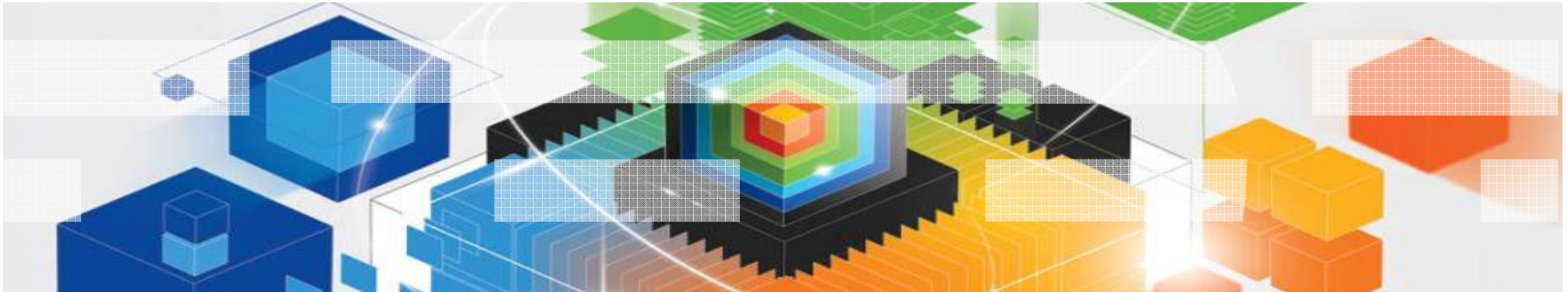


Data Migration Customer Experiences with Linux on System z



David Simpson - Oracle Technical Specialist, IBM (simpson.dave@us.ibm.com)



Copyright and Trademark Information



- For IBM – can be found at <http://www.ibm.com/legal/us/en/copytrade.shtml>
- For Oracle – can be found at <http://www.oracle.com/us/legal/index.html>
- Any performance results/observations in this presentation are purely for education and planning purposes. No Test results should be construed as indicative of any particular customer workload or benchmark result.

Oracle Database Standard Edition 2 (SE2) starting with V12.1.0.2



Is Oracle DB SE2 the right solution for your Oracle database requirements?

- Reference Oracle's website <https://www.oracle.com/database/standard-edition-two/index.html>
 - Reference the [Oracle Feature Availability by Edition](#) whitepaper for details on available features
- SE2 is licensed by the CPU socket, not per core as Enterprise Edition*
- Oracle counts each CPU chip with an activated core as a socket
 - The number of sockets/chips eligible is two
 - If any core is activated on a chip, it requires a SE2 socket license

IBM z13s Model N10 & Rockhopper Model L10 have 2 processor chips meeting the SE2 2 socket limitation

- The IBM z13s Model N20 & Rockhopper Model L20 have a minimum of 4 processor chips and do not meet the SE2 2 socket limitation
- If the z13s N10 or Rockhopper L10 models are upgraded to the z13s N20 or Rockhopper L20 models they will no longer comply to the 2 socket limitation

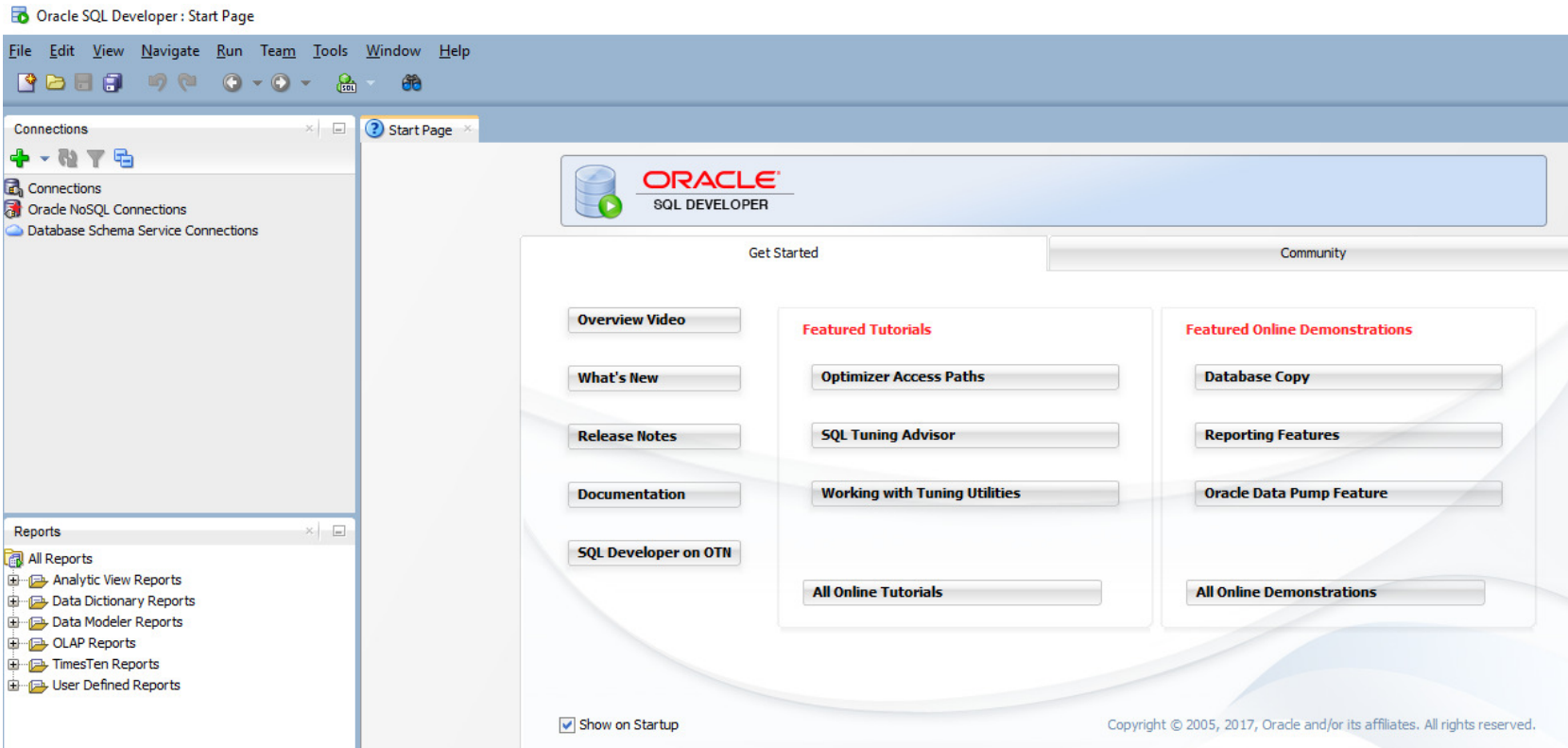
When RAC feature is used with SE2, eligibility is determined by the total number of chips in the cluster, not the individual system and limited to a maximum of two chips (sockets)

