke tahap target.
6. Gulung ke depan cadangan tambahan, konversikan dan terapkan ke database target.
7. Ekspor dan impor metadata untuk semua ruang tabel yang diangkut.

Sebelum cutover, Anda dapat meminimalkan waktu henti dengan melakukan hal berikut:

- Mengekspor dan mengimpor metadata objek berbasis nonsegmen, termasuk,,,, dan `USER PACKAGE_SPEC PACKAGE_BODY PROCEDURE FUNCTION`
- Meningkatkan paralelisme untuk cadangan penuh dan pencadangan tambahan
- Mengonversi file data
- Memajukan cadangan selama migrasi

Dokumen Oracle Reduce Transportable Tablespace Downtime menggunakan Cross Platform Incremental Backup (2471245.1) menjelaskan cara menggunakan Oracle XTTS dengan backup tambahan RMAN. Dokumen ini juga mencakup rincian tentang persyaratan dan rekomendasi. Dokumen tidak menjelaskan cara memigrasikan database Oracle dari lingkungan lokal ke Oracle AWS atau cara memparalelkan setiap langkah migrasi untuk meminimalkan waktu henti.

Panduan ini menyediakan cara untuk memparalelkan fase, meminimalkan downtime migrasi di lingkungan sistem mission-critical dengan ukuran data yang sangat besar.

Setelah fase penyiapan awal, langkah-langkah tingkat tinggi untuk menggunakan Oracle XTTS dengan cadangan tambahan RMAN mencakup fase berikut.

Fase 1 - Mempersiapkan fase

Fase persiapan terdiri dari langkah-langkah berikut:

1. Cadangan penuh awal (level=0) dari tablespaces diambil pada database sumber ke tahap sumber, yaitu penyimpanan NAS.
2. Salinan cadangan ditransfer menggunakan Snowball Edge ke tahap target, yaitu FSx for Lustre terintegrasi dengan Amazon Simple Storage Service (Amazon S3).
3. Ruang tabel Backup dipulihkan dan dikonversi ke database target dengan format endian kecil.

Langkah-langkah dalam fase ini dijalankan hanya sekali selama migrasi. Data yang diangkut sepenuhnya dapat diakses dalam database sumber selama fase ini.

Fase 2 — Roll-forward fase

Fase roll-forward terdiri dari langkah-langkah berikut:

1. Cadangan tambahan diambil dari database sumber ke tahap sumber.
2. Salinan cadangan tambahan ditransfer ke tahap target di atas Direct Connect.
3. Salinan cadangan tambahan dikonversi ke database target dengan format endian kecil. Salinan kemudian diterapkan ke database target awal, yang disebut langkah roll-forward.

Anda dapat menjalankan fase ini beberapa kali. Setiap pencadangan inkremental berturut-turut harus memakan waktu lebih sedikit dan akan membawa salinan file data tujuan lebih terkini dengan basis

data sumber. Seperti pada fase 1, data sumber yang diangkut dapat diakses sepenuhnya selama fase ini.

Fase 3 - Fase transportasi

Fase ketiga meliputi langkah-langkah berikut:

1. Ruang meja yang diangkut diubah menjadi hanya baca.
2. Cadangan inkremental akhir diambil dari database sumber.
3. Metadata diekspor.
4. Cadangan ditransfer dan diterapkan ke tujuan.
5. Metadata objek diimpor.

Pada titik ini, nomor perubahan sistem (SCN) dari database tujuan konsisten dengan database sumber.

Metadata ruang meja yang dapat diangkut diekspor dari database sumber dan diimpor ke database tujuan. Metadata mencakup informasi untuk pengguna, peran, paket, prosedur, fungsi, tabel, dan indeks.

Akhirnya, tablespaces dibuat read/write untuk akses penuh pada database tujuan dari aplikasi.

Fase ini diikuti oleh fase validasi.

Fase migrasi

Panduan ini mencakup migrasi Oracle single instance dan Oracle RAC sebagai basis data sumber dan target. Jika sumber dan targetnya adalah Oracle RAC, Anda dapat memaksimalkan paralelisme untuk setiap langkah migrasi.

