Case Study
RAC on 16 Node Linux Cluster

.... introducing the concept of Flexible Database Cluster ....

Session ID # 37600
Sept 11, 2003
San Francisco, CA
Kevin Closson
PolyServe
Principal Software Engineer (Database Engg)
kevinc@polyserve.com

and

Madhu Tumma
Credit Suisse First Boston
madhu.tumma@csfb.com

Co-author of the book “Oracle 9i RAC - Oracle Real Application Clusters Configuration and Internals” - (Rampant Tech Press)
What do we cover

- Objective of our study
- why RAC anyway ..
- decision to make about adopting Oracle RAC
- Concept of Flexible Database Clusters ‘FDC’
- what we built and tested
  - Architecture and Components
  - Cluster File System and SAN administration
  - Building Multiple Databases in a Cluster
  - Manageability – DB Tasks
  - Dynamism – Threads, Instances and Databases
  - Work Load shifting and Flexibility
  - Tests – IO Load, Scalability, Reconfiguration of node

Based on 16 Node RAC System
Main objective

The main purpose of our study:

- Look at the **manageability, flexibility** of the RAC Cluster in supporting different work loads in a large Linux Cluster
- Can it meet the challenge of **Server consolidation** !!
- Adopting a **large multi-node cluster** based on low-cost commodity servers, instead of **multiple smaller clusters** based on expensive SMP servers
- Proof of concept for **16 and above nodes** based Oracle RAC System
- Existence of **Multiple RAC databases** in a Linux cluster - *Dynamically shifting Servers from One Database to other*
- Administrative **Challenges in** managing large RAC Cluster
- Examine required **infrastructure** tools for smooth administration
Why RAC – Decisions to make ..

Why Adopt or Migrate to Oracle RAC

who needs anyway

• Enterprise Safety – Enterprise Lethargy
• Quite happy with 7.3.4 large databases
• Want to feel the need

Enterprises will adopt , if they see ..

• RAC is overall beneficial
• RAC gives better HA and Better Scalability
• RAC is easy to manage
• RAC gives favorable TCO
Why RAC – Decisions to make ..

At 2003 World Economic Forum in Davos, Switzerland, Bill Joy, the Chief Scientist, Sun Microsystems, while addressing business people, posed a question:

“What if the reality is that people have already bought most of the stuff they want to own”, the stuff he was talking about is IT infrastructure

The New rules for IT management: Nicholas Corr, the HBR’s Editor-at-large in his article ‘IT Does not Matter’

• Spend Less
• Follow, don’t lead
• Focus on vulnerabilities, not opportunities.
Decisions to make …

Critical Examination

Oracle RAC vis-à-vis VCS type of Simple HA Solution

For example

You have 6-node cluster, then 4 Instances

• Each has its own Disk Group

• Virtual IP / Virtual Host Name

• They are managed as separate Unit.

• Media Recovery does not affect others

Simple to manage, easily understood, no complexity involved, every knows about it
Decisions to make ...

Some of the Usual Methods

- Look into Peer industry
- Hire Experts in the field to evaluate
- Set up pilot projects and get a feel for the issues and benefits
- Attend Technical Expositions, technical user groups

Then finally a decision is made by the IT Management to adopt multi-node RAC system..

Then focus more on operational aspects..
Decisions – Operational stuff

**Decide**

- Level 1 - Physical Cluster – ‘n’ number of nodes (or hosts) interconnected
- Level 2 - Shared Storage for the cluster members, Cluster File System, Volume Management and Storage Virtualization
- Level 3 - Database, Mix the application environment, how many instances

**Details to work out**

- Multiple Dept(s) want to use the RAC database !! - How to accommodate them
  - Do you want to just allocate a schema
  - Within the cluster do you want to create Multiple RAC databases
- How big a cluster … you want to build
- How do you support the shared storage for all these nodes – SAN or DAS or NAS
Commoditization of IT infrastructure

- On Demand Era..
- Data Consolidation
- Reduced TCO
  - Fewer servers
  - Fewer storage devices since duplication is avoided
  - Fewer peripherals to perform backup
  - Fewer Units to manager and license
  - Inexpensive and easy to manage

- Flexible Intel based clusters running on Linux
- 1U rack-mounted servers and blade servers
- Increased use Linux O/S for database systems

Giga Information Group Straw Poll:
- 4% open source DB MySQL or PostgreSQL
- 22% have Oracle or DB2 on Linux
- 30% considering Linux for Database Platform
- 44% have no interest in open source
A typical RAC Database

Physical Components
- Nodes / Servers
- Private Interconnect
- Cluster Manager - OSD
- Shared Disk Storage
- CFS / Raw Partition
- Volume Manager
- Public Network

Global cache

Interconnect

Node-A

Node-B
RAC Instances

Node A

SGA

Redo # 1-2

df  df  df  df  df  df  df  df

Node B

SGA

Redo # 3-4

df  df  df  df  df  df  cf  sp

Node C

SGA

Redo # 5-6

Also need Oracle Executables, Admin Directories, UTIL_FILE
Three Layers

Top level
Physical Linux Cluster

Next Level
RAC Database(s)

Then
Each RAC Database will have ‘#’ of threads
Each thread can support an Instance
Case Study – 16 node RAC

1. How we built?
   What is the architecture? Server/Hosts, Storage and Interconnect
   What are the challenges?
   How did we solve? What did we like in this effort?