- Running RAC across multiple LPARs on the same machine is allowed, but not allowed across multiple physical machines (CECs)
 - The number of chips across the LPARS on a single server would not exceed two
 - The total number of chips across multiple physical machines would exceed two

Migrating To Oracle with Oracle SQL Developer



- SQL Developer is a supported product, for all customers with Oracle Database Support.
- Enables you to migrate a database, including the triggers, schema objects, and stored procedures, to an Oracle 12c Database using a drag and drop process.
- Migrate from SYBASE , MSSQL to Oracle 12c



Migration Approaches

Upgrade Preparation / Pre-requisites

Oracle Data Pump



Transportable Database

Transportable Tablespace

RMAN Backupset Migration

Replication (GoldenGate/IBM Migration Factory (Xen))

Start Planning for Oracle 12.1.0.2/12.2.0.1 Upgrade – preupgrd.sql

  **Patch 24732075: DATABASE PATCH SET UPDATE 11.2.0.4.170418**

Last Updated	Apr 18, 2017 2:08 PM (1+ month ago)	Size	105.9 MB
Product	Oracle Database - Enterprise Edition (More...)	Download Access	Software
Release	Oracle 11.2.0.4.0	Classification	Security
Platform	IBM: Linux on System z	Patch Tag	All Database

Recommendations / Certifications

Recommended for Oracle Database 11.2.0.4.0

- Latest PSU included are the latest **preupgrd.sql** with **utluppkg.sql** scripts for 12c upgrades!
- Pre-Upgrade utility provides a list of items which should be reviewed prior to the actual upgrade.
- **4 years newer than the original version you get with a fresh install of Oracle 12.1.0.2!**
- **MOS: How to Download and Run Oracle's Database Pre-Upgrade Utility (Doc ID 884522.1)**
 - Shared_pool_size has gone from 472M to 660M for non-CDB.
 - Memory_target algorithm more robust
 - added pga_aggregate_target to its sizing.
 - Container Database sizing also included.