Fase 0 - Fase pengaturan awal

1. Instal database Oracle 12.1.0.2 atau versi lebih baru pada instans Amazon Elastic Compute Cloud (Amazon EC2) target.
2. Verifikasi basis data target.

   Anda harus memverifikasi apakah instance database target memiliki zona waktu dan karakter yang sama dengan basis data sumber. Jika tidak, pendekatan migrasi ini akan gagal

   Untuk memeriksa apakah versi zona waktu di sumber dan target sama, jalankan perintah berikut.

```
SQL> select * from v$timezone_file;
FILENAME            VERSION    CON_ID
----------------------   -------------   ----------
timezlrg_14.dat        14               0
```

   Untuk memeriksa apakah set karakter DB dan set karakter nasional sama di sumber dan target, jalankan perintah berikut.

```
SQL> select * from nls_database_parameters
         where parameter in ('NLS_CHARACTERSET','NLS_NCHAR_CHARACTERSET');
PARAMETER                                VALUE
-----------------------------------------   --------------------
NLS_NCHAR_CHARACTERSET         UTF8
NLS_CHARACTERSET                    AL32UTF8
```

3. Siapkan utilitas Oracle XTTS.

   Pada sistem sumber, unduh dan ekstrak file skrip, VER4.ziprman_xttconvert_ dari dokumen Oracle 2471245.1. Ubah parameter dalam `xtt.properties` file dengan konfigurasi tertentu. Kemudian salin `xtt.properties` dan `xttdriver.pl` skrip ke sistem tujuan.

Fase 1 - Mempersiapkan fase (database sumber tetap online)

Selama fase ini, file data dari tablespace yang akan diangkut dicadangkan pada sistem sumber. Salinan cadangan diangkut ke sistem tujuan dan dipulihkan dengan menjalankan `xttdriver.pl` skrip.

Langkah 1: Ambil cadangan tablespace penuh (level = 0) pada sistem sumber

Untuk mengurangi waktu durasi pencadangan, salin `xttdriver.pl` file `xtt.properties` dan ke beberapa direktori. Di setiap `xtt.properties` file di bawah setiap direktori, masukkan nama tablespace di `tablespaces` parameter. Kemudian, di bawah setiap direktori, jalankan `xttdriver.pl` dengan `--backup` opsi. Perintah ini mengangkut semua tablespace kecuali untuk `SYSTEMSYSAUX`,`UNDO`,, dan`TEMP`, yang dibuat selama instalasi database target.

Misalnya, setiap `xtt01~xtt04` direktori memiliki semua `xtt` skrip (`xtt.propertiesdanxttdriver.pl`) dengan nilai yang berbeda untuk `tablespaces=` parameter dalam `xtt.properties` file. Saat mengedit `tablespaces` parameter di setiap `xtt.properties` file, pertimbangkan untuk membagi grup tablespace sehingga ukuran total file data di setiap grup tablespace serupa.

Dalam panduan ini, contoh mengandaikan bahwa ada empat grup tablespace. Jika database sumber adalah Oracle single instance, Anda membuat semua `xtt01~04` direktori. Jika database sumbernya adalah Oracle RAC, Anda dapat membuat setiap `xtt` direktori di setiap server Oracle RAC.

```
[oracle@erp expimp]$ ls
out  xtt01  xtt02  xtt03    xtt04
[oracle@erp out]$ ls
out01 out02 out03 out04
```

Tabel berikut menunjukkan contoh `tablespaces=` parameter dalam `xtt.properties` file.

`xtt01`	`tablespaces=APPS_TS_TX_DATA`
`xtt02`	`tablespaces=APPS_TS_TX_IDX`
`xtt03`	`tablespaces=APPS_TS_SUMMARY`

xtt04	tablespaces=APPS_CALCLIP, APPS_OMO, APPS_TS_ARCHIVE, APPS_TS_DISCO, APPS_TS_DISCO_OLAP, APPS_TS_INTERFACE, APPS_TS_MEDIA, APPS_TS_NOLOGGING, APPS_TS_QUEUES, APPS_TS_SEED…

Untuk mencegah salinan cadangan file data yang ada dihapus saat berjalan `xttdriver.pl` secara paralel, komentari baris berikut di`xttdriver.pl`.

```
   if (@present)
   {
      ## Try deleting any existing copied datafiles
##Comment out it for executing rman command in parallel by AWS
##        system("\\rm -f $dfcopydir/*.tf");
      sleep 5;
   }
```

Gunakan `xttdriver.pl` untuk membuat cadangan RMAN dari sistem sumber.

