

Édition numéro 2 pour la Romandie

DOAG SOUG News

Erreur d'I/O sur un datafile:
crash d'instance ou datafile offline?"



Le premier numéro du magazine DOAG/SOUG fait déjà partie de l'histoire ancienne, vous tenez le n°2 entre vos mains.



Gaetano Bisaz
SOUG

Le premier numéro du magazine DOAG/SOUG fait déjà partie de l'histoire ancienne, vous tenez le n°2 entre vos mains.

Le perpétuel changement caractérise notre branche, une situation à laquelle nous sommes nous aussi confrontés, en tant que groupe d'utilisateurs. La nouvelle newsletter est un début, cela va se poursuivre avec de la modernisation, certains points ont été soulevés lors de l'assemblée générale, nous en relaterons dans notre prochain numéro.

Mobility et Cloud sont des notions qui trônent sans doute sur votre "Hype-Cycle". Contrairement à d'autres buzzs, ceux-ci ne disparaissent pas aussi vite qu'ils sont apparus, au contraire, ils font partie de notre vie quotidienne. Il est intéressant d'observer comment des sites Web pour appareils mobiles sont devenus des sites Web normaux, et qu'actuellement beaucoup de ces sites ne sont appréciables que sur les appareils mobiles. Cloud? Presque de l'histoire ancienne - Les frontières entre les services On Premise et le service Cloud sont minces, avec des hybrides et de nouvelles approches entre les deux.

La mise à disposition de "Database as a Service" sur des architectures Oracle consolidées, par exemple avec des bases "Multitenant", doit être bien préparée. Franck Pachot met en évidence un comportement d'Oracle, apparu depuis la version 11.2.0.2. Celui-ci concerne l'état de la base de données suite à une erreur d'écriture sur un datafile. Ce changement fondamental est un peu passé inaperçu car la suppression d'un datafile sous Oracle est plutôt rare. Cependant, pour les versions précédentes au patchset 2 de la 11gR2 le datafile était mis offline et l'instance continuait à tourner. Aujourd'hui, dans les versions supérieures à 11.2.0.2 (12c inclus), dès qu'une erreur d'écriture sur un datafile se produit, toute l'instance est arrêtée. Les conséquences peuvent être importantes en "Multitenant Database" car toute la CDB n'est plus disponible. L'article explique la raison du changement, dans quels cas le nouveau comportement est acceptable, et comment revenir au comportement précédent.

Gaetano Bisaz

Agenda

21.05.2015
**SOUG SIG Tuning | Prangins
(Schweiz)**
sekretariat@soug.ch

11.06.2015
SOUG SIG
sekretariat@soug.ch

03.09.2015
SOUG SIG
sekretariat@soug.ch

17.-20.11.2015
DOAG 2015 Konferenz + Ausstellung



Erreur d'I/O sur un datafile: crash d'instance ou datafile offline?

Franck Pachot, dbi-services

L'arrivée du multitenant rend encore plus critique la disponibilité de l'instance, puisqu'elle peut maintenant faire tourner plusieurs bases. Que pensez-vous qu'il se passe lorsqu'il y a une erreur d'écriture sur un datafile ? Il y a eu un changement majeur en 11.2.0.2 qui est un peu passé inaperçu car on ne supprime pas souvent un datafile. Sauf bien sûr lorsque nous installons une nouvelle infrastructure Oracle chez un client, car nous testons toujours les scénarios de backup/recovery avant de la mettre en production. Voyons donc ce qu'il se passe en supprimant un datafile en RAC et surtout en 12c multitenant...