Oracle Data pump **Export** Recommendations



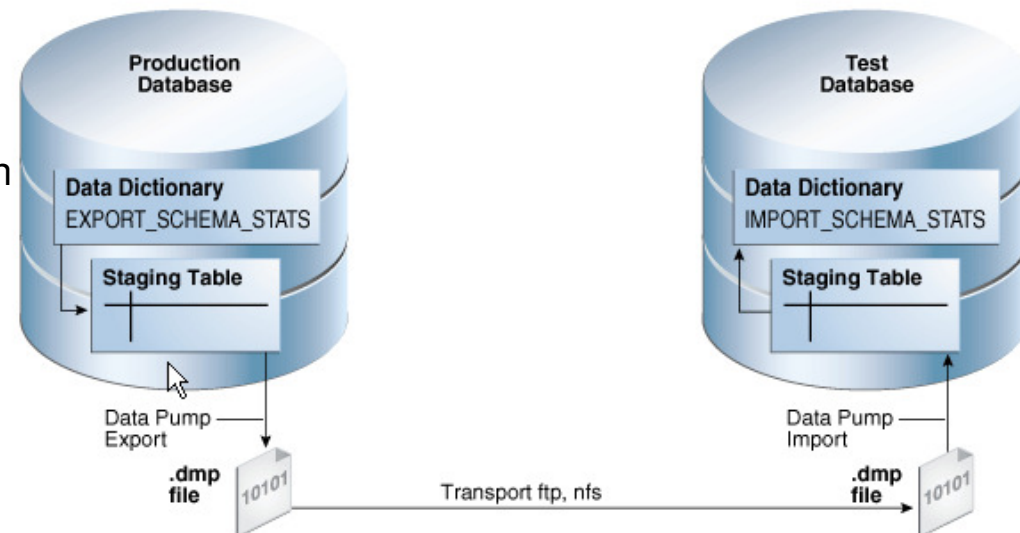
- Exclude statistics on Export

`expdp userid=xxx/xxx full=y EXCLUDE=STATISTICS directory=DBDIR dumpfile=expApr15_%U.dmp parallel=4 logfile=ex_Apr15.log`

- If using Transportable Tablespaces – you need to use **EXCLUDE=TABLE_STATISTICS,INDEX_STATISTICS**

- Consider exporting statistics to a table then migrating the statistics table, to speed up migration time.

- Create staging table in source database with **DBMS_STATS.CREATE_STAT_TABLE**
- Export local stats into staging table using **DBMS_STATS.EXPORT_SCHEMA_STATS**
- Export the staging table and import it into your destination database with Data Pump
- Import the statistics held in the staging table by using **DBMS_STATS.IMPORT_SCHEMA_STATS**



Oracle Data pump **Import** Recommendations



- Datapump import can take a really long time, when importing STATISTICS consuming large amount of TEMP (sorting) tablespace.
 - If Datapump Statistics were not excluded on export you can try and exclude on import with:
EXCLUDE=TABLE_STATISTICS,INDEX_STATISTICS
 - *EXCLUDE=STATISTICS Or EXCLUDE=INDEX_STATISTICS During Datapump Import Still Analyzes The Indexes (MOS ID: 793585.1) Use **alter system set "_optimizer_compute_index_stats" =FALSE;***
- Large Package Bodies Bug **16018315** (fixed in 11.2.0.4, 12.1.0.2)
 - 11.2.0.3: **22179 seconds**
 - 11.2.0.4/12.1.0.2: **2810 seconds**
- Watch parallelization parameters when moving from a system with many CPU cores to fewer.

Hat Tip: <http://www.oraclebuffer.com/oracle/why-datapump-import-impdp-is-not-ignoring-index-statistics/>

Import Performance

- Apply patch for bug **21539301**

Old behavior: build one index at a time with PARALLEL n keyword

New behavior: build n indexes at a time, each without parallelism

12.1.0.2.0 without patch:

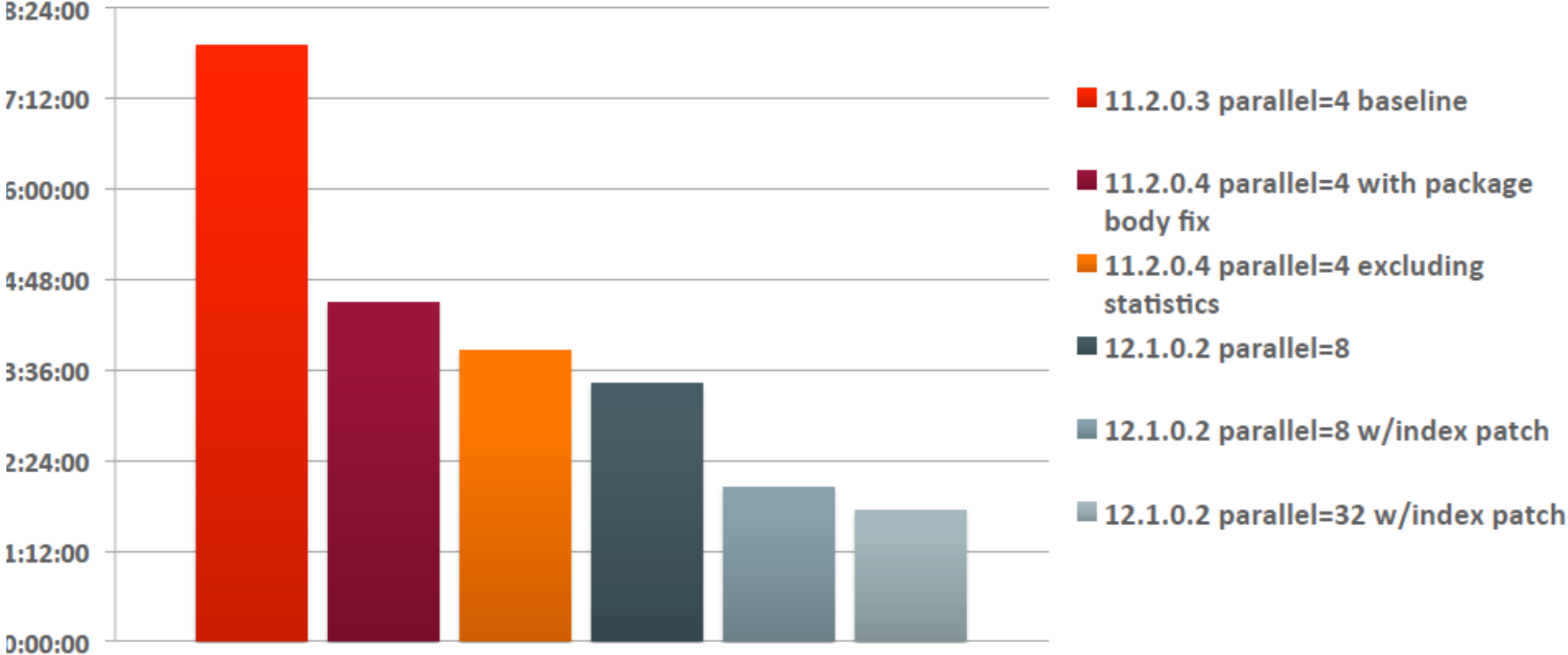
- PARALLEL=4: **5681** seconds (total import 4:14:23)