2. Oracle Installation
   Oracle DB Creation
   Adding Threads (multiple instances)
   Networking Configuration
   Configure SRVCTL and EM

3. What do we miss from the vendor point of view? Tools, Utilities, Some kind of Cluster Verification Tool

4. Tests Conducted
   - Dynamic Insertion of nodes as demand or load increases
   - Once a node goes down, add another one with out much too effort
   - Scalability in terms transaction throughput and IO throughput
   - IO monitoring
   - The usual database activities/tasks
Architecture – what we used for test

the building blocks

16 Nodes / Servers
IBM eServer Model 345
Dual Intel 2.4GHz Xeon Processors
2G RAM – DDR
Dual Gigabit Ethernet Interconnect
FAStT FC2-133 HBA
Suse operating system– SLES8

Disk Storage -
2GB Fibre Technology
2 IBM TotalStorage FAStT700
With 15 FASTT EXP 700 Exp. Units
About 7 TB storage space
206 – 36.4 GB HDSS

PolyServe Matrix Server
DB Optimized
Mount Cluster File system
mxmodstat utility to monitor
CDSL links

Oracle Enterprise Edition 9.2.0.3
With RAC option
Overall Architecture

the building blocks
SAN Administration

IBM Storage
- Total of 206 Disks
- Configure by FastT Storage Manager
- Into 8 arrays of 24 drives across multiple controllers and drive loops
- Into a few Large LUNS with RAID 1+0
- Hardware RAID

Steps to create
- Define Host groups
- Define Hosts
- Define port for each host
- Define storage partition
SAN Administration - LUNS

Shows LUNS from ONE storage server
(Right Side)
PolyServe File System - PSFS

With PolyServe Matrix Server

- LUNS are imported and given a cluster-global name (Track with WW name)

Matrix Server Features

- Allows Common Oracle Home
- All the database files (incl. CF, Redo)
- Supports External Tables
- UTIL File Directory
- Permits CDSL (context dependent symbolic links)
PolyServe File System

Even though some files can be on local f/s, when you have CFS, use CFS for all ... 

....Very Advantageous

<table>
<thead>
<tr>
<th>Data Files</th>
<th>shared</th>
</tr>
</thead>
<tbody>
<tr>
<td>Redo files</td>
<td>shared</td>
</tr>
<tr>
<td>Control files</td>
<td>shared</td>
</tr>
<tr>
<td>Archive logs</td>
<td>Shared</td>
</tr>
<tr>
<td>spfile</td>
<td>shared</td>
</tr>
<tr>
<td>oracle home/</td>
<td>shared</td>
</tr>
<tr>
<td>util dir/</td>
<td></td>
</tr>
</tbody>
</table>
PolyServe File System

- Same File System is Mounted on all nodes
- With DBOPTIMIZED option which permits Direct IO
- For administrative ease, have a large file system and create files as needed

```
rac13 oracle $ df -k
Filesystem 1K-blocks Used Available Use% Mounted on
/dev/sda3 13462052 2880868 10581184 22% / 
shmfs 2069392 0 2069392 0% /dev/shm
/dev/psd/psd1p4 425952820 77307864 348644956 19% /mnt/ps/oradata1
/dev/psd/psd3p1 426025076 197558996 228466080 47% /mnt/ps/oradata2
/dev/psd/psd4p1 426025076 193621772 232403304 46% /mnt/ps/oradata3
/dev/psd/psd5p1 426025076 208329732 217695344 49% /mnt/ps/oradata4
/dev/psd/psd8p1 426025076 216232776 209792300 51% /mnt/ps/oradata5
/dev/psd/psd9p1 426025076 236228768 189796308 56% /mnt/ps/oradata6
/dev/psd/psd12p1 426025076 132482744 293542332 32% /mnt/ps/oradata7
/dev/psd/psd6p1 248501600 13215124 235286476 6% /mnt/ps/shared_apps
```
Physical Cluster Architecture