Jika sistem sumbernya adalah Oracle RAC, itu dapat dijalankan pada setiap instance Oracle RAC secara bersamaan.

Output `xttdriver.pl` disimpan dalam variabel lingkungan `TMPDIR`. Jalankan setiap `xttdriver.pl` perintah dengan `--backup` opsi, dan gunakan `--debug` opsi untuk mengaktifkan mode debug.

```
cd /u01/oracle/expimp/xtt<nn>
export TMPDIR=/u01/oracle/expimp/out/out<nn>
$ORACLE_HOME/perl/bin/perl xttdriver.pl --backup --debug 3
```

Dalam contoh ini, <nn>mewakili kelompok tablespace. Jika ada empat grup tablespace, <nn>menjadi,`01`, `0203`, dan`04`.

Langkah 2: Transfer salinan cadangan lengkap ke sistem tujuan

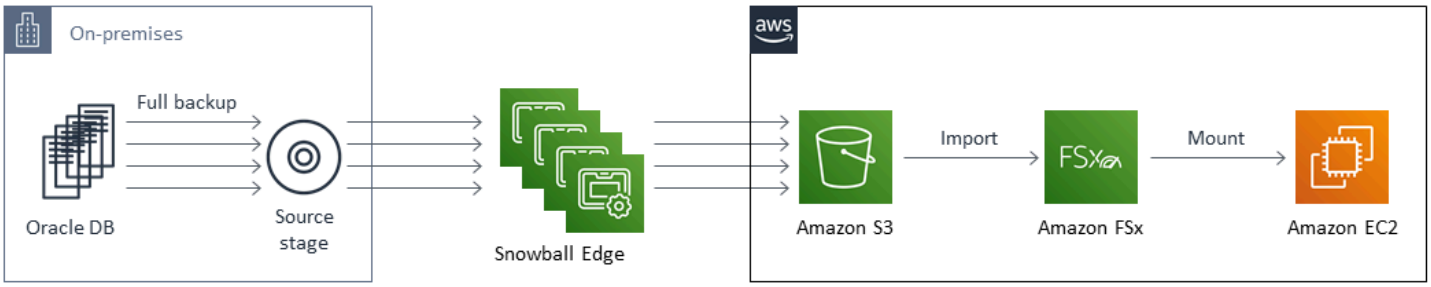
Gunakan salah satu dari dua opsi berikut untuk mentransfer salinan cadangan Anda ke sistem tujuan.

Opsi 1 — Menggunakan AWS Snowball Edge

Jalur: Database sumber -> tahap sumber -> AWS Snowball Edge -> Amazon S3 -> FSx for Lustre

> **Note**
>
> AWS Snowball Edge tidak lagi tersedia untuk pelanggan baru. Pelanggan baru harus AWS DataSyncmencari transfer online, Terminal Transfer AWS Data untuk transfer fisik yang aman, atau AWS Partner solusi. Untuk komputasi tepi, jelajahi AWS Outposts.

Diagram berikut menunjukkan transfer cadangan penuh menggunakan Snowball Edge.



Snowball Edge adalah perangkat penyimpanan dan komputasi fisik tangguh yang dapat Anda gunakan untuk memindahkan data skala besar. AWS Snowball Edge membantu mengatasi tantangan yang mungkin Anda hadapi dengan transfer data skala besar, termasuk waktu transfer yang lama, kurangnya bandwidth yang dapat digunakan, dan masalah keamanan.

Seluruh salinan cadangan file data ditransfer ke Amazon S3 oleh Snowball Edge. Jika ukuran sistem sumber lebih dari 100 TB, Anda harus menggunakan beberapa perangkat Snowball Edge untuk mengurangi durasi transfer.