12.1.0.2.3 with patch:

- PARALLEL=4: **1308** seconds (total import 2:38:29)
- PARALLEL=8: **719** seconds (total import 2:03:11)
- PARALLEL=32: **347** seconds (total import 1:44:41)



Import Time Improves, Step by Step



DataPump Migration with a DB Upgrade

- New feature in 12c: `impdp ... transform=disable_archive_logging:y` (disables logging for import)
 - Apply patch for bug **20778442**
- **If wanting to upgrade a database to 12c from 11.2.0.4 use the VERSION=12 export Parameter**

Overview of Data Pump dumpfile set file versions.

```
Data Pump file version.
=====
```

Version Data Pump Dumpfile Set	Written by database with compatibility	Can be imported into Target:					
		10gR1 10.1.0.x	10gR2 10.2.0.x	11gR1 11.1.0.x	11gR2 11.2.0.x	12cR1 12.1.0.x	12cR2 12.2.0.x
0.1	10.1.x	supported	supported	supported	supported	supported	supported
1.1	10.2.x	no	supported	supported	supported	supported	supported
2.1	11.1.x	no	no	supported	supported	supported	supported
3.1	11.2.x	no	no	no	supported	supported	supported
4.1	12.1.x	no	no	no	no	supported	supported
5.1	12.2.x	no	no	no	no	no	supported

- **Cross-Platform Database Migration (across same endian) using RMAN Transportable Database (Doc ID 1401921.1)**
- Use this option, if downtime is minimal and if your migrating a database with the same Endian (binary) format.
- Only datafiles that contain undo data require conversion including all datafiles belonging to SYSTEM tablespace and all UNDO tablespaces.
- Can mount ASM LUNs of the same endian format temporarily in order to use RMAN Copy utility to the target (new) SAN Luns/storage.
- Can be used to Upgrade a database, but sure to use **startup upgrade** option or else you might have to redo the RMAN copy step.
- Using SAN snapshots you can repeat & test this process multiple times.

- Discovery of SAN LUNs – took us a bit longer than expected...
 - Multipath.conf setup with new LUNs were excluded.
 - Require explicitly to be added to multipath.conf and then restarted.
 - Source system LUNs were defined various partition offsets for alignment reasons.
 - For migration purposes, modify the `/etc/udev/rules.d/99-oracle-udev.rules` file to account for all partitions.
 - Oracle REDO logs created with 512 byte sector did not work with zLinux (use open resetlogs with 4096 sector size).
 - ALTER SYSTEM SET “`_DISK_SECTOR_SIZE_OVERRIDE`”=“TRUE”; was needed with Linux on System z to mount 512 byte sector ASM disk group from Solaris.

Large Endian (Source) to Linux on System z Steps (1):



Large Endian Source system Oracle 11.2.0.4 -> 12.1.0.2 Pre-DB Upgrade Steps:

1) Download from Oracle support "How to Download and Run Oracle's Database Pre-Upgrade Utility (Doc ID 884522.1)"

preupgrade_12.1.0.2.0_15_lf.zip(12.1.0.2)

2) Unzip and run the `preupgrd.sql` script and review the `preupgrade.log` for any issues such as size of tablespaces, or Oracle parameters

sqlplus / as sysdba @preupgrd.sql

3) Review and run any recommendations at a suitable time period e.g.

