|  |  |
| --- | --- |
|  |  |

Audit de la base Oracle MARCELLO

Le Bon Marché

Rapport d’audit

|  |  |
| --- | --- |
|  | **Version :** V 0.21  **Date de la version**: 16/10/2017  **Etat du document :** Draft |

Diffusion

|  |  |
| --- | --- |
| Nom | Société |
| … | UMANIS |
|  |  |

Historique des modifications

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Auteur | Objet de la version et liste des modifications |
| V 1.0 | 03/10/2017 | JEA | Initialisation |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

VAlidation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Version | Emetteur | Date / Visa | Vérificateur | Date / Visa | Approbateur | Date / Visa |
| V 0.21 | JEA | 16/10/2017 |  |  |  |  |

**Date d'application :** 16/10/2017

Table des matières

[1 Objet de l’audit 4](#_Toc495935371)

[2 Environnement 5](#_Toc495935372)

[3 Les points forts 6](#_Toc495935373)

[4 Les points faibles 7](#_Toc495935374)

[5 Les opportunités d’amélioration 8](#_Toc495935375)

[6 Annexes 9](#_Toc495935376)

# Objet de l’audit

Cet audit a pour but de déterminer le comportement global de la base MARCELLO supportant la solution Cylande. Il vise en particulier à expliquer les alertes régulièrement remontées dans le monitoring des applications sur cette base de données

# Environnement

## Os

La base MARCELLO est installée sur un système Microsoft Windows Server 2008 64-bit disposant de 8 Go de RAM.

## RDBMS

MARCELLO est une base en version 10.2.0.5.0. Elle est en rôle PRIMARY\_INSTANCE et fonctionne en NO ARCHIVE LOG.

## Les snapshots statspack

Les prises de mesures sont paramétrées tous les jours, du 03/10/2017 au 20/10/2017, de 10 heures à 20 heures avec une fréquence de 15 minutes ; tous les snapshots sont conservés sur cette période. Il est ainsi possible d’investiguer sur les perturbations et dysfonctionnements perçues dans cet intervalle.

# Les points forts

## Base éfficiente et stable

L’essentiel du modèle de donné est équilibré : les tables et vues les plus utilisées (celles revenant le plus souvent dans les snapshots) disposent de clés primaires et référentielles cohérentes et référencées dans les restrictions des requêtes les plus utilisées. De même, les tables ayant un nombre d’enregistrement négligeable (telles que les tables de paramètres) n’ont généralement pas pour ne pas en dégrader les temps d’accès.

Les ressources systèmes ne sont pas débordées lors des phases d’activité critiques. Nous avons cependant noté un taux d’occupation moyen de la mémoire très élevé, 85% de la RAM.

# Les points faibles

## Version d’Oracle non supportée par le constructeur

La version 10.2 d’Oracle n’est plus supportée par le constructeur (<http://www.oracle.com/us/support/library/lifetime-support-technology-069183.pdf>, en page 4) depuis juillet 2013.

## Quelques objets compilés en mode debug

Ces objets reviennent souvent sans la liste de ceux générant le plus de consommation de ressource CPU et de ressource mémoire. Ils sont aussi parmi ceux qui génèrent le plus de Waits. Nous pouvons ainsi citer par exemple le package PK\_LBM\_VENTES\_GU.

## Présence d’objets invalides

Il y a de nombreux objets invalides dans la base de données. Voici leur répartition par type et par propriétaire.

|  |  |  |
| --- | --- | --- |
| COUNT | OWNER | OBJECT\_TYPE |
| 227 | OLAPSYS | VIEW |
| 1 | OLAPSYS | FUNCTION |
| 32 | OLAPSYS | PACKAGE BODY |
| 173 | PUBLIC | SYNONYM |
| 10 | STORELAND | PROCEDURE |
| 3 | STORELAND | PACKAGE |
| 15 | STORELAND | PACKAGE BODY |
| 1 | SYSMAN | PACKAGE |
| 15 | SYSMAN | PACKAGE BODY |
| 1 | SYSMAN | TRIGGER |
| 4 | SYSMAN | FUNCTION |

# Les opportunités d’amélioration

## Mettre en place un plan de migration vers une version en cours de support

Disposer d’une version qui n’est plus supportée par Oracle entraîne des risques de surcout en cas de nécessité de support par Oracle. Il y a aussi le risque de devoir migrer d’urgence en cas de dysfonctionnement important.