- With Cluster File Systems are mounted, Install ORACM on to single Oracle Home
- Create “CDSL” and create Node-specific cmcfg.ora file and start Cluster manager on all nodes
Physical Cluster is formed..

```
rac1 oracle  $ cat $ORACLE_HOME/oracm/admin/cmcfg.ora
HeartBeat=15000
ClusterName=TEST
PollInterval=1000
MissCount=210
ServicePort=9998
KernelModuleName=hangcheck-timer
PrivateNodeNames= int-rac1 int-rac2 int-rac3 int-rac4 int-rac5 int-rac6 int-rac7 int-rac8 int-rac9 int-rac10
               int-rac11 int-rac12 int-rac13 int-rac14 int-rac15 int-rac16
PublicNodeNames= rac1 rac2 rac3 rac4 rac5 rac6 rac7 rac8 rac9 rac10 rac11 rac12 rac13 rac14 rac15 rac16
HostName=rac1
CmDiskFile=/mnt/ps/db/quorum
rac1 oracle  $
```

```
$ hostname
rac1
$ cd $ORACLE_HOME
$ ls -l oracm
lrwxrwxrwx 1 oracle  dba  17 2003-07-10 13:10 .oracm -> .oracm.{HOSTNAME}
$ ls -ld .oracm*
drvxx-xr-x 5 oracle  dba  120 2003-07-10 13:10 .oracm.rac1
drvxx-xr-x 5 oracle  dba  120 2003-07-10 12:11 .oracm.rac10
drvxx-xr-x 5 oracle  dba  120 2003-07-10 12:11 .oracm.rac11
drvxx-xr-x 5 oracle  dba  120 2003-07-10 11:12 .oracm.rac12
drvxx-xr-x 5 oracle  dba  120 2003-07-10 13:11 .oracm.rac13
drvxx-xr-x 5 oracle  dba  120 2003-07-10 13:10 .oracm.rac14
drvxx-xr-x 5 oracle  dba  120 2003-07-10 13:11 .oracm.rac15
drvxx-xr-x 5 oracle  dba  120 2003-07-10 13:10 .oracm.rac16
drvxx-xr-x 5 oracle  dba  120 2003-07-10 13:11 .oracm.rac17
drvxx-xr-x 5 oracle  dba  120 2003-07-10 13:10 .oracm.rac18
drvxx-xr-x 5 oracle  dba  120 2003-07-10 13:11 .oracm.rac19
$```

CDSL resolved oracm directories

Note: Oracle Cluster manager is not even aware of how many DBs will be created.
How we built 16 nodes Cluster

Create Database: PROD, DSS, DEV
- By Manual Method – easy to script and execute
- Add threads for each Database, Enable
- Start Instances 1- to 12 for PROD, 13 to 14 for DSS and 15 to 16 for DEV

Install Executables
Create CDSL etc
Common init.ora file
Create Database
Add Threads
Enable Threads
Start Instances

alter system set db_create_file_dest = '/usr/data1/';
create database dss
maxinstances 24 maxdatafiles 16384
maxlogfiles 48 noarchivelog
logfile size 100M, SIZE 100M;

ALTER DATABASE ADD LOGFILE THREAD 2 SIZE 100M;
ALTER DATABASE ADD LOGFILE THREAD 2 SIZE 100M;
ALTER DATABASE ENABLE PUBLIC THREAD 2;
ALTER DATABASE ADD LOGFILE THREAD 3 SIZE 100M;
ALTER DATABASE ADD LOGFILE THREAD 3 SIZE 100M;
ALTER DATABASE ENABLE PUBLIC THREAD 3;
Common Oracle Home

**Common ORACLE HOME**

- Easy to install (instead of 16 times)
- Easy to maintain a single copy
- Easy to apply patches / upgrade when needed
- Common Oracle Home is a boon to DBA/s (scripts / logs etc)
- With the help of CDSL, create node-specific directories to support

Imagine .. You have 3 RAC DB(s) in the cluster

- Assume, they are using different versions 9.2.0.1, 9.2.0.3 and 9.2.0.4
- You will end up with 3 * 16 = 48 Oracle Homes, if you do not use the common Oracle Home
### Active Instances in DB ‘PROD’

<table>
<thead>
<tr>
<th>INST_NUMBER</th>
<th>INST_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>rac1:prod1</td>
</tr>
<tr>
<td>2</td>
<td>rac2:prod2</td>
</tr>
<tr>
<td>3</td>
<td>rac3:prod3</td>
</tr>
<tr>
<td>4</td>
<td>rac4:prod4</td>
</tr>
<tr>
<td>5</td>
<td>rac5:prod5</td>
</tr>
<tr>
<td>6</td>
<td>rac6:prod6</td>
</tr>
<tr>
<td>7</td>
<td>rac7:prod7</td>
</tr>
<tr>
<td>8</td>
<td>rac8:prod8</td>
</tr>
<tr>
<td>9</td>
<td>rac9:prod9</td>
</tr>
<tr>
<td>10</td>
<td>rac10:prod10</td>
</tr>
<tr>
<td>11</td>
<td>rac11:prod11</td>
</tr>
<tr>
<td>12</td>
<td>rac12:prod12</td>
</tr>
</tbody>
</table>

12 rows selected.

### Threads in the DB ‘PROD’

<table>
<thead>
<tr>
<th>THREAD#</th>
<th>STATUS</th>
<th>ENABLED</th>
<th>GROUPS</th>
<th>INSTANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>OPEN</td>
<td>PUBLIC</td>
<td>2</td>
<td>prod1</td>
</tr>
<tr>
<td>2</td>
<td>OPEN</td>
<td>PUBLIC</td>
<td>2</td>
<td>prod2</td>
</tr>
<tr>
<td>3</td>
<td>OPEN</td>
<td>PUBLIC</td>
<td>2</td>
<td>prod3</td>
</tr>
<tr>
<td>4</td>
<td>CLOSED</td>
<td>PUBLIC</td>
<td>2</td>
<td>prod4</td>
</tr>
<tr>
<td>5</td>
<td>CLOSED</td>
<td>PUBLIC</td>
<td>2</td>
<td>prod5</td>
</tr>
<tr>
<td>6</td>
<td>OPEN</td>
<td>PUBLIC</td>
<td>2</td>
<td>prod6</td>
</tr>
<tr>
<td>7</td>
<td>OPEN</td>
<td>PUBLIC</td>
<td>2</td>
<td>prod7</td>
</tr>
<tr>
<td>8</td>
<td>OPEN</td>
<td>PUBLIC</td>
<td>2</td>
<td>prod8</td>
</tr>
<tr>
<td>9</td>
<td>OPEN</td>
<td>PUBLIC</td>
<td>2</td>
<td>prod9</td>
</tr>
<tr>
<td>10</td>
<td>OPEN</td>
<td>PUBLIC</td>
<td>2</td>
<td>prod10</td>
</tr>
<tr>
<td>11</td>
<td>OPEN</td>
<td>PUBLIC</td>
<td>2</td>
<td>prod11</td>
</tr>
<tr>
<td>12</td>
<td>OPEN</td>
<td>PUBLIC</td>
<td>2</td>
<td>prod12</td>
</tr>
<tr>
<td>13</td>
<td>CLOSED</td>
<td>PUBLIC</td>
<td>2</td>
<td>prod13</td>
</tr>
<tr>
<td>14</td>
<td>CLOSED</td>
<td>PUBLIC</td>
<td>2</td>
<td>prod14</td>
</tr>
<tr>
<td>15</td>
<td>CLOSED</td>
<td>PUBLIC</td>
<td>2</td>
<td>prod15</td>
</tr>
<tr>
<td>16</td>
<td>CLOSED</td>
<td>PUBLIC</td>
<td>2</td>
<td>prod16</td>
</tr>
</tbody>
</table>

16 rows selected.
Manageability is the key challenge

Next we focus on manageability issues

- Oracle Managed Files - OMF
- External Tables for ETT
- Server Control Utility – SRVCTL
- Analyze large objects
- Transportable Tablespace - PIT recovery
- Server Parameter File - SPFILE
- IO activity Monitoring
- Oracle Streams with in RAC environment

Then.. we discuss the Flexible Database Cluster (FDC)
Use Oracle Managed Files

We used the OMF facility . it simplified the tablespace creation

Advantages :

• administration of the database easier.
• consistent set of names rules
• reduce corruption caused by specifying the wrong file.
• reduce wasted disk space consumed by obsolete files.
• make development of portable third-party tools easier.

Methodology :
Set the DB_CREATE_FILE_DEST desired for data file

Example:

SQL> ALTER SYSTEM SET DB_CREATE_FILE_DEST = '/usr1/oracle/rw/DATA4';
SQL> CREATE TABLESPACE STRMTEST DATAFILE SIZE 1024M EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;
Use Oracle Managed Files

A snapshot of the files we got by this method, it follows the format

Datafile  o1_mf_%t_%u_.dbf  Tempfile  o1_mf_%t_%u_.tmp
Redo log file o1_mf_%g_%u_.log  Control file  o1_mf_%u_.ctl

