Exploration de problèmes de performance d’un entrepôt de données

par

Louisa Demmou

essai présenté au Centre de formation en technologie de l’information (CeFTI)
en vue de l'obtention du grade de Maître en Génie Logiciel MGL

FACULTÉ DES SCIENCES
UNIVERSITÉ DE SHERBROOKE

Sherbrooke, Québec, Canada, avril 2010

Sommaire

Les entrepôts de données ou *DW* (*Data Warehouse*) servent aux systèmes informationnels d'une entreprise. Ils permettent de sauvegarder de façon permanente et sans équivoque tout l'historique de l'entreprise. Cet historique en terme de données est mis sous certaines formes dans une base de données (BD) dite entrepôt, afin de répondre aux besoins d'affaire de la compagnie. Ces systèmes sont qualifiés par l'exploitation des données d'une entreprise afin d'aider à la prise de décision. Aussi importants soient-ils, la majorité des entrepôts de données souffrent pratiquement tous de problèmes de performances, surtout quand il s'agit de gros entrepôts de données. Les problèmes de performance ne se retrouvent pas seulement du côté de l'extraction ou de la restitution. En fait, ils peuvent être présents du côté « *front-end* » (ETC - Extraction Transformation Chargement) que « *back-end* » (restitution). Ils peuvent être plus prononcés d'un côté ou de l'autre ou même parfois des deux.

Le but visé à travers cet essai est de cerner et d'explorer les problèmes de performance d'un entrepôt de données en général et au niveau de l'ETC en particulier. Cette investigation amène à poser un diagnostic de la lenteur et la lourdeur éventuelles de l'exploitation d'un entrepôt de données. Ceci permet d'analyser les problèmes, d'identifier leurs origines et causes diverses et de proposer une solution. À l'aide de certaines métriques, des mesures de performance d'un DW sont élaborées et permettent de fournir une matrice problème/solution, à titre de référence. L'étude inclut l'identification sommaire de certains problèmes de performance au niveau de l'exploitation (restitution).

Le travail présenté dans le présent document n'a pas la prétention de soulever tous les problèmes de performances rencontrés dans un entrepôt de données. Il ne sera en aucun cas question d'aborder en détail les performances d'un entrepôt à tous les niveaux de l'ETC et de la restitution. Il ne sera donné qu'un survol des problèmes de performances rencontrés du côté exploitation. Il n'est nullement prévu non plus d'identifier de façon exhaustive tous les

problèmes pouvant être rencontrés dans les performances d'un DW au niveau de l'ETC.

La méthode proposée est destinée autant à un débutant qu'à un connaisseur en entrepôt de données. Pour le premier, il apprendra que les problèmes de performances sont très fréquents et qu'il n'est pas le seul à être confronté à des situations où ces problèmes prennent beaucoup d'ampleur. La méthode l'orientera vers des étapes qui pourront être exécutées ou validées à son niveau. Pour quelqu'un qui a plus de connaissances dans les DW, des techniques récentes pour l'amélioration des performances d'un entrepôt de données et plus spécifiquement d'un ETC lui sont proposées.

Il n'en demeure pas moins que les problèmes de performance d'un DW sont d'actualité et sont loin d'être tous résolus, qu'ils soient présents au niveau de l'ETC ou de la restitution. Les entreprises possédant un entrepôt de données ne peuvent plus s'en départir même s'il accuse des performances médiocres. Somme toute, il reste beaucoup de travail à faire dans ce domaine.

Remerciements

En tout premier lieu, je remercie grandement Monsieur Domingo Palao Muñoz, chargé de cours de bases de données à l'Université de Sherbrooke au Campus principal, d'avoir accepté d'être le directeur professionnel pour la réalisation de cet essai. Je remercie Domingo pour la confiance qu'il m'a accordée et surtout pour ses encouragements qui m'ont accompagnée durant tout le parcours du travail. Je n'oublierai pas ses recommandations qui m'ont beaucoup aidée pour finaliser ce document.

Je n'oublierai Monsieur Goulet, vice-doyen à l'enseignement, Directeur du centre de formation en technologies de l'information (CeFTI) et professeur titulaire à l'Université de Sherbrooke, pour avoir accepté d'être le directeur académique de mon essai. Je remercie infiniment Monsieur Goulet, pour tous les points pertinents qu'il m'a apportés.