## Augmenter la RAM

Le taux moyen de sollicitation de la RAM étant très élevé, le serveur peut être ralenti par un recours massif au mécanisme de swap. Plus de RAM devrait le soulager lors des pics de traitements.

## Compiler en mode standard les objets compilés en mode débug.

Attention, il peut y avoir des instabilités si un grand nombre d’objets ayant de fortes interdépendances sont ainsi refacturés. Il faudrait le faire avec une session d’administration connectée en mode autonome et redémarrer la base Oracle à la suite de cette séance de recompilation d’objets.

## Corriger ou supprimer les objets invalides

La présence d’objets invalides est source de confusion. En cas de dysfonctionnement des applications, les objets invalides semant la confusion dans l’analyse. Le délai de résolution des pannes sera augmenté des durées de levée des doutes sur la pertinence des objets en cause. Il faut donc supprimer les objets réellement inutiles et corriger ceux qui restent invalides.

## Ajouter des index

Agrandir

## Corriger des requêtes consommatrices de ressources

### Package STORELAND.PK\_LBM\_VENTES\_GU

#### Procédure PK\_LBM\_VENTES\_GU.LBM\_GETTICKETSBM

Cette procédure affiche de mauvaises performances par ce qu’elle est très souvent sollicitée et scanne entièrement une grosse table (LBM\_VENTE\_POSTE), n’exploitant pas son index de clé primaire (LBM\_VENTE\_POSTE\_PK) comme elle le devrait.

/\*

\*\* Corriger la restriction sur lbm\_vente\_poste.stecaisse

\*\* pour ne plus appliquer une fonction sur la clé de l'index

\*\* -- ancien code (2 occurrences) : WHERE stecaisse IN (1, 2)

\*\* -- remplacé par (2 occurrences) : WHERE stecaisse IN ('1', '2')

\*\*/

UPDATE lbm\_vente\_poste v

SET topgu = 2

WHERE **stecaisse IN ('1', '2')**

AND appl = 'SAP'

AND ROWNUM <= p\_rownum

AND hca IS NULL

AND departement = p\_dep

AND ( (typegest != '5' AND typegest != '6' AND p\_typegestion = 'F')

OR ((typegest = '5' OR typegest = '6') AND p\_typegestion = 'G')

)

AND topgu = 0

AND numcdeclt IS NULL

AND TRUNC (datehtrans) =

(SELECT TRUNC (datehtrans)

FROM (SELECT datehtrans

FROM lbm\_vente\_poste v LEFT JOIN lbm\_cp\_histo h

ON v.cp = h.cp

WHERE **stecaisse IN ('1', '2')**

AND v.appl = 'SAP'

AND v.hca IS NULL

AND departement = p\_dep

AND ( ( v.typegest != '5' AND v.typegest != '6' AND p\_typegestion = 'F' )

OR ( (v.typegest = '5' OR v.typegest = '6') AND p\_typegestion = 'G' )

)

AND topgu = 0

AND ( (datehtrans < h.dateblocageinventaire)

OR (h.dateblocageinventaire IS NULL)

AND numcdeclt IS NULL

)

ORDER BY datehtrans)

WHERE ROWNUM <= 1)

AND ( (datehtrans < (SELECT dateblocageinventaire

FROM lbm\_cp\_histo h

WHERE h.cp = v.cp))

OR ((SELECT dateblocageinventaire

FROM lbm\_cp\_histo h

WHERE h.cp = v.cp) IS NULL)

);

Cette correction ne sera cependant pas transcendante puisque la colonne lbm\_vente\_poste.stecaisse n’est pas suffisamment discriminante. Or la seconde colonne dans la clé d’index, lbm\_vente\_poste.datehtrans qui devrait rendre le recours à l’index plus efficace subit aussi une fonction TRUNC. Il faudra évaluer la validité de la réécriture proposée ci-dessous pour s’assurer que la requête conserve sa valeur fonctionnelle.

UPDATE lbm\_vente\_poste v

SET v.topgu = 2

WHERE v.stecaisse IN ('1', '2')

AND v.appl = 'SAP'

AND ROWNUM <= p\_rownum

AND v.hca IS NULL

AND v.departement = p\_dep

AND ( (v.typegest != '5' AND v.typegest != '6' AND p\_typegestion = 'F')

OR ((v.typegest = '5' OR v.typegest = '6') AND p\_typegestion = 'G')

)

AND v.topgu = 0

AND v.numcdeclt IS NULL

AND exists -- TRUNC (v.datehtrans) =

(SELECT 1 -- TRUNC (w.datehtrans)

--FROM (SELECT w.datehtrans

FROM lbm\_vente\_poste w

LEFT JOIN lbm\_cp\_histo h

ON w.cp = h.cp

WHERE w.stecaisse IN ('1', '2')

and w.datehtrans = v.datehtrans

----- ou encore and w.datehtrans between trunc(v.datehtrans)

and trunc(1 + v.datehtrans)

AND w.appl = 'SAP'

AND w.hca IS NULL

AND w.departement = p\_dep

AND ( ( w.typegest != '5' AND w.typegest != '6' AND p\_typegestion = 'F' )

OR ( (w.typegest = '5' OR w.typegest = '6') AND p\_typegestion = 'G' )

)

AND w.topgu = 0

AND ( (w.datehtrans < h.dateblocageinventaire)

OR (h.dateblocageinventaire IS NULL)

AND w.numcdeclt IS NULL

)

--ORDER BY w.datehtrans

--)

--WHERE ROWNUM <= 1

)

AND ( (v.datehtrans < (SELECT h.dateblocageinventaire

FROM lbm\_cp\_histo h

WHERE h.cp = v.cp

)

)

OR ((SELECT h.dateblocageinventaire

FROM lbm\_cp\_histo h

WHERE h.cp = v.cp

) IS NULL

)

);

#### Procédure STORELAND.PK\_LBM.LBM\_EXTRACTIONS

Lancement des traitements (dans un contexte de lock applicatif) :

1. PRC\_LBM\_VENTE
2. PRC\_LBM\_TypeRemise
3. PRC\_LBM\_TypeAM

#### Procédure STORELAND.PK\_LBM.LBM\_INTEGRATIONCLIENT

Réduire les accès disques et les FULL SCANS

Rectifier le code extrait ci-dessous comme indiqué :

---------------------------------------------------------

-- 3.) mettre a jour LBM\_MESSAGE\_ID si id\_batch est null (nouveaux clients)

-- ou id\_batch est negatif (si plantage trt precedent) avec - nouveau code traitement,

---------------------------------------------------------

/\*

\*\* Remplacement de la ligne : ((NVL(ID\_BATCH\_STL, 0) = 0) OR (ID\_BATCH\_STL < 0))

\*\* par : NOT (NVL(ID\_BATCH\_STL, 0) > 0)

\*\* pour permettre le recours à l’index

\*\*/

UPDATE LBM\_MESSAGE\_ID

SET ID\_BATCH\_STL = LeCodeTraitementBizTalkNegatif

WHERE NOT (NVL(ID\_BATCH\_STL, 0) > 0)

-- ((NVL(ID\_BATCH\_STL, 0) = 0) OR (ID\_BATCH\_STL < 0))

AND TYPEFLUX = 'CLI'

;

#### Procédure STORELAND.PK\_LBM\_CRM.LBM\_GETCUSTOMERSFORRCU

#### Procédure STORELAND.PK\_LBM\_VENTES\_GU.LBM\_GETTICKETSBM

### Storland.exe

Une requête tout aussi consommatrice en temps et ressource se trouve dans STORLAND.EXE. Elle nécessite les mêmes modifications que celles décrites dans le paragraphe précédent, savoir,

1. Changer le type en varchar2 des éléments énumérés dans stecaisse IN ('1', '2')
2. Ne plus appliquer de fonction au champ datehtrans

SELECT COUNT(\*)

FROM lbm\_vente\_poste v

WHERE **stecaisse IN (1, 2)**

AND appl = 'SAP'

AND hca IS NULL

AND departement = '741'

AND (v.typegest != '5' AND v.typegest != '6')

AND topgu = 0

AND numcdeclt IS NULL

AND **TRUNC (datehtrans)** =

(SELECT TRUNC (datehtrans)

FROM (SELECT datehtrans

FROM lbm\_vente\_poste v2

LEFT JOIN lbm\_cp\_histo h

ON v2.cp = h.cp

WHERE **stecaisse IN (1, 2)**

AND v2.appl = 'SAP'

AND v2.hca IS NULL

AND departement = '741'

AND (v2.typegest != '5' AND v2.typegest != '6')

AND topgu = 0

AND ( (datehtrans < h.dateblocageinventaire)

OR (h.dateblocageinventaire IS NULL)

AND numcdeclt IS NULL

)

ORDER BY datehtrans

)

WHERE ROWNUM <= 1

)

AND ( (datehtrans < (SELECT dateblocageinventaire

FROM lbm\_cp\_histo h

WHERE h.cp = v.cp

)

)

OR ((SELECT dateblocageinventaire

FROM lbm\_cp\_histo h

WHERE h.cp = v.cp

) IS NULL

)

;

Une solution alternative consiste à créer un index comportant des fonctions (function-based index) :

1. Créer l’index

CREATE INDEX LBM\_VENTE\_POSTE\_FB\_STEC\_DATEH

ON STORELAND.LBM\_VENTE\_POSTE (To\_Number(stecaisse), Trunc (datehtrans)) ;

1. Calculer les statistiques de la table ou au moins du nouvel index

EXEC DBMS\_STATS.gather\_table\_stats

(

'STORELAND'

, 'LBM\_VENTE\_POSTE'

, cascade => TRUE

);

EXEC DBMS\_STATS.GATHER\_INDEX\_STATS

(

ownname => 'STORELAND'

, indname => 'LBM\_VENTE\_POSTE\_FB\_STEC\_DATEH'

, force => TRUE

);

# Annexes

## Extrait de V$DATABASE

select NAME

, LOG\_MODE

, CONTROLFILE\_TYPE

, CONTROLFILE\_CREATED

, CONTROLFILE\_TIME

, OPEN\_RESETLOGS

, VERSION\_TIME

, OPEN\_MODE

, PROTECTION\_MODE

, PROTECTION\_LEVEL

, REMOTE\_ARCHIVE

, DATABASE\_ROLE

, ARCHIVELOG\_COMPRESSION

, SWITCHOVER\_STATUS

, DATAGUARD\_BROKER

, GUARD\_STATUS

, FORCE\_LOGGING

, PLATFORM\_ID

, PLATFORM\_NAME

, FLASHBACK\_ON

, SUPPLEMENTAL\_LOG\_DATA\_FK

, SUPPLEMENTAL\_LOG\_DATA\_ALL

, DB\_UNIQUE\_NAME

FROM v$database

;