```
select  substr(a.file_name,1,60) filename, substr(a.tablespace_name,1,20) tablesp, (a.bytes/1024/1024) bytes  
from v$datafile b, dba_data_files a where a.file_id = b.file#
```

<table>
<thead>
<tr>
<th>FILENAME</th>
<th>TABLESP</th>
<th>BYTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>/usr1/oracle/rw/DATA/o1_mf_system_3f0dae3c-3_.dbf</td>
<td>SYSTEM</td>
<td>308.5</td>
</tr>
<tr>
<td>/usr1/oracle/rw/DATA/o1_mf_sys_undo_3f0dae49-4_.dbf</td>
<td>SYS_UNDOTS</td>
<td>170.375</td>
</tr>
<tr>
<td>/usr1/oracle/rw/DATA/o1_mf_temp_3f0daf80-0_.dbf</td>
<td>TEMP</td>
<td>1000</td>
</tr>
<tr>
<td>/usr1/oracle/rw/DATA2/o1_mf_item_3f0daf9d-1_.dbf</td>
<td>ITEM</td>
<td>10000</td>
</tr>
<tr>
<td>/usr1/oracle/rw/DATA3/o1_mf_warehous_3f0db0b4-2_.dbf</td>
<td>WAREHOUSE</td>
<td>10000</td>
</tr>
<tr>
<td>/usr1/oracle/rw/DATA4/o1_mf_customer_3f0db1cb-3_.dbf</td>
<td>CUSTOMER</td>
<td>10000</td>
</tr>
<tr>
<td>/usr1/oracle/rw/DATA5/o1_mf_orders_3f0db2e0-4_.dbf</td>
<td>ORDERS</td>
<td>10000</td>
</tr>
<tr>
<td>/usr1/oracle/rw/DATA6/o1_mf_product_3f0db441-5_.dbf</td>
<td>PRODUCT</td>
<td>1000</td>
</tr>
<tr>
<td>/usr1/oracle/rw/DATA6/o1_mf_delcust_3f0db463-6_.dbf</td>
<td>DELCUST</td>
<td>10000</td>
</tr>
<tr>
<td>/usr1/oracle/rw/DATA2/o1_mf_warehous_3f0db5c6-7_.dbf</td>
<td>WAREHOUSE</td>
<td>10000</td>
</tr>
<tr>
<td>/usr1/oracle/rw/DATA2/o1_mf_item_3f0db6dc-8_.dbf</td>
<td>ITEM</td>
<td>10000</td>
</tr>
<tr>
<td>/usr1/oracle/rw/DATA2/o1_mf_customer_3f0db7f3-9_.dbf</td>
<td>CUSTOMER</td>
<td>10000</td>
</tr>
<tr>
<td>/usr1/oracle/rw/DATA2/o1_mf_orders_3f0db90b-10_.dbf</td>
<td>ORDERS</td>
<td>10000</td>
</tr>
<tr>
<td>/usr1/oracle/rw/DATA2/o1_mf_product_3f0dba23-11_.dbf</td>
<td>PRODUCT</td>
<td>1000</td>
</tr>
<tr>
<td>/usr1/oracle/rw/DATA3/o1_mf_warehous_3f0dba3e-12_.dbf</td>
<td>WAREHOUSE</td>
<td>10000</td>
</tr>
<tr>
<td>/usr1/oracle/rw/DATA3/o1_mf_item_3f0dbb56-13_.dbf</td>
<td>ITEM</td>
<td>10000</td>
</tr>
<tr>
<td>/usr1/oracle/rw/DATA3/o1_mf_customer_3f0dbc6d-14_.dbf</td>
<td>CUSTOMER</td>
<td>10000</td>
</tr>
<tr>
<td>/usr1/oracle/rw/DATA3/o1_mf_orders_3f0dbd85-15_.dbf</td>
<td>ORDERS</td>
<td>10000</td>
</tr>
<tr>
<td>/usr1/oracle/rw/DATA3/o1_mf_product_3f0dbe9c-16_.dbf</td>
<td>PRODUCT</td>
<td>1000</td>
</tr>
<tr>
<td>/usr1/oracle/rw/DATA4/o1_mf_warehous_3f0dbeb8-17_.dbf</td>
<td>WAREHOUSE</td>
<td>10000</td>
</tr>
</tbody>
</table>
Handling External Tables

Great Tool for ETL

Well it is not just the UTIL_FILE any more .. to read / write, Now you have External TABLES to deal with .. for ETL

EXT. Table a new and flexible way of loading data ..

Methodology

• Create Directory and Create External Table
• Keep replacing the File at different intervals
• Then make a SELECT to load into Regular DW tables
• PARALLEL Clause can be specified to read the DATA SOURCE in parallel
• Reading of data from external files using the Oracle loader technology.
• Good for basic Extraction, transformation, and transportation (ETT) tasks

To make it possible, Cluster File System is needed – We used the PSFS based file system to store the test data files
Handling External Tables

CREATE OR REPLACE DIRECTORY admin_load_dir as '/app/home/oracle/work/';

GRANT READ ON DIRECTORY admin_load_dir TO POLREC;

GRANT WRITE ON DIRECTORY admin_load_dir TO POLREC;

CREATE TABLE ext_employees (employee_id NUMBER(4), first_name VARCHAR2(20), last_name VARCHAR2(25), job_id VARCHAR2(10), manager_id NUMBER(4), hire_date DATE, salary NUMBER(8,2))

ORGANIZATION EXTERNAL (

TYPE ORACLE_LOADER DEFAULT DIRECTORY admin_load_dir

ACCESS PARAMETERS ( records delimited by newline badfile
       admin_load_dir:'empxt%a_%p.bad', logfile
       admin_load_dir:'empxt%a_%p.log', fields terminated by ',
       missing field values are null( employee_id, first_name,
       last_name, job_id, manager_id, hire_date char date_format date
       mask "dd-mon-yyyy", salary ))

LOCATION ('testfile1.dat')

PARALLEL REJECT LIMIT UNLIMITED;
SRVCTL – great tool to manage