Amazon FSx for Lustre sangat terintegrasi dengan Amazon S3. Anda dapat membuat sistem file FSx for Lustre yang ditautkan ke bucket S3 Anda dalam hitungan menit. Saat ditautkan ke repositori data Amazon S3, sistem file FSx for Lustre secara transparan menyajikan objek Amazon S3 sebagai file. Dalam lingkungan nyata, kinerja FSx for Lustre tergantung pada kapasitas penyimpanannya, ukuran setiap file, dan seberapa baik file didistribusikan ke dalam sistem file. Ketika FSx for Lustre digunakan sebagai penyimpanan tahap target, Anda tidak perlu mengembalikan salinan cadangan tablespace ke Amazon Elastic Block Store (Amazon EBS) atau Amazon Elastic File System dari Amazon S3.

1. Impor bucket S3 ke FSx for Lustre.

Gunakan Snowball Edge untuk mentransfer dan menyimpan area panggung sumber (misalnya,`src_backups`) ke dalam ember S3 (misalnya,). `s3-src-backups`

Buat sistem file FSx for Lustre (`dest_backups`misalnya,) sebagai area tahap tujuan untuk salinan cadangan file data.

Instal klien Lustre open-source pada sistem tujuan (Amazon EC2). Untuk informasi selengkapnya, lihat Panduan Pengguna Amazon FSx for Lustre.

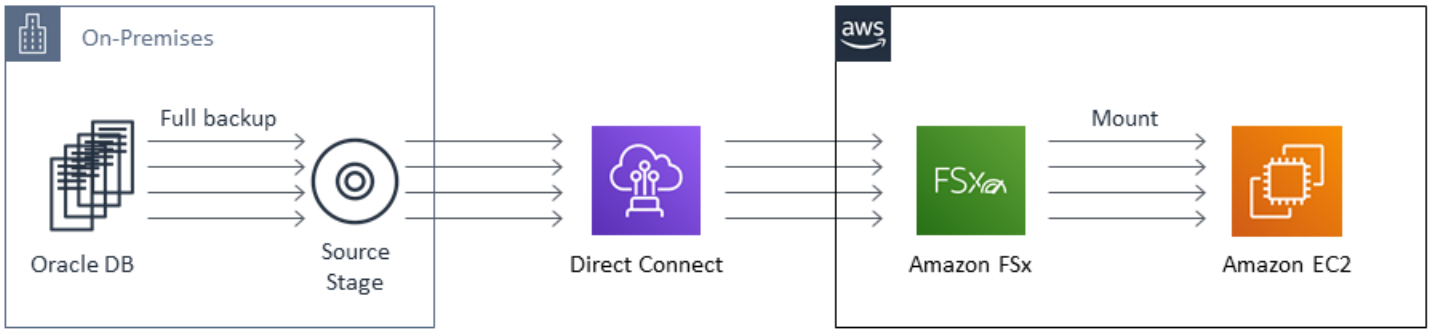
Setelah klien Lustre diinstal, pasang sistem file FSx for Lustre ke sistem tujuan (Amazon EC2).

2. Transfer `res.txt` dari `$TMPDIR` sumber `$TMPDIR` ke tujuan.

Opsi 2 — Menggunakan AWS Direct Connect

Jalur: Database sumber -> tahap sumber -> AWS Direct Connect -> FSx for Lustre

Diagram berikut menunjukkan transfer cadangan penuh dengan menggunakan AWS Direct Connect.



Direct Connect memberi Anda kemampuan untuk membuat koneksi jaringan pribadi antara pusat data, kantor, atau lingkungan colocation Anda dan AWS. Koneksi jaringan pribadi ini mengurangi biaya jaringan, meningkatkan throughput, dan memberikan pengalaman yang konsisten.

Anda dapat langsung mentransfer salinan cadangan file data dari sistem sumber ke sistem tujuan (Amazon EC2) di mana sistem file Amazon FSx for Lustre dipasang. Opsi ini lebih cepat daripada opsi Snowball Edge jika Anda dapat mengakomodasi bandwidth tinggi. Direct Connect

Setelah salinan cadangan file data ditransfer oleh Snowball Edge atau Direct Connect, Anda dapat melihatnya di sistem file FSx for Lustre di area tahap target.

