SCAN, VIP, HAIP etc

Introduction

This paper is to explore few RAC abbreviations and explain the concepts behind these acronyms.

This paper is NOT designed as a step-by-step approach, rather as a guideline.

1. VIP

VIP (Virtual IP Address) is quite simply an IP address that is logically plumbed on a logical network interface. A listener usually listens on that IP address on a specific port. Clusterware monitors both the listener and IP address for high availability, as a resource.

Listing 1-1: VIP

```
$ /sbin/ifconfig -a
...

e1000g0:1:
flags=1040843<UP,BROADCAST,RUNNING,MULTICAST,DEPRECATED,IPv4> mtu
1500 index 2 inet 172.16.140.151 netmask ffff0000 broadcast
172.16.255.255
...
```

Listing 1-1 shows a VIP `172.16.140.151` plumbed on e1000g0 network interface (logical interface is e1000g0:1). IP address of this VIP is 172.16.140.151 in a sub network. This IP address and interface is constantly monitored by the clusterware for any failures. Note that my subnet is 255.255.0.0, but this subnet can be on any subnet that your network administrator designates.

Listing 1-2: Listener

```
$ lsnrctl status listener

LSNRCTL for Solaris: Version 11.2.0.2.0 - Production on 18-FEB-
2012 15:31:27
…
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=LISTENER)))
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=172.16.140.151)(PORT=15
21)))
…
```

Why do you need VIP?

In Listing 1-2, output of lsnrctl command shows that a listener is listening on this virtual IP address. To illustrate the importance of VIP, we will perform a small test case. In this test case, we will stop the listener, unplumb the IP address, and then make a connection to that IP address using the connect string printed in Listing 1-3.

Listing 1-3: Connect String

```
just_vips=
 (description=
  (load_balance=off)
   (address=(protocol=tcp)(host=172.16.140.151)(port=1521))
   (address=(protocol=tcp)(host=172.16.140.152)(port=1521))
  (connect_data=
     (service_name=racdb)
   )
 )
```

Tracing the UNIX system calls from that sqlplus process, we shall understand the internal details about a connection process. Listing 1-4 prints the output of truss command of a sqlplus session connecting to *just_vips* connect string.

1. In this example, sqlplus is sending a TCP connection request to the port 1521, IP address 172.16.140.151.
2. But, we unplumbed the IP address and so, the process is waiting for a timeout period of 60 seconds before proceeding. Essentially, your application also will wait for 60 seconds before proceeding if the IP address is not available.

That timeout is, precisely, what clusterware is trying to reduce with the use of VIP.

Listing 1-4: Truss output with failed IP

```
truss -d -E -vall sqlplus rs@just_vips
...
3.0730  0.0002 connect(9, 0x00539F10, 16, SOV_DEFAULT)  Err#150 EINPROGRESS
        AF_INET  name = 172.16.140.151  port = 1521
3.0731  0.0000 brk(0x00540D90)                                = 0
3.0732  0.0000 brk(0x00544D90)                                = 0
3.0732  0.0000 brk(0x00544D90)                                = 0
pollsys(0xFFFFFD7FFFDEFC60, 1, 0xFFFFFD7FFFDEFB40, 0x00000000) (sleeping...)
        fd=9  ev=POLLOUT rev=0xFFFFFFFF
        timeout: 60.000000000 sec
63.0735  0.0000 pollsys(0xFFFFFD7FFFDEFC60,1,0xFFFFFD7FFFDEFB40, 0x00000000)= 0
        fd=9  ev=POLLOUT rev=0
        timeout: 60.000000000 sec
63.0740  0.0001 close(9)
```

Clusterware constantly monitors the database node. If the database node goes down, then the connection process must wait for the connection timeout. To avoid the connection timeout, clusterware will relocate the Public Virtual IP address of the failed node immediately to a surviving node. Note that only VIP is failed over, and the listener is not started on the failed-over IP address.

