DBA Best Practices from the Field

Arup Nanda
Starwood Hotels
Who am I

• Oracle DBA for 13 years and counting
• Speak at conferences, write articles, 4 books
• Brought up the Global Database Group at Starwood Hotels, in White Plains, NY
Why this Session

• I have seen too many Rules of Thumb and “Best” Practices.
• Some of them could be questionable or even downright wrong!
• Warning: I am questioning everything, leaving nothing to “expert opinions”.
• proligence.com/downloads.html
Advice is seldom welcome. And those who want it most always like it the least.

Earl of Chesterfield
1694-1773
Guidelines

• Best practices must be justified
  - “Use Oracle Flexible Architecture”
  - “Ummm … why?”
  - “Because, it’s the best practice, stupid!”
  No Justification → Not Acceptable

• It must apply to all cases or show clearly where it’s applicable
  - Best practice is to hot-backup the database
  - Unless, there is a performance gain by running in NOARCHIVELOG mode
Different Oracle Homes

• **Traditional Approach:**
  – `/u01/app/oracle/10.2`
  – Patches are applied to the same OH

• **Suggestion:**
  – Use a *different* Oracle Home for each upgrade and patching
    `/u01/app/oracle/10.2/db1`
  – In next patch, create a different OH
    `/u01/app/oracle/10.2/db2`
  – Apply the patch here, not on db1
New OH Each Time

Current OH: /u01/app/oracle/10.2/db4
New OH: /u01/app/oracle/10.2/db5

- Reduction in risk of new code.
  - Patch fails -> reset the OH to the old one – db4 and restart the db
- Diff is possible, what changed by the patch
- Catalog scripts (c*,u*,catalog, catproc) preserved
- Some binaries are available in older OH
- OPatch "bugs" are also mitigated
ASM Home ≠ DB Home

- ASM is embedded in the Database code; so no need to have a different Ora Home
- Suggestion:
  - Create a different OH for ASM, even though it will be initially identical to DB Ora Home
    - /u01/app/oracle/10.2/db1
    - /u01/app/oracle/10.2/asm1
  - Apply patches specific to product to OH
Set Audit Trail

• Set audit_trail = db
  Even if you do not need to audit
• True or False: Setting audit_trail to DB will start auditing and increase IO?
• Answer: FALSE! You need to issue AUDIT statements as well.
• This parameter needs a recycle; so set it even if you don’t plan on using it.
• 11g already has it by default!
Set some parameters

- Set some parameters (These are not modifiable by ALTER SYSTEM).
- `_trace_files_public = TRUE`
  - Sooner or later you need to give access to some trace files to developers
- `utl_file_dir = '/tmp'`
  - Don’t need that because of directory objects
  - Req'd for creating Log Miner Dictionary on Flat File
Dump "OFA"

- Oracle Flexible Architecture (OFA)
  /u01/app/oracle
  - admin/SID/bdump
  - admin/SID/udump
  - oradata/SID/datafiles

- Does not allow separation of filesystems for security, performance, physical location, etc.
- Does not allow for passive failovers
Non-OFA Layout

• Suggested Layout
  /oracle
    ➔ admin/SID/* – not allowed to users
  /u01/udump – allowed for users
  /prddata/SID/mount1/datafiles – high performance
  /prddata/SID/mount2/datafiles – low performance

• This mount point naming convention – /prddata/SID/mount1– allows passive failover. MP
  unmounted from one host and mounted to the passive node.
• On QA, use /qadata/SID/mount1/… naming convention; so both prod and qa can be mounted without risk of
  name collision
Analyze CPU

- Auditing is expensive; we need biggest bang for the buck - Session Auditing
  
  SQL> audit session;