Je remercie spécialement Monsieur Patrice Roy, chargé de cours à l'Université de Sherbrooke, sur les trois Campus, pour ses encouragements et ses conseils, dans un moment difficile de ma vie où son intervention a joué un rôle déterminant pour terminer mon essai.

Je tiens à remercier chaleureusement Monsieur Claude Cardinal, directeur adjoint au CeFTI, pour le rôle qu'il joué dans tout le processus du cours rattaché à l'essai. Sans sa coordination bénéfique, plusieurs tâches n'auraient pas abouti.

Table des matières

Introduction 1

Chapitre 1 Contexte des problèmes de performance d'un DW 5

1.1 Système décisionnel 5

1.2 Entrepôt de données 6

1.2.1 DW versus Data mart 7

1.3 Composantes d'un entrepôt de données 8

1.4 Principes d'un entrepôt de données 9

1.4.1 En amont et en aval (« *front-end* » et « *back-end* ») 9

1.4.2 Niveau des données 10

1.4.3 Dimensions et faits 10

1.4.4 Architecture d'un entrepôt 11

1.4.5 Outils utilisés au niveau d'un DW 11

1.4.6 Structure de données 12

1.4.7 Vocation d'un entrepôt de données 12

1.4.8 Résumé 12

Chapitre 2 Exploration de problèmes de performance d'un DW 13

2.1 Mise en contexte. 14

2.2 Acteurs au niveau d'un entrepôt de données 14

2.3 Problèmes de performance : en amont et en aval du DW 17

2.4 Performance « *back-end* » du DW 19

2.4.1 Les défis d'un ETC 23

2.4.2 Métriques d'évaluation de performance d'un ETC 24

2.4.3 Identification des problèmes de performance « *back-end* » d'un DW 25

2.5 Performance « *front-end* » du DW 30

2.6 Résumé 33

Chapitre 3 Solutions aux problèmes de performance d'un DW 35
3.1 Facteurs clé de performance d'un DW 36
3.2 Caractéristiques de la BD pour l'amélioration de la performance 37
3.3 Description de la configuration de l'environnement des tests.....37
3.4 Création de tables de fait avec l'option NOLOGGING - Solution S1 38
3.5 Amélioration des requêtes SQL – Solution S2.....39
3.6 Utilisation du plan de la requête – Solution S3 39
3.7 Performance de la requête ou script SQL – Solution S4 40
3.7.1 Commande HINT APPEND dans la commande INSERT.....40
3.7.2 Commande HASH JOIN 40
3.7.3 Commande NOT EXISTS au lieu du NOT IN.....41
3.7.4 Boucles imbriquées - NESTED LOOP 41
3.7.5 Favorisation de l'usage BULK COLLECT dans un curseur.....41
3.8 Gestion des index - Solution S5 41
3.9 Gestion des partitions des tables – Solution S6.....42
3.10 Utilisation appropriée des tables partitionnées - Solution S6.....43
3.11 Utilisation du parallélisme - Solution S7.....44
3.11.1 Description de l'exécution en parallèle 45
3.11.2 Résultat de tests avec le parallélisme 46
3.12 Utilisation de la cache des requêtes - Solution S8.....48
3.13 Utilisation des vues matérialisées - Solution S9.....50
3.14 Compression des données - Solution S10 50
3.15 Utilisation de BD chargement en colonnes - Solution S11 51
3.16 Technologie DW monofonctionnel - Solution S12 52
3.17 Technologie de l'accélérateur de requêtes - Solution S13 53
3.18 Technologie BD spécialisées – Solution S14.....54
3.19 Utilisation d'outil de surveillance de l'ETC - Solution S15.....54
3.20 Utilisation d'un outil de monitoring de la BD - Solution S16.....55
3.21 Mécanisme de reprise d'une chaîne ETC - Solution S17.....55

3.22 Mécanisme de validation des données - Solution S18 57
3.23 Analyse de performances de l'outil BI - Solution S19 57
3.24 Matrice problèmes –solutions 58
3.25 Résumé 58

Conclusion 60

Liste de références 62

Annexe A Scripts des requêtes de test de parallélisme 64

Annexe B Exemple d'un plan de requête 70

Annexe C Modèle de données des tests 71

Annexe D Configuration de la BD et de l'environnement de test 73

Annexe E Volumétrie des données de test 77

Annexe F Fonctionnement du parallélisme 78
F.1 Opérations pouvant être parallélisées 78
F.2 Paramètre d'initialisation du parallélisme 78
F.3 Exemples d'utilisation du parallélisme 79
F.4 Quand faut-il implémenter l'exécution parallèle? 80

