Reducing downtime using incremental backups and X-Platform TTS

Martin Bach, Practice Director, Enkitec
Reducing downtime using incremental backups and X-Platform TTS

Martin Bach, Practice Director, Enkitec

Thank you for joining us
Presenters

Martin Bach
@MartinDBA
Oracle ACE, member of the OakTable, Practice Director at Enkitec.

James Murtagh @jmurts
Part of the Oracle Tools’ team at Red Gate, creator of All Things Oracle.com.
Before we start

• You will be on mute for the duration of the event
• We are now talking so please type a message in the Chat box in the Control Panel if you can’t hear us (please check your speakers and GoToWebinar audio settings first)
• There will be a Q&A session at the end but please type your questions in the Questions box in the Control Panel in advance
• A recording of the full webinar will be put up online
About Red Gate

• Over 600,000 technology professionals worldwide use Red Gate’s software, including 93% of Fortune 100 companies.
About Red Gate

- Our philosophy is to design highly usable, reliable tools that solve problems commonly faced by DBAs and developers.

“Source Control for Oracle is a huge time saver for me and for teams that I work with. I am blown away!”

Lewis Cunningham, Database Architect, Oracle ACE Director
About Enkitec

• Global systems integrator with a focus on the Oracle platform
• Exadata Specialized Partner status (one of a handful globally)
• 200+ Successful Exadata Implementations
• Dedicated, In-house Engineered Systems Lab (POV, Patch Validation)
• Additional practices include
  - Oracle RAC Implementation
  - Oracle Golden Gate
  - Oracle Applications Administration
  - On-Call Database Administration

This webinar is brought to you by:
Reducing downtime using incremental backups and X-Platform TTS

Martin Bach, Practice Director, Enkitec
About Me

- I work for Enkitec
- Co-author of “Pro Oracle Database 11g RAC on Linux” by Apress
- Oak Table Member
- Oracle ACE
- Oracle Certified Master
- Blogger
Agenda

• Why bother with incremental backups and cross-platform transportable tablespaces?
• The old way of migrating to little-endian
• The new way
• Demo
Why you should care

Or: why you *do* care and want to know more
Trends & Facts

• IBM pSeries, SPARC servers, HP-UX Itanium
  – Have been losing market share
  – Lots of workloads taken over by industry standard components
  – Mostly running Linux

• POWER-based systems seem to have done better defending their market share

• SPARC suffered from low performance/$$$ in the past

• AMD appears to have become a niche vendor
Reasons

- Majority of workloads ok with 2-4 socket systems
- Established UNIX platforms expensive
  - In contrast to equivalent x86-64 cores
  - Oracle core factor!
- Advantages are being lost
  - Largest memory available with E7 Xeons (2-4 TB/node)
  - Largest current Xeon-EP system is 2s24c48t (1.5 TB RAM)
  - AMD 63xx pack 16 “cores”/socket, E7-8870 10+HT
  - RAS features not as important as they were
So what does that mean

- Many organisations are revisiting their standards
  - New building blocks emerging
  - Consolidation & DBaaS

- Oracle’s Engineered Systems
  - Exadata
  - Oracle Database Appliance

- Or roll your own?
No really now

- Well I guess all I want to say is

You will be moving Systems to Linux soon!
Migrating to Linux
That includes Exadata too
Strategies for Linux migrations

- Strategy depends on source platform
  - It is easy if the data encoding remains the same
  - Endianness matters!

- Important Big Endian platforms
  - SPARC, AIX, HP-UX, …

- Mainstream Little Endian platforms
  - Linux, Windows
Endianness

- Term is based on Jonathan Swift’s “Gulliver’s Travels”
- In CS it means the byte order in a word (=memory)
- Big endian
  - Store the most significant byte in the smallest address
- Little endian
  - Store the least significant byte in the smallest address
- Conversion problems
Data Guard!

- Who would have thought...that it can be so easy!
  - Heterogeneous Data Guard can be a solution

- Physical Standby
  - MOS DocID 413484.1
  - Contains table with valid combinations

- Logical Standby
  - MOS Doc ID 108568.1

- Even support for some 32bit/64bit combinations!
Some possible Data Guard combinations

- **Windows**
  - Mixed 32bit/64bit, including IA64
  - Linux 32bit/64bit, including IA64

- **HP-UX**
  - Mixed PA-RISC and Itanium configuration

- **AIX and Solaris SPARC**
  - Might happen: Enhancement Request in bug 12702521

- **Respect restrictions as outlined in My Oracle Support**
If no Data Guard then

- There are many options in 11g, including
- Recovery Manager/Transportable Tablespaces
- DBMS_FILE_TRANSFER
- Logical export/import
- Replication
  - Golden Gate
  - Streams
  - others
The Endian Conversion

Using RMAN to convert from big endian to little endian
Using TTS for a migration

• Simple in theory
  1. Set tablespace to read-only on source
  2. Export tablespace metadata from source
  3. Transport files to destination
  4. Create necessary metadata (users, grants)
  5. Import tablespace metadata to destination
  6. Make tablespace available for reading and writing

• If you have a large enough downtime window
Let’s do it then!