- Purpose:
  - Calculate CPU consumption and profile users
  - Calculate I/O used by users
  - Identify if someone’s account was locked after repeated wrong passwords
Understand the CPU Usage

```sql
select username, to_char(logoff_time,'mm/dd') ts,
       count(1) cnt,
       sum(session_cpu) sum_cpu,
       avg(session_cpu) avg_cpu,
       min(session_cpu) min_cpu,
       max(session_cpu) max_cpu
from dba_audit_trail
where logoff_time between '&start_date' and '&end_date'
  group by username, to_char(logoff_time,'mm/dd')
order by username, to_char(logoff_time,'mm/dd')
```

Output

<table>
<thead>
<tr>
<th>USERNAME</th>
<th>TS</th>
<th>CNT</th>
<th>SUM_CPU</th>
<th>AVG_CPU</th>
<th>MIN_CPU</th>
<th>MAX_CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>USER1</td>
<td>04/04</td>
<td>3</td>
<td>918</td>
<td>306</td>
<td>17</td>
<td>859</td>
</tr>
<tr>
<td>USER2</td>
<td>04/04</td>
<td>36</td>
<td>15,286</td>
<td>425</td>
<td>0</td>
<td>4,094</td>
</tr>
<tr>
<td>USER3</td>
<td>04/04</td>
<td>3</td>
<td>794</td>
<td>265</td>
<td>174</td>
<td>379</td>
</tr>
<tr>
<td>USER4</td>
<td>04/04</td>
<td>187</td>
<td>396,299</td>
<td>2,119</td>
<td>1</td>
<td>124,274</td>
</tr>
</tbody>
</table>
Know Activity by Users

```sql
select username, to_char(logoff_time,'mm/dd') ts,
    sum(logoff_lread) lread,
    sum(logoff_pread) pread,
    sum(logoff_lwrite) lwrite,
    sum(session_cpu) scpu
from dba_audit_trail
where logoff_time between '&start_date' and '&end_date'
group by username, to_char(logoff_time,'mm/dd')
order by username, to_char(logoff_time,'mm/dd')
```

**Output**

<table>
<thead>
<tr>
<th>USERNAME</th>
<th>TS</th>
<th>LREAD</th>
<th>PREAD</th>
<th>LWRITE</th>
<th>SCPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>USER1</td>
<td>04/04</td>
<td>283,271</td>
<td>10,858</td>
<td>33</td>
<td>918</td>
</tr>
<tr>
<td>USER2</td>
<td>04/04</td>
<td>4,570,965</td>
<td>6,225</td>
<td>2,854</td>
<td>15,286</td>
</tr>
<tr>
<td>USER3</td>
<td>04/04</td>
<td>601,838</td>
<td>1,988</td>
<td>26</td>
<td>794</td>
</tr>
<tr>
<td>USER4</td>
<td>04/04</td>
<td>33,639,028</td>
<td>4,545,505</td>
<td>1,083,473</td>
<td>396,299</td>
</tr>
</tbody>
</table>
Trace Account Locks

- Identify when someone’s account was locked

```sql
select to_char(timestamp,'mm/dd/yy hh24:mi') ts,
       os_username, userhost, returncode
from dba_audit_trail
where username = 'ARUP'
order by timestamp;
```

**Output**

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Username</th>
<th>Userhost</th>
<th>Returncode</th>
</tr>
</thead>
<tbody>
<tr>
<td>01/10/07 14:12</td>
<td>arupnan</td>
<td>CORP\UPNANT</td>
<td>0</td>
</tr>
<tr>
<td>01/10/07 15:12</td>
<td>arupnan</td>
<td>CORP\UPNANT</td>
<td>0</td>
</tr>
<tr>
<td>01/11/07 04:00</td>
<td>orandsp</td>
<td>hndspdb1</td>
<td>1017</td>
</tr>
<tr>
<td>01/12/07 04:00</td>
<td>orandsp</td>
<td>hndspdb1</td>
<td>1017</td>
</tr>
<tr>
<td>01/13/07 04:00</td>
<td>orandsp</td>
<td>hndspdb1</td>
<td>1017</td>
</tr>
<tr>
<td>01/14/07 04:00</td>
<td>orandsp</td>
<td>hndspdb1</td>
<td>1017</td>
</tr>
<tr>
<td>01/15/07 04:00</td>
<td>orandsp</td>
<td>hndspdb1</td>
<td>28000</td>
</tr>
</tbody>
</table>

- **Wrong Password**
- **Login OK**
- **Account Locked**
Audit DDL

• Because someone will always complain, what happened to his/her table
  .... and you are the DBA and you are saying you don’t know what happened to it?!!!!!!!!!!!

