

Introduction à la réplication de data avec Oracle Version 1.0.3*

Rolland Balzon Philippe
Department of Computer Science
 λ^∞ Free Corp.
prolland@free.fr

29 juillet 2002

1 Réplication par l'applicatif

Le problème est le suivant : 3 instances, un applicatif. Cet applicatif écrit dans une des instances. Deux cas l'applicatif en dur va écrire dans les deux autres, ou par trigger le moteur sgbd va synchroniser l'écriture dans les deux autres.

Pour cette dernière option, on remarquera que si la deuxième instance tombe, on aura un défaut d'intégrité.

Pour pallier à cette éventualité de défaut d'intégrité On peut simuler un Commit à deux phases ; i.e. utiliser des mécanismes d'écriture quasi-parallèle sur tous les instances quand toutes celle -ci sont prêtes à écrire.

2 Réplication par le moteur du SGBD

Nous sommes dans un cas similaire à une intance qui dois arroser d'autre instance. Il y a deux mécanismes de réplication proposé par Oracle. Le pull et le push. Dans le premier cas c'est les instances esclaves qui viennent chercher les news ; dans le second c'est le maître qui dirige.

2.1 Pull version

2.1.1 Incrémental

Nous sommes sur un esclave. Sur cet esclave, par dblink par exemple, on voit les tables des autres instances maîtres. Pour chacune des tables esclaves on va spécifier plusieurs éléments dont ceux-ci

- une table maître
- une requête sur celle-ci
- une fréquence de rafraîchissement

On donne un exemple de déclaration d'une structure pl/sql de type snapshot :

```
; création du snapshot S_COURS_DEVISE sur l'instance fille SCPE  
;à partir de la table COURS_DEVISE de l'instance mère ECP.ANTARES  
;(ECP.ANTARES est le nom du lien précédemment ;déclaré à partir de  
;l'instance fille)
```

*This document can be found to <http://prolland.free.fr/works/oracle/replication.pdf>

```

CREATE SNAPSHOT S_COURS_DEVISE STORAGE

(INITIAL 64K NEXT 32K PCTINCREASE 0)

REFRESH COMPLETE START WITH SYSDATE NEXT
to_date(to_char(SYSDATE,'DD/MM/YYYY')||'|' 13:00:00', 'DD/MM/YYYY
HH24:MI:SS') + 1

AS SELECT TO_NUMBER(SOC_CODE) SOC_CODE, DEV_CODE, CDE_DDEB,
CDE_DFIN, CDE_COURS, CDE_EURO, CDE_STD, CDE_CRT FROM
COURS_DEVISE@ECP.ANTARES

```

On remarquera que les journaux de snapshot sont indépendants des journaux redolog. Dans le pire des cas on trace donc 2 fois la même chose d'où un déficit d'optimisation sur ce point. Les développeur d'Oracle cherche visiblement à recentrer les snapshot sur les redlog.

2.1.2 Not Incrémental

Nous n'avons pas beaucoup d'information sur ce mode. Ce que nous savons c'est que l'esclave viens chercher des blocs Oracle (modifié ou pas) sur le maître pour les importer. Cela conviens visiblement pour des rafraichissements peu nombreux (un quotidien nocturne par exemple), mais où le maître est très modifié.

2.2 Push version

Cela utilise un journal par instance. Impossible si trop de mise à jour. En fait elle fonctionne par trigger et en pl/sql.

3 Schema

Dans certain cas on peut passer outre la création de plusieurs instances centralisées ou distribuées. On peut par exemple avoir une seule instance, avec des schemas (user) différents. Un user accédera à un ensemble de tables, un autre user à un autre ensemble de table. Cela permet d'avoir un seul dictionnaire.

4 dblink

Les limitations :

- Pas de respect de contraintes entre table locales et tables dblink.
 - Lenteurs dans les requêtes mettant en oeuvre des tables de différentes instances.
- Une déclaration de dblink peut être comme suit :

```

CREATE PUBLIC DATABASE LINK GRH.MARCUS connect to grh_link identified by
grh_link using 'grhmrc';

```

5 Virtual Private Database

La solution Virtual private database (8i) permet plusieurs mécanismes, par exemple :

- Ajout d'une clause where automatique. C'est le dba qui gère cette clause. Ainsi par exemple, un utilisateur pourra voir que les occurrences d'une table pays ou pays-code = "fr".
- Possibilité de préfixer le nom d'une table avec un schéma automatiquement.

6 MiddleWare

Il existe un middleware livré avec Oracle 8i qui permet l'envoi de messages inter-applicatif. Cela se révélera très utile dans un environnement multi-applicatifs.