| NAME | LOG\_MODE | CONTROLFILE\_TYPE | CONTROLFILE\_CREATED | CONTROLFILE\_TIME | OPEN\_RESETLOGS | VERSION\_TIME | OPEN\_MODE | PROTECTION\_MODE | PROTECTION\_LEVEL | REMOTE\_ARCHIVE | DATABASE\_ROLE | ARCHIVELOG\_COMPRESSION | SWITCHOVER\_STATUS | DATAGUARD\_BROKER | GUARD\_STATUS | FORCE\_LOGGING | PLATFORM\_ID | PLATFORM\_NAME | FLASHBACK\_ON | SUPPLEMENTAL\_LOG\_DATA\_FK | SUPPLEMENTAL\_LOG\_DATA\_ALL | DB\_UNIQUE\_NAME |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| STOR | NOARCHIVELOG | CURRENT | 30-MAY-14 | 06-oct-17 | NOT ALLOWED | 30-MAY-14 | READ WRITE | MAXIMUM PERFORMANCE | UNPROTECTED | ENABLED | PRIMARY | DISABLED | SESSIONS ACTIVE | DISABLED | NONE | NO | 12 | Microsoft Windows x86 64-bit | NO | NO | NO | STOR |

## Script de compilation des objets en mode debug

Declare

cVc\_owner constant varchar2 (30) := 'STORELAND' ;

begin

for laChose in (select name, type

from all\_plsql\_object\_settings

where type in ( 'PACKAGE'

, 'PACKAGE BODY'

, 'PROCEDURE'

, 'FUNCTION'

, 'TRIGGER'

)

and owner = cVc\_owner

and plsql\_debug = 'TRUE'

order by type asc, name asc

) loop

begin

case laChose.type

when 'PACKAGE' then

execute immediate 'alter package ' || laChose.name || ' compile' ;

when 'PACKAGE BODY' then

execute immediate 'alter package ' || laChose.name || ' compile body' ;

when 'PROCEDURE' then

execute immediate 'alter procedure ' || laChose.name || ' compile' ;

when 'FUNCTION' then

execute immediate 'alter function ' || laChose.name || ' compile' ;

when 'TRIGGER' then

execute immediate 'alter trigger ' || laChose.name || ' compile' ;

end case;

exception

when others then

dbms\_output.put\_line('Erreur sur [' || laChose.name || ', ' || laChose.type || '] : ' || sqlerrm);

end;

end loop;

end;

/

## Script de compilation des objets invalides

Declare

cVc\_owner constant varchar2 (30) := 'STORELAND' ;

begin

for laChose in (select object\_name, object\_type

from all\_objects

where object\_type in ( 'PACKAGE'

, 'PACKAGE BODY'

, 'PROCEDURE'

, 'FUNCTION'

, 'VIEW'

, 'TRIGGER'

)

and owner = cVc\_owner

and status = 'INVALID'

order by object\_type asc

) loop

begin

case laChose.object\_type

when 'PACKAGE' then

execute immediate 'alter package ' || laChose.object\_name || ' compile' ;

when 'PACKAGE BODY' then

execute immediate 'alter package ' || laChose.object\_name || ' compile body' ;

when 'PROCEDURE' then

execute immediate 'alter procedure ' || laChose.object\_name || ' compile' ;

when 'FUNCTION' then

execute immediate 'alter function ' || laChose.object\_name || ' compile' ;

when 'VIEW' then

execute immediate 'alter view ' || laChose.object\_name || ' compile' ;

when 'TRIGGER' then

execute immediate 'alter trigger ' || laChose.object\_name || ' compile' ;

end case;

exception

when others then

dbms\_output.put\_line('Erreur sur [' || laChose.object\_name || ', ' || laChose.object\_type || '] : ' || sqlerrm);

end;

end loop;

end;

/

## Reste à faire

### Vérifications de la sort\_area\_size ou (si elle est utilisée de la pga\_aggregate\_target)

NB1.. Attention vérifier si *workarea\_size\_policy =auto*=> dans ce cas, ce paramétrage n’est plus utilisé.

