High Availability Infrastructure of Database Cloud: Architecture, Best Practices

Kai Yu
Oracle Solutions Engineering, Dell Inc.
About Me

• **Kai Yu, Senior Architect, Dell Oracle Solutions Lab**
  - 17 years Oracle DBA and Solutions Engineering
  - Specializing in Oracle RAC, Oracle VM and Oracle EBS
  - Oracle ACE Director, Oracle papers author/presenter
  - IOUG RAC SIG President (2009-2010)
  - Co-founder and board member of IOUG Virtualization SIG
  - Winner of 2011 OAUG Innovator of Year Award
  - Winner of 2012 Oracle Excellence Award: Technologist of the Year: Cloud Architect by Oracle Magazine
  - Oracle Blog: http://kyuoracleblog.wordpress.com/
Dell Global Solutions Engineering Lab
Dell Global Solutions Engineering Lab

- Dell | Oracle Solutions Engineering: [www.dell.com/oracle](http://www.dell.com/oracle)
  - Dell | Oracle Partnership offers customers complete a solution
  - Solutions Deliverables:
    - Tested and Validated Configuration
    - Deployment Guild
    - Oracle Advisor
    - Detailed Oracle Solutions Deliverables List (SDL)
  - Dell Oracle Solutions Reference Configuration
    - Oracle RAC and High Availability
    - Oracle Virtualization Technology
    - Oracle Enterprise Manager & Cloud
    - Oracle Applications
    - Proof of Concepts
Agenda

- High Availability for Database Cloud
- High Availability Architecture Design
- Best Practices of Infrastructure Configuration
- Reduce Planned Downtime
- Oracle Maximal Availability Architecture (MAA)
- Q&A
High Availability for Database Cloud
High Availability for Database Cloud

• Consolidating databases into a private database cloud
  – Challenges to the traditional computing architecture
  – Consolidate multiples databases on private cloud Infrastructure
  – Provide Platform as a Service for database applications
  – Provide Database Services based on shared based on cloud Infrastructure.
  – Integrate all the resources to allow provisioning on demand: dynamically provisioning to meet the workload needs
High Availability for Database Cloud

- **What is meant by High Availability?**
  - Defined by Service Level Agreement(s) (SLAs):
  - HA goal is to meet SLA requirement
  - Balance between the availability and implementation cost
  - SLA: for example, 99.95%, annual 4 hrs 22 minutes downtime
    Downtime window: first Saturday: 8pm-10pm every quarter

- **High Availability SLAs in a Cloud Environment**
  - Consolidating many databases in a single cloud infrastructure
  - Great business impact due to the infrastructure downtime
  - Databases may have different SLAs for different business:
    - Different business requirements
    - Different time zones
    - Infrastructure downtime means downtime for all the databases
    - Very difficult to find the downtime for maintenance that meets every SLA
High Availability for Database Cloud

- **Causes of Impacting System Availability**
  - Service outage by unplanned downtime: hardware or software failure, human error, nature disaster, etc.
  - Service disruption by planned downtime: hardware/software upgrade, patching and migration from old system to new system
  - Service performance degradation: violate performance SLA for example, 99% transactions finished in a 2 seconds window

- **Architect a High Availability Cloud Infrastructure**
  - Design a highly available cloud architecture
  - Architect hardware infrastructure to reduce unplanned outage
  - Use configuration and implementation best practices for HA
  - Administration and troubleshooting tips for High Availability
  - Methods/Options to minimize planned downtime
  - Establish the pre-active real time monitoring system
High Availability Architecture Design
High Availability Architecture Design

- **Oracle Real Application Clusters: Active-active cluster database**
  - Protect database availability against up to N-1 server failure with Virtual IP (VIP) automatic failover by Oracle clusterware
  - Add node or remove node based on demand of capacity
  - Reduce planned downtime for hardware, OS, software upgrade
  - Application load balancing
  - Key components: Interconnect heartbeat, and shared storage
  - A recorded Swingbench RAC high availability demo
High Availability Architecture Design

- **Oracle RAC one node database: active-passive database**
  - Single node database on clusterware; no load balancing
  - Require the same system architecture as RAC: network/storage
  - Install Oracle 11gR2 Grid Infrastructure and RAC on all nodes
  - Specify RAC One node Database during the database creation
High Availability Architecture Design

- Protect database against server failure with VIP automatic failover
- Reduce planned downtime: Online relocating database to another node for hardware, OS, database software upgrade
  
  $\texttt{srvctl relocate database} -d <\texttt{dbname}> -n <\texttt{nodename}> -w -15