```
rac1 oracle $ srvctl status database -d prod
Instance prod1 is running on node rac1
Instance prod2 is running on node rac2
Instance prod3 is running on node rac3
Instance prod4 is running on node rac4
Instance prod5 is running on node rac5
Instance prod6 is running on node rac6
Instance prod7 is running on node rac7
Instance prod8 is running on node rac8
Instance prod9 is not running on node rac9
Instance prod10 is not running on node rac10
Instance prod11 is running on node rac11
Instance prod12 is running on node rac12
Instance prod13 is not running on node rac13
Instance prod14 is not running on node rac14
Instance prod15 is not running on node rac15
Instance prod16 is not running on node rac16
```
rac1 oracle $ srvctl config database -d prod
```
rac1 prod1 /usr1/oracle
rac2 prod2 /usr1/oracle
rac3 prod3 /usr1/oracle
rac4 prod4 /usr1/oracle
rac5 prod5 /usr1/oracle
rac6 prod6 /usr1/oracle
rac7 prod7 /usr1/oracle
rac8 prod8 /usr1/oracle
rac9 prod9 /usr1/oracle
rac10 prod10 /usr1/oracle
rac11 prod11 /usr1/oracle
rac12 prod12 /usr1/oracle
rac13 prod13 /usr1/oracle
rac14 prod14 /usr1/oracle
rac15 prod15 /usr1/oracle
rac16 prod16 /usr1/oracle
```
srvctl stop instance -d prod -i prod1
srvctl stop instance -d prod -i prod2, prod3, prod4
srvctl start instance -d prod -i prod3, prod2
srvctl start instance -d prod -i prod4
```

….. from any node, you can manage all the instances and databases ….
Analyze large objects

