Oracle RAC Connection String Rewrite

Jason Rahm, 2012-22-10

My buddy Brent Imhoff is back with another excellent customer solution to follow up his IPS traffic inspection solution. This solution is a modification of the original iRule for db context switching from the deployment guide for Oracle Database and RAC.

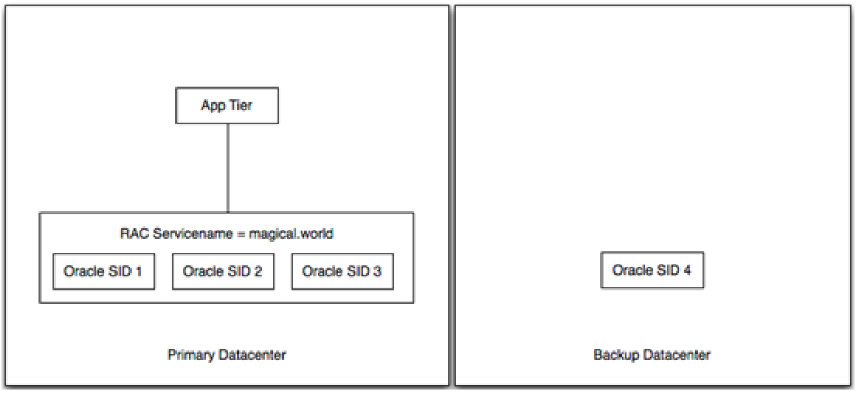
Problem Description

Customer has two Oracle instances, one in a primary datacenter and one in the secondary datacenter. The primary datacenter is comprised of three Oracle instances, each with a unique Oracle System ID (SID), as part of an Oracle Real Application Cluster (RAC). The secondary datacenter is setup as a single node with it's own SID.

The RAC is essentially a load balancer for Oracle nodes, but somewhat limited in that it doesn't work between data centers. As part of the RAC implementation, a Service Name is defined in the RAC configuration and clients use the service name in the connect string. RAC evaluates the service name and makes a determination regarding which SID could handle the request appropriately. Alternatively, clients can use the SID to connect directly to an Oracle node.

The customer has some 800 applications that rely on these databases. Because there is RAC only in one location, they are forced to change connect strings each time they need to change the data sources for things like maintenance. Additionally, it removes the possibility of an automated failover if something goes wrong unexpectedly. The customer asked F5 if we could be an Oracle proxy to handle the rewrite of connect strings and essentially make the clients unaware of any changes that might be happening to the Oracle nodes.

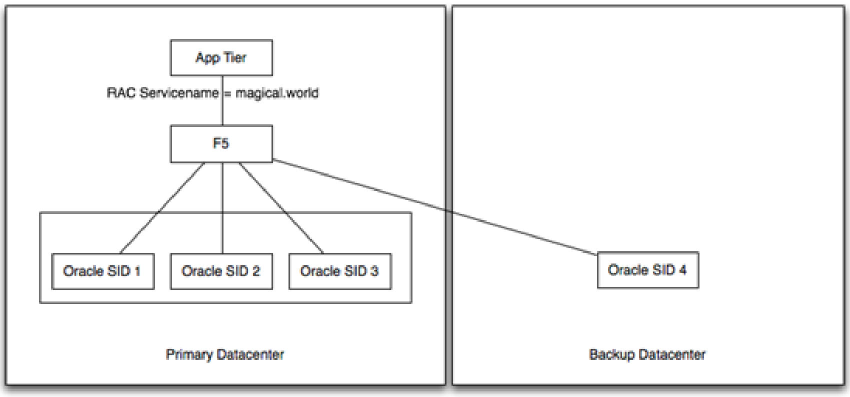
Old and Busted (aka Oh Crap! My database died. Quick! Change 800 configurations to get things working again!)



Solution

Inserting the F5 solution, the customer was able to create a single IP/port that was highly available to be used by the app tier. Using priority groups, we were able to balance traffic to each oracle node and dynamically rewrite the connect strings to match the node being used to service the request. We tested two scenarios: minimal traffic and gobs of traffic. When we failed an Oracle node under minimal traffic, the application tier was completely unaware of any change at all. No log messages, no errors, it just went along it's merry way. Running the same test with gobs of traffic (couple of GBs of transactions), the application noticed something wasn't quite right, resent the transactions that didn't make it, and happily continued. No oracle DBA intervention required.