Listing 1-5: Truss output after VIP relocated

```
0.0799  0.0000 so_socket(PF_INET, SOCK_STREAM, IPPROTO_IP, "", SOV_DEFAULT) = 9
0.0800  0.0000 ioctl(9, FIONBIO, 0xFFFFFD7FFFDEF9D8)          = 0
                 write 4 bytes
0.0801  0.0001 connect(9, 0x0053BDF0, 16, SOV_DEFAULT) Err#146 ECONNREFUSED
        AF_INET  name = 172.16.140.151  port = 1521
0.0802  0.0000 close(9)                                       = 0
0.0802  0.0000 getsockopt(9, SOL_SOCKET, SO_SNDBUF, 0xFFFFFD7FFFDF0214,
```

From Listing 1-5, we can see that failed IP address returns immediately with *ECONNREFUSED* and the connection process tries to connect using next IP address in the list. In a nutshell, connection timeout has been eliminated by monitoring VIP and failing VIP to a surviving node.

Network, VIP as resource

In clusterware, VIP is created as a resource and monitored. There are subtle differences between implementation of network type resources in 11.2 compared to 11.1. We will discuss 11.2 implementation in this paper.

Listing 1-6: crsctl

```
$ crsctl status resource ora.solrac1.vip
NAME=ora.solrac1.vip
TYPE=ora.cluster_vip_net1.type
TARGET=ONLINE
STATE=ONLINE on solrac1

$ crsctl status resource ora.solrac1.vip -p |grep USR_ORA_VIP
GEN_USR_ORA_VIP=
USR_ORA_VIP=solaris1_vip

$ grep solaris1_vip /etc/hosts
172.16.140.151  solaris1_vip.solrac.net solaris1_vip
```

In Listing 1-6, output of *crsctl* command shows that VIP resource is online on *solrac1* server. Static configuration (using –p flag) shows that this VIP resource is associated with *solaris1_vip*. Solaris1_vip is an alias defined in /etc/hosts file for the 172.16.140.151 IP address.

Listing 1-7: Dependency

```
$ crsctl status resource ora.net1.network -p |more
NAME=ora.net1.network
TYPE=ora.network.type
…
USR_ORA_IF=e1000g0
USR_ORA_NETMASK=255.255.0.0
USR_ORA_SUBNET=172.16.0.0
```

Listing 1-7 shows that VIP resource is dependent upon network resource named ora.net1.network. Network subnet, netmask, and the interface information are kept in the network resource definition.

These two resources are checked by clusterware to monitor the VIP and network. During normal operating conditions, VIP will be up in the defined node. If a node fails, then the VIP of the failed node is failed over to a surviving node, but listener is not started.

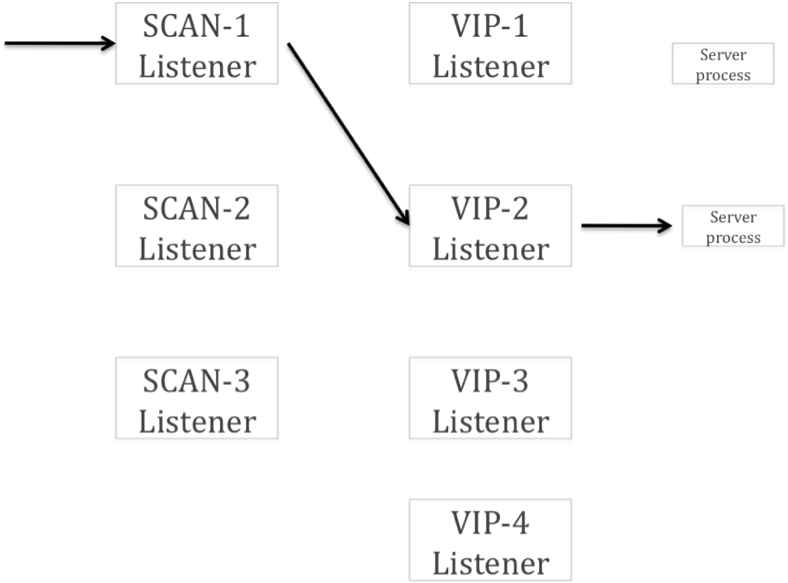
2. SCAN

SCAN listeners, introduced in 11gR2, provide an abstraction layer of cluster configuration. SCAN listeners also act as a load balancing mechanism to manage workload. SCAN listeners simply acts as a redirection mechanism redirecting incoming connections to a local VIP listener.

