

Faster Cross-Platform Database Migration with RMAN

John Larkin

JP Morgan Chase

Who is John Larkin

- Originally a mainframe COBOL programmer
- DBA for the last 23 years, the last 15 with Oracle.
 - UNIX (Solaris, Aix, Windows, Linux)
- Recently completed a year-long project to migrate 88 databases from AIX and Solaris servers to a Veritas-clustered Solaris environment.
- Employed by chemical manufacturers, publishers, retailers as both employee and independent contractor.
- 10g OCA
- Contact – john.x1.larkin@chase.com / john.larkin1@comcast.net

Cross-Platform Migration

- DATAPUMP or EXPORT and IMPORT
 - Time consuming
- Cold backup
 - For same OS
- RMAN
 - Automated
 - » Less chance for errors
 - Parallelizable
 - » Conversion time savings of up to 88%

Cross-Platform Migration

- Some challenges
- May only need to convert a fraction of all the datafiles in the database.
- Testing
 - » serial runtime - 150 min
 - » parallelized runtime - 18 min.
 - » worst case scenario

Cross-Platform Migration

- Year-long technology refresh project
 - migrate and upgrade 88 Oracle 10g databases
 - 6 standalone Solaris / AIX servers
 - Veritas-clustered environment with 10 Solaris servers
- Review RMAN behavior and performance
- Main goal for migration
 - balance the downtime
 - assurance of a successful migration.
 - Use iterative approach
 - modify and improve plan as we progress.

The PROJECT

- Migrate 88 databases to common Solaris 10 env.
- Disparate operating systems (50/50 split)
 - Solaris 8
 - AIX 5.3
- Databases running Oracle Enterprise 10.2.0.2
 - a few at 10.1.0.4 and 9.2.0.6
 - Nine with a manual dataguard solution for DR.
- Map databases to new cluster
 - attempt to balance relative load
 - related databases on the same node
- Disk shared
- IP / DNS – 1 instance per IP

The PLAN

- 2 different approaches for migration
- 2 main migration alternatives from each OS
 - with some minor variations
 - Not necessarily inconsequential

Solaris to Solaris Migrations

- Simplest options “Same OS conversion”
- EXPORT / IMPORT is always an option
 - Time an issue
 - No incremental EXP/IMP to minimize complexity/risk
 - Staggering migrations of related databases
 - » Increase overall outage
- FTP / NFS mounts
 - Time an issue
 - Used for development databases (smaller)

Solaris to Solaris Migrations

■ Physical Standby Database

- Create on target server
- Migration complete by a flip of the switch

■ Activate vs Switchover

- Switchover - allow for easier fall back to the original database
- Issue - standby controlfiles – recreate

AIX to Solaris Migrations

- more complex - “Different OS conversion”
- EXPORT / IMPORT always an option
 - Resolved the cross-platform issue
 - Resource contention
- FTP cold database
 - File conversion needed – partially correct
 - » Missed opportunity
- RMAN Cross-Platform Transportable Database (TDB)
 - Datafile conversions
 - Parallelizable
 - Ease (maybe) and speed
 - » Simplicity – sort of...
 - » Speed – definitely

RMAN Conversion notes

- Was not my favorite tool
 - What a difference a year makes.
- RMAN – track backup and recovery information
 - RMAN repository
 - Recovery Catalog
 - Store settings
 - target database – the database RMAN will act upon.
- **rman target /**
 - **catalog=**

RMAN Conversion notes

- **CONFIGURE DEVICE TYPE DISK
PARALLELISM 6 BACKUP TYPE TO
BACKUPSET;**
- **PARALLELISM** parameter - **CONVERT
DATABASE** stmt
- Conversion type – **SOURCE** or **TARGET**

SOURCE CONVERSION

```
run
{
  CONVERT DATABASE
  new database      'oradb1p'
  TRANSPORT script
    'oradb1p_transportscript.sql'
  to platform      'Solaris[tm] OE (64-bit)'
  db_file_name_convert  '/app/oraf' ,
    '/dbmig/oradb1p_migr' ;
}
```

SOURCE CONVERSION cont'd

- on Source server this produces:
- a TRANSPORT script
- a set of CONVERTED data files.
- Specific instructions to complete the migration

SOURCE CONVERSION cont'd

- The MESSAGES:
- Run SQL script oradb1P_transportscrip.sql on the target
 - Creates database
- Edit init.ora \$OMF_NAME.ora
 - Modify to create new database
 - Recommend using original init.ora and apply changes
- Recompile all PL/SQL modules
 - utlirp.sql and utlrp.sql on the target platform
- Change the internal database identifier if desired
 - DBNEWID Utility.