NB2. Dans init.ora, **sort\_area\_size=1048576**

-- Sur l'utilisation de la mémpoire pour les tris :

-- objet : produire un graphique Nombre de tris en fonction des jours de la semaine

set pages 9999

column sorts\_memory format 999,999,999

column sorts\_disk format 999,999,999

column ratio format .99999

select to\_char (snap\_time, 'day')

, avg (newmem.value - oldmem.value) sorts\_memory)

, avg (newdsk.value - olddsk.value) sorts\_disk

from perfstat.stats$sysstat oldmem

, perfstat.stats$sysstat newmem

, perfstat.stats$sysstat olddsk

, perfstat.stats$sysstat sn

where newdsk.snap\_id = sn.snap\_id

and olddsk.snap\_id = sn.snap\_id -1

and mewmem.snap\_id = sn.snap\_id

and oldmem.snap\_id = sn.snap\_id -1

and oldmem.name = 'sorts (memory)'

and newmem.name = 'sorts (memory)'

and olddsk.name = 'sorts (disk)'

and newdsk.name = 'sorts (disk)'

and newmem.value - oldmem.value > 0

group by to\_char (snap\_time, 'day')

;

----

-- objet : produire un graphique Nombre de tris en fonction des tranches horaires de la journée

set pages 9999

column sorts\_memory format 999,999,999

column sorts\_disk format 999,999,999

column ratio format .99999

select to\_char (snap\_time, 'hh24')

, avg (newmem.value - oldmem.value) sorts\_memory)

, avg (newdsk.value - olddsk.value) sorts\_disk

from perfstat.stats$sysstat oldmem

, perfstat.stats$sysstat newmem

, perfstat.stats$sysstat olddsk

, perfstat.stats$sysstat sn

where newdsk.snap\_id = sn.snap\_id

and olddsk.snap\_id = sn.snap\_id -1

and mewmem.snap\_id = sn.snap\_id

and oldmem.snap\_id = sn.snap\_id -1

and oldmem.name = 'sorts (memory)'

and newmem.name = 'sorts (memory)'

and olddsk.name = 'sorts (disk)'

and newdsk.name = 'sorts (disk)'

and newmem.value - oldmem.value > 0

group by to\_char (snap\_time, 'hh24')

;

Mail de Christophe GONZALVEZ du 20171004

Il faudrait qu’en préambule tu décrives l’environnement OS et Rdbms en donnant le versioning.

Il faut également que tu donnes le délai de rétention des rapports statpack.

Il faudra aussi que

-          Le client te fournisse le plan de production afin que tu puisses faire le lien entre les activités constatées dans les statpacks et l’activité de production. Cela te permettra d’identifier des traitements gourmands.

-          Tu fasses un tour d’horizon du paramétrage des bases de données et de la configuration network

Lorsque tu auras fini tes différentes analyses, il faudra que pour chaque point tu donnes une action à réaliser.

Mail de Christophe GONZALVEZ du 20170927

De mémoire, l’audit est motivé par une envie du client d’avoir une vision extérieure sur leur base de données en terme d’implémentation logique ( répartition tables, index, lob) et physique ( localisation tablespace, nbr d’extents, etc..).

Une analyse de l’activité de la base de données devra aussi être fait à partir de rapport STATPACK. Normalement ils doivent l’avoir installé, c’est un point à vérifier avec eux s’ils l’ont installé ou si il faut que tu le fasses. Il faudra également voir si la configuration network est correct.

:

<http://dbspecialists.com/not-licensed-for-awr-use-statspack-instead/>

<https://perfstat.wordpress.com/2014/09/04/capturing-long-running-sql-in-statspack/>

<https://blog.dbi-services.com/awrrpt-and-spreport-in-multitenant/>

<https://www.akadia.com/services/ora_statspack_survival_guide.html>

<https://jonathanlewis.wordpress.com/category/oracle/statspack/>

<http://www.remote-dba.net/t_op_sql_high_use.htm>

Aussi :

<http://orafrance.developpez.com/dbahelp/>