

CLUSTERING DATABASE MENGUNAKAN TEKNOLOGI DBMS

Rengga Asmara, Isbat Uzzin Nadhori, Moh. Nizam Thursana

Jurusan Teknik Informatika

Politeknik Elektronika Negeri Surabaya

Institut Teknologi Sepuluh Nopember

Kampus PENS-ITS Keputih Sukolilo Surabaya 60111

Telp (+62)31-5947280, 5946114, Fax. (+62)31-5946114

Email : rengga@eepis-its.edu, isbat@eepis-its.edu, thursana@gmail.com

ABSTRAK

Salah satu keunggulan Oracle sebagai DBMS (Database Management System) adalah mampu menangani banyak sekali data yang masuk dan keluar dari database. Dengan adanya keunggulan tersebut, database Oracle digunakan oleh banyak perusahaan yang mempunyai kapasitas data yang sangat besar. Tapi, jika pada Server yang menghubungkan antara client dan database Oracle mengalami traffic yang sangat tinggi, maka akan terjadi kemacetan pada Server tersebut yang bisa membuat database menjadi down (mati). Hal tersebut bisa membuat perusahaan tersebut mengalami kerugian besar.

Oracle mengatasi masalah tersebut dengan mengimplementasikan sebuah teknologi Clustering Database yang disebut Real Application Cluster (RAC) dimana pengaksesan database Oracle dibagi menjadi beberapa server dengan tujuan server - server tersebut mampu mengatasi masalah traffic yang sering terjadi apabila banyak sekali client yang mengakses ke database.

Kata kunci : RAC, oracle, database.

1. PENDAHULUAN

1.1 Latar Belakang

Salah satu keunggulan Oracle sebagai DBMS (Database Management System) adalah mampu menangani banyak sekali data yang masuk dan keluar dari database. Dengan adanya keunggulan tersebut, database Oracle digunakan oleh banyak perusahaan yang mempunyai kapasitas data yang sangat besar. Tapi, jika pada Server yang menghubungkan antara client dan database Oracle mengalami traffic yang sangat tinggi, maka akan terjadi kemacetan pada Server tersebut yang bisa membuat database menjadi down (mati). Hal tersebut bisa membuat perusahaan tersebut mengalami kerugian yang sangat besar.

Pada penelitian ini akan dibuat sebuah sistem database dimana dengan menggunakan komputer yang lebih dari satu sebagai server (instance), dan satu komputer lagi sebagai database untuk menyimpan data. Komputer server tersebut saling terkoneksi satu sama lain. Arsitektur seperti ini bisa disebut juga dengan cluster. Sedangkan di teknologi Oracle, sistem ini biasa disebut dengan RAC (Real Application Cluster). Diharapkan setiap perubahan data dikoordinasikan dengan baik

oleh instance - instance tersebut. Sehingga antara satu instance dengan instance yang lain bisa saling mengambil informasi dan data image dengan bebas.

Pada arsitektur ini memungkinkan user dan aplikasi mempunyai keuntungan dari tenaga memproses pada mesin yang banyak. Arsitektu RAC juga mempunyai kelebihan, seperti apabila terjadi tabrakan pada sebuah sistem atau sistem tersebut tidak siap, aplikasi tetap bisa mengakses database dengan menggunakan instance lain yang masih ada.

1.2 Rumusan Permasalahan

Permasalahan yang akan dibahas dalam penelitian ini adalah bagaimana melakukan instalasi RAC (Real Application Cluster) dengan menggunakan 3 PC, dengan rincian, 2 PC sebagai komputer server (instance) dengan sebuah shared storage.

Batasan-batasan pada penelitian ini adalah pada penelitian ini akan digunakan:

1. Real Application Cluster
2. Oracle 10g
3. Sample data dengan record banyak yang dibuat secara random.

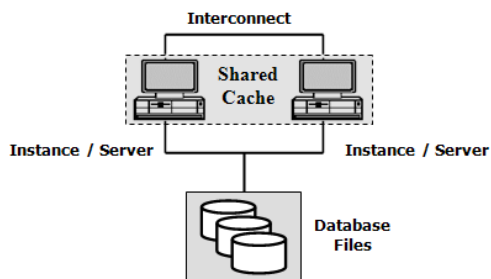
4. Sebagai simulasi akan dibuat sebuah aplikasi yang melakukan request pada database.