EXECUTE dbms_stats.gather_dictionary_stats;

purge dba_recyclebin;

4) Run pre-migration check scripts:

e.g. select count(1) from dba_objects where status='INVALID';

5) Verify Time Zone level (Doc ID 1585343.1)

SQL> select * from v\$timezone_file;

Large Endian (Source) to Linux on System z Steps (2):



1) Backup control file to trace:

```
alter database backup control file to trace as '/tmp/control.sql' resetlogs;
```

2) Create a server parameter file from the spfile and copy the parameter file to zLinux system with
copy command:

```
create pfile='initpfile_SOURCE.ora' from spfile;
```

3) Verify files requiring conversion:

```
spool /tmp/convert.lst  
set pagesize 9999  
set linesize 132  
select FILE_NAME "Datafiles requiring Conversion" from DBA_DATA_FILES  
where TABLESPACE_NAME in (select distinct TABLESPACE_NAME from DBA_ROLLBACK_SEGS);
```

4) Gather data file names requiring conversion:

```
spool /tmp/noconvert.lst  
set pagesize 9999  
set linesize 132  
column file_name format a70  
select FILE_NAME "Datafiles NO Conversion" from DBA_DATA_FILES  
where TABLESPACE_NAME NOT in (select distinct TABLESPACE_NAME from DBA_ROLLBACK_SEGS);
```

Large Endian (Source) to Linux on System z Steps (2):



5) Check for any external datapump directories that require setup on target system:

```
spool /tmp/datapump.lst
column directory_path format a50
column directory_name format a25
set linesize 200
set pagesize 9999
set serveroutput on;

declare x boolean; begin x := dbms_tdb.check_external; end;
/
select directory_name, directory_path from dba_directories;
```

6) Review disk space requirements (As Oracle ASM user):

```
spool /tmp/asmdisk.lst
set linesize 200
set pagesize 9999
column name format a25
select name, total_mb, free_mb, state from v$asm_diskgroup;
```

7) Copy any DB/Instance specific files to target server

tnsnames.ora, listener.ora, sqlnet.ora, /tmp/convert.lst, /tmp/noconvert.lst, /tmp/datapump.lst

Copy controlfile (/tmp/upgrade/control.sql) to Linux on z system /tmp

Copy \$ORACLE_HOME/dbs/initpfile_SOURCE.ora (Source) to \$ORACLE_HOME/dbs/initTARGET.ora (Linux on z)

Target System Setup Steps:

Target Linux on System z - 11.2.0.4 -> 12.1.0.2 DB Pre-Migration Steps:

1) Install and Patch (latest PSU) Oracle 12.1.0.2 Grid & DB code on zLinux box:

Configure ASM with +DATANEW and +FRA ASM disk groups on new storage.

2) Make required changes to in \$ORACLE_HOME/dbs/initpfile_TARGET.ora:

- control_files (comment out)
- any lines that require +ASM diskgroup changes e.g. log_archive_dest, db_file_create_dest
- comment out any Underscore Oracle parameters for Upgrade (
_enable_space_preallocation = 0
_trace_files_public = FALSE
- create any required file system directories (diag_dest, audit directories)

3) Modify /tmp/control.sql (control file):

- add to line: STARTUP NOMOUNT pfile='initpfile_TARGET.ora
- comment out ALTER DATABASE OPEN RESETLOGS; and ALTER TABLESPACE TEMP ADD TEMPFILE lines
- Modify any ASM diskgroup names changes
- Change REDO Log BLOCKSIZE to 4096 e.g. -> LOGFILE GROUP 1... SIZE 512M BLOCKSIZE 4096,

4) Ensure any ASM diskgroups are pre-created on target system with sufficient Disk space: **select name, total_mb, free_mb, state from v\$asm_diskgroup;**

Target System Setup Steps (2):

5) Create any DBA_DIRECTORIES (diagnostic directories as well) from source pre-migration step at the Oracle file system level in Linux with correct permissions:

select directory_name, directory_path from dba_directories; (output from Source)

6) Modify configure sql*net files - tnsnames.ora/sqlnet.ora/ etc:

7) pre-create SQL scripts in target migration section (steps 6 & 7)

```
CONVERT FROM PLATFORM 'Solaris[tm] OE (64-bit)' PARALLELISM 2
DATAFILE '+DATA/TEST/DATAFILE/SYSTEM.xxx.xxx' FORMAT '+DATA02'
DATAFILE '+DATA/TEST/DATAFILE/UNDOTBS1.xxx.xxx' FORMAT '+DATA02';
```

(get name of all datafiles other than SYSTEM & UNDO from v\$datafile in /tmp/noconvert.lst file)

```
CONVERT PARALLELISM 4
DATAFILE '+DATA/TEST/DATAFILE/USERS.xxx.xxx' FORMAT '+DATA02';
```

Target System Setup Steps (3):

- 8) Linux SA - configure `/etc/multipath.conf` for temporary source system Luns
- ensure multipath alias names are also configured in multipath.conf file

- 9) Ensure `/etc/udev/rules.d/12-dm-permissions.rules` is updated with New LUNs

- 10) Ensure `/etc/fcp.conf` is updated with BCV / FlashCopy LUN ids. (Filipe's script)

- 11) run `multipath -ll` command to verify new LUNs are discovered and run:
ls -la /dev/mapper/<new lun name> (get dm-XX name / number)
ls -lato /dev/dm-XX (verify Oracle disk permissions)

Source System Downtime- time required to sync SAN LUNs (1)

1) Shutdown application servers, shutdown any GoldenGate replication processes, then shutdown database and run transportable DB check:

```
shutdown immediate;  
startup mount;  
alter database open read only;  
DECLARE db_ready BOOLEAN;  
BEGIN  
    db_ready := DBMS_TDB.CHECK_DB('IBM zSeries Based Linux', DBMS_TDB.SKIP_NONE);  
END;  
/
```

