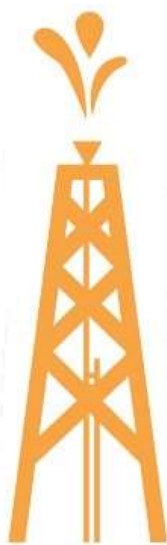




AUDIT DE LA BASE ORACLE MARCELLO

Le Bon Marché

Rapport d'audit



Version : V 0.11

Date de la version : 03/11/2017

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 0.11	JEA	16/10/2017				

Date d'application : 16/10/2017

TABLE DES MATIERES

1 OBJET DE L’AUDIT..... 4

2 ENVIRONNEMENT 5

3 LES POINTS FORTS 6

4 LES POINTS FAIBLES..... 7

5 LES OPPORTUNITES D’AMELIORATION 8

6 ANNEXES..... 12

Draft

1 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

Draft

2 ENVIRONNEMENT

2.1 OS

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

2.2 RDBMS

MARCELLO est une base en version 10.2.0.5.0. Elle est en rôle PRIMARY_INSTANCE et fonctionne en NO ARCHIVE LOG.

2.3 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 perçus dans cet intervalle.

3 LES POINTS FORTS

3.1 BASE EFFICIENTE ET STABLE

L'essentiel du modèle de données 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.

Draft

4 LES POINTS FAIBLES

4.1 VERSION D'ORACLE NON SUPPORTEE 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.

4.2 QUELQUES OBJETS COMPILES 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.

4.3 PRESENCE 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

5 LES OPPORTUNITES D'AMELIORATION

5.1 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.

5.2 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.

5.3 COMPILER EN MODE STANDARD LES OBJETS COMPILES EN MODE DEBUG.

Attention, il peut y avoir des instabilités si un grand nombre d'objets ayant de fortes interdépendances sont ainsi refactorisés. 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.

5.4 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.

5.5 AJOUTER DES INDEX

Agrandir

5.6 CORRIGER DES REQUETES CONSOMMATRICES DE RESSOURCES

5.6.1 Package STORELAND.PK_LBM_VENTES_GU

5.6.1.1 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' )
                  )
              )
     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.numdeclet 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
    )
);

```

5.6.1.2 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

5.6.1.3 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'
;

```

5.6.1.4 Procédure STORELAND.PK_LBM_CRM.LBM_GETCUSTOMERSFORRCU

5.6.1.5 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 numdeclet 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 numdeclet 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
    )
  )
;

```

6 ANNEXES

6.1 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
;
```

DB_UNIQUE_NAME	STOR
LOG_MODE	NO
CONTROLFILE_TYPE	NO
CONTROLFILE_CREATED	NO
CONTROLFILE_TIME	NO
OPEN_RESETLOGS	NO
VERSION_TIME	NO
OPEN_MODE	NO
PROTECTION_MODE	NO
PROTECTION_LEVEL	NO
REMOTE_ARCHIVE	NO
DATABASE_ROLE	NO
ARCHIVELOG_COMPRESSION	NO
SWITCHOVER_STATUS	NO
DATAGUARD_BROKER	NO
GUARD_STATUS	NO
FORCE_LOGGING	NO
PLATFORM_ID	NO
PLATFORM_NAME	NO
FLASHBACK_ON	NO
SUPPLEMENTAL_LOG_DATA_FK	NO
SUPPLEMENTAL_LOG_DATA_ALL	NO
DB_UNIQUE_NAME	NO

6.2 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;
/

```

6.3 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;
/
```