1.3 Batasan Permasalahan

Pada penyelenggaraan penelitian ini, batasan permasalahannya adalah :

1. Data input yang digunakan hanya bersifat simulasi, tidak berupa data real.
2. Hasil yang akan dianalisa adalah kinerja dari server database yang melayani request banyak dan bagaimana server melakukan failover.
3. Hasil yang akan dianalisa juga perbedaan kinerja antara database single instance dan sistem RAC

2. PERANCANGAN SISTEM



Gambar 2.1 perancangan sistem secara logika

2.1. Gambaran Sistem

Perancangan sistem secara logika bisa terlihat seperti gambar di atas. Secara logika sistem cluster database ini terdiri dari 2 buah server yang pada oracle server tersebut menjalankan instance pada tiap – tiap server. Instance tersebut memproses permintaan dari client.

Masing – masing server juga mengakses database yang sama. Database tersebut bisa diakses oleh kedua server. Sehingga jika pada salah satu server melakukan perubahan pada database, maka server lain bisa mengakses data yang sama dengan hasil update dari salah satu server tadi.

2.2 Pemasangan Hardware

Hardware akan terpasang pada sistem jaringan. Komputer server terhubung satu sama lain dengan switch. Dengan salah satu server terpasang sebuah media penyimpanan yang bisa diakses bersama – sama oleh semua server.

2.3 Persiapan pembuatan sistem

Pada pembuatan sistem cluster, terdapat beberapa hal yang harus terpenuhi. Seperti tertulis di bawah ini :

1. Cek semua aspek pada cluster dan tiap titik. Semua parameter kernel system operasi, koneksi dan pengaturan jaringan, paket – paket yang harus dipenuhi, dan lainnya yang diperlukan untuk dicek. Jika error atau kesalahan kecil ditemukan, maka akan keluar peringatan untuk mengoreksi ulang kesalahan sebelum melanjutkan proses instalasi.
2. Mendefinisikan alamat pada tiap titik, titik virtual IP dan interkoneksi secara private.
3. Mendefinisikan dua lokasi file Oracle Cluster Registry (OCR).
4. Mendefinisikan dua lokasi file untuk Voting Disk.

Untuk itu perlu dipersiapkan sebelum melakukan pembuatan sistem. Persiapan tersebut adalah :

2.3.1 Hardware Requirement

1. Sedikitnya dibutuhkan minimal 1 gigabyte (GB) memory fisik. Untuk mengetahui besar memory fisik, ketikkan perintah berikut pada terminal : `grep MemTotal /proc/meminfo`
2. Minimal dibutuhkan 1 GB ruang kosong pada swap atau 2 kali dari besar memori fisik. Pada sistem dengan memori 2 GB atau lebih, ruang kosong pada swap bias ditentukan antara 1 atau 2 kali dari besar memori fisik. Untuk mengetahui keterangan ruang kosong pada swap, ketikkan perintah berikut pada terminal : `grep SwapTotal /proc/meminfo`
3. Minimal 400 megabytes ruang kosong yang harus tersedia pada directory /tmp . untuk bias mengetahui besar ruang kosong yang ada pada directory /tmp, ketikkan perintah berikut pada terminal : `df -k /tmp`.
4. Lebih dari 4 GB ruang kosong yang harus tersedia untuk Oracle Database 10g Software, tergantung pada tipe instalasi. Perintah `df` bias digunakan untuk mengetahui ruang kosong tersebut.

2.3.2 Network requirements

1. Masing – masing computer mempunyai minimal 2 NIC, dimana 1 NIC untuk

public network dan NIC yang lain digunakan untuk private network atau sebagai interkoneksi antar computer.

