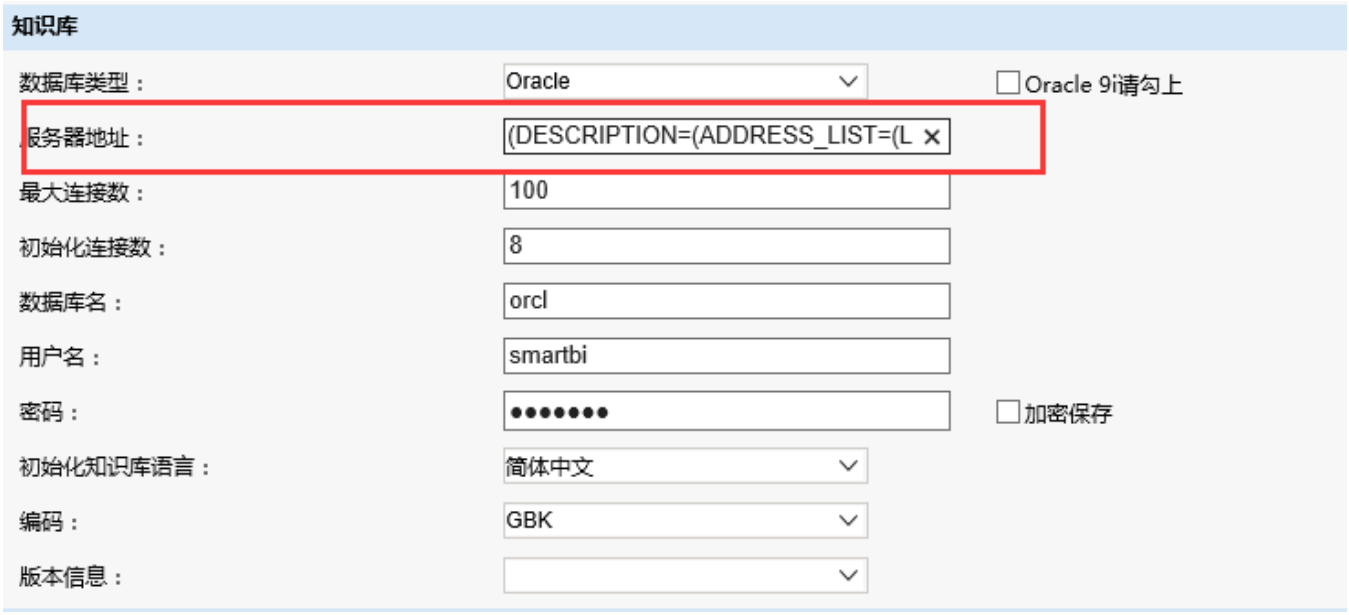
Smartbi使用Oracle RAC数据库做知识库

RAC，全称real application clusters，译为"真正应用集群"，是Oracle新版数据库中采用的一项新技术，也是Oracle数据库支持网格计算环境的核心技术。Oracle RAC主要支持Oracle 10g、11g、12c版本，可以支持24 x 7 有效的数据库应用系统，在低成本服务器上构建高可用性数据库系统。在Oracle RAC环境下，Oracle集成提供了集群软件和存储管理软件，为用户降低了应用成本。当应用规模需要扩充时，用户可以按需扩展系统，以保证系统的性能。

1、在Smartbi中配置知识库时，如何连接到Oracle RAC数据库呢，其实和连接普通的Oracle数据库没什么区别，只是所用的"服务器地址"写法稍有不同。数据库类型选择"**Oracle**"，将"**服务器地址**"设置为后面所说明的格式。



2、针对Oracle RAC数据库，所用的"**连接字符串**"格式如下。需要将其中的**rachost1**、**rachost2**分别修改为集群中对应节点的IP地址，**orcl**修改为RAC集群的服务名。

```
(DESCRIPTION=
        (ADDRESS_LIST=
                (LOAD_BALANCE=ON)
                (ADDRESS=(PROTOCOL=TCP)(HOST=rachost1)(PORT=1521))
                (ADDRESS=(PROTOCOL=TCP)(HOST=rachost2)(PORT=1521))
        )
        (CONNECT_DATA=
                (SERVICE_NAME=orcl)
        )
)
```

3、接着将上述内容放到一行上变成一个字符串，将其粘贴到"**服务器地址**"文本框中，然后"**测试知识库连接**"。如果能"测试通过"，则说明配置正确。

```
(DESCRIPTION=(ADDRESS_LIST=(LOAD_BALANCE=ON)(ADDRESS=(PROTOCOL=TCP)(HOST=192.168.1.101)(PORT=1521))(ADDRESS=
(PROTOCOL=TCP)(HOST=192.168.1.102)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=urpdb)))
```

4、注意：在用JDBC方式连接Oracle RAC数据库时，必须确保所用的JDK是1.6以上、JDBC驱动是classes12.jar以上版本，否则在"**测试知识库连接**"时会报各种错误。

```
***  ***

        at smartbi.freequery.basicdata.MetaDataServiceImpl.testConnection(MetaDataServiceImpl.java:246)
        at ...(...)
        at smartbi.connectionpool.ConnectionPool$2.createConnection(ConnectionPool.java:317)
Caused by: java.sql.SQLException: Io : Got minus one from a read call
        at oracle.jdbc.dbaccess.DBError.throwSqlException(DBError.java:169)
        at oracle.jdbc.dbaccess.DBError.throwSqlException(DBError.java:211)
        at oracle.jdbc.dbaccess.DBError.throwSqlException(DBError.java:324)
        at oracle.jdbc.driver.OracleConnection.<init>(OracleConnection.java:266)
        at oracle.jdbc.driver.OracleDriver.getConnectionInstance(OracleDriver.java:365)
        at oracle.jdbc.driver.OracleDriver.connect(OracleDriver.java:260)


***  ***

        at smartbi.freequery.basicdata.MetaDataServiceImpl.testConnection(MetaDataServiceImpl.java:246)
        at ...(...)
        at smartbi.connectionpool.ConnectionPool$2.createConnection(ConnectionPool.java:317)
Caused by: java.sql.SQLException: Io : Connection refused(DESCRIPTION=(TMP=)(VSNNUM=169870592)(ERR=12505)
(ERROR_STACK=(ERROR=(CODE=12505)(EMFI=4))))
        at oracle.jdbc.dbaccess.DBError.throwSqlException(DBError.java:169)
        at oracle.jdbc.dbaccess.DBError.throwSqlException(DBError.java:211)
        at oracle.jdbc.dbaccess.DBError.throwSqlException(DBError.java:324)
        at oracle.jdbc.driver.OracleConnection.<init>(OracleConnection.java:266)
        at oracle.jdbc.driver.OracleDriver.getConnectionInstance(OracleDriver.java:365)
        at oracle.jdbc.driver.OracleDriver.connect(OracleDriver.java:260)


***  ***

        at smartbi.freequery.basicdata.MetaDataServiceImpl.testConnection(MetaDataServiceImpl.java:246)
        at ...(...)
        at smartbi.connectionpool.ConnectionPool$2.createConnection(ConnectionPool.java:317)
Caused by: java.lang.ArrayIndexOutOfBoundsException: 7
        at oracle.security.o3logon.C1.r(C1)
        at oracle.security.o3logon.C1.l(C1)
        at oracle.security.o3logon.C0.c(C0)
        at oracle.security.o3logon.O3LoginClientHelper.getEPasswd(O3LoginClientHelper)
        at oracle.jdbc.ttc7.O3log.<init>(O3log.java:290)
        at oracle.jdbc.ttc7.TTC7Protocol.logon(TTC7Protocol.java:251)
        at oracle.jdbc.driver.OracleConnection.<init>(OracleConnection.java:252)
        at oracle.jdbc.driver.OracleDriver.getConnectionInstance(OracleDriver.java:365)
        at oracle.jdbc.driver.OracleDriver.connect(OracleDriver.java:260)


***  ***

        at smartbi.freequery.basicdata.MetaDataServiceImpl.testConnection(MetaDataServiceImpl.java:246)
        at ...(...)
        at smartbi.connectionpool.ConnectionPool$2.createConnection(ConnectionPool.java:317)
Caused by: java.sql.SQLException: Io : Connection refused(DESCRIPTION=(ERR=1153)(VSNNUM=169870592)(ERROR_STACK=
(ERROR=(CODE=1153)(EMFI=4)(ARGS='(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=10.22.19.14)(PORT=1521))
(CONNECT_DATA=(CID=(PROGRAM=)(HOST=__jdbc__)(USER=))null))'))(ERROR=(CODE=303)(EMFI=1))))
        at oracle.jdbc.dbaccess.DBError.throwSqlException(DBError.java:169)
        at oracle.jdbc.dbaccess.DBError.throwSqlException(DBError.java:211)
        at oracle.jdbc.dbaccess.DBError.throwSqlException(DBError.java:324)
        at oracle.jdbc.driver.OracleConnection.<init>(OracleConnection.java:266)
        at oracle.jdbc.driver.OracleDriver.getConnectionInstance(OracleDriver.java:365)
        at oracle.jdbc.driver.OracleDriver.connect(OracleDriver.java:260)
```

附录：Oracle RAC集群数据库介绍

Oracle RAC是一个集群数据库，可以实现负载均衡和故障无缝切换。如何知道RAC数据库已经实现了这些功能呢，下面就对此进行功能测试。

一、 负载均衡测试

RAC数据库的负载均衡是指对数据库连接的负载均衡，当一个新的会话连接到RAC数据库时，通过指定的分配算法将请求分配到集群的任一节点上，这就是RAC数据库完成的功能。负载均衡在RAC中分为两种：一种是基于客户端连接的负载均衡；一种是基于服务器端的负载均衡。

1. RAC客户端负载均衡

客户端连接的负载均衡配置起来非常简单，与RAC数据库的实例负载和监听没有任何关系，因此也就不需要在集群节点进行任何设置，只要在客户端机器上的tnsnames.ora文件中添加负载均衡策略配置即可。这里以Linux客户端为例进行介绍。

（1） 修改/etc/hosts文件

编辑/etc/hosts文件，将RAC数据库相关的IP地址信息添加进去，例如：

```
192.168.12.231 node-rac1
192.168.12.232 node-rac2
192.168.12.230 node-vip1
192.168.12.240 node-vip2
```

（2） 查看RAC数据库的service_names

```
[oracle@node-rac1 ~]$ sqlplus "/as sysdba"
SQL*Plus: Release 11.1.0.6.0 - Production on Sun Sep 12 22:05:53 2010
Copyright (c) 1982, 2007, Oracle. All rights reserved.
Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - Production
With the Partitioning, Real Application Clusters, OLAP, Data Mining
and Real Application Testing options
NAME TYPE VALUE

-------------- -------------
service_names string racdb
```

这里需要说明的是：

- 在配置RAC负载均衡时，客户端连接的是RAC数据库的服务名，而不是实例名，也就是SERVICE_NAME必须设置为"SERVICE_NAME = racdb"。

（3） 修改Oracle客户端的配置文件tnsnames.ora

```
RACDB=
(DESCRIPTION =
      (ADDRESS_LIST =
            (ADDRESS = (PROTOCOL = TCP)(HOST = node-vip2)(PORT = 1521))
            (ADDRESS = (PROTOCOL = TCP)(HOST = node-vip1)(PORT = 1521))
            (LOAD_BALANCE = yes)
      )
      (CONNECT_DATA =
            (SERVER = DEDICATED)
            (SERVICE_NAME = racdb)
      )
)
```

这个配置文件的说明如下：

- LOAD_BALANCE = yes，表示启用连接负载均衡。在默认情况下"LOAD_BALANCE = no"，因此如果要配置负载均衡，必须添加设置"LOAD_BALANCE = yes"。启用负载均衡后，SQLNet会随机选择ADDRESS_LIST列表中的任意一个监听，然后将请求分发到此监听上，通过这种方式完成负载均衡。如果"LOAD_BALANCE = no"，那么SQLNet会按照ADDRESS_LIST列表中的顺序选择监听，只要这个监听正常就一直使用该监听。
- SERVICE_NAME = racdb，这个"racdb"是RAC数据库的服务名，而非实例名。

（4）在客户端测试负载均衡

在客户端开启一个sqlplus连接，执行如下操作：

```
[oracle@client ~]$ sqlplus system/xxxxxx@racdb
SQL*Plus: Release 11.1.0.7.0 - Production on Sun Sep 12 21:24:55 2010
Copyright (c) 1982, 2008, Oracle. All rights reserved.
Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - Production
With the Partitioning, Real Application Clusters, OLAP, Data Mining
and Real Application Testing options
SQL> show parameter instance_name
NAME TYPE VALUE

---------------- --------------------
instance_name string racdb1
```

继续开启第二个sqlplus连接，执行如下操作：

```
[oracle@client ~]$ sqlplus system/xxxxxx@racdb
SQL*Plus: Release 11.1.0.7.0 - Production on Sun Sep 12 21:31:53 2010
Copyright (c) 1982, 2008, Oracle. All rights reserved.
Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - Production
With the Partitioning, Real Application Clusters, OLAP, Data Mining
and Real Application Testing options
SQL> show parameter instance_name
NAME TYPE VALUE

------------ ------------------
instance_name string racdb2
```

按照这种方法，陆续打开多个sqlplus连接，可以看到，每次连接到的实例都在racdb1和racdb2之间变化，这样就实现了RAC数据库连接的负载均衡。

2. 服务器端的负载均衡

客户端的负载均衡解决了连接数据库的负载问题，但是由于连接是由客户端发起的，它并不知道RAC数据库集群中各个节点的繁忙状态和连接信息，因此负荷较大的节点仍然会增加新的连接，这样就可能导致RAC节点无法真正做到负载均衡。不过幸运的是，从Oracle 10g开始，服务器端负载均衡就可以根据RAC中各节点的负荷及连接数情况，将新的请求分配到集群中负载较低、连接数较少的节点上来，这样就从根本上实现了数据库的负载均衡，并且使客户端连接的负载均衡与服务器端的负载均衡可以配合使用，互不影响。

每个集群节点的负载情况是由PMON进程来定期更新的。PMON进程每3秒会将集群中每个节点的负载信息及连接数写入service_register中，当节点的负载发生变化时，将会立刻通知监听程序，最后由监听程序来决定将新的客户端连接分配到哪个节点上，通过这种方式，RAC数据库实现了真正的负载均衡。服务器端负载均衡配置也非常简单，只需在各节点的tnsnames.ora文件中添加一个对连接到各个节点进行监听的配置，然后在初始化参数中设置remote_listener即可。

（1）修改服务器端的tnsnames.ora

只需添加如下内容即可：

```
LISTENERS_RACDB =
(ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = node-vip2)(PORT = 1521))
      (ADDRESS = (PROTOCOL = TCP)(HOST = node-vip1)(PORT = 1521))
)
```

（2）修改参数remote_listener

查看RAC数据库的参数remote_listener：

```
SQL> show parameter remote_listener
NAME TYPE VALUE

------------ -----------
remote_listener string LISTENERS_RACDB
```

可以看到，remote_listener已经设置为"LISTENERS_RACDB"了。

如果remote_listener的值为空，可以通过如下命令修改每个实例的remote_listener参数：

```
SQL> alter system set remote_listener='LISTENERS_RACDB' sid='node-rac1';
SQL> alter system set remote_listener='LISTENERS_RACDB' sid='node-rac2';
```

这样，服务器端的负载均衡就配置完成了。

二、 透明应用失败切换测试

透明应用失败切换（Transparent Application Failover，TAF），这是客户端的一种功能。TAF包含两层意思：失败切换是指客户端连接到某个实例，如果连接失败，可以连接到另外一个实例；透明应用是指客户端应用程序在连接失败后可以自动重新连接到另一个数据库实例，而这个过程对应用程序是不可见的。要使用TAF功能，只需修改客户端的tnsnames.ora文件中的设置即可，结合前面介绍的客户端负载均衡功能，一个包含负载均衡和TAF功能的客户端设置如下：

```
RACDB =
(DESCRIPTION =
      (ADDRESS_LIST =
            (ADDRESS = (PROTOCOL = TCP)(HOST = node-vip2)(PORT = 1521))
            (ADDRESS = (PROTOCOL = TCP)(HOST = node-vip1)(PORT = 1521))
            (LOAD_BALANCE = yes)
       )
       (CONNECT_DATA =
            (SERVER = DEDICATED)
            (SERVICE_NAME = racdb)
            (FAILOVER_MODE =
                  (TYPE=SELECT)
                  (MODE=BASIC)
                  (RETRY=3)
                  (DEALY=5)
            )
       )
)
```

其中的几个参数的含义如下：

- TYPE，用于指定FAILOVER_MODE的类型，有3种类型可选，分别是SESSION、SELECT和NONE。

- SESSION，表示当一个正在连接的会话实例发生故障时，系统可以自动将会话切换到其他可用的实例，而应用程序无需再次发起连接请求，但是实例故障时正在执行的SQL需要重新执行。
- SELECT，表示如果正在连接的实例发生故障，将使用游标和之前的快照继续执行SELECT操作，其他操作必须要重新执行。
- NONE，这个是客户端默认值，表示禁止SQL接管功能。
- MODE，表示连接模式，有两种类型，分别是BASIC和PRECONNECT。
- BASIC表示在建立初始连接时仅连接到一个节点，并且只有在发生节点故障时才连接到备用节点。
- PRECONNECT表示在建立初始连接时就连接到主节点和备用节点。
- RETRY：表示当前节点失败后，失败切换功能尝试连接备用节点的次数。
- DELAY：表示两次尝试之间等待的秒数。

设置完客户端监听后，重启客户端服务，然后执行下面的操作：

```
[oracle@client ~]$sqlplus system/xxxxxx@racdb
SQL*Plus: Release 11.1.0.7.0 - Production on Sun Sep 12 23:23:15 2010
Copyright (c) 1982, 2008, Oracle. All rights reserved.
Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - Production
With the Partitioning, Real Application Clusters, OLAP, Data Mining
and Real Application Testing options
SQL> COLUMN instance_name FORMAT a10
SQL> COLUMN host_name FORMAT a10
SQL> COLUMN failover_method FORMAT a15
SQL> COLUMN failed_over FORMAT a10
SQL> SELECT instance_name, host_name, NULL AS failover_type, NULL AS failover_method, NULL AS failed_over FROM v$instance UNION SELECT
NULL, NULL, failover_type , failover_method, failed_over FROM v$session WHERE username = 'SYSTEM';
INSTANCE_ NAME HOST_NAME FAILOVER_TYPE FAILOVER_METHOD FAILED_OVER

---------- ---------- ---------- ---------- ----------
racdb2 node-rac2 SELECT BASIC NO
```

此时，不断开此连接，然后在RAC数据库的任意一个节点上执行如下语句：

```
[oracle@node-rac2 ~]$ srvctl stop instance -d racdb -i racdb2
```

关闭node-rac2节点的racdb2实例后，继续执行与前面那个SQL命令相同的语句，结果如下：

```
INSTANCE_ NAME HOST_NAME FAILOVER_TYPE FAILOVER_METHOD FAILED_OVER

---------- ---------- ---------- ---------- ----------
racdb1 node-rac1 SELECT BASIC YES
```

从输出可以看到，上面的SQL会话已经切换到了node-rac1的实例racdb1上，也就是实现了故障自动切换功能。
至此，关于RAC数据库的功能测试已经验证完毕了。

原始文档：http://pan.baidu.com/s/1eQ5Uyp8