TARGET CONVERSION

- run
- {
- CONVERT DATABASE
- new database 'oradb1p'
- → ON TARGET PLATFORM ¹
- → CONVERT script 'oradb1p_convertscript.sql' ²
- TRANSPORT script 'oradb1p_transportscript.sql'
- to platform 'Solaris[tm] OE (64-bit)'
- db_file_name_convert
- '/app/oraf/u000', '/app/oraf/u031/tempxfer/u000',
- '/app/oraf', '/app/orax'; }

TARGET CONVERSION cont'd

- on the SOURCE server this produces :
 - a TRANSPORT script
 - a CONVERT script
- ** NO converted datafiles will be created.**

TARGET CONVERSION cont'd

- The following messages from RMAN (source):
- Run SQL script oradb1p_transportscript.sql on the target platform
- Edit init.ora \$OMF_NAME.ora
- Run RMAN script oradb1p_convertscript.sql on target platform to convert datafiles ³
- Run utlirp.sql and utlrp.sql on the target platform
- Change the internal database identifier - DBNEWID
 - 1. TARGET conversion is to be performed.
 - 2. The RMAN script to run on target server to complete the conversion.
 - 3. The extra step required when you perform a TARGET conversion.

TARGET CONVERSION cont'd

- -- Notes
- db_file_name_convert - order matters
- need db_file_name_convert or format
 - \$ORACLE_HOME/dbs/omf_names
- must be some difference between the FROM and TO mappings or you will get OMF names.

Initial Setup

- backup controlfile from the source database
 - for the RMAN conversion.
- single temporary filesystem to store the converted files before moving to the destination server
- directory for each database being converted and
 - Sub directories for mount points
 - maintain the mount point groupings from the source database

The Cross-Platform Migration Process

■ Source conversion

- Initial development database conversions
- The basic steps for conversion :
- Get a location for the READ ONLY data files.
- Build directories for the files (either 1 big dir or 1 for 1).
- Set the AUDIT trail parameter to none
- Disable Standby database log shipping
 - » log_archive_dest_state_2 defer
- Get file sizes by mountpoint

» FILE_NAME	FILES	MAXBYTES
» -----		
» /app/oraf/u006	8	12,280,922,112

- Restart the database in READONLY mode

Cross-Platform Migration Process cont.

- prerequisite checks - verify the db is transportable
- What platforms are valid for TDB migration
 - select platform_name from v\$db_transportable_platform ;
 - Solaris[tm] OE (32-bit)
 - Solaris[tm] OE (64-bit)
 - AIX-Based Systems (64-bit)
- Do any files need conversion (cheat)
 - select distinct(file_name)
 - from dba_data_files a, dba_rollback_segs b
 - where a.tablespace_name = b.tablespace_name ;

Cross-Platform Migration Process cont.

- Can the database be transported to the target OS?
 - `retcode := dbms_tdb.check_db('Solaris[tm] OE (64-bit)'
,dbms_tdb.skip_none); --(want TRUE)`
 - “The following directories exist in the database:
SYS.APPL_FILE_IN, SYS.DATA_PUMP_DIR”
- Identify external tables and bfiles – manual
 - `select directory_path||'/'||location External_file_path from
dba_directories a, dba_external_locations b
where a.directory_name=b.directory_name ;`
 - Identify BFILE files that will need to be transferred
 - `@tdb_get_bfile_dirs.sql`

Cross-Platform Migration Process cont.