2. Untuk public network, masing – masing NIC harus mendukung TCP/IP. Untuk private network, interkoneksi harus mendukung UDP menggunakan *high-speed* NIC dan switch yang mendukung TCP/IP.
3. Sebelum proses instalasi, masing – masing titik membutuhkan alamat IP dan nama pada masing – masing titik didaftarkan pada DNS atau file */etc/hosts* untuk masing – masing titik. Satu alamat virtual IP yang tidak terpakai dan sebuah nama VIP yang didaftarkan pada DNS atau file */etc/hosts* yang dikonfigurasi untuk NIC utama dibutuhkan pada masing – masing titik. Virtual IP harus terletak pada subnet yang sama dengan alamat IP public.
4. Konfigurasi Virtual IP dilakukan setelah proses instalasi selesai. Jika terdapat sebuah titik yang fails, maka virtual IP ini digunakan untuk mengalihkan ke titik yang lain.

2.3.3 RAC Network Software Requirements

1. Masing – masing titik pada cluster membutuhkan sebuah interkoneksi yang mendukung Cache Fusion, dan TCP/IP untuk kebutuhan Oracle Clusterware polling.
2. Kebutuhan terbaik untuk UDP adalah minimal sebuah gigabit Ethernet untuk perfoma terbaik, kabel crossover tidak mendukung UDP (gunakan switch dengan kecepatan tinggi), tambahkan UDP buffer size pada system operasi, mengaktifkan UDP checksumming.

2.3.4 Package requirements

1. Tergantung pada cluster yang akan diinstal, terdapat paket – paket yang harus sudah terinstal pada system operasi. The Oracle Universal Installer (OUI) melakukan pengecekan pada system, apakah paket – paket yang dibutuhkan sudah terpenuhi atau belum. Untuk memastikan pengecekan yang dilakukan oleh OUI berhasil, disarankan untuk melakukan pengecekan sendiri sebelum menjalankan OUI.

2. Untuk mengetahui paket – paket yang sudah diinstal, ketikkan perintah berikut pada terminal : `rpm -q <nama paket>` atau `rpm -qa | grep <sekmen nama paket>`
3. Sebagai contoh, untuk melakukan pengecekan pada gcc, ketikkan pada perintah berikut pada terminal:
 - `rpm -qa |grep compat`
 - dan hasil yang akan keluar kurang lebih seperti di bawah ini :
 - `compat-db-4.0.14.5`
 - `compat-gcc-7.3-2.96.122`
 - `compat-gcc-c++-7.3-2.96.122`
 - `compat-libstdc++-7.3-2.96.122 is not installed`
 - `compat-libstdc++-devel-7.9-2.96.122 is not installed`
4. Jika ada yang belum terinstal, bisa dilakukan proses instalasi paket dengan menggunakan perintah sebagai berikut dengan akses sebagai *root* : `rpm -Uvh <nama paket>`
 - Sebagai contoh, melakukan instalasi pada `compat-libstdc++-devel` adalah sebagai berikut : `rpm -Uvh compat-libstdc++-devel-7.9-2.96.122`

2.3.5 Hangcheck-timer Module Configurasi

1. Komponen lain untuk system yang harus dipenuhi adalah modul kernel hangcheck-timer.
2. Modul hangcheck-timer digunakan untuk memberikan hang pada system operasi yang dapat mempengaruhi *reliability* pada sebuah titik RAC dan menyebabkan database rusak. Jika terjadi hang, maka modul akan merestart computer.
3. Untuk mengetahui apakah hangcheck-timer sudah terpasang pada system, bisa menggunakan perintah `lsmod`. Seperti contoh berikut : `/sbin/lsmod |grep -i hang`
4. Jika modul belum terpasang, bisa dilakukan instalasi dengan mengetikkan perintah sebagai berikut : `/sbin/insmod hangcheck-timer hangcheck_tick=30 hangchec_margin=180`
5. Parameter `hangcheck_tick` mendefinisikan seberapa sering, dalam detik, hangcheck-timer mengecek hang pada computer. Default value adalah 60 detik.