• SQL: AUDIT TABLE BY SESSION;

• stmt_audit_option_map shows the statements

• AUDIT ALL BY SESSION does most DDLs

• Caveat: in DW environments, users create and drop a large number of tables; so this may not be advisable.
No .LOG for Redos

• Common Practice:
  – Redo logs are named <Name>.log

• Problem:
  – Deletion of log files via some cron that deletes “.log” files generated, e.g. sqlnet.log.
  – Listener Attack that can change the listener log to redo1.log

• Suggestion:
  – Choose .redo or .rdo for redo log files.
Listener

• Set Admin Restrictions in LISTENER.ORA
  – ADMIN_RESTRICCTIONS_LISTENER=on
  – This prevents online modification of the listener parameters
  – Modify the listener.ora file and use
    $ lsnrctl reload

• Use a different listener for External Procedures
Build a Metadata Repository

• Use Data Pump to Create a Repository of Objects:
  
  $ expdp u/p content=metadata_only full=y 
  directory=tmp_dir dumpfile=md.dmp

• Import this to create an SQL File
  
  $ impdp u/p directory=tmp_dir 
  dumpfile=md.dmp sqlfile=md.sql

• See my paper: Datapump: Not Just for Data Movement
Validate Database

• Use RMAN Validation Option
  RMAN> backup validate database archivelog all;
  Then check for corrupt blocks in view v$database_block_corruption

• Logical Corruption
  RMAN> backup validate check logical database archivelog all;
Preview RMAN Restore

• Always preview a restore
  RMAN> restore tablespace users preview;
• Does not actually restore but checks the availability of files
• Not the same as VALIDATE
  – Checks what files are required
  – Validate assumes you know that
• Not the same as TEST
  RMAN> restore tablespace users test;
  – Does not actually start the recovery process; so the tablespace need not be offline
RMAN> restore tablespace users preview;

List of Datafile Copies
Key    File S Completion Time    Ckp SCN    Ckp Time       Name
------- ---- - --------------- ---------- --------------- ----
173716  238 A 30-MAR-07  62872433554 30-MAR-07 /f.rman
... And so on ...
173775  2074 A 31-MAR-07  62918498516 31-MAR-07 /j.rman

no backup of log thread 1 seq 92170 lowscn 62872343042 found to restore
... And so on ...
no backup of log thread 1 seq 92173 lowscn 62902345362 found to restore

List of Archived Log Copies
Key    Thrd Seq S Low Time    Name
------- ---- ------- - --------- ----
92212   1    92174   A 30-MAR-07 /PROPRD1_1_92174_525355299.arc
... And so on ...
92239   1    92201   A 01-APR-07 /PROPRD1_1_92201_525355299.arc

Media recovery start SCN is 62872433554
Recovery must be done beyond SCN 62948207913 to clear data files fuzziness
Finished restore at 06-APR-07
Save RMAN Log

• You copy to tape:
  – RMAN backup files
  – Init file
  – Archived logs
  – But not RMAN Log files, do you?
• RMAN Logs contain information about the backup pieces, names, location, etc.
• Proves invaluable during recovery
  input datafile fno=00084 name=/f1.dbf
  output filename=/backup/loc3/data_D-CRMPRD_I-79785763_TS-DWT_ODS8_RES_FN
  0-96_43ie2scm.rman tag=FULLBKPFS recid=174298 stamp=618757792
• Allows you to look for specific files from backup sets
DBID

- Important for Recovery
- Note the DBID and keep it in a separate place
- Write DBID to alert log every time backup is taken

```sql
declare
    l_dbid number;
begin
    select dbid into l_dbid from v$database;
    dbms_system.ksdwrt(2,'DBID='||l_dbid);
end;
```
Do Not Use SPFILE

- **SPFILE Advantages:**
  - Can be on shared filesystem, incld. on ASM
  - Can be backed up by RMAN
  - Can be updated automatically by command line by `ALTER SYSTEM SET ... SCOPE = SPFILE;`

- **SPFILE Disadvantage**
  - Older version overwritten
  - Comments possible; but only for the current entry
PFiLe Advantages

• Place comments in the init.ora file
  # AKN 3/20/06 added because ...
  # RJN 4/10/06 changed from 1M to 2M
  # JER 10/3/06 changed from 2M to 4M
  # DFW 12/7/06 changed from 4M to 6M SR# ...
  log_buffers = 6M

• Has a history of changes, with the names and dates of changes

• Very useful for troubleshooting and for record keeping
If you must use SPFILE

• Make sure you have a version control system in place to track parameter changes
  
• Example:
  – SQL> create pfile=’/tmp/a’ from spfile;
  – Check diff between this and the previous
  – Write the differences to a log file

• In Oracle 11g, you can create PFILE from memory:
  SQL> create pfile='...' from memory;
New Oracle User for Clients

• Problem:
  – App running on the DB server, needs SQL*Plus
  – $OH/bin/sqlplus is not accessible to world

• Common Solution:
  – Change $OH permissions to allow all others
  – Make app part of the “dba” group

• Suggestion:
  – Create a separate Oracle user: “appora”
  – Install the Oracle client under that user
Separate Instance and DB Names

• Common Practice:
  – DB_NAME same as Instance Name

• Suggestion:
  – Append “1” after DB Name for Instance, e.g.
    • DB Name: PRODB
    • Instance: PRODB1
  – If you ever need to convert the DB to RAC, you will not need to change the Instance Name
  – No need to change Init.ora, PW File, etc.
Archivelog Location

• Rate the most important
  – Datafiles
  – Archivelogs
  – Backup of datafiles

• Most important is archivelogs
  – If datafiles are lost, they can be recreated
  – Archived logs are *never* recreatable
  – Missing archived logs = halted recovery

• Flash Recovery Area
  – Not for Archived Logs
Create a Controlfile on Trace

• **Execute:**
  
  SQL> alter database backup controlfile to trace as '/path/cr_db.sql' reuse;

• It creates a CREATE CONTROLFILE script
  – You can use it to recreate controlfile
  – Or, the database itself
  – Self documenting the datafiles and redo logs

• **Change Control:**
  – Write a separate file for each day
  – Do a diff to find the added files, redo logs, etc.
Use oraenv

• Oracle supplied tool, in $OH/bin
• Look up the OH in /etc/oratab or /var/opt/oracle/oratab (in Solaris)
• Why this?
  – It makes your job easier while changing OH
  – It makes a consistent interface – jobs, commands
To ASSM, or not?

• Automatic Segment Space Management
  – Uses bitmap of free space on the block; no need to check the UET$ table
  – Great for Performance
• But, bitmap is only for 25, 50 and 75% free
• Potentially lose up to 25% space on each block
• Suggestions:
  – Use ASSM for non-DW databases
  – Use MSSM for DW databases
    • Buffer busy waits not common on DW anyway
Kill Inactive Sessions

• Problem:
  – Some apps, especially web apps under connection pool, remain inactive draining resources.

• Suggestion:
  – Use resource manager and set the inactive session disconnect timeout

• Why RM, why not Profiles?
  – RM allows you to turn on and off via scheduling and event. Profiles are hard coded.
  – RM allows service name based control
Check Listener Log

• Create External Tables on Listener Logs to identify issues, profile users, etc.

