

Etude Optim for Z/OS

Extrait d'un rapport d'intervention



Siège commercial :
110 Boulevard de Sébastopol
75003 Paris
Tél. : +33 1 44 54 67 16
Fax. : +33 1 44 54 02 30

Date : 31 juillet 2009
Version finale : v1.0

Table des matières	
Executive summary	3
Préambule	
1 Analyse de l'existant	
1.1 Les environnements de qualification des applications	
1.2 Sécurité des données DB2 et Optim for z/OS	
1.3 Optim for z/OS	
2 Analyse des scenarii d'extraction	
2.1 Gestion des jeux d'essai	
2.2 Description du noyau des extractions actuelles	
3 Préconisations	
3.1 Les axes d'investigation	
3.2 Préconisations et évolutions fonctionnelles	
3.2.1 Simplification et optimisation des processus	
3.2.2 Généralisation et industrialisation	
3.2.3 Préconisation de sécurité et de confidentialité	
3.2.4 Préconisation évolutives et fonctionnelles	
4 Conclusions	
5 Annexes	
5.1 Proposition de Monitorat	
5.2 DATA SLICER	5

Executive summary

Suite à notre intervention dans le cadre de cette mission d'optimisation Optim, voici dans ce paragraphe la liste des préconisations classifiées en 4 domaines qui sont détaillées en « §3. Préconisations » :

- 1 . Simplification et optimisation des processus
 - Optimisation du point d'entrée ou « noyau » des extractions
 - Préconisations pour le cycle Contrats/Clients/Contrats
 - Suppression des relations parasites
 - Tri des identifiants pour extraction
 - Réduction du nombre d'extraction pour les Tables Pleines
 - Standardisation du vidage des Tables Vides
- 2 . Généralisation et industrialisation
 - Utilisations des définitions génériques
 - Maintenance du transcodage des Tables techniques (pleines)
 - Optimisation du traitement de purge pour remplacement
 - Génération dynamique et automatique des utilitaires LOAD
- 3 . Préconisation de sécurité et de confidentialité
 - Deux méthodes de protection des définitions Optim
 - Maquillage en Production des données confidentielles Personnes
 - Maquillage en Recettes des données techniques
- 4 . Préconisations évolutives et fonctionnelles
 - Solution pour le mode de rafraichissement annuel dit « silencieux »
 - Création d'un environnement de Référence pour les tests de non-régression

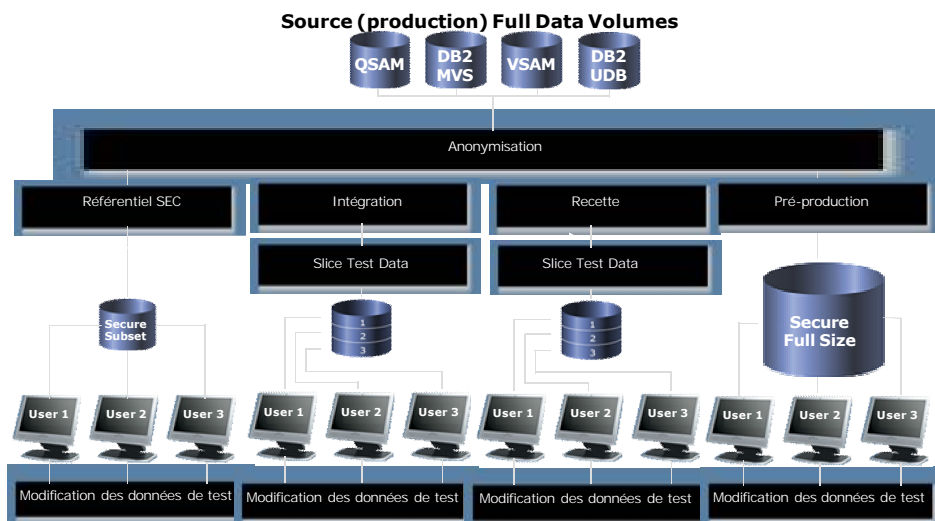
Solution pour le mode de rafraîchissement annuel dit « silencieux »

Le mode de rafraîchissement annuel entraîne une neutralisation des environnements 3 à 5 jours. Les préconisations de performance et d'automatisation réduiront sensiblement les phases de chargement des données. Les déchargements sont quant à eux contraints de travailler de façon séquentielle à partir des sauvegardes de production. Le mode silencieux induit une notion d'impact zéro pour les chargements dans votre cas puisque les extractions sont réalisées à partir des sauvegardes. Une solution envisageable est la création d'un **référentiel** qui est à la fois un jeu de données cohérentes, et un échantillon de production. Cet environnement sera placé en amont des environnements d'intégration et de recette, voire en amont des environnements de développement. Pour la suite de la description cet environnement prendra l'appellation « Référentiel SEC » (SEC : Sous Ensemble Cohérent).

Une fois réalisée l'opération de constitution de l'environnement, les utilisateurs pourraient utiliser le principe des processus actuels équivalents aux traitements quotidiens et hebdomadaires pour alimenter les environnements d'intégration et de recette. Le rafraîchissement annuel de Production vers le Référentiel SEC n'imposera pas de contraintes de neutralisation de l'intégration et de la recette. Les utilisateurs des environnements d'intégration et de recette pourront planifier plus facilement leur rafraîchissement annuel qui sera une extraction du Référentiel SEC vers l'environnement cible choisi. De plus L'environnement Référentiel SEC sera plus proche de la version du modèle de donnée de développement, d'intégration et de recette. La prise en compte du delta de définition des données pour la constitution des jeux d'essai sera effectuée lors de l'extraction/chargement entre la production et le référentiel SEC. Les traitements quotidiens et hebdomadaires pourraient continuer à passer de la production vers l'intégration et la recette en plus du Référentiel SEC.