Langkah 3: Kembalikan salinan cadangan lengkap dan konversi file data

Kembalikan salinan cadangan lengkap dan konversi file data ke format endian sistem tujuan. Untuk mengurangi durasi pemulihan dan konversi, jalankan `xttdriver.pl` perintah dengan `--restore` opsi untuk setiap grup tablespace.

```
cd /u01/oracle/expimp/xtt<nn>
export TMPDIR=/u01/oracle/expimp/out/out<nn>
$ORACLE_HOME/perl/bin/perl xttdriver.pl --restore --debug 3
```

Setelah langkah selesai, file data ditempatkan di dest_datafile_location yang ditentukan dalam file pada sistem tujuan. `xtt.properties`

Fase 2 — Roll-forward fase (database sumber tetap online)

Anda dapat mengulangi fase roll-forward sebanyak yang diperlukan untuk menangkap file data tujuan hingga database sumber.

Fase roll-forward terdiri dari langkah-langkah berikut.

1. Buat cadangan tambahan dari database sumber.
2. Transfer cadangan ke sistem tujuan.
3. Konversi cadangan ke format endian sistem tujuan.
4. Terapkan cadangan ke salinan file data tujuan yang dikonversi untuk menggulungnya ke depan.

Setiap backup inkremental berturut-turut harus memakan waktu lebih sedikit dan akan membawa salinan file data tujuan lebih terkini dengan database sumber.

Langkah 1: Ambil backup inkremental secara paralel

Ambil cadangan tambahan dari grup tablespace yang diangkut pada sistem sumber secara paralel.

Langkah ini membuat backup tambahan untuk semua tablespaces= yang tercantum dalam file. `xtt.properties`

Jika Anda dapat mengaktifkan fitur BLOCK CHANGE TRACKING pada sistem sumber, Anda dapat sangat mengurangi waktu pencadangan tambahan.

Perintah berikut mengenali SCN berikutnya setelah pencadangan penuh secara otomatis.

```
cd /u01/oracle/expimp/xtt<nn>
export TMPDIR=/u01/oracle/expimp/out/out<nn>
$ORACLE_HOME/perl/bin/perl xttdriver.pl --backup --debug 3
```

Langkah 2: Transfer backup tambahan dan file res.txt ke sistem tujuan

Jika Anda memiliki bandwidth yang cukup, Anda dapat mengurangi durasi transfer dengan menggunakan Direct Connect.

Transfer cadangan tambahan antara dan. `src_scratch_location` `dest_scratch_location` Transfer `res.txt` file antara `$TMPDIR` pada sistem sumber dan `$TMPDIR` pada sistem tujuan.

Jika Anda mengambil beberapa cadangan tambahan, `res.txt` file harus disalin setelah pencadangan tambahan terakhir sebelum dapat diterapkan pada sistem tujuan.

```
[source]$ scp $TMPDIR/res.txt oracle@[dest]:/u01/oracle/expimp/out/out<nn>
[source]$ scp <src_scratch_location>/* oracle@[dest]:<dest_scratch_location>
```

Langkah 3: Konversi dan terapkan cadangan tambahan

Ubah cadangan tambahan ke format endian sistem tujuan dan terapkan pencadangan tambahan ke salinan file data tujuan pada sistem tujuan.

Dalam panduan ini, anggaplah bahwa kelompok tablespace adalah empat. Jalankan setiap `xttdriver.pl` perintah dengan `--restore` opsi untuk setiap grup tablespace.

Pada langkah ini, utilitas Oracle XTTS mengambil database target untuk dimulai ulang. Untuk menjalankan perintah secara paralel, Anda harus menggunakan kustomisasi berikut dalam skrip Perl,`xttdriver.pl`.