- Output: “The following directories contain external files for BFILE columns, copy to target..
There are 0 directories, 0 total BFILES “
- @tdb_get_bfiles.sql – lists the bfiles
- Run CONVERT DATABASE on the SOURCE server.
- Move to the TARGET server
- Get the datafiles to the TARGET server.
- Modify init.ora as needed – use original.

Cross-Platform Migration Process cont.

- Create a temporary control file / database
- Run the RMAN convert script.
- Run the SQL transport script - creates the real db.
- Run utlirp.sql and utlrip.sql to recompile all PL/SQL
- Run the NEWID utility if you needed
- Your migrated database is ready to use.

Parallelism Issues

- Appears to be relatively straight forward
- bugs encountered : SOURCE conversion
 - Bug 5016125 - RMAN convert with parallelism > 1 fails
 - » Note:417455.1 – workaround
 - Use a single channel,
 - patch to 10.2.0.4
 - Upgrade to 11g
- bugs encountered : TARGET conversion
 - Bug 5641099 - convert script created in a way that prevents the use of RMAN PARALLELISM.
 - Need multiple datafiles under 1 CONVERT DATAFILE statement for parallel execution.
 - May be a feature ? - OMF type names for LOGFILE and TEMPFILE

TRANSPORT Script

- -- create a new control file and open the database.
- STARTUP NOMOUNT
- CREATE CONTROLFILE REUSE SET DATABASE "ORADB1P"
- RESETLOGS ARCHIVELOG
- LOGFILE GROUP 1 (
■ '/app/oraf/u003/redo_logs/ORADB1P/ORADB1P.g1.m1.rdo')
- DATAFILE '/app/oraf/u033/datafiles/ORADB1P/ORADB1P.system.01.dbf'
- CHARACTER SET US7ASCII ;
- -- Db can now be opened zeroing the online logs. Add tempfiles
- ALTER DATABASE OPEN RESETLOGS;
- ALTER TABLESPACE ORACLE_TMP_SRT ADD TEMPFILE
 'xx/temp1.dbf'
- prompt * Your database has been created successfully!

- We're not done yet.

TRANSPORT Script cont'd

- SHUTDOWN IMMEDIATE
- STARTUP UPGRADE PFILE='../dbs/init_00jvr8a7_1_0.ora'
- @@ ?/rdbms/admin/utlirp.sql
- SHUTDOWN IMMEDIATE
- STARTUP PFILE='../dbs/init_00jvr8a7_1_0.ora'
- -- Recompile all PL/SQL modules.
- @@ ?/rdbms/admin/utlirp.sql

CONVERT Script

- The CONVERT script in it's native form :
- ```
RUN { CONVERT DATAFILE
 '/oracle/oraf/u517/datafiles/SID1/APP1_OPN_TBL.10.dbf'
FROM PLATFORM 'AIX-Based Systems (64-bit)'
FORMAT
 '/dbmig/temp_migr/u517/datafiles/SID1/APP1_OPN_TBL.10.dbf';
CONVERT DATAFILE
 '/oracle/orafs/u519/datafiles/SID1/APP1_OPN_TBL.11.dbf'
FROM PLATFORM 'AIX-Based Systems (64-bit)'
FORMAT
 '/dbmig/temp_migr/u519/datafiles/SID1/APP1_OPN_TBL.11.dbf';
```
- Changes needed to make the Convert script useable :
- ( Use FORMAT if using intermediate file names)
- ```
cat oradb1p_convertscript.sql | awk '{ if($1 == "FORMAT") print  }' >
oradb1p_convertscript_t2
```
- ```
vim oradb1p_convertscript_t2
```
- ```
:%s/CONVERT DATAFILE/,/
```
- ```
:%s/;/
```

# CONVERT Script cont'd

- at the top of the script add :

```
RUN
```

```
{
```

```
 CONVERT DATAFILE
```

- To the bottom of the script add :

```
FROM PLATFORM 'AIX-Based Systems (64-bit)'
```

```
db_file_name_convert
```

```
 '/app/oraf/u031/tempxfer/u000','/app/oraf/u000',
```

```
 '/app/oraf/u032/tempxfer/u033','/app/oraf/u033',
```

```
 '/app/orax', '/app/oraf';
```

```
};
```