- RAC and RAC one node can be switched back and forth
  Convert to RAC from RAC one node
  
  $\texttt{srvctl convert database} -d <\texttt{dbname}> -c \texttt{RAC} -n <\texttt{nodename}>
  $\texttt{srvctl add instance} -d <\texttt{dbname}> -i <\texttt{instname}> -n <\texttt{nodename}>

  Convert to RAC one node from RAC
  
  $\texttt{srvctl remove instance} -d <\texttt{dbname}> -i <\texttt{instname}>
  $\texttt{srvctl convert database} -d <\texttt{dbname}> -c \texttt{RAC} -n <\texttt{hostname}>

- Refer to my C#12 presentation on RAC One Node for more details
- Why RAC one node:
  Low cost alternative to RAC: $10k/p vs. $17k/p (SE) vs $23k/p (EE)
  two nodes 2 X8 core servers, $80K vs. $512K(EE) vs. $272k(SE)
High Availability Architecture Design

- Consolidate multiple database on a database cloud
  - Multiple databases (Single/RAC/ RAC one node) on a single cluster
  - Possible multiple versions Oracle Homes on 11g R2 Clusterware:
  - Example: An EBS Database Grid has 100 Oracle EBS databases *
    (refer to my OOW11 presentation)
High Availability Architecture Design

• **Oracle VM provides HA against physical server failure**
  – Database server runs on virtual machine
  – Virtual machines run on a pool of VM servers (VM server pool)
  – Enable HA on the VM server pool as well as on virtual machines
  – Virtual Machine images/storage stored in shared storage
  – Failover to another VM server to reduce unplanned downtime
High Availability Architecture Design

- Oracle VM provides live migration to prevent planned downtime
  - Live migration to another VM server without downtime of VM
  - Migrating the virtual machine to other physical server for:
    - Physical server maintenance
    - Balancing the workload by moving VM to another physical server
    - Zero downtime for the virtual machine: OS & applications
High Availability Architecture Design

- **Oracle RAC one node works with Oracle VM:**
  - RAC One Node fully supported in Oracle VM environment.
  - The database will failover to another VM if this VM fails.
  - Oracle VM live migration for hardware maintenance.
  - RAC One node online relocation for VM and OS maintenance.
High Availability Architecture Design

- **Oracle RAC database works on Oracle VM**
  - Two possible HA configurations: HA by VM and HA by RAC
  - Only HA by RAC is supported and HA by VM is not supported for RAC database configuration
High Availability Architecture Design

- **Oracle RAC Database works on Oracle VM**
  - Consoliation of multiple RAC databases in fewer physical servers
  - Each Database instance runs on its own VM independently
  - One database instance node eviction will not impact other databases
  - Less impact of downtime during OS and Oracle software upgrade
High Availability Architecture Design

- **Protect Data Availability**
  - Oracle flashback can reverse the data lost due to human error
  - Data recovery at all levels: row, table, even the entire database
  - Protect from data corruption:
    - Oracle ASM lib adds integrity metadata for data integrity check
    - Enable corruption on primary/standby database by setting DB_ULTRA_SAFE=DATA_AND_INDEX
    - RMAN backup verifies all data blocks to ensure no corrupted blocks are saved to the backup files
    - ASM block repair: relocates the valid block from the mirrored copy to an uncorrupted portion of the disk
  - Raid configuration in database storage: Rail 1, Raid 10, Raid 5, etc
  - RMAN backup and recovery

- **Data Guard to protect database again site failure**
  - Physical standby database kept in sync with the primary database with Oracle data guard
High Availability Architecture Design

- Protect Modes: maximal protection; maximum availability; maximum performance
- Switchover/ Failover to standby database
- Active Data Guard enables queries against the standby database while redo logs are applied in real time
- Disaster recovery solutions across different data centers

- Oracle RAC + Oracle Data Guard: protect database against server failure, storage failure and site failure
Infrastructure Configuration Best Practices
Infrastructure Configuration Best Practices

- **High Availability Storage Infrastructure**
  - Storage HA plays a key role in the infrastructure HA
  - Redundant IO paths from servers to storage array
  Server <-> HBAs <-> Switches <-> Storage Controllers<-> Disk Enclosures
Infrastructure Configuration Best Practices

– Redundant IO paths from servers to storage array
  
  Multiple physical paths: (HBA1, Switch1, Volume), (HBA1, Switch2, Volume),
  (HBA2, Switch1, Volume), (HBA2, Switch2, Volume),