1. Komentari baris berikut:

   - Baris 4867: `my $outputstart = `sqlplus -L -s \"/ as sysdba\" \@xttstartupnomount.sql`;`
   - Baris 4868: `checkError ("Error in executing xttstartupnomount.sql", $outputstart);`
   - Baris 4992: `my $outputstart = `sqlplus -L -s \"/ as sysdba\" \@xttdbopen.sql`;\`

2. Buat target DB dalam NOMOUNT status.

```
sqlplus / as sysdba @xttstartupnomount.sql
```

3. Lakukan roll forward dari incremental backup secara parallel.

```
cd /u01/oracle/expimp/xtt<nn>
export TMPDIR=/u01/oracle/expimp/out/out<nn>
$ORACLE_HOME/perl/bin/perl xttdriver.pl --restore --debug 3
```

4. Setelah memajukan semua cadangan tambahan, atur status DB target. OPEN

```
sqlplus / as sysdba @xttdbopen.sql
```

Pada titik ini, Anda dapat mengulangi fase 2 hingga SCN pada basis data sumber dan tujuan cukup dekat. Anda harus memutuskan apakah maju ke fase 3 atau mengulang fase 2, tergantung pada target downtime yang diberikan.

Untuk mengurangi waktu henti selama fase 3 (fase transportasi), Anda dapat mengekspor dan mengimpor metadata objek berbasis nonsegmen, seperti,,, dan USER PACKAGE PROCEDUREFUNCTION, sebelum Anda mengatur database sumber sebagai. READ ONLY

Jika jumlah objek database dalam database sumber sangat besar (ratusan ribu), mengekspor dan mengimpor metadata akan memakan waktu beberapa jam. Dalam hal ini, pertimbangkan untuk mengekspor metadata.

Mengekspor objek berbasis nonsegmen dijalankan read/write pada sistem sumber. Maka Anda harus mengimpor objek ke sistem tujuan sebelum sistem sumber diatur keREAD ONLY. Pada saat ini, Anda harus menyimpan objek sumber database sepertiPACKAGE,PROCEDURE, dan FUNCTION tidak berubah.

Ini adalah contoh file parameter dump yang digunakan untuk mengekspor metadata objek berbasis nonsegmen, termasukUSER,,,, dan. PACKAGE_SPEC PACKAGE_BODY PROCEDURE FUNCTION

```
directory=dmpdir
dumpfile=xttsmsc%U.dmp
full=y
filesize=1048576000
logfile=expmsc.log
metrics=y
```

```
exclude=TABLE,INDEX,CONSTRAINT,COMMENT,
MATERIALIZED_VIEW,MATERIALIZED_VIEW_LOG,SCHEMA_CALLOUT
```

Gunakan perintah berikut untuk mengekspor metadata.

```
SQL> create directory dmpdir as <location>;
expdp system/<system password> parfile=<parameter file>
```

Ini adalah contoh file parameter dump untuk mengimpor metadata objek nonsegmen.

```
directory=dmpdir
dumpfile=xttsmsc%U.dmp
full=y
logfile=impmsc1.log
EXCLUDE=TABLESPACE, PROCOBJ, RLS_CONTEXT, RLS_GROUP, RLS_POLICY, TABLESPACE_QUOTA
metrics=y
remap_tablespace=
APPS_TS_ARCHIVE:SYSTEM,
APPS_TS_INTERFACE:SYSTEM,
APPS_TS_SEED:SYSTEM,
APPS_TS_SUMMARY:SYSTEM,
…..
```

Pada saat ini, tidak ada tablespace kecualiSYSTEM,SYSAUX,UNDO, dan TEMP pada sistem tujuan. Jadi, Anda harus memetakan ulang semua objek yang akan diimpor ke SYSTEM tablespace.

Gunakan perintah berikut untuk mengimpor metadata.

```
SQL> create directory dmpdir as <location>;
impdp system/<system password> parfile=<parameter file>
```

Sekarang Anda dapat melihat objek yang dibuat pada database tujuan.

```
SQL> select object_type, count(*) from dba_objects group by object_type order by
 count(*) desc;
OBJECT_TYPE                  COUNT(*)
------------------------------ ----------
SYNONYM                          89327
PACKAGE                           55670
PACKAGE BODY                   54447
VIEW                                 41378
```

```
JAVA CLASS                          31978
SEQUENCE                           12766
…..
```

Tidak ada ruang meja kecualiSYSTEM,, SYSAUXUNDO, danTEMP. USERObjek dibuat dengan USERS sebagai tablespace default. Selama fase 3, tablespace yang dapat diangkut akan dibuat, dan kemudian objek USER akan diubah dengan tablespaces default asli.