# CONVERT Script cont'd

- Now the convert script (oradb1p\_convertscript\_t2) looks like :

```
RUN {
```

```
 CONVERT DATAFILE
```

```
 '/app/oraf/u032/tempxfer/u022/dataf/ORADB1P/ODBP.app.dbf'
```

```
 ,'/app/oraf/u031/tempxfer/u006/dataf/ORADB1P/ODBP.ora.dbf'
```

```
 ,'/app/oraf/u032/tempxfer/u033/dataf/ORADB1P/ODBP.sys.dbf'
```

```
 ,'/app/oraf/u031/tempxfer/u006/dataf/ORADB1P/ODBP.sax.dbf'
```

```
 FROM PLATFORM 'AIX-Based Systems (64-bit)'
```

```
 db_file_name_convert
```

```
 '/app/oraf/u031/tempxfer/u006','/app/oraf/u006',
```

```
 '/app/oraf/u032/tempxfer/u022','/app/oraf/u022',
```

```
 '/app/oraf/u032/tempxfer/u033','/app/oraf/u033',
```

```
 '/oracle/orafx', '/oracle/orafs';
```

```
 }
```

# CONVERT Script cont'd

- convert all datafiles 88% faster , using 6 degrees of parallelism on a Target system conversion with Oracle 10.2.0.3
- a 24 hour conversion could be cut down to less than 12 hours overall.
- Source system conversion method could offer a greater reduction in overall runtime had we been able to take the source database files directly as input to the CONVERT DATAFILE statement and route the output to mount points that were NFS mounts of the destination server filesystem.



# CONVERT Script cont'd

- A means to further reduce conversion time is described in Metalink Note 732053.1 – Avoid Datafile Conversion during Transportable Database.
- only datafiles that contain UNDO data require conversion
- any files that make up the SYSTEM and UNDO tablespaces and any other tablespaces that contain ROLLBACK segments
- ```
select distinct(file_name)
from dba_data_files a, dba_rollback_segs b
where a.tablespace_name=b.tablespace_name;
```

CONVERSION Results

- review of the output from the RMAN CONVERT DATABASE and CONVERT DATAFILE statements
- confirm the runtime improvement
- provide additional insight into the conversion process
- Recovery Manager: Release 10.2.0.2.0 - Production
connected to target database: ORADB1P(DBID=9329)
RMAN> run {
3> CONVERT DATABASE NEW DATABASE 'oradb1p'
4> ON TARGET PLATFORM
.....
46> }

CONVERSION Results cont'd

- Starting convert at 16-NOV-08
- using target database control file instead of recovery catalog
- allocated channel: ORA_DISK_1
- channel ORA_DISK_1: sid=549 devtype=DISK
- Dir SYS.DATA_PUMP_DIR, SYS.APPL1_FILE_IN found in db
- User SYS with SYSDBA and SYSOPER priv in password file
- channel ORA_DISK_1: starting to check datafiles
- input datafile fno=00078 name=ORADB1P.appl1_tbl.01.dbf
- channel ORA_DISK_1: datafile chk complete, elapsed time: 00
- channel ORA_DISK_1: starting to check datafiles
- input datafile fno=00079 name=ORADB1P.appl1_tbl.02.dbf
- channel ORA_DISK_1: datafile checking complete, elapsed time: 00:00:00
- channel ORA_DISK_1: starting to check datafiles
- input datafile fno=00106 name=ORADB1P.appl2m_opn_tbl.37.dbf
- channel ORA_DISK_1: datafile checking complete, elapsed time: 00:00:00
- Finished backup at 16-NOV-08