6. Parameter pada `hangcheck_margin` mendefinisikan seberapa panjang, dalam detik, waktu menunggu untuk sebuah respon dari kernel. Default value adalah 180 detik.
7. Jika kernel gagal dalam merespon, maka `hangcheck-timer` akan melakukan restart pada system.
8. Menggunakan default value, computer akan melakukan restart jika kernel gagal merespon dalam waktu 240 detik.
9. Modul ini harus diinstal pada masing-masing computer.

2.3.5 Required UNIX group dan user

1. Pertama kali dibuat group `oinstall` untuk melakukan instalasi pada Oracle database software pada system. Grup ini memiliki inventori Oracle, dimana semua catalog pada semua software database Oracle terpasang pada system.
 - `groupadd -g 500 oinstall`
2. Membuat grup `dba`. Grup ini mengidentifikasi user UNIX yang memiliki hak untuk melakukan administrasi pada database.
 - `groupadd -g 501 dba`
3. Membuat user `oracle`. User ini mempunyai hak milik atas semua software yang terinstal. User ini harus mempunyai inventori Oracle dan mempunyai hak akses dalam melakukan administrasi database. Oleh karena itu, user `oracle` ini harus dimasukkan sebagai grup `oinstall` dan grup `dba`.
 - `useradd -u 500 -d /home/oracle -g "oinstall" -G \`
 - `"dba" -m -s /bin/bash oracle`
4. Melakukan pengecekan apakah user `nobody` sudah terdaftar dalam system. User `nobody` harus mempunyai hak milik atas external jobs (`extjob`) yang harus dieksekusi setelah proses instalasi.
 - `grep nobody /etc/passwd`
 - `nobody:x:99:99:Nobody:/:/sbin/nobody`

2.3.6 The oracle User Environment

1. Untuk menjalankan Oracle Universal Installer (OUI), diharuskan login sebagai `oracle`. Bagaimanapun juga, sebelum menjalankan OUI, perlu dilakukan konfigurasi environment pada user

`oracle`. Konfigurasi yang diperlukan adalah :

- Set default file mode creation mask (`umask`) menjadi 022 dalam file startup
 - Set variabel environment `DISPLAY` dan `ORACLE_BASE`
 - Ruang kosong pada temporary yang cukup untuk OUI
2. Jika `/tmp` direktori kurang dari 400 megabytes dalam ruang kosong, identifikasi sebuah system file yang besarnya cukup dan set variabel environment `TMP` dan `TMPDIR` untuk spesifikasi sebuah direktori temporary pada file system ini.
 3. Gunakan perintah `df-k` untuk identifikasi sebuah system file yang cocok dengan ruang disk yang kosong. Pastikan user `oracle` dan grup `oinstall` bisa melakukan perubahan pada direktori tersebut.
 - `df -k`
 - `/mkdir /u01/mytmp`
 - `chmod 777 /u01/mytmp`

2.3.7 User Shell Limits

1. Untuk meningkatkan performa pada software, bisa ditambahkan batas shell di bawah ini untuk user `oracle`:
 - **nofile** : angka maximum pada file open description diharuskan menjadi 65536
 - **noproc** : angka maximum pada proses tersedia pada sebuah user tunggal tidak boleh kurang dari 16384
2. Nilai pada parameter – parameter di atas bisa ditulis pada file `/etc/security/limits.conf`.
 - `soft nproc 2047`
 - `hard nproc 16384`
 - `soft nofile 1024`
 - `hard nofile 65536`

2.3.8 Configuring for Remote Installation

1. User Equivalence
 - OUI mendeteksi apakah computer yang digunakan untuk menjalankan OUI adalah bagian dari cluster. Jika iya, maka akan OUI akan memberikan peringatan untuk memilih titik dari cluster yang akan

ditambahkan untuk dilakukan instalasi. Untuk membuat proses ini berjalan dengan baik, user equivalence pasti akan berpengaruh pada user oracle di setiap titik dalam cluster.