2) shutdown database for migration sync (FlashCopy) of Production SAN Storage LUNS:

```
SQL> shutdown immediate;
```

3) Shutdown all CRS services OR shutdown ASM instance:

```
set ORACLE_HOME & ORACLE_SID to ASM instance  
sqlplus / as sysasm  
shutdown immediate;
```

4) Synchronize SAN backup disks with FlashCopy

Target System Downtime- (1)

1) Verify DB SAN Resync is complete

2) As Oracle ASM user mount source system ASM LUNs (read-only):

```
sqlplus / as sysasm  
alter diskgroup DATA mount;
```

3) As DB user, startup database NOMOUNT for modified parameter file:

```
sqlplus / as sysdba  
startup nomount pfile='initpfile_target.ora'  
(ensure initpfile_test.ora control_files line is commented out )
```

4) Run conversion of system and UNDO tablespaces to new ASM diskgroups. Run script 6.sql:

```
CONVERT FROM PLATFORM 'Solaris[tm] OE (64-bit)' PARALLELISM 2  
DATAFILE '+DATA/system01.dbf' FORMAT '+DATA02'  
DATAFILE '+DATA/undotbs01.dbf' FORMAT '+DATA02';
```

```
rman target / @6.sql
```

5) Copy the remainder of the ASM datafiles to the target ASM diskgroup (will take time)

```
rman target / @7.sql
```

6) ensure the new Oracle initpfile_TARGET.ora file points to the new zLinux ASM diskgroups:

Target System Downtime- (2)

7) Modify/Update the control file, with newly created file names. The following script can be used to update the **/tmp/control.sql** script:

```

set pagesize 999
set linesize 132
column full_alias_path format a70
select concat('+'||gname, sys_connect_by_path(aname, '/')) full_alias_path
from ( select b.name gname, a.parent_index pindex, a.name aname,
      a.reference_index rindex , a.system_created, a.alias_directory,
      c.type file_type
      from v$asm_alias a, v$asm_diskgroup b, v$asm_file c
      where a.group_number = b.group_number
            and a.group_number = c.group_number(+)
            and a.file_number = c.file_number(+)
            and a.file_incarnation = c.incarnation(+)
      )
start with (mod(pindex, power(2, 24))) = 0
and rindex in
( select a.reference_index
  from v$asm_alias a, v$asm_diskgroup b
  where a.group_number = b.group_number
        and (mod(a.parent_index, power(2, 24))) = 0
        and a.name = '&DATABASENAME'
  )
connect by prior rindex = pindex;

```

Recreate the Control File using the updated **/tmp/control.sql** script.

8) Open database with resetlogs and upgrade option
alter database open resetlogs upgrade;

Target System Downtime- (3)

9) Recreate the TEMP tablespace (can get commands from **/tmp/control.sql**)
create script 11.sql:

```
ALTER TABLESPACE TEMP ADD TEMPFILE .... AUTOEXTEND OFF;  
@11.sql
```

10) review oracle alert.logs for any errors:
tail -f <insert path to oracle alert log>

Target System Upgrade steps - 11.2.0.4 -> 12.1.0.2 (1):

1) Ensure DB is ALWAYS restarted with "**alter database open upgrade**" until DB is fully upgraded to 12.1.0.2

2) Run the new Parallel Upgrade Utility (catctl.pl) once system is started in UPGRADE mode:

```
export ORACLE_HOME=/opt/app/oracle/product/12.1.0.2/db_1
```

```
mkdir $HOME/upgrade
```

```
cd $ORACLE_HOME/rdbms/admin
```

```
export PATH=$ORACLE_HOME/bin:$PATH
```

```
nohup $ORACLE_HOME/perl/bin/perl catctl.pl -n 8 -l $HOME/upgrade catupgrd.sql &
```


Target System Upgrade steps - 11.2.0.4 -> 12.1.0.2 (2):

3) Startup upgraded database in UPGRADE mode (again), run the "utlu121s.sql" script to check the summary of the upgrade results:

```
cd $ORACLE_HOME/rdbms/admin
```

```
sqlplus / as sysdba
startup mount pfile='initpfile_TARGET.ora';
alter database open upgrade;
SQL> @utlu121s.sql
```

4) The "catuppst.sql" script would have been run as part of the upgrade. If there are ANY errors, in the Upgrade that we're fixed, we need to run it manually:

```
SQL> @catuppst.sql
```

5) If the "postupgrade_fixups.sql" file contained any recommendations, run it now:

```
SQL> @$ORACLE_HOME/cfgtoollogs/orcl/preupgrade/postupgrade_fixups.sql
```

6) The following items are not essential, but the upgrade manual suggests it is a best practice to run them. The following item is probably included in postupgrade_fixups.sql script:

```
EXECUTE DBMS_STATS.gather_fixed_objects_stats;
```

Target System Upgrade steps - 11.2.0.4 -> 12.1.0.2 (3):

7) Recompile invalid objects:

@utlrp.sql & @utluiobj.sql -- Check for newly invalid objects.