A typical connection process is:

1. Connection request is made by the application connecting to the SCAN IP address and Port.
2. SCAN listener receives the service name and redirects the connection to the VIP listener servicing that service_name. SCAN listener uses load balancing statistics to choose a node if the service is preferred in more than one instance.
3. VIP listener bequeath connection process and the connection continue to create a new DB connection.

Figure 2-1: SCAN redirection



As shown in Figure 1-1, SCAN listeners act as simple redirection mechanism and so, they are considered a lightweight process. In contrast, VIP listeners are not lightweight processes since they must fork a new process to create a database connection. Number of SCAN listeners in a RAC cluster can't exceed three.

Configuration

SCAN listeners and SCAN IP addresses are mandatory from 11gR2 Grid Infrastructure onwards. During installation process, Installer requires a SCAN IP and SCAN listener to be configured. SCAN IP address and SCAN listeners are managed as resources in Oracle Clusterware.

Listing 2-1: SCAN IP address as a resource

```
$ crsctl stat resource ora.scan1.vip
NAME=ora.scan1.vip
TYPE=ora.scan_vip.type
TARGET=ONLINE
STATE=ONLINE on solrac1

$ crsctl stat resource ora.scan1.vip -p |grep
'^USR_ORA_VIP'
USR_ORA_VIP=172.16.140.150
```

Listing 2-1 shows the configuration of SCAN IP address. Resource ora.scan1.vip is used to monitor a SCAN IP address, in this example, 172.16.140.150 is the SCAN IP address.

Listing 2-2: SCAN Listener as a resource

```
$crsctl stat res ora.LISTENER_SCAN1.lsnr -p|egrep'ENDPOINTS|START_DEP'

ENDPOINTS=TCP:1521
START_DEPENDENCIES=hard(ora.scan1.vip)
dispersion:active(type:ora.scan_listener.type) pullup(ora.scan1.vip)
```

Listing 2-2 shows that the SCAN listener listening on port 1521. SCAN listener is dependent upon the resource *ora.scan1.vip*, which is the SCAN IP address.

Deep review of connection process

1. Application uses a connection string specifying the DNS name of a connection string.

```
solrac_po=
 (description=
   (address=(protocol=tcp)(host=solscan)(port=1521))
  (connect_data=
     (service_name=po)
   )
 )
```

2. SCAN listener is listening on solscan IP address. Services on the SCAN listener shows the PO service is servied by a VIP listener, as the server is remote. Notice the keyword indicating that connections to the SCAN listener for PO service will be redirected to the remote server listening on the IP:Port `172.16.140.151:1521`

```
Service "po" has 1 instance(s).
  Instance "solrac1", status READY, has 1 handler(s) for this service...
    Handler(s):
      "DEDICATED" established:4 refused:0 state:ready
         REMOTE SERVER
        (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=172.16.140.151)(PORT=1521)))
```

3. Services serviced by that VIP listener shows that VIP listener will hande the PO service.

```
$ lsnrctl services listener
Service "po" has 1 instance(s).
  Instance "solrac1", status READY, has 1 handler(s) for this
service...
    Handler(s):
      "DEDICATED" established:4 refused:0 state:ready
         LOCAL SERVER
```

Essentially, SCAN listener is redirecting connection requests to the VIP listener; VIP listener services the incoming connection requests.