Analyze Large Object
- Runs in parallel
- Use dbms_stats package in preference to analyze statement
- When 8 instance were up we ran ANALYZE
- 40 G table with 10 mil rows (with four partitons) took 5 min

```sql
DBMS_STATS.GATHER_TABLE_STATS ( 'POLREC', 'POLICYREC', NULL , 10 ) ;
```

```
SQL> SELECT INST_ID, QCSID, SID, SERVER_GROUP "Group", SERVER_SET "Set", Degree "Req Degree" FROM GV$PX_SESSION ORDER BY INST_ID, QCSID, QCINST_ID, SERVER_GROUP, SERVER_SET;
```

Showed DOP of 28 (on 8 instances)

<table>
<thead>
<tr>
<th>INST_ID</th>
<th>QCSID</th>
<th>SID</th>
<th>Group</th>
<th>Set</th>
<th>Degree</th>
<th>Req Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>23</td>
<td>24</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>25</td>
<td>14</td>
<td>1</td>
<td>1</td>
<td>28</td>
<td>32</td>
</tr>
<tr>
<td>1</td>
<td>25</td>
<td>17</td>
<td>1</td>
<td>1</td>
<td>28</td>
<td>32</td>
</tr>
<tr>
<td>1</td>
<td>25</td>
<td>21</td>
<td>1</td>
<td>1</td>
<td>28</td>
<td>32</td>
</tr>
</tbody>
</table>
Transportable Tablespace

**From PROD database**
(Nodes1 to 12)
Schema : POLREC
TBS - 40g size
(POL1, POL2, POL3, POL4)
Place in R-O
Export Meta data
Copy the Files

**Test Observations**
- Within the same Linux Cluster
- But between different Oracle RAC Databases (PROD to DSS)
- From 8K block buffer to 16K block buffer (we had to add db_8k_cache_size)

IMP-00003: ORACLE error 29339 encountered
ORA-29339: tablespace block size 8192 does not match configured block sizes
ORA-06512: at "SYS.DBMS_PLUGTS", line 1503

**To DSS database (Node 13 to 16)**
Create User : POLREC
Import Meta data
Alter TBS to Read Write
Transportable Tablespace

Steps at source database ‘PROD’

SQL> EXECUTE
    DBMS_TTS.TRANSPORT_SET_CHECK('POL1, POL2, POL3, POL4', TRUE);

SQL> SELECT * FROM TRANSPORT_SET_VIOLATIONS;

SQL> ALTER TABLESPACE POL1 READ ONLY;

$ imp parfile=imp_polrec.par

Copy all the Files at o/s level

cp /usr1/oracle/rw/DATA2/POL1_file1.dbf
    /mnt/ps/oradata5/TTS

cp /usr1/oracle/rw/DATA2/POL1_file2.dbf
    /mnt/ps/oradata5/TTS

cp /usr1/oracle/rw/DATA2/POL1_file3.dbf
    /mnt/ps/oradata5/TTS etc ..

At Source Database :
Create USER/schema on Destination Database ‘DSS’

$ imp parfile=imp_polrec.par

$ cat imp_polrec.par
TRANSPORT_TABLESPACE=y
FILE=polrec_metadata.dmp
TABLESPACES=(POL1, POL2, POL3, POL4)
USERID='sys/sys as sysdba'
DATAFILES=(
    '/mnt/ps/oradata5/TTS/POL1_file1.dbf',
    '/mnt/ps/oradata5/TTS/POL1_file2.dbf',
    '/mnt/ps/oradata5/TTS/POL1_file3.dbf',
    '/mnt/ps/oradata5/TTS/POL2_file1.dbf',
    '/mnt/ps/oradata5/TTS/POL2_file2.dbf',
    '/mnt/ps/oradata5/TTS/POL2_file3.dbf',
    '/mnt/ps/oradata5/TTS/POL3_file1.dbf',
    '/mnt/ps/oradata5/TTS/POL3_file2.dbf',
    '/mnt/ps/oradata5/TTS/POL3_file3.dbf',
    '/mnt/ps/oradata5/TTS/POL4_file1.dbf',
    '/mnt/ps/oradata5/TTS/POL4_file2.dbf',
    '/mnt/ps/oradata5/TTS/POL4_file3.dbf' )
TTS_OWNERS=(POLREC)
FROMUSER=(POLREC)
TOUSER=(POLREC)

$ cat exp_polrec.par
TRANSPORT_TABLESPACE=y
FILE=polrec_metadata.dmp
TABLESPACES=(POL1, POL2, POL3, POL4)
TTS_FULL_CHECK=Y
USERID='sys/sys as sysdba'
Server Parameter File

Use in preference to a pfile

- SPFILE provides a common repository of init parameters
- Stored in a binary file
- Parameters stored in a server parameter file are persistent
- Combine all of your instance specific init parameter files into a single init parameter file
- SPFILE is located on a shared file available to all instances
- ALTER SYSTEM statement allows you to set, change, or delete
- client-side parameter files (pfiles) increases your parameter administration overhead.
Streams in RAC

Issues

• Uses Archive Logs, rather than Redo Logs to propagate with a potential for delayed replication

• If instance dies where Capture process running, you have to restart again on a surviving instance
How to monitor IO Activity

In a large Cluster with multiple Databases and Multiple Instances, it becomes difficult to separate the IO activity and assess the performance impact

- Real challenge will be to assess
- We need a tool which is flexible enough
- That can give statistics component wise
- Is it Cluster Aware and RAC database aware !!