Fase 3 — Fase transportasi (database sumber hanya dibaca)

Selama fase ini, sistem sumber menjadi hanya baca. File data pada sistem tujuan dibuat konsisten dengan sistem sumber dengan menerapkan cadangan tambahan akhir. Kemudian, ekspor metadata objek dari sistem sumber dan impor ke sistem tujuan.

Langkah 1: Buat ruang tabel di database sumber hanya dibaca

Sebagai SYSDBA, buat semua tablespace ditransfer `READ ONLY` pada sistem sumber.

Untuk mengurangi downtime, Anda dapat menjalankan dua langkah berikut secara bersamaan.

Langkah 2: Buat cadangan inkremental akhir

Pada sistem sumber, buat cadangan tambahan akhir dari ruang tabel yang ditransfer.

```
cd /u01/oracle/expimp/xtt<nn>
export TMPDIR=/u01/oracle/expimp/out/out<nn>
$ORACLE_HOME/perl/bin/perl xttdriver.pl --backup --debug 3
```

Langkah ini mengembalikan kesalahan, seperti "ORA-20001: TABLESPACE (S) IS READONLY”; kesalahan diharapkan, dan Anda dapat dengan aman mengabaikannya.

Langkah 3: Ekspor metadata

Ekspor metadata ruang tabel yang dapat diangkut dari database sumber.

Ini adalah contoh file parameter untuk mengekspor metadata ruang tabel yang dapat diangkut.

```
directory=dmpdir
metrics=y
dumpfile=xttsmeta%U.dmp
```

```
filesize=1048576000
logfile=expxtts.log
transport_tablespaces=
APPS_TS_ARCHIVE,
APPS_TS_INTERFACE,
APPS_TS_MEDIA,
APPS_TS_NOLOGGING,
….
exclude=table_statistics,index_statistics
```

Selain itu, jika sistem sumber memiliki banyak tabel dan indeks, Anda dapat menghemat waktu selama mengekspor dengan mengecualikan statistiknya. Setelah Anda mengimpor tablespace yang dapat diangkut, impor statistik ke sistem tujuan.

Sebelum Anda menjalankan`expdp`, buat direktori database tempat file dump disimpan di sistem sumber.

```
SQL> create directory dmpdir as <location>;
expdp system/<system password> parfile=<parameter file>
```

Dua langkah berikut adalah langkah terakhir untuk ruang meja yang dapat diangkut lintas platform dengan cadangan tambahan RMAN. Langkah-langkah ini harus dilakukan secara berurutan.

Langkah 4: Transfer file dan terapkan cadangan tambahan akhir

Transfer cadangan tambahan akhir dan ekspor file dump ke sistem tujuan, konversi, dan terapkan cadangan tambahan akhir.

Gunakan Direct Connect untuk mentransfer salinan cadangan tambahan akhir dan file res.txt ke tujuan. Anda dapat menggunakan konektivitas VPN, tetapi menggunakan Direct Connect akan mengurangi downtime secara signifikan jika memiliki bandwidth yang cukup.

Untuk mengembalikan cadangan tambahan akhir, pada sistem tujuan, jalankan perintah berikut, menggunakan `--restore` opsi, untuk setiap grup tablespace.

```
cd /u01/oracle/expimp/xtt<nn>
export TMPDIR=/u01/oracle/expimp/out/out<nn>
$ORACLE_HOME/perl/bin/perl xttdriver.pl --restore --debug 3
```

Langkah 5: Impor metadata objek

Impor metadata objek ke sistem tujuan dengan menggunakan Oracle Data Pump. Jalankan perintah berikut untuk mendapatkan daftar file data untuk `transport_datafiles=` parameter pada sistem tujuan.

```
cd /u01/oracle/expimp/xtt<nn>
export TMPDIR=/u01/oracle/expimp/out/out<nn>
$ORACLE_HOME/perl/bin/perl xttdriver.pl -e
```

Setiap kali Anda menjalankan perintah sebelumnya, Anda mendapatkan `xttplugin.txt` file, yang memiliki `transport_datafiles=` parameter. Gabungkan `transport_datafiles=` dalam satu baris dari semua `xttplugin.txt` file, dan masukkan daftar file data ke dalam `transport_datafiles` argumen file parameter untuk metadata impor.