CONVERSION Results cont'd

- CONVERT DATABASE statement
 - 1 channel allocated
 - lists directories defined in the database
 - users in the password file.
 - all of the datafiles are checked (not converted) in less than one second each. (TARGET system conversion)
- FTP the datafiles in parallel (4 separate FTP sessions)
 - throughput of more that 2 GB / minute for each FTP session
- 150 Opening data connection for
ORADB1P.appl2m_opn_idx.02.dbf (2097160192 byte)
local: ORADB1P.appl2m_opn_idx.02.dbf remote:
ORADB1P.appl2m_opn_idx.02.dbf
2097160 kbytes recvd in 40 secs (51241.76 Kbytes/s)

CONVERSION Results cont'd

- partial log from the CONVERT DATAFILE statement on Target server :
- =>rman target / nocatalog
- Recovery Manager: Release 10.2.0.2.0 - Production on Sun Nov 16 15:36:51 connected to target database: ORADB1P (DBID=393..829)
using target database control file instead of recovery catalog
RMAN> @oradb1p_convertscript_t2.sql

continued...

CONVERSION Results cont'd

```

RMAN> RUN {
  1> CONVERT DATAFILE
71>'/app/oraf/u032/tempx/u033/dataf/ODP/ODBP.sys.dbf'
72>,'/app/oraf/u031/tempx/u006/dataf/ORDP/ODBP.sya.dbf'
83>,'/app/oraf/u031/tempx/u006/dataf/ORDP/ODBP.ord.dbf'

109> FROM PLATFORM 'AIX-Based Systems (64-bit)'

110> db_file_name_convert
117> '/app/oraf/u031/tempxfer/u006' , '/app/oraf/u006',
133> '/app/oraf/u032/tempxfer/u022' , '/app/oraf/u022',
144> '/app/oraf/u032/tempxfer/u033' , '/app/oraf/u033',
145> '/app/orax' , '/app/oraf';
146> }

```

CONVERSION Results cont'd

- Starting backup at 16-NOV-08

allocated channel: ORA_DISK_1

channel ORA_DISK_1: sid=547 devtype=DISK

allocated channel: ORA_DISK_2

channel ORA_DISK_2: sid=513 devtype=DISK

...

allocated channel: ORA_DISK_6

channel ORA_DISK_6: sid=557 devtype=DISK

continued

CONVERSION Results cont'd

```
channel ORA_DISK_1: starting datafile conversion
channel ORA_DISK_2: starting datafile conversion ....
channel ORA_DISK_6: starting datafile conversion
input filename=ORADB1P.appl1_opn_tbl.01.dbf
input filename=ORADB1P.appl1_opn_tbl.02.dbf
input filename=/app/oraf/tx/u031/u06/datf/ODB.usr.dbf
converted datafile=
/app/oraf/u006/dataf/ORDB/ORADBP.ora_usr.dbf
channel ORA_DISK_6: datafile conv compl, elapsed
time: 00:00:45
```

continued

CONVERSION Results cont'd

channel ORA_DISK_6: starting datafile conversion

input filename=ORADB1P.appl2m_opn_tbl.06.dbf

converted datafile=ORADB1P.appl1_opn_tbl.02.dbf

channel ORA_DISK_2: datafile conversion complete, elapsed time: 00:00:49

channel ORA_DISK_2: starting datafile conversion

input filename=ORADB1P.appl2m_opn_tbl.07.dbf

converted datafile=ORADB1P.appl1_opn_tbl.01.dbf

channel ORA_DISK_1: datafile conversion complete, elapsed time: 00:00:52

channel ORA_DISK_1: starting datafile conversion

converted datafile=ORADB1P.appl1_opn_tbl.03.dbf

channel ORA_DISK_3: datafile conversion complete, elapsed time: 00:00:52

channel ORA_DISK_3: starting datafile conversion

input filename=ORADB1P.appl2m_opn_tbl.08.dbf

input filename=ORADB1P.appl2m_opn_tbl.09.dbf

input filename=ORADB1P.appl2m_opn_tbl.10.dbf

converted datafile=ORADB1P.appl1_opn_tbl.05.dbf

channel ORA_DISK_5: datafile conversion complete, elapsed time: 00:00:53

Finished backup at 16-NOV-08

12/11/2008 8:55 PM

CONVERSION Results cont'd

- average throughput > 2 GB/min per channel.
- “input filename” - the first parameter passed to the db_file_name_convert parameter
- “converted datafile” - the second parameter.
- the various channels are each processing files in parallel.