Le point suivant à adresser par ces principes est le maintien constant de la cohérence sur les environnements où sont réalisés les tests tout en limitant le nombre d'environnement physique de bases DB2. Pour atteindre cet objectif nous vous proposons de mettre en œuvre ce que nous appelons des SLICES. Ce sont des données cloisonnées au sein d'une même table DB2 et partagées pour plusieurs programmes en même temps. Ces techniques peuvent être mises en œuvre et gérées à l'aide de la solution logicielle DATA SLICER.

Vous trouverez ci-après une représentation d'un ensemble d'environnements qui sont gérés avec des SLICES :



La partie haute du schéma est géré avec Optim jusqu'aux chargements dans les environnements Référentiel SEC et Pré-production. Pour alimenter les environnements Slicés d'Intégration et de Recette, la source pourra être l'environnement de Production et/ou l'environnement Référentiel SEC pour chacun des Slices.

Vous trouverez en annexe une description de la solution DATA SLICER.

DATA SLICER

Avec quelques conventions simples, la solution permet aux développeurs/contrôle qualité de partager leurs données de test dans un ensemble unique de tables dans leur environnement de test pour éviter les collisions pendant les phases de test. Ces techniques peuvent réduire le besoin d'avoir des centaines de copies des mêmes tables et permettre des tests simultanés avec des données partagées.

Data Slicer permet de simplifier :

- les recettes applicatives
- les tests de non régression
- les tests d'intégration
- la présentation pour démonstration des résultats (rapports)

Slice s'appuie les autorisations DB2 standard, chaque utilisateur peut utiliser ses propres slice de données ou ceux d'un autre avec les autorisations nécessaires. Les programmes ne sont pas modifiés, les slice sont accédées de façon transparente en fonction du SQL-ID et des autorisations DB2. Data Slicer fonctionne en TP et en Batch.

Le diagramme suivant illustre le concept de Slice :

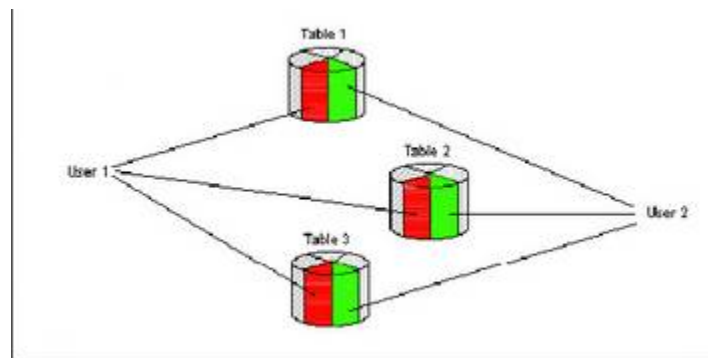


Figure 1 : Chaque utilisateur ou groupe d'utilisateurs possède son propre jeu de données « Slice » au sein de la même base de données.

Ce diagramme montre les Slices (rouge et vert) utilisés par chacun des utilisateurs (user-1 et user-2). Ils sont indépendants l'un de l'autre. Les utilisateurs peuvent créer plusieurs versions de leur Slice, ayant ainsi la capacité de tester et comparer les images avant/après lors de comparaisons de données de test pour vérifier exactement ce qui a changé. Cette fonctionnalité est indispensable lors des tests de non régression.

Les typologies de phases de test suivantes précisent les cas d'utilisations possibles :

Tests Unitaires : les utilisateurs peuvent partager ou isoler des données de test unitaires dans un ensemble unique de table DB2.

Ceci permet à chaque développeur/testeur de gérer un ou plusieurs slices avec leurs propres données de test. Chaque Slice est taillé pour répéter des tests pour un ou plusieurs programmes donnés. Les données peuvent aussi être copiées et partagées par les programmeurs. Pour vérifier le bon fonctionnement des programmes, cette fonction élimine ainsi des tâches ennuyeuses et répétitives de création et de recréation des cas de test. Une copie des Slices peut être générée et gelée pour servir de référence, dans ce cas elle pourra être partagée mais non modifiable. Les tables chargées à partir d'un Slice sont elles modifiables.

Tests d'intégration : Les tests d'intégration ou plans de tests sont organisés et cadencés en phases ou en journées de test.

Chaque phase ou journée se déroule en séquence, la deuxième après la première, etc.... Les tests d'intégration sont pour la plus part linéaires. Les tests de la première journée doivent être terminés pour commencer ceux de la journée suivante. Avec la solution **Slice**, les tests peuvent

être réalisés de manière concurrente ou simultanée. Non seulement le premier jour de test peut être fait en même temps que le »deuxième », mais ils peuvent être faits sur le même jeu de tables DB2 par un ou plusieurs utilisateurs.

Tests de non Régression : Slice/Compare simplifie la maintenance et la vérification des étapes nécessaires aux tests de non régression. Un test de non régression vérifie que les changements faits dans un ensemble de programmes impactent **Uniquement** les données qui doivent être changées par les modifications apportées par les programmes. Pour vérifier que les modifications produisent les bons effets, les modifications de données sont comparées manuellement aux résultats escomptés.