PolyServe Matrix Server comes with very useful a tool: “mxmodstat”

- It is cluster aware
- Made for Oracle Clustered RAC Databases
- It knows the Clustered RAC Databases
- ODM compliant
How to monitor IO Activity

A sample output:
- Monitoring IO activity on account of 6 instances of DSS database

```
 rac11 oracle $ mmcdnstat -i60 -l dss1 dss2 dss3 dss4 dss5 dss6

<table>
<thead>
<tr>
<th></th>
<th>dss1</th>
<th>dss2</th>
<th>dss3</th>
<th>dss4</th>
<th>dss5</th>
<th>dss6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Syn</td>
<td>Asyn</td>
<td>KB/s</td>
<td>Ave</td>
<td>n</td>
<td>Syn</td>
</tr>
<tr>
<td>1</td>
<td>552</td>
<td>69689</td>
<td>11</td>
<td>3</td>
<td>1065</td>
<td>132862</td>
</tr>
<tr>
<td>2</td>
<td>569</td>
<td>71836</td>
<td>11</td>
<td>1</td>
<td>642</td>
<td>807705</td>
</tr>
<tr>
<td>3</td>
<td>575</td>
<td>72520</td>
<td>11</td>
<td>1</td>
<td>546</td>
<td>687565</td>
</tr>
<tr>
<td>4</td>
<td>560</td>
<td>70722</td>
<td>11</td>
<td>0.35</td>
<td>348</td>
<td>43814</td>
</tr>
<tr>
<td>5</td>
<td>571</td>
<td>72084</td>
<td>11</td>
<td>0.30</td>
<td>250</td>
<td>32444</td>
</tr>
<tr>
<td>6</td>
<td>568</td>
<td>71115</td>
<td>12</td>
<td>3</td>
<td>1110</td>
<td>139035</td>
</tr>
<tr>
<td>7</td>
<td>567</td>
<td>70902</td>
<td>12</td>
<td>2</td>
<td>586</td>
<td>72805</td>
</tr>
<tr>
<td>8</td>
<td>563</td>
<td>70415</td>
<td>12</td>
<td>2</td>
<td>565</td>
<td>72786</td>
</tr>
<tr>
<td>9</td>
<td>561</td>
<td>70254</td>
<td>12</td>
<td>2</td>
<td>553</td>
<td>68794</td>
</tr>
<tr>
<td>10</td>
<td>552</td>
<td>66295</td>
<td>12</td>
<td>2</td>
<td>540</td>
<td>67265</td>
</tr>
<tr>
<td>11</td>
<td>574</td>
<td>71140</td>
<td>12</td>
<td>2</td>
<td>575</td>
<td>72184</td>
</tr>
<tr>
<td>12</td>
<td>571</td>
<td>71176</td>
<td>11</td>
<td>1</td>
<td>564</td>
<td>70217</td>
</tr>
</tbody>
</table>

rac11 oracle $ date
Tue Aug 12 12:27:52 EDT 2003
```

Great Tool ..

rac1 oracle  $ mxodmstat -h
mxodmstat: invalid option -- h
PolyServe MxS Oracle9i Option statistics V1.0-0 / SuSE SLES 8 (powered by UnitedLinux 1.0) (i586) 07/17/2003-15:27:08 (SYMS,NON-OPT)
Usage: mxodmstat -l[v]
-or-
mxodmstat [-Q <query>][-p][-i <interval>][-c <count>][-s <select>...][-a <expand>...]-[I|D|N|A [names]]
Where:
- l List processes
- lv List processes and file summary
- Q <query> Perform query. Possible queries: "lgwr" "dbwr" "pqo"
- p Add percent of cluster to report
- i <interval> Number of seconds between samples. Defaults to 5.
- c <count> Do <count> iterations and quit
- s <select> Restrict to one or more categories:
  restrict by ProcType: "fg" "bg" "lgwr" "dbwr" "pqo"
  restrict by FileType: "small" "large" "olg" "alg" "tsort"
  restrict by Operation: "read" "write"
  restrict by WaitType: "sync" "async"
- a <expand> Expand by one or more categories: "proc" "file" "op" "wait"
- I [names] Select by instance name(s)
- D [names] Select by database name(s)
- N [names] Select by node name(s)
- A [names] Select by application name(s)
Dynamism of Nodes ..

Physical Linux Cluster: Cluster Manager running

PROD Instances 1 to 12

DSS Instances 13 - 14

Node 01
Node 02
Node 03
Node 04
Node 05
Node 06
Node 07
Node 08
Node 09
Node 10
Node 11
Node 12
Node 13
Node 14
Node 15
Node 16

Time1: 12 + 2 + 2 Instances
Time2: 8 + 6 + 2 Instances
Time3: 8 + 8 + 0 instances
Dynamic Node management

How do you do that : ?

- Ensure that all the nodes are in the physical cluster
- First pre-Create threads for all of the databases (PROD, DEV, DSS)
- Bring up instances only as needed. (Example : 12 + 2 + 2)
- Shutdown some instances and startup new instances as needed
- It is all script based !! Can even be automated ..... 

Environment

- PROD for OLTP activity
- DSS for data warehousing and analysis of data
- DEV for development access (sqlload data occasionally
- All are using the OMF based files
Stress Tests ..