Cuplikan kode berikut menunjukkan file parameter untuk mengimpor tablespace yang dapat diangkut pada sistem tujuan.

```
directory=dmpdir
metrics=y
dumpfile=xttsmeta%U.dmp
logfile=impxtts.log
exclude=TYPE
transport_datafiles= '+EBSDATA/APPS_TS_TX_DATA_2.dbf','+EBSDATA/
APPS_TS_TX_DATA_11.dbf','+EBSDATA/APPS_TS_TX_DATA_22.dbf','+EBSDATA/
APPS_TS_TX_DATA_183.dbf','+EBSDATA/APPS_TS_TX_DATA_204.dbf','+EBSDATA/
APPS_TS_TX_DATA_219.dbf','+EBSDATA/APPS_TS_TX_DATA_227.dbf'…..
```

Sebelum menjalankan`impdp`, buat direktori database yang menunjuk ke lokasi file dump ekspor.

```
SQL> create directory dmpdir as <location>;
impdp system/<system password> parfile=<parameter file>
```

Fase 4 - Validasi data yang diangkut

Langkah 1: Periksa ruang tabel untuk korupsi

Pada langkah ini, ruang tabel yang diangkut hanya dibaca di database tujuan. Anda dapat memvalidasi apakah tablespace valid atau tidak dengan menjalankan perintah RMAN sebagai berikut.

```
sqlplus / as sysdba;
set feedback off
set pagesize 0
set heading off
spool tbs_val.sql;
select 'validate tablespace '||tablespace_name||' check
 logical;' from dba_tablespaces where tablespace_name not in
 ('SYSTEM','SYSAUX','TEMP','USERS','UNDOTBS1','UNDOTBS2','UNDOTBS3','UNDOTBS3','UNDOTBS4');
spool off;
exit;

--Remove the first and last line in the spool file, tbs_val.sql
rman target /
RMAN> @tbs_val.sql
```

Selain itu, Anda dapat memeriksa jumlah objek, seperti tabel, indeks, sinonim, tampilan, dan objek paket.

Langkah 2. Buat semua ruang meja yang diangkut dalam database tujuan read/write

```
sqlplus / as sysdba;
set feedback off
set pagesize 0
set heading off
spool tbs_rw.sql
select 'alter tablespace '||tablespace_name||' read write;' from dba_tablespaces where
 status='READ ONLY';
spool off;
exit;
--Remove the first and last line in the spool file, tbs_rw.sql
sqlplus / as sysdba;
```

```
SQL> @tbs_rw.sql
```

Kesimpulan

Dalam panduan ini, Anda mempelajari cara meminimalkan waktu henti dengan menggunakan ruang meja portabel lintas platform Oracle dan cadangan tambahan RMAN dengan,, dan Amazon untuk Lustre. AWS Snowball Edge AWS Direct Connect FSx

Saat memigrasikan database Oracle dari tempat ke lokasi AWS, pertimbangkan variabel berikut:

- Bandwidth jaringan yang dibutuhkan oleh Direct Connect
- Kapasitas FSx untuk Lustre
- Ukuran metadata dari database Oracle
- Ukuran cadangan tambahan

Dengan menggunakan metode yang direkomendasikan dalam panduan ini, saat bermigrasi dari database Oracle 100 TB yang berjalan pada sistem Unix ke AWS, Anda dapat mengurangi waktu henti menjadi sekitar 10 jam.

Riwayat dokumen

tabel berikut menjelaskan perubahan signifikan pada panduan ini. Jika Anda ingin diberi tahu tentang pembaruan masa depan, Anda dapat berlangganan umpan RSS.

Perubahan	Deskripsi	Tanggal
–	Berubah menyebutkan heterogen dalam Pendahuluan dan Tinjauan Umum.	14 Juli 2021
–	Publikasi awal	2 Maret 2021

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.