- Untuk mengaktifkan user equivalence, pastikan file `/etc/hosts.equiv` ada di setiap titik yang berisi hostname masing – masing titik yang ada dalam cluster.
 - Sebagai contoh, jika cluster mempunyai 2 titik, ex0044 dan ex0045. Maka file `/etc/hosts.equiv` berisi seperti di bawah ini:
 - `[root@ex04]# cat/etc/hosts.equiv`
 - ex0044
 - ex0045
 - `[root@ex05]# cat /etc/hosts.equiv`
 - ex0044
 - ex0045
2. Menggunakan SH
- OUI juga mendukung ssh dan scp (OpenSSH) untuk install remote. Perintah `ssh` adalah sebuah pengganti yang aman untuk perintah `rlogin`, `rsh` dan `telnet`.
 - Untuk bisa melakukan koneksi ke sebuah server OpenSSH dari sebuah mesin client, paket `openssh` harus sudah terinstal pada mesin client. Untuk mengetahui apakah paket `openssh` sudah terinstal, ketikkan perintah berikut.
 - `rpm -q|grep -i openssh`
 - Sebagai contoh, untuk koneksi menggunakan `rsh` adalah sebagai berikut
 - `rsh ex0044 uname -r`
 - `rsh ex0045 uname -r`
3. kita asumsikan cluster menggunakan 2 titik, ex0044 dan ex0045. Bisa dijalankan langkah – langkah berikut ini untuk melakukan konfigurasi SSH menggunakan DSA pada cluster tersebut.
- Sebagai user oracle, membuat kunci public dan private pada kedua titik:
 - `[ex0044] $ /usr/bin/ssh-keygen -t dsa`
 - `[ex0045] $ /usr/bin/ssh-keygen -t dsa`

- Berikan lokasi penyimpanan secara default untuk file kunci. Jika ada perintah untuk memasukkan path, cukup tekan enter.
- Masukkan isi dari file `id_dsa.pub` dari masing – masing titik ke dalam file `authorized_keys` pada titik pertama.
 - `[ex0044] $ ssh ex0044 “cat ~/.ssh/id_dsa.pub” >> \`
`~/.ssh/authorized_keys`
 - `[ex0044] $ ssh ex0045 “cat ~/.ssh/id_dsa.pub” >> \`
`~/.ssh/authorized_keys`
- Kopikan file `authorized_keys` ke dalam lokasi yang sama pada titik kedua.
 - `[ex0044] $ scp ~/.ssh/authorized_keys ex00045:/home/oracle/.ssh/`
- Tes konfigurasi
 - `[ex0044] $ ssh ex0045 hostname ex0045.us.oracle.com`
 - Pastikan hostname muncul tanpa ada perintah untuk memasukkan password

2.3.9 Parameter Sistem Operasi Linux

Periksa parameter yang ditunjukkan pada table di atas, nilainya ditentukan lebih besar atau sama dengan nilai yang dianjurkan. Gunakan perintah `sysctl` untuk menampilkan nilai default dari berbagai parameter. Contohnya untuk menampilkan parameter `semaphore`, jalankan perintah berikut :

```
#sysctl -a|grep sem
Kernel.sem = 250 32000 32 128
```

Nilai yang ditampilkan merepresentasikan `semmsl`, `semmns`, `semopm` dan `semmni`. Kernel parameter yang dapat diset secara manual meliputi :

- **SEMMNS** : Jumlah semaphore pada system
- **SEMMNI** : Jumlah semaphore identifier yang mengontrol jumlah semaphore yang dapat dibuat pada waktu tertentu.
- **SEMMSL** : Semaphore dikelompokkan ke dalam kumpulan semaphore, dan `SEMMSL` mengontrol ukuran array, atau jumlah semaphore yang berisi tiap

kumpulan semaphore. Jumlahnya harus sekitar 10 atau lebih dari jumlah maximum dari Oracle processes.