  A storage volume should be able to fail over to another controller

  Software multipathing: Two redundant IO paths seen in OS:
  
  to the same storage volume: /dev/sdb and /dev/sdc
  
  Linux Device Mapper (DM) or from storage vendors

  Group them together to alias Data1 using multipathing software

  ```
  multipath {
    wwid 36090a028e093fc906099540639aa2149 #<---- for sdb and sdc
    alias Data1 }
  }
  
  #service multipathd restart
  
  ls -lt /dev/mapper/*
  
  brw-rw---- 1 root disk 253, 8 Feb 18 02:02 /dev/mapper/votingdisk1
  
  – SAN Disk Array RAID for Redundancy: Raid 10/5 Configuration

  – Redundant Storage Controllers for high availability

  – Oracle ASM diskgroup redundancy settings
Infrastructure Configuration Best Practices

- ASM failure group: mirroring/redundancy level setting:
  - External: no ASM mirroring, rely on external redundancy
  - Normal: 2-way mirroring, 2 failure groups
  - High: 3-way mirroring three failure groups.

- ASM diskgroup for OCR and Voting disks:
  - External Redundancy: relay on external RAID configuration
  - Normal Redundancy (3 failure groups): 3 voting disks
  - High Redundancy (5 failure group): 5 voting disks
  - 1 OCR + up to 5 copies: one per diskgroup
  - A quorum failure group: only used for OCR and votingdisk
Infrastructure Configuration Best Practices

- Network High Availability Configuration
  - fully redundant interconnects for cluster configuration
  - Network bonding vs Oracle Highly Available Virtual IP (HAIP)
  - Dedicated switches for private interconnects

- Redundant Hardware Infrastructure for Cluster Database
Infrastructure Configuration Best Practices

- **Managing Clusterware**
  - Create an additional copy of OCR: (One copy per diskgroup)

```
[root@k2r720n1 bin]# ./ocrconfig -add+OCR
[root@k2r720n1 bin]# ./ocrcheck
Status of Oracle Cluster Registry is as follows:
  Version : 3
  Total space (kbytes) : 262120
  Used space (kbytes)  : 3448
  Available space (kbytes) : 258672
  ID     : 1318929051
  Device/File Name     : +OCR
  Device/File integrity check succeeded
```

- Remove a diskgroup for OCR `ocrconfig -delete old_location`
- Backup OCR: automatic: `ocrconfig -showbackup`
- Backup OCR: manual: `ocrconfig -manualbackup`
- Restore OCR: stop clusterware: `crsctl stop`
  
  run restore command: `crs ocrconfig -restore`
- Diagnose OCR problem: `OCRDUMP` and `OCRCHECK`
- **Export OCR**: `ocrconfig -export`
- **Import OCR**: `ocrconfig -import`
Infrastructure Configuration Best Practices

- **Set odd number of voting disks, 1/3/5 by defaults**
- Store Voting disks on ASM: only one diskgroup for voting disk
- One voting disk per one failure group for redundancy

```
SQL> select path ||'' || name ||'' || GROUP_NUMBER ||'' || FAILGROUP
  2   from v$asm_disk where name like 'OCR%';

PATH||''||NAME||''||GROUP_NUMBER||''||FAILGROUP
-----------------------------------------------------------------------
ORCL:OCR1 OCR1 4 OCR1
ORCL:OCR2 OCR2 4 OCR2
ORCL:OCR3 OCR3 4 OCR3
ORCL:OCR4 OCR4 4 OCR4
ORCL:OCR5 OCR5 4 OCR5

SQL> !crsctl query css votedisk
## STATE File Universal Id File Name Disk group
--- ------ ------------------ -------------
1. ONLINE 6f26b70a88a84f56bf41a1d4d6e87661 (ORCL:OCR5) [OCRVOTDISK]
2. ONLINE 19546d7b121e6fbfc6f224f4fd9de7 (ORCL:OCR4) [OCRVOTDISK]
3. ONLINE 30e8391c66654f82bfc5bc8ed239683c (ORCL:OCR3) [OCRVOTDISK]
4. ONLINE 06f600a529ad4f02bf4508b04c58683 (ORCL:OCR2) [OCRVOTDISK]
5. ONLINE 6bcd818a35d4f3ebf95aa31831c8f5 (ORCL:OCR1) [OCRVOTDISK]
Located 5 voting disk(s).
```

- Voting disk is backed up in OCR automatically in 11gR2 if
- crsctl commands to add, delete, replace voting disk.
- Restore voting disk: Restore OCR if OCR corrupted
  - Start crs in exclusive mode: `crsctl start crs -excl`
  - Add new votingdisk: `crsctl replace votedisk +asm_disk_group`
  - Restart crs
Infrastructure Configuration Best Practices

- **Clusterware Troubleshooting**
  - *Clusterware health Verification Utility: CLUFY*
  - *Clusterware utility: crsctl for check crs status and start/stop*
  - *Log files: $GRID_HOME/log/<host>/alert<host>.log and $GRID_HOME/log/<host>/<process>/log*