Annexe G Points généraux d'amélioration de performance 81

Annexe H Matrice problèmes - solutions 83
H.1 Matrice problèmes « *back-end* » – solutions 84
H.2 Matrice problèmes « *front-end* » – solutions 86

Annexe I Outils de monitoring de la base de données 87

Liste des tableaux

Tableau 2-1: Identification de problèmes de performance « *back-end* » du DW 26
Tableau 2-2: Identification de problèmes de performance « *front-end* » du DW.....32
Tableau 3-1: Données et résultats de test de parallélisme 47
Tableau 3-2: Données pour exemple de BD orientée colonnes.....51
Tableau 3-3: Scripts des requêtes du test de parallélisme 64
Tableau 3-4: Données de configuration de la BD de test 73
Tableau 3-5: Données de configuration l'environnement de test.....76
Tableau 3-6: Matrice problèmes de performance « *back-end* » - solutions.....84
Tableau 3-7: Matrice problèmes de performance « *front-end* » – solutions 86

Liste des figures

Figure 1-1: Schématisation d'un magasin de données 7
Figure 1-2: Composantes de base sommaires d'un entrepôt de données 8
Figure 1-3: Composantes de base détaillées d'un entrepôt de données 9
Figure 2-1: Identification de problèmes de performance en amont et en aval d'un DW . 19
Figure 2-2: Différentes sources de données prises en charge par l'ETC 20
Figure 2-3: Quadrant magique Gartner- outil ETC juin-2009 22
Figure 2-4: Complexité des processus ETC 24
Figure 2-5: Quadrant magique Gartner - Outils BI - janvier-2009 31
Figure 3-1: Résultat de test option NOLOGING 39
Figure 3-2: Traitement parallèle versus non parallèle et traitement intra-requête 45
Figure 3-3: Graphique des résultats de tests de parallélisme 48
Figure 3-4: Tests Cache de la requête 49
Figure 3-5: Exemple accès aux données avec Columnar DB 52
Figure 3-6: Modèle de données de test - Inventaire 70
Figure 3-7: Modèle de données de test -Ventes 71
Figure 3-8: Modèle de données de test - Inventaire 72

Glossaire

Ad hoc (requêtes) Requêtes formulées par l'utilisateur sous l'inspiration du moment. Les requêtes ad hoc sont, par nature, difficilement prévisibles, puisque les comportements des utilisateurs sont inconnus au stade de la conception.

Agrégats Lignes physiques d'une base données, presque toujours créées par totalisation d'autres enregistrements de la base de données, dans le but d'améliorer la performance des requêtes. On les appelle parfois données pré-totalisées.

Base de données multidimensionnelle

Base de données dans laquelle les données sont présentées sous forme de cubes, par opposition aux tables d'une base de données relationnelle.

Bottom-up Technique ou approche dite ascendante (de bas en haut) se caractérise par une suite de processus qui apportent chacun une partie fondamentale essentielle à l'aboutissement du processus final.

Business Intelligence

Terme générique désignant les informations internes ou externes d'une organisation et qui lui servent à prendre de meilleures décisions.

Dashboard Tableau de bord. Peut être considéré comme un outil offrant une visualisation simplifiée des situations décrites et des constats effectués par les indicateurs. Il permet à son utilisateur de se construire une représentation synthétique de la réalité. C'est un rapport avec plusieurs graphiques dont la jauge, l'histogramme, les indicateurs.

Data mart Magasin de données qui est un sous ensemble d'une base de données relationnelle, utilisé en informatique décisionnelle. Il est généralement exploité en entreprise pour restituer des informations ciblées sur un métier spécifique, constituant pour ce dernier un ensemble d'indicateurs à

vocation de pilotage de l'activité et d'aide à la décision.

Datapumping — Ensemble des tâches consistant à détecter, à sélectionner, à extraire et à filtrer les données brutes issues de sources hétérogènes

Front-end / back-end — Termes génériques qui référent à un début et à une fin d'un processus. Le front-end est responsable de collecter des inputs d'un usager, sous diverses formes et de les traiter afin de les rendre conformes aux spécifications exigées par le back-end. Le front-end est une sorte d'interface entre l'utilisateur et le back-end.

Full extract — Extraction complète de dimensions ou de faits à partir des systèmes sources, contrairement à une extraction itérative (*incremental extract*), (dite aussi delta) qui ne ramène que les données nouvelles.

Granularité — Niveau de détail saisi dans l'entrepôt de données.

Hachage (hashing) — Opération qui consiste à appliquer une fonction mathématique permettant de créer l'empreinte numérique d'un message, en transformant un message de taille variable en un code de taille fixe, en vue de son authentification ou de son stockage.

Incremental extract — Extraction itérative permettant de ramener les données qui ont changé depuis la dernière itération.

Index — Une structure de données associée à une table ordonnée logiquement selon les valeurs d'une clé et servant à améliorer les performances d'une base de données et la vitesse de réponse aux requêtes. Les index en arbre binaire sont utilisés pour les champs de haute cardinalité et les index bitmap servent pour les champs de cardinalité moyenne ou basse.

Index bitmap — Technique d'indexation relationnelle convenant très bien à des colonnes ayant un faible nombre de valeurs possibles (faible cardinalité).

Index en arbre binaire

Index relationnel particulièrement utile pour les colonnes de haute cardinalité. Cet index construit une arborescence de valeurs avec une liste

	des codes de ligne ayant la valeur de la feuille.
Late arriving fact	Des faits (mesures) qui arrivent tard. Il s'agit d'un processus typique aux entrepôts de données.
Type LOB	*Large Object*. C'est un type de données qui permet de stocker dans un champ un gros volume d'informations. Il existe CLOB (*Character LOB*), BLOB (*binary LOB*), NCLOB (*National CLOB*). La taille maximum d'un LOB est de 4G.
Lookup table	Table de correspondance, qui permet d'associer des valeurs, se comportant un peu comme une table de vérité et de déterminer sa sortie en fonction de ses entrées et du contenu de la table. C'est une structure de données, employée pour remplacer un calcul par une opération plus simple de consultation. Le gain de vitesse peut être significatif, car rechercher une valeur en mémoire est souvent plus rapide qu'effectuer un calcul important.
Métadonnées	Toutes les informations tenues à jour pour aider au fonctionnement ou à l'utilisation d'un entrepôt de données, représentant une véritable encyclopédie de l'entrepôt de données.
Partitionnement	Séparation logique ou physique des machines virtuelles en diverses entités indépendantes.
Pipelining	Élément d'un circuit électronique dans lequel les données avancent les unes derrière les autres, au rythme du signal d'horloge.
Power user	Usagers d'applications qui ont le privilège de faire des tests d'acceptation et d'être impliqués dans la définition des besoins.
Requêtage	L'outil de requêtage permet d'exécuter des commandes SQL arbitraires.
Scorecarding	Indicateurs de performance.
Slowly changing dimension	Dimension des données qui changent peu souvent. De ce fait, leur changement arrive une fois de temps en temps, utile pour garder des traces sur le changement.

Star schema	Schéma en étoile est un modèle de données d'un entrepôt de données. La table située au centre de l'étoile est la table de faits ou mesures (ou encore métriques). Les tables situées à l'extrémité de l'étoile sont des tables de dimensions (ou encore axes d'analyse).
Surrogate key	Clé de substitution, qui est une clé non intelligente utilisée afin de substituer la clé naturelle (Business Key) qui provient des systèmes opérationnels.
Tables partitionnées	Tables (et les index associés) qui sont gérées comme des tables physiques distinctes mais apparaissent du point de vue logique comme une seule et même table. Les grandes tables de faits sont candidates au partitionnement, souvent par date. Cette approche peut faire gagner du temps au niveau des requêtes et des mises à jour.
TDWI	*The Data Warehouse Institute*. Une excellente ressource d'information en ligne sur tout ce qui a trait à l'entrepôt de données, BI. www.tdwi.org
Téraoctets (To)	Un millier de milliards d'octets (2^{40}).
Vue matérialisée	Dans les systèmes de gestions de base de données, de type relationnel, une vue est une table virtuelle représentant le résultat d'une requête sur la base de données. Comme son nom l'indique et à la différence d'une vue standard, dans une vue matérialisée les données sont dupliquées. On l'utilise essentiellement à des fins d'optimisation et de performance dans le cas où la requête associée est particulièrement complexe ou lourde, ou pour faire des réplications de table.
Webinar	Conférence en ligne, une application Internet permettant d'organiser des conférences, réunions de travail ou des formations virtuelles avec des personnes distantes.