- **SEMOPM** : Jumlah maximum dari operasi tiap memanggil operasi semaphore
- **SHMAX** : Ukuran maximum dari shared-memory segment. Jika memungkinkan, ukuran ini harus lebih besar sedikit daripada ukuran yang terbesar yang diantisipasi oleh System Global Area(SGA).
- **SHMNI** : Jumlah dari shared memory identifier
- **RMEM_MAX** : maximum TCP menerima window(buffer) size
- **RMEM_DEFAULT** : default TCP menerima ukuran window
- **WMEM_MAX** : maximum TCP mengirim window size
- **WMEM_DEFAULT** : default TCP mengirim ukuran window

Anda dapat mengatur parameter semaphore tersebut secara manual dengan menuliskan isi dari file /proc/sys/kernel/sem:

```
#echo SEMMSL_value
SEMMSL_value
SEMNS_value
SEMOPM_value\SEMNI_value >
/proc/sys/kernel/sem
```

Untuk merubah nilai parameter tersebut dan membuatnya tetap seperti itu, edit file /etc/sysctl.conf sbb :

```
#vi /etc/sysctl.conf
...
Kernel.sem=250 32000 100
128
Kernel.shmall=2097152
Kernel.shmmax=2147483648
Kernel.shmni=4096
Fs.file-max=65536
Rmem_max=262144
Rmem_default=262144
Wmem_default=262144
Wmwm_max=262144
Net.ipv4.ip_local_port_range=1024 65000
```

Parameter kernel diatas direkomendasikan hanya berupa nilai. Untuk system database produksi, dianjurkan agar anda menentukan nilai untuk mengoptimalkan performance dari system.

Catatan : Karena ada banyak parameter yang harus dicek, anda dapat menggunakan Cluster Verification Utility untuk pemeriksaan secara otomatis.

2.3.10 Menggunakan Raw Partitions

Walaupun Red Hat dan SLES 9 menyediakan Logical Volume Manager(LVM), LVM ini tidak selalu mengetahui cluster. Alasannya, Oracle tidak secara umum mendukung penggunaan logical volume dengan RAC baik untuk Oracle Clusterware atau file database pada Linux

Untuk membuat raw partition yang dibutuhkan, lakukan langkah2 berikut :

1. Install shared disk yang ingin anda gunakan, dan reboot system bila perlu
2. Untuk mengidentifikasi nama device untuk disk yang ingin anda gunakan untuk database, masukkan perintah berikut :

```
#!/sbin/fdisk -l
Disk /dev/sda:9173
MB,9173114880 bytes
255 heads,63
sectors/track,1115 cylinders
Units = cylinder of 16065 *
512 = 8225280 bytes
Disk /dev/sdb:9173
MB,9173114880 bytes
255 heads,63
sectors/track,1115 cylinders
Units = cylinders of 16065 *
512 = 8225280 bytes
...
```

3. Partisi device. Anda dapat membuat raw partition yang diperlukan baik untuk device baru yang anda tambahkan ataupun device yang dipartisi sebelumnya yang mempunyai free space yang belum dipartisi. Untuk mengidentifikasi device yang mempunyai free space yang belum dipartisi, periksalan awal dan akhir jumlah cylinder dari partisi yang ada dan tentukan apakah device berisi cylinder yang tidak digunakan. Identifikasi jumlah dan ukuran dari raw files yang anda perlukan untuk instalasi. Gunakan chart pada slide sebagai starting point dalam menentukan storage yang diperlukan. Gunakan petunjuk berikut saat membuat partisi :
 - Gunakan command **p** untuk mendaftarkan table partisi pada device
 - Gunakan command **n** untuk membuat partisi baru
 - Setelah anda sudah membuat semua partisi yang diperlukan pada device, gunakan command **w** untuk

menuliskan table partisi yang dimodifikasi pada device.

```
#fdisk /dev/sda
Command (m for help) : n
E extended
P primary partition (1-4) p
Partition number (1 - 4) : 1
First cylinder (1-1020,default 1) : 1
Last cylinder or + size or + sizeM or + sizeK (1-1020) : 500M #
System TB
Command (m for help) : w
The partition table has been altered!
```