Pour rappel, nous avons l'habitude de rencontrer les scénarios suivants:

- Perte d'un membre du controlfile = crash d'instance.
- Perte d'un membre de redo log = message dans alert.log. S'il n'y a pas d'autre membre, on devra redémarrer l'instance (avec perte de transactions).
- Perte d'un datafile système = crash d'instance
- Perte d'un datafile non-système = le datafile passe offline

Le dernier cas n'est plus vrai depuis la 11.2.0.2, voici un exemple sur un 12c:

```
RMAN> host "rm -f /u02/oradata-  
ta/CDB/PDB1/PDB1_users01.dbf";
```

```
host command complete  
  
RMAN> alter system checkpoint;  
RMAN-00571: =====  
=====
```

```
RMAN-00569: =====  
ERROR MESSAGE STACK FOLLOWS  
=====
```

```
RMAN-00571: =====  
=====
```

```
RMAN-00601: fatal error in  
recovery manager  
RMAN-03004: fatal error during  
execution of command  
ORA-01092: ORACLE instance  
terminated. Disconnection  
forced  
RMAN-03002: failure of  
sql statement command at  
02/19/2015 22:51:55  
ORA-03113: end-of-file on com-  
munication channel
```

```
Process ID: 19135  
Session ID: 357 Serial number:  
41977  
ORACLE error from target da-  
tabase:  
ORA-03114: not connected to  
ORACLE
```

J'ai lancé le 'alter system' à partir de RMAN car en 12c on peut exécuter toute requête SQL à partir de RMAN. Mais l'important ici est que mon instance est stoppée. Tous les services (de toutes les PDB) sont arrêtés, même si personne n'a besoin du tablespace USERS de la PDB1.

Avant, depuis les plus vieilles versions d'Oracle, ce scénario aurait simplement mis le tablespace USERS offline mais la plupart des

utilisateurs pouvaient continuer à travailler.

Pourquoi mettre le datafile offline?

On est tellement habitués au comportement des anciennes versions que l'on trouve souvent normal de mettre le datafile offline s'il n'y a pas besoin de stopper l'instance. Mais pourquoi continuer à laisser l'instance tourner, en mettant seulement le datafile offline, lorsqu'on ne peut pas écrire les 'dirty buffers' lors du checkpoint ? Pourquoi ne pas stopper l'instance directement, comme ce qu'il se passe lorsque c'est un datafile système qui manque ? Et pourquoi ce comportement seulement lorsqu'on est en archive log mode ?

La raison, c'est que pour un tablespace non-système, et lorsqu'on est en archive log mode, et si le problème vient d'une panne disque, alors vous pouvez faire un 'restore datafile', 'recover datafile' et 'alter datafile online' sans avoir à stopper/redémarrer l'instance.

S'il s'agit d'un tablespace système, la base ne peut pas rester ouverte. Si vous n'êtes pas en archive log mode, le datafile ne peut pas être récupéré. Et si tout votre storage n'est plus disponible, alors de toute façon plus personne ne peut travailler – autant arrêter l'instance.

Mais si vous avez consolidé plusieurs applications sur une base de données, les applications qui n'utilisent pas le tablespace perdu peuvent continuer à être utilisés. Donc ce choix de mettre le datafile offline au lieu de stopper l'instance est guidé par le choix de laisser disponible une partie de l'application ouverte.

Ce choix était valable pour beaucoup d'installations par le passé, mais aujourd'hui :

- Quelle est la probabilité de perdre un seul datafile plutôt que tout le storage? Aujourd'hui tout est stripé. Et en ASM, on ne touche pas aux datafiles donc pas de risque d'en supprimer un par erreur.

- Avez-vous encore beaucoup d'utilisateurs qui peuvent continuer à travailler lorsqu'un datafile est offline ? On a souvent une seule application par base. Nous verrons le cas du multitenant plus loin.
- N'avez-vous pas des mécanismes de failover qui permettent de rendre la totalité du service en quelques minutes – donc beaucoup moins que le temps de récupération d'un fichier ? Si l'instance ne voit plus le fichier mais qu'il est toujours là, RAC maintient l'accès sur une autre instance. Si le fichier est perdu, DataGuard (ou Dbvisit standby lorsqu'on est en Standard Edition) permet d'ouvrir une copie de la base rapidement, et de manière automatique si elle est en fast-start failover avec un observateur.

Plutôt que de mettre le datafile offline – et donc de rendre l'application non disponible même sur tous les nœuds d'un RAC – en attendant une opération manuelle de restore, ne vaut-il pas mieux stopper l'instance et rendre la base disponible ailleurs si un autre serveur peut voir le fichier ?

11gR2 – Patchset 2

C'est donc l'idée du 'bug 7691270', plutôt considéré comme 'enhancement request', apparu en 11.2.0.2 et mettant fin à ce comportement particulier qu'on connaissait depuis longtemps: toute erreur d'écriture au checkpoint provoque maintenant l'arrêt immédiat de l'instance. Et c'est bien sûr toujours le cas en 12c.

Evidemment, c'est un 'shutdown abort' qui est fait puisqu'on ne peut plus écrire les 'dirty buffers'. Mais les modifications sont protégées par le redo. C'est aussi la raison pour laquelle le comportement précédent – datafile offline – n'était implémenté qu'en archive log mode, afin d'être sûr de pouvoir faire le recovery du fichier une fois récupéré. En noarchive log mode, le redo serait prochainement écrasé lors d'un futur log switch, donc le crash de l'instance est indispensable pour éviter de générer plus de redo.

On en reparle plus bas à propos des pluggable databases.

A noter que ceci ne concerne que les écritures de 'dirty buffers', lors d'un checkpoint. Les erreurs de lectures, ou d'écriture en direct-path, sont écrites dans l'alert.log mais ne changent rien au status du datafile ni de l'instance.

_datafile_write_errors_crash_instance

Ce nouveau comportement est activé par le paramètre :

```
_datafile_write_errors_crash_instance = true
```

True est la valeur par défaut. On peut revenir au mode précédent avec:

```
alter system set "_datafile_write_errors_crash_instance"=false
```

Le nouveau comportement correspond d'avantage à ce qu'on souhaite aujourd'hui, mais si vous avez consolidé plusieurs applications dans une base, que vous n'avez pas de mécanisme automatique de failover, et que vous estimez qu'il est possible de perdre un datafile sans les perdre tous, alors il peut être souhaitable de mettre ce paramètre à false.

Deux rappels :

- C'est un paramètre non-documenté, c'est une bonne idée de demander au support oracle s'il n'y a pas de contre-indication dans votre contexte.
- Si vous n'êtes pas en ARCHIVELOG mode, alors peu importe, puisque de toute façon vous ne tenez pas à vos données.

RAC

Ce changement a principalement été motivé par le fait que de plus en plus de bases sont en haute disponibilité.

Mettre un datafile offline en RAC le rend inaccessible à toutes les instances. Or souvent, l'erreur d'écriture ne vient pas de l'absence du fichier, mais d'un problème d'accès à partir d'un nœud. Il est donc préférable de laisser les autres instances tourner.

Si en RAC on préfère mettre le datafile offline, avec "_datafile_write_errors_crash_instance" à false, alors il faut s'assurer d'être en 11.2.0.4 ou 12c car sinon, à cause du bug 13745317, l'agent du cluster stoppe les instances lorsqu'un datafile est offline. C'est donc pire : plus aucune instance n'est disponible dans le cluster.

Multitenant 12c

Nous avons vu plus haut qu'une raison de mettre le datafile offline était la consolidation de plusieurs applications dans une même base, ce qui n'est plus très courant aujourd'hui, d'où la décision d'arrêter en 11.2.0.2.

Mais la 12c a amené une nouvelle manière de consolider avec le multitenant. Si l'on fait du multitenant pour consolider plusieurs applications sur une même instance, veut-on toujours stopper toute l'instance (donc toute la CDB) à la première erreur d'écriture sur un datafile user d'une pluggable database (PDB) ?

Si la Haute Disponibilité est assurée par DataGuard en fast-start failover, alors oui, il est peut-être préférable de basculer tous les utilisateurs vers le site qui n'a pas de problèmes d'I/O. Si on est en RAC par contre, il est possible de ne basculer que les services associés à cette pluggable database. Tout dépend aussi si l'on doit assurer le niveau de service (SLA) par PDB ou pour toute la CDB.

Donc en multitenant, il est possible qu'on préfère stopper seulement la PDB.

Pour cela, il faudra accepter de mettre les datafiles offline :

```
alter system set "_datafile_write_errors_crash_instance"=false
```

Mais ce n'est pas suffisant. Cela ne concerne que les datafiles non-système. On ne veut pas non plus stopper

toute la CDB si on perd le tablespace système d'une seule PDB. J'avais ouvert un bug là-dessus (19001390), qui est corrigé dans le dernier PSU.

Pluggable database

Bien, avec la correction de ce bug, l'instance CDB ne s'arrête pas. Mais ce n'est pas encore suffisant... si on a perdu un datafile système, il va falloir le restaurer. Et lorsqu'on perd un datafile système, on fait un shutdown abort puisqu'on ne peut pas mettre un datafile système offline. En non-CDB, pas de problème. L'instance étant arrêtée, on ne risque pas de perdre les online redo logs.

Mais faire un 'shutdown abort' d'une PDB est différent. Les online redo logs couvrent toute la CDB et vont être écrasés aux prochains log switches. Par défaut Oracle nous empêche de faire ça. Il faudra donc attendre de pouvoir stopper toute la CDB afin de faire le restore/recover du fichier manquant.

C'est un gros problème de disponibilité. Si on est en archivelog mode, on peut accepter de faire ce 'shutdown abort' de PDB puisqu'on garde le redo. On peut autoriser ça avec :

```
alter system set "_enable_pdb_close_abort"=true
```

Si l'on n'est pas en archivelog mode, alors c'est plus risqué. Donc même avec ce paramètre à true, la PDB ne sera pas mise offline. On peut malgré tout autoriser cela avec un paramètre équivalent – mais pour mode noarchivelog :

```
alter system set "_enable_pdb_close_noarchivelog"=true
```

Bien sûr, on risque alors de perdre la PDB dans ce cas. C'est acceptable seulement si on peut la recréer.

Ces deux paramètres peuvent se définir au niveau de chaque PDB, mais il faut les définir au niveau CDB si on ne veut pas que l'instance soit arrêtée au moment du checkpoint. C'est donc

le comportement de toutes les PDB qu'il faut définir.

Exemple

Pour résumer, voici le comportement après avoir supprimé le datafile système d'une PDB lorsque la CDB (avec dernier PSU) a les paramètres suivants

```
"_datafile_write_errors_crash_instance"=false
"_enable_pdb_close_abort"=true
```

Je vais utiliser le Recovery Advisor pour voir l'erreur :

```
RMAN> list failure;

Database Role: PRIMARY

List of Database Failures
=====

Failure ID Priority Status
Time Detected Summary
-----
897          CRITICAL OPEN
23-MAR-15    System datafile
8: '/u02/oradata/CDB/PDB/datafile/o1_mf_system_bk04nchr_.dbf' is missing
```

Ce fichier est le datafile du tablespace système de la pluggable database PDB. Grâce au paramétrage précédent, l'instance n'a pas été stoppée au checkpoint.

```
RMAN> advise failure;

Database Role: PRIMARY

List of Database Failures
=====

Failure ID Priority Status
Time Detected Summary
-----
897          CRITICAL OPEN
23-MAR-15    System datafile
8: '/u02/oradata/CDB/PDB/datafile/o1_mf_system_bk04nchr_.dbf' is missing

analyzing automatic repair op-
```

```
tions; this may take some time
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=50
device type=DISK
analyzing automatic repair op-
tions complete
```

```
Mandatory Manual Actions
=====
no manual actions available
```

```
Optional Manual Actions
=====
1. If file /u02/oradata/CDB/
PDB/datafile/o1_mf_system_
bk04nchr_.dbf was unintention-
ally renamed or moved, restore
it
2. Automatic repairs may be
available if you shutdown the
database and restart it in
mount mode
```

```
Automated Repair Options
=====
Option Repair Description
-----
1 Restore and recover
datafile 8
Strategy: The repair in-
cludes complete media recovery
with no data loss
Repair script: /u01/app/
oracle/diag/rdbms/cdb/CDB/hm/
reco_2804473397.hm
```

Le Recovery Advisor propose le resto-
re/recover:

```
RMAN> repair failure;

Strategy: The repair includes
complete media recovery with
no data loss
Repair script: /u01/app/ora-
cle/diag/rdbms/cdb/CDB/hm/
reco_2804473397.hm
```

```
contents of repair script:
# restore and recover da-
tafile
sql 'PDB' 'alter database
datafile 8 offline';
restore ( datafile 8 );
recover datafile 8;
sql 'PDB' 'alter database
datafile 8 online';
```

```
Do you really want to execute
the above repair (enter YES or
NO)? YES
executing repair script
```

```
sql statement: alter database
datafile 8 offline
```

Mais:

```
RMAN-00571: =====
=====
RMAN-00569: =====
ERROR MESSAGE STACK FOLLOWS
=====
RMAN-00571: =====
=====
RMAN-03002: failure of repair
command at 03/23/2015 14:22:34
RMAN-03015: error occurred in
stored script Repair Script
RMAN-03009: failure of sql
command on default channel at
03/23/2015 14:22:34
RMAN-11003: failure during
parse/execution of SQL state-
ment: alter database datafile
8 offline
ORA-01541: system tablespace
cannot be brought offline; shut
down if necessary
```

Mais malheureusement – encore un bug
– la pluggable database n'est pas fer-
mée et le Recovery Advisor a oublié cette
étape. Il faut donc le faire à la main :

```
RMAN> alter pluggable database
PDB close;

Statement processed

Ceci est possible grâce à "_
enable_pdb_close_abort"=true
(je suis en mode archivelog).
```

Le recovery peut maintenant se faire :

```
RMAN> repair failure;
...
media recovery complete,
elapsed time: 00:00:01
Finished recover at 23-MAR-15

sql statement: alter database
datafile 8 online
repair failure complete

Il nous reste à ouvrir la PDB.
```

Comme il y a eu un shutdown abort,
il faut aussi faire un recover sur tous
les fichiers :

```
RMAN> alter pluggable database
PDB open;

RMAN-00571: =====
=====
RMAN-00569: =====
ERROR MESSAGE STACK FOLLOWS
=====
RMAN-00571: =====
=====
RMAN-03002: failure of
sql statement command at
03/23/2015 14:40:27
ORA-01113: file 9 needs media
recovery
ORA-01110: data file 9: '/u02/
oradata/CDB/datafile/o1_mf_
sysaux_bjdxgygdj_.dbf'

RMAN> recover pluggable data-
base PDB;

Starting recover at 23-MAR-15
using channel ORA_DISK_1

starting media recovery
media recovery complete,
elapsed time: 00:00:00

Finished recover at 23-MAR-15

RMAN> alter pluggable database
PDB open;

Statement processed
```

Ce n'est donc pas tellement automa-
tisé, mais c'est possible sans impac-
ter la disponibilité des autres PDB.
Bien sûr, il faut avoir confiance dans
le monitoring. On aura besoin des ar-
chivelogs, il est préférable de ne pas
trop attendre.

Conclusion

J'ai détaillé les différents scenarios et
paramètres car il faut bien compren-
dre les conséquences en termes de dis-
ponibilité. Et il n'y a pas de cas géné-
ral : la haute disponibilité est assurée
de manière différente sur chaque site
(RAC, Dataguard avec ou sans obser-
ver,...), et les besoins de disponibilité

du service et des données sont propres à chaque application.

Il est toujours préférable de garder les paramètres par défaut. Mais dans certains cas décrits ici, on peut préférer mettre le paramètre "_datafile_write_errors_crash_instance" à false. Et dans ce cas, si l'on est en multitenant archivelog, il est recommandé de passer le dernier PSU et de définir aussi "_enable_pdb_close_abort" à true afin de pouvoir restaurer le fichier manquant sans interrompre le service des autres PDB.

Il est par contre important dans tous les cas de s'assurer que le monitoring de l'alert.log envoie une alerte dès qu'il y a une erreur d'écriture sur un fichier. Plus on attend et plus la récupération sera longue et risquée.

Dernière remarque. Aujourd'hui, beaucoup considèrent que la disponibilité des fichiers est assurée par le stockage, avec les niveaux de RAID assurant le mirroring, les SAN synchro-

nisés, etc. On oublie malheureusement trop souvent l'erreur humaine. La perte d'un fichier peut simplement venir d'une mauvaise manipulation, d'un bug dans un script, etc. Et dans ce cas, le responsable peut rapidement faire un restore/recover et remettre le datafile online, sans devoir stopper toutes les applications.

Pour la même raison, par notre expérience chez les clients, nous conseillons toujours de multiplexer les controlfiles et redo log files même sont mirrorés au niveau des disques.



Franck Pachot
franck.pachot@dbi-services.com

Franck Pachot is senior consultant at dbi services in Switzerland. He has 20 years of experience in Oracle databases, all areas from development, data modeling, performance, administration, and training. Oracle Certified Master and Oracle ACE, he tries to leverage knowledge sharing in forums, publications, presentations.



<http://tinyurl.com/njespw9>

SIG Tuning - le 21 mai 2015

Rencontrez les experts en Romandie!

Hôtel La Barcarolle - CH-1197 Prangins (hotel-labarcarolle.ch)

présentations:

- Modern Linux Tools for Oracle Troubleshooting, de Luca Canali et Zbigniew Baranowski - CERN
- SQL Plan Directives, de Franck Pachot, dbi services
- Adaptive Dynamic Sampling, de Christian Antognini - Oracle
- Anatomy of a slow query, de Lothar Flatz - DISO

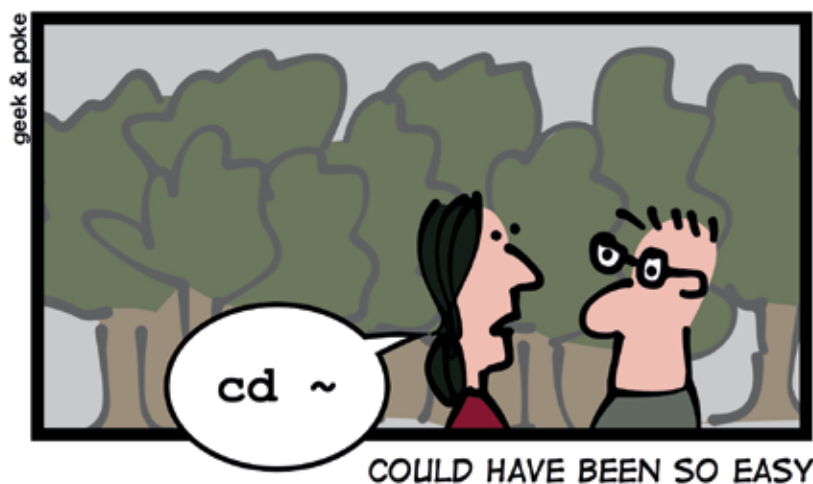
*** Aperitif dinatoire ***



Registration: www.soug.ch/events/sig-150521-anmeldung.html



HANSEL AND GEEKEL



Link: <http://geek-and-poke.com/geekandpoke/2014/7/31/hansel-and-geekel>

License: http://creativecommons.org/licenses/by/3.0/deed.en_US

Plaisir anticipé:

Dans le prochain numéro du DOAG/SOUG News vous trouverez un article sur Oracle et Docker, rédigé par l'auteur de l'article de cette édition. Cet artic-

le sera le premier disponible en trois langues! Nous nous réjouissons d'ores et déjà de lire cette approche passionnante et novatrice d'Oracle et Docker.

Mentions légales

secrétariat SOUG:

Dornachstrasse 192, 4053 Basel
Tel.: 061 367 93 30, Fax: 061 367 93 31
sekretariat@soug.ch

rédaktion:

Geatano Bisaz
gaetano.bisaz@soug.ch

réalisation / DTP:

DOAG Dienstleistungen GmbH
Tempelhofer Weg 64, 12347 Berlin
office@doag.org

impression:

Druckerei Rindt GmbH & Co. KG
www.rindt-druck.de

rédaction Newsletter:

nl@soug.ch

inscription aux événements SOUG:

event@soug.ch

questions des membres:

sekretariat@soug.ch

site du web:

www.soug.ch

Photo couverture: © alphaspirt / 123RF.com

Photo p. 3: © Kirill Makarov / 123RF.com

Newbies CH

Sascha Körner, Rudolph AG
Rolf Maag, Banque Pictet & Cie SA
Ricardo Santandrea, Banque Pictet & Cie SA
David Schoch, Six Group Services AG
Shahid Abdul, Six Group Services AG