New Hotness - (aka The Magic of iRules)



Configuration

Use the Oracle deployment guide to ensure TCP profiles are created correctly. Also included are good procedures for building node monitors for each Oracle member. Once those are all in place, use an iRule similar to the following. This could be made more generic to accurately calculate the length of the payload replacement. The service_name and replacement SID's could also be defined as variables to make the deployment more straight forward. There's also a hack that limits the SID patching length to 1 time. In the Oracle deployment guide, the iRule is written to accommodate multiple rewrites of connect strings in a given flow. In our testing, it seemed to be adding the same offset to the list twice (which screwed things up pretty nicely). I'm not sure why that was happening, but the hack fixed it (at least in this instance).

```
when CLIENT_DATA {

  if { [TCP::payload] contains "(CONNECT_DATA=" } {
    log local0. "Have access to TCP::Payload"
    set service_match [regexp -all -inline -indices "\(SERVICE_NAME=some_datasource_service.world\)" [TCP::payload]]
    log "Found a service_match = $service_match"

    set tmp [lindex $service_match 1]
    set newservice [list $tmp]

    foreach instance $newservice {
      log local0. "Iterating through connect strings in the payload.  Raw:  $instance"
      set service_start [lindex $instance 0]

      set original_tcp_length [TCP::payload length]
      TCP::payload replace $service_start 34 $sid
      log local0. "Inserted SID at $service_start offset."

      TCP::payload replace 0 2 [binary format S1 [TCP::payload length]]
      log local0. "Updated packet with new length: [TCP::payload length] - original $original_tcp_length"

      ##
      ##set looking_for_connect [findstr [TCP::payload] "(DESCRIPTION=(ADDRESS=" 0]
      set looking_for_connect [findstr [TCP::payload] "(DESCRIPTION" 0]
      log local0. "Looking for connect:  $looking_for_connect"
     ##set connect_data_length [string length [findstr [TCP::payload] "(DESCRIPTION=(ADDRESS=" 0]]
      set connect_data_length [string length [findstr [TCP::payload] "(DESCRIPTION" 0]]
      TCP::payload replace 24 2 [binary format S1 $connect_data_length]
      log local0. "New Oracle data length is $connect_data_length"

    }
   }
  if { [TCP::payload] contains "(CONNECT_DATA=" } {
      set looking_for_connect [findstr [TCP::payload] "(DESCRIPTION" 0]
      log local0. "2. Looking for connect:  $looking_for_connect"
  }

  TCP::release
  TCP::collect

}
when LB_SELECTED {

  log local0. "Entering LB_SELECTED"
  if { [TCP::payload] contains "(CONNECT_DATA=" } {
      set looking_for_connect [findstr [TCP::payload] "(DESCRIPTION" 0]
      log local0. "1. Looking for connect:  $looking_for_connect"
  }


  switch [LB::server addr] {
    10.10.10.152 { ;
      set sid "SID=ORAPRIME1"
    }
    10.10.10.153 { ;
      set sid "SID=ORAPRIME2"
    }
    10.10.10.154 { ;
      set sid "SID=ORAPRIME3"
    }
    10.44.44.44 { ;
      set sid "SID=ORABACKUP"
    }
  }
  TCP::collect
  log local0. "Exiting LB_SELECTED"
}
```

Related Articles

- DevCentral Groups - Oracle / F5 Solutions
- Oracle/F5 RAC Integration and DevArt - DevCentral - DevCentral ...
- Delivering on Oracle Cloud
- Oracle OpenWorld 2012: The Video Outtakes
- Oracle OpenWorld 2012: That's a Wrap
- HA between third party apps and Oracle RAC databases using f5 ...
- Oracle Database traffic load-balancing - DevCentral - DevCentral ...
- Oracle OpenWorld 2012: BIG-IP APM Integration - Oracle Access ...

Technorati Tags: Oracle, RAC

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com