```bash
[grid@k4r815n1 ~]$ crsctl check crs
CRS-4638: Oracle High Availability Services is online
CRS-4537: Cluster Ready Services is online
CRS-4539: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
[grid@k4r815n1 ~]$ crsctl check cluster -all
*******************************************************************************
k4r815n1:
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
*******************************************************************************
k4r815n2:
CRS-4537: Cluster Ready Services is online
CRS-4529: Cluster Synchronization Services is online
CRS-4533: Event Manager is online
*******************************************************************************
```

```
[grid@k4r815n1 k4r815n1]$ pwd
/opt/app/11.2.0/grid/log/k4r815n1
[grid@k4r815n1 k4r815n1]$ ls -lrt | cut -d total
```

```
--- grid oinstall diskmon
--- root oinstall ohasd
--- grid oinstall cssd
--- root oinstall crsd
--- grid oinstall gpnpg
--- grid oinstall srvm
-rw-rw-r-- root root alertk4r815n1.log
--- grid oinstall client
--- root oinstall ctssd
--- root oinstall gnsd
--- grid oinstall racg
--- grid oinstall mdnsd
--- grid oinstall gipcd
--- grid oinstall evmd
--- root oinstall agent
--- grid oinstall admin
```
Node Eviction:
Cluster split brain condition: a node failure partitions the cluster into multiple sub-clusters without knowledge of the existence of others.
Possible causes: not responding network heartbeat, disk heartbeat, a hung node or hung ocssd.bin process.
- Consequence: data collision and corruption
- IO fencing: fencing the failed node off from all the IOs: STOMITH (Shoot The Other Machine In The Head) algorithm
- Node eviction: pick a cluster node as victim to reboot.
  Always keep the largest cluster possible up, evicted other nodes:
two nodes: keep the lowest number node up and evict other
- Two CSS heartbeats and misscounts to detect node eviction
  1. Network HeartBeat (NHB) over private interconnect to check node membership; misscount: 30 secs
  2. Disk heartbeat: between the cluster node and voting disk: misscount: 200 secs
Infrastructure Configuration Best Practices

- Troubleshooting node eviction
  - Common causes for OCSSD eviction:
    - network failure latency exceeds CSS miscount 30 seconds
    - access disk issue: CSS miscount 200 sec OCSSD failure,
  - Common causes of: CSSDAGENT OR CSSDMONITOR eviction: OS scheduler problem caused by OS locked up in driver or hardware or the heavy loads; thread of CSS demon hung
  - Review the log files, refer to metalink note [1050693.1]

Node Eviction Diagnosis Examples
- Case 1: Node 2 was rebooted in a 2-node 11g R2 cluster on Linux: OCSSD log: $CRS_HOME/log/<hostname>/cssd/ocssd.log file in Node1:

```
2010-11-23 17:11:55.221: [ CSSD] [1342572864] clssmnPollingThread: node k4r815n2 (2) at 75% heartbeat failure, removal in 7.500 seconds
2010-11-23 17:11:59.231: [ CSSD] [1353062720] clssmnSendingThread: sending status msg to all nodes
2010-11-23 17:11:59.231: [ CSSD] [1353062720] clssmnSendingThread: sent 5 status msgs to all nodes
2010-11-23 17:12:00.232: [ CSSD] [1342572864] clssmnPollingThread: node k4r815n2 (2) at 90% heartbeat failure, removal in 2.490 seconds, seedhbimpd 1
2010-11-23 17:12:02.718: [ CSSD] [1342572864] clssmnPollingThread: Removal started for node k4r815n2 (2), flags 0x3040c, state 3, wt4c 0
2010-11-23 17:12:02.718: [ CSSD] [1342572864] clssmnDiscHelper: k4r815n2, node(2) connection failed, endp (0x264), probe (0x10000000), ninf->endp 0x264
2010-11-23 17:12:02.718: [ CSSD] [1342572864] clssmnDiscHelper: node 2 clean up, endp |0x264|, init state 5, cur state 5
```
Infrastructure Configuration Best Practices

- Case 2: node 1 reboot: 
  $CRS_HOME/log/<hostname>/cssd/ocssd.log file

- Case 3: One node rebooted once a month in 11 nodes cluster:
  
  ------ /var/log/message:
  Jul 23 11:15:23 racdb7 logger: Oracle CRS failure. Rebooting for cluster integrity

  ------ OCSSD log: 
  clssnmPollingThread: node racdb7 (7) at 90% heartbeat fatal, eviction in 0.550 seconds

  …
  clssnmDoSyncUpdate: Terminating node 7, racdb7, misstime(60200) state(3)
Reduce Planned Downtime
Reduce Planned Downtime

- **Reduce Upgrade Downtime**
  - Online patching: patches that can applied to a running database instance [MOS #761111.1] for database one-off patches
  - Rolling upgrade in RAC: one instance is shutdown for upgrade while other are functioning:
    - rolling upgrade of hardware, firmware, BIOS, OS
    - rolling upgrade of clusterware: CRS upgrade
    - rolling upgrade of ASM
    - rolling upgrade of RAC: database patch, PSU, CPU patches
  - Rolling Database Upgrade to reduce downtime:
    - Use Data Guard SQL Apply feature
    - Use 11g Transient Logical Standby feature
    - MOS note #949322.1
Reduce Planned Downtime

1. Start with physical standby
2. Convert to logical standby and queued all redo data
3. Upgrade the logical standby
4. Apply all redo data to Standby
5. Switch over roles
6. Upgrade the old production (the new standby)
7. Apply all redo data to the new standby (old production)
8. Switch over roles
9. Convert back to physical standby
Reduce Planned Downtime

- **System downtime during the migration**
  - Server migration, platform migration (OS), storage migration, database migration
  - Database migration involves migrating all the data to the new system
  - Significant downtime involved. Normally it requires the application downtime during the migration process.

- **Options and considerations to reduce the migration downtime**
  - Applications: build the new system and do the last minute switch
  - Database migration includes the last minute data to the new system
    - Export/import and data pump: need a long database time
    - Transportable tablespaces for across platform migration
    - Data Guard based migration method for across server migration
    - Oracle Golden Gate for online migration
    - Use ASM disk rebalance for online storage migration: online migrate database from old SAN storage to new SAA.
    - Storage replication method by the storage vendor
Oracle Maximal Availability Architecture (MAA)

- Unplanned Downtime
  - Server Availability
  - Data Availability

- Planned Downtime
  - System Changes
  - Data Changes
  - App Changes

Real Application Clusters
- Flashback
- RMAN & Oracle Secure Backup
- ASM
- Data Guard
- GoldenGate

Oracle MAA Best Practices
- Online Reconfiguration
- Rolling Upgrades
- Online Redefinition
- Edition-based Redefinition

Reference:
1. Oracle Database 11g Release 2 High Availability, Oracle whitepaper, March 2012
3. Oracle Data Guard 11g Data Protection and Availability for Oracle Database
Thank you and QA
Contact me at kai_yu@dell.com or visit my Oracle Blog at http://kyuoracleblog.wordpress.com/

**My Speaking Topics at Oracle OpenWorld 2012**

**I’m Speaking**

Oracle OpenWorld 2012 is just around the corner. This is my 7th consecutive year of speaking at this world largest IT event. This year my presentations are related to high availability, database cloud, Oracle virtualization and deploying Oracle virtual assembly in a self service cloud environment. The followings are my speaking sessions:

1. UGF700 – Oracle on Oracle VM: Expert Panel, Sunday Sept 30, 12:30pm-2:00pm, Moscone West 2012.
   On this panel I will share some experience of configuring an Oracle VM virtual infrastructure and an infrastructure cloud based on commodity hardware with Oracle VM3.1.1 and Oracle Enterprise Manager 12.1.

2. CONS875 – High-Availability Infrastructure of the Database Cloud: Architecture and Best Practices, Monday Oct 1, 3:15pm-4:15pm, Moscone West- 3020
   This session will start with the architecture design and then discuss various techniques and options for achieving high availability including Oracle Clusterware/Oracle Real Application Clusters (Oracle RAC), Oracle Data Guard, and Oracle Maximum Availability Architecture. Then it will explore the configuration best practices for ensuring fault tolerance of the entire stack of the infrastructure. It will also cover the methods for reducing downtime during events such as system upgrades and migration and how to do troubleshooting during system outages.

3. CONS205 – Simplify App Deployment in the Cloud with Virtual Assemblies and Oracle Enterprise Manager 12c: Wednesday, Oct3, 5:00pm-6:00pm Moscone South- 307
   In this session I will share some experience from my Infrastructure cloud project that I have been working in last few months. I will start with how to create an Oracle virtualization assembly with Oracle Virtual Assembly Builder 11g Release 1 in the following process: