

Spfile unique pour instances Oracle en cluster



par Fabien Celaia ([Fadace](#))

Date de publication : 11.6.2006

Dernière mise à jour : 2.12.2006

Plusieurs instances, une base de donnée... Comment s'en sortir avec les fichiers de configuration Oracle lorsque le cluster y met son grain de sel ?

I - Problématique.....	3
II - Hypothèse.....	4
III - Ce qui ne marche pas.....	5
IV - La démarche.....	6
V - Précautions.....	8

I - Problématique

Une instance SGBDR d'Oracle permet de travailler avec 2 types de fichiers de configurations.

Type de fichier	Nom du fichier	Avantages	Désavantages
pfile	initORACLE_RAC1.ora	Fichier en mode texte, éditable sans soucis	Les modifications passées via éditeur de commande SQL ne l'impactent pas
spfile	spfileORACLE_RAC1.ora	Gait et garde les modifications systèmes apportées par les commandes ALTER SYSTEM du moment que le scope est déterminé (et déterminable) à 'spfile' ou 'both'.	Fichier binaire, non éditable (ou très peu). Une modification non autorisée peut le rendre totalement inutilisable.

Il convient de manipuler ce type de fichier avec prudence, sa corruption ayant souvent pour effet de ne plus permettre le redémarrage de l'instance ou/et l'ouverture de la base de données.

Pour rappel, lors d'un startup, l'instance Oracle suit le chemin suivant:

- 1 Vérifie les variable ORACLE_SID et ORACLE_HOME
- 2 Vérifie si la variable PFILE est renseignée
- 3 Recherche dans \$ORACLE_HOME/dbs (sous Windows, dans %ORACLE_HOME%\database) un éventuel spfileORACLE_SID.ora
- 4 Recherche dans \$ORACLE_HOME/dbs (sous Windows, dans %ORACLE_HOME%\database) un éventuel initORACLE_SID.ora
- 5 Recherche dans \$ORACLE_HOME/dbs (sous Windows, dans %ORACLE_HOME%\database) un éventuel spfile.ora
- 6 Recherche dans \$ORACLE_HOME/dbs (sous Windows, dans %ORACLE_HOME%\database) un éventuel init.ora
- 7 Détecte, selon le contenu de ces fichiers, les différents répertoires et fichiers nécessaires à son startup
- 8 Monte l'instance
- 9 Détecte, selon l'existence de fichiers de contrôles dans le fichier de configuration, les fichiers nécessaires à remonter la base
- 10 Monte la base de données
- 11 Ouvre la base

Dans une installation locale, il est souvent possible de passer de l'un à l'autre aisément.

Sur une installation RAC, le problème vient du fait que plusieurs instances accèdent à une même base de données. Chaque instance ayant par défaut son propre fichier de configuration, la moindre désynchronisation de ce fichier risque d'empêcher une des instances de redémarrer.

Tous les spfiles devant être en grande partie similaires, la solution élégante consiste à n'en utiliser qu'un seul global.

II - Hypothèse

Nom de la base	ORACLE_RAC
Nom de la 1ère instance	ORACLE_RAC1
Nom de la 2ème instance	ORACLE_RAC2
Nom du host de la 1ère instance	host1
Nom du host de la 1ème instance	host2
Version minimum du cluster et de la base	10.1

III - Ce qui ne marche pas

Gravitant dans un environnement Unix, nous serions tenté d'utiliser simplement des liens symboliques.

```
$ cd $ORACLE_HOME/dbs
$ mv spfileORACLE_RAC1.ora /NouvelleLocalisation/spfileORACLE_RAC.ora
$ ln -s /NouvelleLocalisation/spfileORACLE_RAC.ora spfileORACLE_RAC1.ora
$ ls -l spfileORACLE_RAC1.ora
lrwxrwxrwx  1 oracle  dba      41 Mar 30 14:58 spfileORACLE_RAC1.ora -> /NouvelleLocalisation/
spfileORACLE_RAC.ora
```

Cette méthode est tout à fait appropriée pour des fichiers de type *fichier de mots de passe*, mais impossible à faire pour un spfile. Lors de la modification dynamique du spfile, l'instance Oracle ne sera alors pas en mesure d'aller écrire dans le fichier qui n'est qu'un pointeur Unix. Il vous gratifiera alors, lors du premier *create spfile from pfile* d'un *Unable to open file \$ORACLE_HOME/dbs/pfileORACLE_RAC2.ora*.

Trouvons donc quelque chose de plus futé, et qui à l'avantage de fonctionner quel que soit le système d'exploitation utilisé.

IV - La démarche

Se connecter à la base, vérifier l'instance sur laquelle le cluster nous a en fait connecté et créer un pfile

```
$ connect system/**@ORACLE_RAC
SQL> select instance_name, host_name from v$instance ;

INSTANCE_NAME          HOST_NAME
-----
ORACLE_RAC2            host2

SQL> create pfile from spfile ;
```

Se connecter à la machine sur laquelle a été créé le fichier spfile.

```
telnet host1
```

Arrêt de la base de données

```
$ srvctl stop database -d ORACLE_RAC
```

Modification du pfile selon mes besoins.

```
$ cd $ORACLE_HOME/dbs
$ vi initORACLE_RAC2.ora
```

Redémarrage de la base de données en forçant l'utilisation de ce nouveau pfile. Sans cela, la base remontera avec les ancienne valeurs du spfile local, toujours prioritaire sur un pfile.

```
$ cd $ORACLE_HOME/dbs
$ srvctl start database -d ORACLE_RAC -o pfile=oraORACLE_RAC2.ora
```

Régénération d'un nouvel spfile contenant les valeurs du pfile. Celui-ci est bien évidemment créé dans \$ORACLE_HOME/dbs.

```
$ connect system/**@ORACLE_RAC
SQL> create spfile from pfile ;
```

Nouvel arrêt de la base de données pour déplacement du spfile

```
$ srvctl stop database -d ORACLE_RAC
$ mv spfileORACLE_RAC2.ora /LocationCentralisee/spfileORACLE_RAC.ora
```

Création d'un nouveau fichier pfile pointant sur le spfile centralisé, et ce sur les 2 instances, en prenant bien garde de spécifier les bons noms d'instances physiques (1 et 2 sans croiser). Prendre aussi soin de supprimer dès à présent tout spfile dans les répertoires \$ORACLE_HOME/dbs

```
telnet host2
$ cd $ORACLE_HOME/dbs
$ rm spfileORACLE_RAC*.ora
$ vi initORACLE_RAC2.ora
spfile='/LocationCentralisee/spfileORACLE_RAC.ora'
:wq!

telnet host1
$ cd $ORACLE_HOME/dbs
$ rm spfileORACLE_RAC*.ora
$ vi initORACLE_RAC1.ora
spfile='/LocationCentralisee/spfileORACLE_RAC.ora'
:wq!
```

Redémarrage de la base. Choix par défaut : le pfile à ligne unique puisque plus aucun spfile ne doit y être.

```
srvctl start database -d ORACLE_RAC
```

Redémarrage du TAF (la partie service haute disponibilité pour la base)

```
srvctl stop database -d ORACLE_RAC  
srvctl start service -d ORACLE_RAC
```

V - Précautions

Voilà, le passage de plusieurs spfiles locaux à un global a été effectué. Simple, non ?

Il reste néanmoins 2 pièges à éviter maintenant:

```
CREATE PFILE FROM SPFILE ;
```

Cette commande va écraser votre pfile du répertoire `$ORACLE_HOME/dbs` ne comprenant que le pointeur sur le spfile centralisé avec toutes les informations comprises dans ledit spfile.

Au prochain redémarrage, vous vous retrouverez donc toujours avec un spfile correct, mais désynchronisé par rapport à l'autre instance du RAC.

```
CREATE SPFILE FROM PFILE ;
```

Cette commande va créer un spfile d'une ligne unique dans le répertoire `$ORACLE_HOME/dbs`, ne comprenant que le pointeur sur le spfile centralisé.

Au prochain redémarrage, il primera donc sur le pfile.

Ensuite à la moindre modification via `alter system`, il se désynchronisera donc de la version centralisée.

Pour s'assurer ensuite que le RAC redémarrera directement avec le fichier centralisé, sans utiliser notre redirection, il ne nous reste plus qu'à exécuter un

```
srvctl modify database -d ORACLE_RAC -p '/LocationCentralisee/spfileORACLE_RAC.ora'
```