8) Run again to check the final outcome of the upgrade:

@utlu121s.sql

9) Verify system checks:

select count(1) from dba_objects where status=='INVALID';

10) Verify Oracle PSU's are applied to upgraded databases:

select * from dba_registry_sqlpatch;

11) Re-enable any Underscore Oracle parameters that are needed, create spfile from pfile.

Startup and restart the database from the spfile:

- **show parameter control_files / select name from v\$controlfile;**
- **make any necessary changes to pfile, update control_files, re-enable underscore parameters**
- **create spfile='+DATANEW' from pfile='initpfile_TARGET.ora'**

Target System Upgrade steps - 11.2.0.4 -> 12.1.0.2 (4):



12) If manual upgrade for timezone needed Scripts to automatically update the RDBMS DST (timezone) version in an 11gR2 or 12cR1 database . (Doc ID 1585343.1):

```
SQL> select * from v$timezone_file;
```

13) Re-configure Golden Fate Disk replication processes (if applicable)

14) Re-point application servers to new DB server (tnsnames.ora) & Restart/Test applications.

Transportable Tablespace:

- Transportable tablespace is similar migration technique to Transportable Database, but used for migrations to between platforms with a different Endian format. (e.g small to Big)
- Similar to Transportable database to help limit downtime.
- Uses Data Pump to migrate the meta data / schema objects (triggers, procedures, views etc)
- With 12c, you can now use full transportable export/import to move complete databases for sources that are 11.2.0.3 and up to 12c targets. (**Upgrading a Database Using Transportable Tablespaces (Doc ID 1457743.1)** i.e. including the metadata
 - Only for source databases with RDBMS version 11.2.0.3+
 - separate copy of the datafiles is needed (space consideration)

RMAN Backup Set Migration:

- Cross-platform migration of 11.2.0.4 database to 12c on Linux on System z using RMAN convert database. Oracle Doc: https://docs.oracle.com/cd/B28359_01/backup.111/b28270/rcmxplat.htm#BRADV89986
- Cross-platform migration of 12c database to 12c on Linux on System z using RMAN compressed backup set. Oracle Doc: <https://docs.oracle.com/database/121/BRADV/rcmxplat.htm#BRADV724>

Source System:

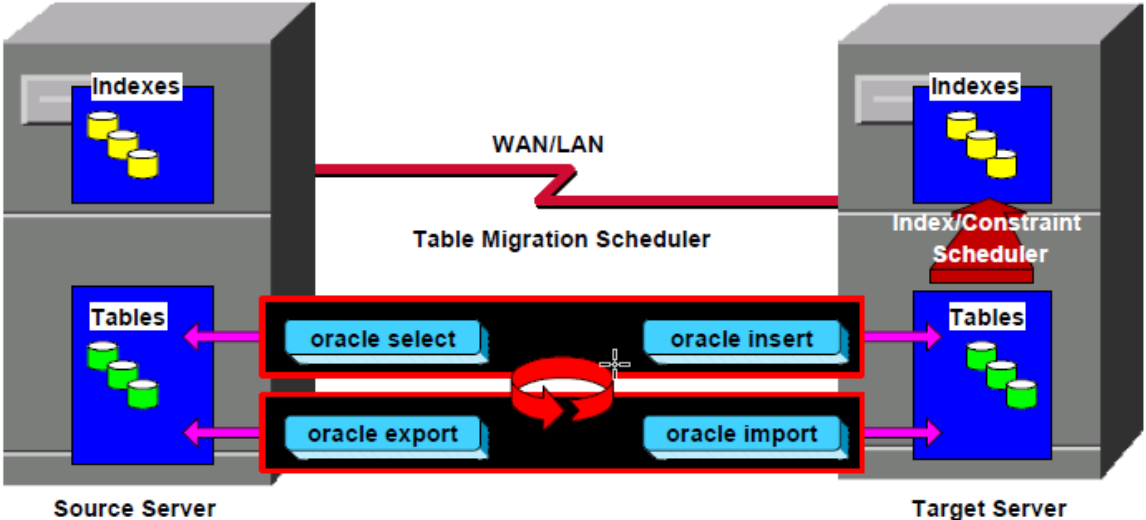
1. Ran pre-upgrade script (Removed EM and Apex)
2. alter database open read only
3. Ran dbms_tdb.check_db('IBM zSeries Based Linux',dbms_tdb.skip_readonly);
4. Ran dbms_tdb.check_external;
5. RMAN> convert database new database 'migbd' transport script '/ggate/11to12/transport' to platform 'IBM zSeries Based Linux' format '/u01/11to12/db_%d_%s_%U.bkp';
6. back up controlfile

Target System (z System):