2.3.12 Membinding Partisi

1. Setelah anda sudah membuat partisi yang diperlukan,anda harus membind (mengikat) partisi ke raw device.Akan tetapi,pertama anda harus menentukan raw device mana yang telah dibatasi untuk device lainnya,masukkan command berikut :
#/usr/bin/raw -qa
Raw devices mempunyai nama device dalam form /dev/raw/rawn,dimana **n** adalah jumlah yang mengidentifikasi raw device.
2. Bukafile
/etc/sysconfig/rawdevices
pada text editor,dan tambahkan baris di bawah pada tiap partisi yang telah anda buat :
/dev/raw/raw1 /dev/sda1
Tentukan raw device yang tidak digunakan pada tiap partisi
4. Untuk raw device yang anda buat untuk Oracle Cluster Registry (OCR),masukkan perintah di bawah untuk mengeset pemilik,group,dan izin pada file device :
#chown root:dba /dev/raw/rawn
#chmod 640 /dev/raw/rawn
5. Untuk setiap raw device tambahan yang anda tentukan dalam file rawdevices,masukkan perintah di bawah untuk mengeset pemilik,group,dan izin pada file device :
#chown oracle:oinstall /dev/dev/rawn
#chmod 660 /dev/raw/rawn
6. Untuk membind partisi ke raw device,masukkan perintah berikut :
#/sbin/service rawdevices restart

Dengan mengedit file rawdevice,system membind partisi ke raw device saat reboot.

2.3.12 Mapping File Raw Device

Untuk memungkinkan DBCA untuk mengidentifikasi raw partition yang tepat pada setiap file database,anda harus mempertimbangkan membuat mapping file raw device sbb :

1. Buat subdirectory file database di bawah directory Oracle base,dan tentukan pemilik,group,dan izin yang tepat :

```
#mkdir -p $ORACLE_BASE/oradata/dbname
#chown -R oracle:oinstall $ORACLE_BASE/oradata
#chmod -R 775 $ORACLE_BASE/oradata
```

2. Ubah directory ke \$ORACLE_BASE/oradata/dbname directory dan edit file dbname_raw.conf pada text editor untuk membuat file seperti berikut :

```
system=/dev/raw/raw1
sysaux=/dev/raw/raw2
example=/dev/raw/raw3
users=/dev/raw/raw4
temp=/dev/raw/raw5
undotbs1=/dev/raw/raw6
undotbs2=/dev/raw/raw7
...
```

Gunakan petunjuk berikut saat membuat atau mengedit file tersebut :

- Setiap baris pada file harus mempunyai format berikut :
Database_object_identifie
r=raw_device_path
 - Untuk database RAC,file harus menentukan satu automatic undo tablespace data file (undotbsn) dan dua redo log files (redo_n_1,redo_n_2) untuk setiap instance
 - Tentuka paling tidak dua control file(control1,control2).
 - Gunakan manual daripada automatic undo management,tentukan single RBS tablespace data file(rbs) daripada automatic undo management tablespaces
3. Simpan file,dan perhatikan nama file yang anda tentukan.Saat anda mengkonfigurasi oracle user's environment dalam pelajaran ini,set variable DBCA_RAW_CONFIG

environment untuk menentukan full path file ini

2.3.11 Verifikasi Setup Cluster dengan cluvfy

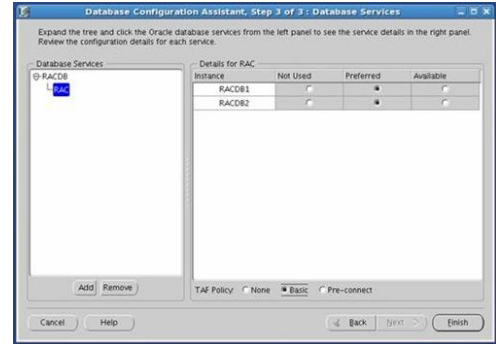
Cluster Verification Utility (cluvfy) memungkinkan anda untuk menjalankan banyak preinstalasi dan postinstalasi mengecek berbagai langkah-langkah pada instalasi database RAC anda. cluvfy utility tersedia pada Oracle 10g Release 2. Untuk mengecek kesiapan cluster anda untuk instalasi Oracle Clusterware, jalankan cluvfy sbb :

```
$runcluvfy.sh stage -post
hwos -n all -verbose
Performing post-checks
for hardware and operating
system setup
Checking node
reachability...
...
Result : Node
reachability check passed
from node "ex004".
Checking user
equivalence...
...
Result: User equivalence
check passed from user
"oracle".
Checking node
connectivity...
...
Result:Node connectivity
check passed.
Checking shared storage
accessibility...
...
Shared storage check
passed on nodes
"ex0045,ex0044".
...
Post-check for hardware
and operating system setup
was successful on all the
nodes.
```