• See
  
  http://www.dbazine.com/oracle/or-articles/nanda14
Service Names

• Oracle database can be accessed via SID or Service Name

• Conventional TNS Entry

```plaintext
prodb1 =
    (DESCRIPTION =
        (ADDRESS_LIST =
            (ADDRESS = (PROTOCOL = TCP)(HOST = prolin1)
            (PORT = 1521)))
        (CONNECT_DATA = (SID = PRODB1)))
```

• Service Name

```plaintext
(CONNECT_DATA = (SERVICE_NAME = PRODB1)))
```
Enable Service Names

• In the instance, check service names present already:
  SQL> show parameter service_names

• Create additional service names:
  SQL> alter system set service_names = 'SVC1', 'SVC3', 'SVC3';

• Check is listener is listening for these:
  $ lsnrctl services

• In RAC, you should use SRVCTL:
  $ srvctl add service -d MYDB -s SVC1 ...
Why Service Names?

• No change in functionality
• Separates use from user, e.g. SCOTT logging from laptop uses service SVC1; but from app server SVC2.
• Enhances resource manager use
• Allows load balancing and failover in RAC or Data Guard databases
• Allows fine grained failover capabilities
  – Service SVC1 fails from node1 to node2; but SVC2 fails to node3
OS Specific Tweaks

- On HP/UX, use sched_noage
  - Necessary for setting right priorities for processes
  - Make "dba" group part of MLOCK

- On Solaris use Intimate Shared Memory
  - Optimizes the memory management
Raw Devices

• Use one size for devices and add them to tablespaces.
  • Common Use:
    – Create a raw device of 100GB in name /dev/..../users01.dbf
    – Create tablespace USERS with the raw device
    – When USERS need more room, expand the raw device.
  • Recommended Use:
    – Create raw devices of 30GB named /dev/.../d1, d2, etc.
    – Create tablespace with the devices d1, d2 and d3.
    – When USERS need more room, add a new device
• Advantages
  – No outage
  – Reuse devices
Using ORADEBUG

• Problem:
  – Database Issue; you want to use oradebug; but SQL*Plus hangs!

• When SQL*Plus does not work, use
  $ sqlplus -prelim
  It does not establish a connection
  You can run ORADEBUG now
Dumping

- Data block
  ```sql
  alter system dump datafile d block b;
  ```
- The rest:
  ```sql
  alter session set events 'immediate trace name <Key> level 10';
  ```
- Controlfile CONTROLF
- File Headers FILE_HDRS
- Redo Headers REDOHDR
- System State SYSTEMSTATE
- Process State PROCESSSTATE
- Library Cache LIBRARY_CACHE
  ```sql
  alter session set events 'immediate trace name LIBRARY_CACHE level 10';
  ```
Scripts

• Deletion of trace files older than some days.

```bash
DAYS=2
find /u02/app/oracle/admin -name "*.log" -ctime ${DAYS} -exec rm {} \;
find /u02/app/oracle/admin -name "*.trc" -ctime ${DAYS} -exec rm {} \;
find /u02/app/oracle/admin -name "*.trw" -ctime ${DAYS} -exec rm {} \;
find /u02/app/oracle/admin/*/cdump -ctime ${DAYS} -exec rm -r {} \;
```

• This clears up enough log files and trace files from OH, a major cause of failure.
**Aliases**

- Aliases make some repetitive job faster and quicker

```bash
alias bdump='cd $ORACLE_BASE/admin/$ORACLE_SID/bdump'
alias pfile='cd $ORACLE_BASE/admin/$ORACLE_SID/pfile'
alias obase='cd $ORACLE_BASE'
alias tns='cd $ORACLE_HOME/network/admin'
alias oh='cd $ORACLE_HOME'
alias os='echo $ORACLE_SID'
```
Remember

• It’s not a best practice, if it is not justified
• You have to understand why; not just what
• Best practice needs to be situation-aware
  – Which goes back to “you have to understand”
• Always question whenever someone tells you it’s a best practice