1. Startup nomount pfile='/u01/db/dbs/initmigbd.ora';
2. RMAN > restore controlfile from '/ggate/11to12/2erp8uo1_1_1';
3. RMAN > alter database mount;
4. RMAN > catalog start with '/ggate/11to12/' ; where unzipped backup files reside
5. RMAN > restore database;
6. alter database open resetlogs upgrade; (from sqlplus)
7. cd \$ORACLE_HOME/rdbms/admin
\$ORACLE_HOME/perl/bin/perl catctl.pl -n 8 catupgrd.sql
8. run postupgrade_fixups.sql
9. run utltp.sql
10. run utluiobj.sql
11. run utlu121s.sql

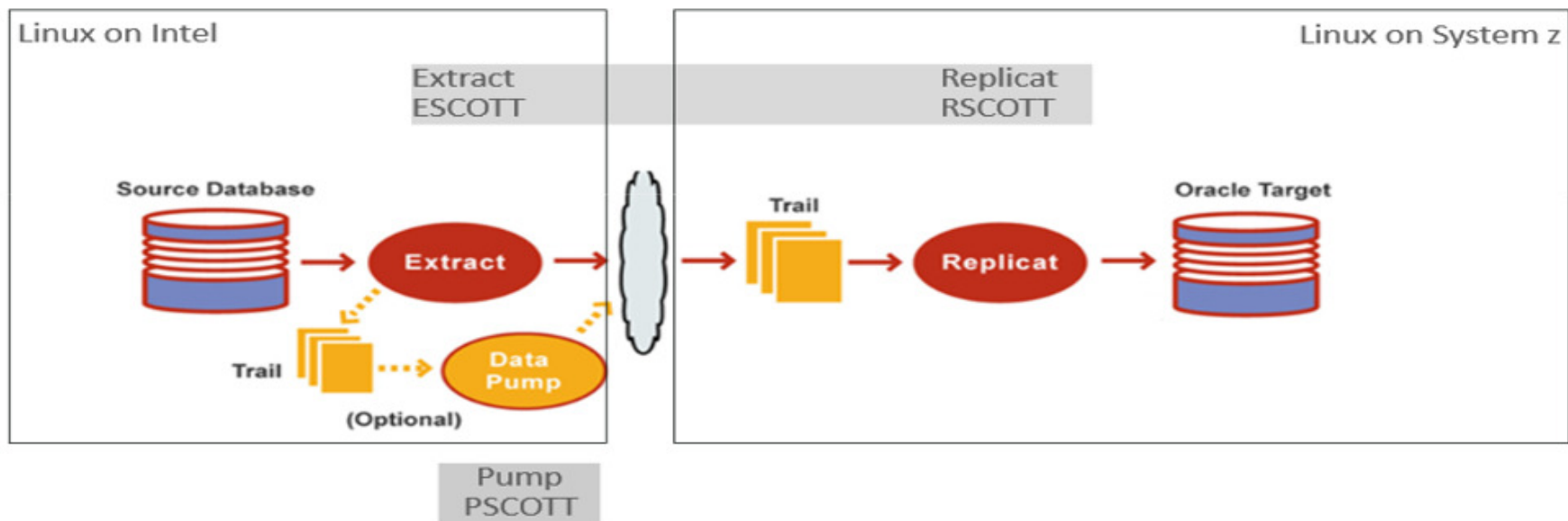
Using Replication Tools for Migration:

- Several tools available such as Oracle Golden Gate, IBM Infosphere CDC and IBM Migration Factory (Xenobridge).



Oracle Golden Gate (Replication Tool):

- Oracle GoldenGate can help migrate across platforms, different DB Version and Types
- Some additional overhead (cpu) on source and target
- Licensed product from Oracle
- CLOB & BLOB
- Thomas Niewel of Oracle – great presentation on SIG website



Source: http://www.oraclezsig.org/2015presentations/WED_1345_2015_Niewel_Migrating_to_LOZ.pdf © 2017 IBM Corporation

Oracle Migration Approaches To Linux on System z



Migration Technique	Pro's	Con's
Oracle Data Pump	<ul style="list-style-type: none"> • Simple to use • Can be used to upgrade to 12c 	<ul style="list-style-type: none"> • Temporary disk space on source and target + transferring files. • Downtime to rebuild DB
Transportable Database	<ul style="list-style-type: none"> • Copy Datafiles, UNDO tablespaces only require a conversion. • Faster than Data Pump/tablespace • Possible to mount DB Luns on Target 	<ul style="list-style-type: none"> • Source and Target Systems must be same endian.
Transportable Tablespace	<ul style="list-style-type: none"> • Requires RMAN conversion of database files. 	<ul style="list-style-type: none"> • Must use Data Pump for schema objects • Requires Temporary Disk Space on source and Target
RMAN Backupset Migration	<ul style="list-style-type: none"> • Simple to create an RMAN backupset • RMAN incremental (12c+) can be used to keep source & target in sync. 	<ul style="list-style-type: none"> • Time to run backup, copy files and restore • Temporary disk space on source/target
Replication (GoldenGate/IBM Migration Factory)	<ul style="list-style-type: none"> • Good for Zero downtime migration 	<ul style="list-style-type: none"> • additional software software costs • Oracle BLOBs/CLOBs can sometimes have issues. (Verify with Oracle SR)

Questions?