3. UJI COBA DAN ANALISA

3.1 UJI COBA

3.1.1 Transparent Application Failover



Gambar 3.1 pengaturan sistem failover

Pengaturan untuk sistem failover dari RAC bisa terlihat seperti gambar di atas.

Untuk melakukan pengujian, dapat dilakukan langkah – langkah sebagai berikut:

1. Login ke database dan lakukan query ke salah satu tabel

```
[oracle@raclinux1 ~]$ sqlplus scott/tiger@rac
SQL*Plus: Release 10.2.0.1.0 - Production on Fri May 2 14:51:04 2008
Copyright (c) 1982, 2005, Oracle. All rights reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 - Production
With the Partitioning, Real Application Clusters, OLAP and Data Mining options
SQL>
SQL>
SQL> select count(*) from emp;

COUNT(*)
-----
14
SQL>
```

2. Login sebagai sysdba dan lakukan query pada tabel gv\$session untuk mengetahui server yang diakses oleh user scott

```
1 select inst_id , failover_type, failover_method, failed_over
2* from gv$session where username = 'SCOTT'
SQL> /

INST_ID FAILOVER_TYPE FAILOVER_M FAI
-----
2 SELECT BASIC NO
SQL>
```

User scott terkoneksi pada server 2 (raclinux2).

3. Lakukan shutdown pada server2 (raclinux2). Kemudian lakukan query ulang pada tabel gv\$session untuk mengetahui server yang diakses oleh user scott

```
SQL> 1
1 select inst_id , failover_type, failover_method, failed_over
2* from gv$session where username = 'SCOTT'
SQL> /

no rows selected
```

Record masih kosong. Karena setelah server2(raclinux2) di-shutdown, user scott belum melakukan query

- Melakukan query lagi ke salah satu tabel dengan user scott

```
[oracle@raclinux1 ~]$ sqlplus scott/tiger@RAC
SQL*Plus: Release 10.2.0.1.0 - Production on Fri May 2 14:51:04 2008
Copyright (c) 1982, 2005, Oracle. All rights reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 - Production
With the Partitioning, Real Application Clusters, OLAP and Data Mining options
SQL>
SQL>
SQL> select count(*) from emp;

  COUNT(*)
-----
         14
SQL> █
```

- Lakukan query lagi pada tabel gv\$session. Maka akan terlihat user telah melakukan koneksi ke server1(raclinux1)

```
SQL> 1
    2 select inst_id , failover_type, failover_method, failed_over
    3 * from gv$session where username = 'SCOTT'
SQL> /

no rows selected

SQL> /

  INST_ID  FAILOVER_TYPE  FAILOVER_M  FAI
-----
         1  SELECT          BASIC        YES
```

4. KESIMPULAN DAN SARAN

4.1 Kesimpulan

Dari hasil uji coba Clustering Database ini dapat ditarik beberapa kesimpulan:

- Koneksi antar server dilakukan melalui public network dan private network
- Koneksi akan dialihkan melalui private network ke server yang standby jika akses terhadap server yang sibuk terjadi kegagalan.

4.2 Saran

5. DAFTAR PUSTAKA

Website.

<http://arjudba.blogspot.com/2008/08/configuration-shared-storage-in-oracle-rac.html>
[15 April 2009].

<http://kbase.redhat.com/faq/docs/DOC-6133> [16 April 2009].