Liste des abréviations

BA	*Business Analyst* : analyste d'affaire
BD	Base de données
BI	*Business Intelligence* : intelligence d'affaire
CAST	*Create Table As Select* : abréviation de la commande SQL pour la création de table
CRM	*Customer Relation Management Tool* : outil de gestion de la relation client
CSV	*Comma Separated Values* : fichier de données séparées par des virgules
CUP	Code universel des produits, est le premier système d'identification numérique pour les articles vendus en magasin
DBA	*Database Administrator* : administrateur de base de données
DOLAP	*Desktop On-Line Analytical Processing*
DOP	*Degree Of Parallelism* : degré du parallélisme
DW	*Data Warehouse* : entrepôt de données
ERP	*Enterprise Resource Planning* : progiciel de gestion intégré
Etc.	Et caetera locution signifiant : et le reste
ETC	Extraction Transformation Chargement
ETL	*Extraction Transformation Load* : extraction transformation chargement
Exp	Exemple
FTP	*File Transfer protocol* : protocole de transfert de fichiers
Go	Unité de mesure d'espace de stockage de données. Un Giga octets est équivalent à 1 073 014 813 octets.
HOLAP	*Hybrid On-Line Analytical Processing*
MOLAP	*Multidimensional On-Line Analytical Processing*
OLAP	*On-Line Analytical Processing* : traitement d'analyse en ligne. C'est un ensemble de principes définis en termes généraux pour fournir un cadre

	dimensionnel à l'aide de la décision
OLAP	
Multidimensionnel	Implémentation OLAP non basée sur des bases de données relationnelles. Ces systèmes ne permettent pas de faire face aux mêmes volumes que les bases de données relationnelles, mais ils offrent le plus souvent de meilleures performances et des outils plus étroitement intégrés que les systèmes relationnels
OLTP	*On-Line Transactional Processing*, est un processus de transaction en ligne qui permet de qualifier les bases de données pouvant réaliser rapidement des transactions SQL (parfois en temps réel) à destination d'autres applications telles que des outils de suivi de production ou de contrôle de stock. Ce terme provient du monde des *mainframes* (ordinateur central)
RAID	*Redundant Array of Independant/Inexpensive Disk* : réseau redondant de disques. Technique qui consiste à stocker les mêmes données à différents endroits sur plusieurs disques durs, de manière à augmenter leur accessibilité et leur sécurité
RAM	*Random Access Memory* : mémoire primaire d'un ordinateur où les données peuvent être lues, écrites ou effacées
ROLAP	*Relational On-Line Analytical Processing*
SCD	*Slowly Changing Dimension* : changement à évolution lente
SGBD	Système de gestion de base de données
SQL	*Structured Query Language* : langage structuré d'interrogation
XML	*Extensible Mark-up Language* : langage de balise extensible

Introduction

Un entrepôt de données ou *DW* (*Data Warehouse*), est considéré beaucoup plus comme un environnement que comme une application ou une base de données. Plusieurs composantes définissent un DW : les systèmes sources, les différentes connexions, le réseau, les traitements d'extraction - chargement, le calcul des agrégations, l'interrogation des données. Les performances d'un DW consistent en l'orchestration de toutes ces composantes en un temps record. En effet, qui dit DW dit manipulation de données colossales. L'attente des usagers face à l'interrogation de toutes ces données est avant tout une réponse fiable mais aussi très rapide ou du moins dans un délai acceptable.

À l'heure actuelle, la majorité des gros entrepôts de données souffrent de problèmes de performance. Ils dérangent les usagers, les administrateurs et les développeurs du DW. Il n'y a pas de recettes magiques qui résolvent en un seul coup tous ces problèmes. Il faut quasiment les étudier un par un. Par contre, ils sont pratiquement communs à des entrepôts de données différents. Ils ne s'appliquent pas à un domaine bien spécifique. Qu'il s'agisse d'un entrepôt de données de ventes au détail ou de celui du domaine de distribution de l'électricité, les problèmes de performances sont du même type.

L'étude entreprise dans cet essai est le fruit d'une expérience vécue avec trois DW différents appartenant à des domaines différents. Le premier exploite les données de la vente au détail des produits de rénovation. Le deuxième utilise les données du tourisme : vente de voyages, de séjours et de billets d'avion. Pour le dernier, il manipule des données de distribution d'électricité. Les problèmes de performances rencontrés sont du même type :

- Un ETC (Extraction Transformation Chargement) qui accuse un retard dépassant les limites permises;
- Des exécutions de rapports qui sont loin d'être instantanées;

- Des exécutions de tableaux de bord lentes, lourdes et fastidieuses;
- Les données récentes qui sont parfois non disponibles aux heures habituelles;
- Le chevauchement d'ETC du jour avec celui du lendemain sans signe d'alarme.