- That does indeed sound possible!
- Slight complications
  - I.e. you can’t take a tablespace from AIX and plug it into Linux
  - Tablespaces must be self-contained
- You may need to convert the data files
- But what about the source system?
  - You will have application downtime during the migration!
  - “We do not use (a lot of) downtime”
X-Platform incremental backup TTS now
Potentially reducing the downtime window considerably at no extra cost
What is this tongue twister?

• Cross Platform incremental backups + TTS addresses:
  – The need for extended downtime
  – Read-Only phase can be shortened
  – Less/no need for replication tools

• Cross Platform incr backup + TTS does NOT help you (much) in these cases:
  – If you are exporting excessive meta-data (EBS!)
  – If you are not migrating to Exadata
RMAN incremental backups

- Standard RMAN feature
  - Create an image copy of a data file
  - Take incremental backups of the data file
  - Apply incremental on *datafile copy* rolling it forward
  - Designed with “switchover to copy” in mind
  - Source remains online & available

- Storage admins will find that amusing
The New World Order

- Incremental backups and x-platform TTS are cool
- Procedure explained in Doc ID 1389592.1
- Multiple phases
  - Initial setup
  - Prepare phase
  - Roll forward phase
  - Transport phase
  - Cut over
Prerequisites

- Incremental Convert Home
  - Separate 11.2.0.2 + Exa BP12 home
  - Patch 13340675 for core functionality
  - Patch 1459711 if source contains IOTs

- Cannot overcome limitations of TTS

- Source database must have compatible set to >= 10.2
  - + Archivelog mode
  - + no compressed backups for device type disk
  - + no offline files in TBS; TBS must not be read only
Initial setup (1)

- On the source host
  - Download rman_xttconvert_1.3.zip from MOS
  - Store in $HOME/xtt
  - Edit xtt.properties
  - Create necessary directories
  - Copy $HOME/xtt to destination
Initial setup (2)

- On the destination host
  - Create incremental RDBMS software home as described
  - Start a temporary instance in nomonut mode
  - Create/identify RDBMS home for the destination database
  - Create/define database to import tablespace into
  - Apply any other patches
    - QDPEs
    - Overlay patches
  - Create necessary directories
xtt.properties used for demo

tablespace=MBH1,MBH2
platformid=20
dfcopydir=/u01/xtt/stage_source
backupformat=/u01/xtt/stage_source
stageondest=/u01/xtt/stage_dest
storageondest=/u01/oradata/little/
backupondest=/u01/reco
cnvinst_home=/u01/app/oracle/product/11.2.0.2/xtt_home
cnvinst_sid=xtt
parallel=1
Conversion instance

[oracle@linux dbs]$ cat initxtt.ora
db_name=xtt
compatible=11.2.0.2.0

[oracle@linux dbs]$ tail –n2 /etc/oratab
xtt:/u01/app/oracle/product/11.2.0.2/xtt_home:N
little:/u01/app/oracle/product/11.2.0/dbhome_1:N

This webinar is brought to you by:
Prepare Phase (1)

• Run prepare step on source system
  – Creates datafile copies locally to “dfcopydir”
  – Does some validation and sanity checking
  – Creates files xttplan.txt and rmanconvert.cmd
  – $ perl xttdriver.pl -p
Prepare Phase (2)

oracle@solaris:~/xtt$ . oraenv
ORACLE_SID = [bigendia] ?
The Oracle base remains unchanged with value /u01/app/oracle

oracle@solaris:~/xtt$ $ORACLE_HOME/perl/bin/perl xttdriver.pl -p
Prepare source for Tablespaces:
   'MBH1','MBH2' /u01/xtt-stage_dest
xttpreparesrc.sql for 'MBH1','MBH2' started at Wed Jun 19 08:50:14 2013
xttpreparesrc.sql for ended at Wed Jun 19 08:50:15 2013
oracle@solaris:~/xtt$
Prepare Phase (3)

• Transfer datafile copies to destination
  – Either use NFS or scp to copy files
  – `$ scp /u01/xtt/stage_source/* linux:/u01/xtt/stage_dest`
  – No ASMCMD or DBMS_FILE_TRANSFER possible

• Convert copies on destination system
  – Copy `rmanconvert.cmd` and `xttplan.txt` to destination
  – `$ perl xttdriver.pl -c`
  – Datafiles will be created in storageondest location
[oracle@linux xtt]$ $ORACLE_HOME/perl/bin/perl xttdriver.pl -c
Converted datafiles listed in: /tmp//xttnewdatafiles.txt

[oracle@linux xtt]$ cat /tmp//xttnewdatafiles.txt
##MBH1
5,/u01/oradata/little/MBH1_5.xtf
##MBH2
6,/u01/oradata/little/MBH2_6.xtf
[oracle@linux xtt]$
This webinar is brought to you by:

FYI-RMAN conversion script

oracle@Solaris:/tmp$ cat rmanconvert.cmd
host 'echo ts##MBH1';
  convert from platform 'Solaris Operating System (x86-64)'
  datafile
   '/u01/xtt/stage_dest/MBH1_5.tf'
   format '/u01/oradata/little/%N_%f.xtf'
  parallelism 1;
host 'echo ts##MBH2';
  convert from platform 'Solaris Operating System (x86-64)'
  datafile
   '/u01/xtt/stage_dest/MBH2_6.tf'
   format '/u01/oradata/little/%N_%f.xtf'
  parallelism 1;
oracle@Solaris:/tmp$
Prepare Phase Summary

• The following actions were performed
  – Initial image copy of tablespaces created
  – Data file copies are available on destination
  – Files have been converted to destination platform

• Source system was ONLINE all the time

• Now keep calm and carry on
Roll Forward Phase (1)

- Source system carries on as normal
- Incremental backups of the source tablespaces are key
  - Take backup
  - Transfer to the destination
  - Convert incremental backup and apply to datafile copies
  - Repeat as often as you like
- At the cutover date continue with Transport Phase
Roll Forward Phase (2)

- Take initial incremental backup
  - `$ perl xttdriver -i`
  - Creates the incremental backup
  - Meta-information in tsbkupmap.txt and incrbackups.txt

- Transfer files to destination
  - `$ scp $(cat incrbackups.txt) linux:/u01/xtt/stage_dest`
  - `$ scp xttplan.txt tsbkupmap.txt linux:$(pwd)`
Roll Forward Phase (3)

- Convert and apply: on destination
  - Recently transferred backups are converted to Linux
  - And applied to the datafile copies
- Based on xttplan.txt tsbkupmap.txt
- `$ perl xttdriver.pl -r`
  - Environment variables point to destination database
  - Connection made against xtt instance for conversion
  - Connection against destination db for merge
- Meta files need to be copied for each iteration
Roll Forward Phase (4)

- Get the next SCN for further backups

```
oracle@ssolaris:~/xtt$ $ORACLE_HOME/perl/bin/perl \ 
> xttdriver.pl -s
Prepare newscn for Tablespaces: 'MBH1','MBH2'
New /tmp//xttplan.txt with FROM SCN's generated
oracle@ssolaris:~/xtt$
```

<table>
<thead>
<tr>
<th>Datafile#</th>
<th>TBS name, current SCN</th>
</tr>
</thead>
<tbody>
<tr>
<td>MBH1####992746</td>
<td>5</td>
</tr>
<tr>
<td>MBH2####992764</td>
<td>6</td>
</tr>
</tbody>
</table>

This webinar is brought to you by:
Roll Forward Phase (5)

- Repeat this phase as often as you like
- Then make a decision to cut over
  - Enter the transport phase
  - This is where you *will* encounter an outage
  - Outage duration depends on the amount of metadata
- In the meantime
  - Pre-create users and grants in destination database
Transport Phase (1)

• Identical to “traditional” TTS now
  – Make tablespaces on source database read only
  – Create a final incremental backup
  – Ship it over including required files
  – Roll forward + recovery
  – Import metadata
  – Validate tablespaces in destination
  – Make them read-write
  – DONE!
Transport Phase (2)

oracle@soliris:/xtt$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.3.0 Production on Wed Jun 19 09:37:52 2013

Copyright (c) 1982, 2011, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit
With the Partitioning option

SQL> alter tablespace mbh1 read only;
Tablespace altered.

SQL> alter tablespace mbh2 read only;
Tablespace altered.

SQL>
Transport Phase (3)

- Final incremental backup
- No big difference to previous phase
- Steps to follow
  - Create the incremental backup
  - Ship the RMAN backup sets
  - Ship xttplan.txt tsbkupmap.txt
  - Roll the backup forward
Transport Phase (4)

- Prepare metadata import
  - The code uses expdp
  - It assumes a NETWORK_LINK

```
[oracle@linux xtt]$ $ORACLE_HOME/perl/bin/perl xttdriver.pl -e
[oracle@linux xtt]$ cat /tmp/xttplugin.txt
impdp directory=<DATA_PUMP_DIR> logfile=<tts_imp.log> 
  network_link=<ttslink> transport_full_check=no 
  transport_tablespaces=MBH1,MBH2 
  transport_datafiles='/u01/oradata/little/MBH1_5.xtf','/u01/oradata/little/MBH2_6.xtf';
[oracle@linux xtt]$
```

- Change for your needs and run
Finishing touches

- Run RMAN validate tablespace xyz check logical;
- Make tablespace read-write on destination
- Test
- Sign off
- Have a beverage of choice
References

Read on! Test it! Do it!
Useful DocIDs and other material

- 1166561.1-Master Note for TTS
- 1389592.1-cross platform incremental backup TTS master note
Q&A

www.red-gate.com/oracle
www.enkitec.com
@jmurts
@MartinDBA
Thank you

www.red-gate.com/oracle
www.enkitec.com
@jmurts
@MartinDBA