DNS setup of SCAN address

SCAN IP address are setup in DNS for name resolution. Three scan listeners and SCAN IP addresses can be configured at the most and it is a better practice to configure three SCAN IP addresses and three SCAN listeners. Reason is that

```
nslookup   solscan
 Name : solscan
   Address: 172.16.140.150
 Name : Scan-ip
   Address: 172.16.140.149
 Name : Scan-ip
   Address: 172.16.140.148
```

Parameter setup

Two parameters are used to configure these listeners in the database. Parameter remote_listener is set to scan_ip:1521 and the local_listener is set to a connect string connecting to local VIP listener.

```
NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
remote_listener                      string      solscan.solrac.net:1521
```

Parameter local_listener is populated by the agent connecting to the VIP listener.

```
NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
local_listener                       string      (DESCRIPTION=(ADDRESS_LIST=(AD
                                                 DRESS=(PROTOCOL=TCP)(HOST=172.
                                                 16.140.151)(PORT=1521))))
```

PMON process registers the services with the listeners using local_listener and remote_listener. Configuration of these two parameters is of paramount importance since PMON service registration is the key for consistent successful connections.

Salient points about SCAN

Few important points about SCAN

1. It is a better practice to have three SCAN IP addresses and three scan listeners to provide fault tolerance.
2. These three SCAN IP addresses can be alive in any node of the cluster. If you have more than three nodes in a cluster, then nodes joining initially to the cluster will have SCAN resources running.
3. SCAN is an abstraction layer. In a huge cluster, client connect string do not need to specify all nodes in a cluster. Even if the topology of the cluster changes, still, there is no reason to change connect string.
4. SCAN and VIP addresses must be in the same subnet. Multiple subnets can be created using *listener_networks*, but redirection is not contained within the same subnet.

3. HAIP

HAIP, High Availability IP, is the Oracle based solution for load balancing and failover for private interconnect traffic. Typically, Host based solutions such as Bonding (Linux), Trunking (Solaris) etc is used to implement high availability solutions for private interconnect traffic. But, HAIP is an Oracle solution for high availability.

During initial start of clusterware, a non-routeable IP address is plumbed on the private subnet specified. That non-routable IP is used by the clusterware and the database for private interconnect traffic.

```
$ oifcfg getif
e1000g0  172.16.0.0  global  public
e1000g1  1.3.1.0  global  cluster_interconnect
```

Output of oifcfg command shows that e1000g1 interface is used for cluster interconnect. Clusterware will plumb IP addresses on this interface.

```
$ ifconfig -a
e1000g1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu
1500 index 3
        inet 1.3.1.170 netmask ffffff00 broadcast 1.3.1.255
e1000g1:1: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu
1500 index 3
        inet 1.3.1.70 netmask ffffff00 broadcast 1.3.1.255
```

Clusterware plumbed two IP addresses on 169.254.x.x subnet on e1000g1 private interface as shown below. These two IP addresses will be used by the clusterware and RAC database for private interconnect traffic.

```
$ifconfig -a
...
```

```
e1000g1:2: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu
1500 index 3
        inet 169.254.201.54 netmask ffff8000 broadcast
169.254.255.255
e1000g1:3: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4> mtu
1500 index 3
        inet 169.254.106.96 netmask ffff8000 broadcast
169.254.127.255
…
```

Review of database shows that these two IP addresses are used for private interconnects in node 1.

```
  1* select * from gv$cluster_interconnects
SQL> /

   INST_ID NAME            IP_ADDRESS       IS_ SOURCE
---------- --------------- ---------------- --- -------------------------------
         1 e1000g1:3       169.254.106.96   NO
         1 e1000g1:2       169.254.201.54   NO
```

Essentially, even if one of the physical interface is offline, private interconnect traffic can be routed through the other available physical interface. This leads to highly available architecture for private interconnect traffic.

This IP address is monitored by a clusterware as a resource.

```
$ crsctl stat res ora.cluster_interconnect.haip -init |more
NAME=ora.cluster_interconnect.haip
TYPE=ora.haip.type
TARGET=ONLINE
STATE=ONLINE on solrac1
```

Summary

In Summary, VIP, SCAN, and HAIP are important terminologies to understand for RAC concepts.