Stress Test Scenario and observations

• 16 instances running for PROD (OLTP type) database
• Varying .. 100, 250 and 500 users per instance
• As the number of users go high, it is no longer linear
• Other factors kick in - In the end, vmmstat showed swapping – PGA accounted for bulk of Physical RAM

Results :

<table>
<thead>
<tr>
<th>Users Per Node</th>
<th>Total Users</th>
<th>Trans per min</th>
<th>% of linear Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>1600</td>
<td>19140</td>
<td>--</td>
</tr>
<tr>
<td>250</td>
<td>4000</td>
<td>47160</td>
<td>98 %</td>
</tr>
<tr>
<td>500</td>
<td>8000</td>
<td>70740</td>
<td>74 %</td>
</tr>
</tbody>
</table>

![Graph of 8000 User Stress Testing](image-url)
Advanced Tests .. suite-1

Higher Availability with RAC

• Start with PROD on 1-12, DSS on 13-14 and DEV on 15-16
• 3600 users on PROD nodes – with good amount of enqueue waits
• At 15:48:44 EDT, number of users (see below)

```
SQL> HOST date
Thu Jul 24 15:48:44 EDT 2003
SQL> @users

<table>
<thead>
<tr>
<th>MACHINE</th>
<th>COUNT(*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>rac1</td>
<td>325</td>
</tr>
<tr>
<td>rac10</td>
<td>312</td>
</tr>
<tr>
<td>rac11</td>
<td>312</td>
</tr>
<tr>
<td>rac12</td>
<td>312</td>
</tr>
<tr>
<td>rac2</td>
<td>312</td>
</tr>
<tr>
<td>rac3</td>
<td>312</td>
</tr>
<tr>
<td>rac4</td>
<td>312</td>
</tr>
<tr>
<td>rac5</td>
<td>312</td>
</tr>
<tr>
<td>rac6</td>
<td>312</td>
</tr>
<tr>
<td>rac7</td>
<td>312</td>
</tr>
<tr>
<td>rac8</td>
<td>312</td>
</tr>
<tr>
<td>rac9</td>
<td>312</td>
</tr>
</tbody>
</table>
```

After allowing this work load for 40 min, node-5 was abruptly switched off .......
Advanced Tests .. suite-1

Shows the reconfiguration of cluster manager
Advanced Tests .. suite-1

Shows alert_prod1.log during reconfig.

Results

• In 4 sec. OCMS reconfigured (16:43:47 to 16:43:51)

• Alert log shows : Node1 started IR and completed in 10 sec. (16:43:53 to 16:44:03)

Then Added Node-13 to Cluster DB
**Advanced Tests .. suite-1**

**Observations : During a period of 12 min ..**

- The OLTP portion of the FDC suffered a server failure on node rac5
- Users suffered a momentary pause in service during PolyServe Matrix Server, OCMS and Oracle instance recovery.
- Users on the remaining 11 nodes maintained their connections
- Node 13 was chosen to replace node 5 and since tnsnames.ora was set up appropriately, users were able to connect to the instance on that node just as soon as it was brought online.

*Detailed screen shots and study results are available in the White Paper*
Scalability with RAC

- DSS Query: Table scan throughput and to a lesser degree, processor bandwidth for filter, sort/join, grouping and aggregation. (with Parallel Query processing)
- For Test, employed Query of Varied Intensity – and un-tuned queries
- Focus on I/O throughput
- Started with PROD on 1-12, DSS on 13-14 and DEV on 15-16
- Went on adding two DSS Nodes at time
- Studies the physical MB transferred (via gv$ views)

Results in the next slide ..
Advanced Tests .. suite-2

2 DSS instances 13, 14 nodes

add 2 nodes

84 % scalability

4 DSS instance 13, 14, 15, 16 nodes

add 2 nodes

70 % scalability

6 DSS instances 11-16 nodes

* Detailed screen shots and study results are available in white paper
Advanced Tests .. suite-3

Slightly changed the test scenario

- DSS was initially set up on nodes rac11 and rac12. PROD was contained to nodes 1 through 10. Nodes 13 to 16 were a hot standby “pool” of server resources that could be dedicated to either PROD or DSS on demand
- When went from 2 to 4 nodes: 100% scalability was noticed
- Better parallelization!!
Advanced Tests .. suite-3

IO Monitoring Tool : ODM based ‘mxodmstat’

- mxodmstat was used to monitor all the instances of the DSS
- elements chosen for this command was “Total I/O”
- the dynamic nature of adding instances to service the query queue can be seen below screen capture (in three batches)
General Suggestions !! …

• work out number of nodes and assess the Physical / System resources

• Assuming that there is quite a lot flexibility
  • Start with a limited set of nodes with an option to plug in at later stage as demand grows
  • Consider new technology like Blade Architecture which gives total flexibility in growing

• Use best possible interconnects, storage switches etc.

• Plan the Shared Storage - what happens when you want to add Nodes. Does you infrastructure permit easier connectivity?

• Cluster Manager software - Stable and adoptive

• Then decide the type of Physical Structures for database use
  • want to use Cluster File System or settle down for raw-partitions

• Then plan for how many Databases with in the Cluster and how many Instances per database
Questions & Answers

Session ID # 37600

Case Study – RAC on 16 Node Linux Cluster

Thank you!!

Kevin Closson and Madhu Tumma