ERROR Messages

- Missing a destination directory on a mount point.- The message is fairly explicit and it didn't take long to find and fix the error :
- input filename=ORADB1IP.appl2m_opn_tbl.15.dbf
RMAN-00569: ==ERROR MESSAGE STACK FOLLOWS==
RMAN-03009: failure of backup command on OR_Dsk_1
ORA-19504: failed to create file "ORADBI.appl2_tbl.dbf"
ORA-27040: file create error, unable to create file
SVR4 Error: 2: No such file or directory

ERROR Messages

- Incomplete cleanup – The message above required a rerun of the CONVERT DATAFILE statement. We tested to see how RMAN would react to existing destination files. The error message below gave us the answer.
- RMAN-00569: ==ERROR MESSAGE STACK FOLLOWS==
RMAN-03009: failure of backup command on ORA_DISK_1 channel at 10/15/2008 15:32:30
ORA-19504: failed to create file
/app/oraf/u022/datafiles/ORDBI/ORDBI.appl1_tbl.dbf
ORA-27038: created file already exists

Conclusion

- Drive to reduce operational costs – server consolidation
- Cross-platform database moves becoming common
- Our move was not the worst case (endianness)
- We learned more about the inner workings of RMAN and datafile structures.
- parallelized processing greatly improves RMAN throughput
- push beyond what I thought was reasonable.

Tom Kyte “How we know what we know”

Acknowledgements

- special thanks to the entire Marketing DBA group (Devesh, Ravi, Dave, Suresh, Linda, Pradeep, Hanmanth, Sadiq) who have provided invaluable input into the design and testing of the various methods considered during the course of this project and recognition of the Marketing System Administrators for their willingness to humor us and our requests for system modifications to help us improve the migration process. Our project manager and resource managers helped us overcome many bureaucratic obstacles over the life of this project.

References

- Oracle - Backup and Recovery
Advanced User's Guide
Chapter 15 – RMAN Cross-Platform
Transportable Databases and Tablespaces
- Metalink – Bug reports, Technical Notes, How-To articles. The specific numbers are listed in this paper.

More Fun

■ PLATFORM_NAME	ENDIAN_FMT
AIX-Based Systems (64-bit)	Big
Apple Mac OS	Big
HP-UX (64-bit)	Big
IBM * Linux	Big
Solaris[tm] OE *	Big
HP *	Little
Linux *	Little
Microsoft Windows *	Little
Solaris Operating * non-Sparc	Little

Endian discussion cont'd

- A 32 bit quantity is split into 4 bytes.
i.e. x90AB12CD
- The 4 bytes are: 90, AB, 12, CD where each byte requires 2 hex digits.
- Two ways to store this in memory.
- **Big Endian** - most significant byte in smallest addr.

Mem Addr 00 ==> FF

Data 90, AB, 12, CD

- **Little Endian** - *least* significant byte in smallest addr.

Mem Addr 00 ==> FF

Data CD, 12, AB, 90

Endian discussion cont'd

- Little Endian is the reverse order compared to big endian of the bytes – the bits are still in the same order. Mnemonic : 3-L's It's Little endian if the Least significant digit is in the Lowest address (Big edian - the most significant byte is stored first).
- The term "endian" comes from Jonathan Swift, a satirist, wrote "Gulliver's Travels" in 1727. He talks about how certain people prefer to eat their hard boiled eggs from the little end first (thus, little endian), while others prefer to eat from the big end (thus, big endians) and how this lead to various wars.

Endian discussion cont'd

- Most modern computer processors agree on bit ordering "inside" individual bytes (was not always the case). So any single-byte value will be read the same on almost any computer.

Miscellaneous Ramblings

- TNSNAMES.ora vs Oracle Internet Directory
 - Looking for anyone who is using it
- Staggered file migration
 - Require less space at one time

THANK YOU !

- Contact information
 - John.X1.Larkin@chase.com