La méthode préconisée dans le présent document vise à donner une structure, des étapes ordonnées et des solutions à certains problèmes rencontrés au niveau de l'ETC. L'approche adoptée est une démarche incrémentale et évolutive (*bottom-up*). La méthode élaborée pourra servir à plusieurs intervenants d'un DW. Elle a le mérite d'encadrer et d'offrir des solutions techniques à certains problèmes identifiés.

Le premier chapitre du rapport est consacré aux définitions, concepts, principes et règles liés aux entrepôts de données. Ces éléments permettent de situer l'environnement dans lequel l'étude a été faite. Le but n'est pas de définir un entrepôt de données et de décrire son fonctionnement. Ce sont des éléments essentiels qui jouent un rôle dans la description de la problématique de performance d'un entrepôt de données. Ils sont aussi cités dans les solutions données dans le chapitre consacré à cette fin.

Le deuxième chapitre est dédié à l'identification des problèmes de performance rencontrés dans un DW. Il recense ces problèmes avec détail au niveau de l'ETC. Il décrit les situations dans lesquelles il est éventuellement possible de les rencontrer avec une cause probable et une incidence éventuelle au niveau de la restitution.

Le chapitre trois présente les solutions envisagées pour résoudre les problèmes identifiés dans le chapitre précédent.

La dernière partie du document donne un constat de ce qui a été énoncé dans les chapitres précédents, à savoir la matrice problèmes / solutions. Elle produit des points non couverts par l'étude et ce qui pourrait être envisagé dans le futur quant aux problèmes de performance d'un DW en général.

Les sources utilisées comme support au présent document proviennent principalement de nombreux articles scientifiques consultés dans les différentes banques de données de l'Université ainsi que certains articles du TDWI (*The Data Warehouse Institute* *http://tdwi.org/*). Certains schémas, figures et définitions sont issues de livres, principalement ceux de Ralph Kimball [8] [9].

Chapitre 1

Contexte des problèmes de performance d'un DW

Le travail du présent essai repose sur les entrepôts de données en général et plus précisément sur l'ETC. Les entrepôts de données représentent le cœur des systèmes décisionnels de l'entreprise. Ils sauvegardent toutes les données historiques de l'entreprise. Celles-ci sont disponibles sept jours sur sept, de façon évolutive, de telle sorte à répondre aux besoins de l'entreprise à tout moment. L'entrepôt de données constitue la connaissance de l'entreprise. Elle est prépondérante au rendement de la compagnie. La performance de l'entrepôt de données constitue le fond du présent sujet. Pour se situer dans le contexte du projet, il est essentiel de présenter certaines définitions, quelques concepts et principes ainsi que certaines règles régissant un DW. C'est le contenu de ce premier chapitre. Il permet au lecteur ou à la lectrice de cerner des éléments fondamentaux et importants liés au sujet traité.

1.1 Système décisionnel

Le système décisionnel (*Decision Support System* ou *Business Intelligence BI)* exploite des données d'une entreprise dans le but de faciliter la prise de décision. Son but ultime est d'offrir une aide à la décision et de permettre au responsable de la stratégie d'entreprise d'avoir une vue d'ensemble de l'activité traitée. Il va de soi qu'à travers l'utilisation de ce genre de système, on vise l'amélioration, l'évolution, l'efficacité, l'avancement, les performances, la productivité de l'entreprise.

Cette technologie est apparue dans les années 1990. Les entreprises sont confrontées à l'exploitation de données colossales dont la qualité et l'exhaustivité sont primordiales aux gestionnaires. Malgré l'existence assez récente de ce genre de système, beaucoup d'entreprises sont actuellement équipées de système décisionnel. Avec tout ce qui a été élaboré pour ces

systèmes, il reste encore un problème fondamental auquel ils sont confrontés : leurs performances. Les gestionnaires d'entreprises qui sont habitués à l'exploitation des données de système décisionnel comprennent que ces données se doivent d'être de qualité et aussi disponibles rapidement. La productivité n'est plus la seule clé de la réussite; les entreprises se doivent d'être flexibles, rapides et surtout réactives.

Les systèmes décisionnels permettent de produire des rapports ou des tableaux de bord, qui répondent à la question « Que s'est-il passé ? », mais ils peuvent être également conçus pour répondre à la question analytique « Pourquoi est-ce que cela s'est passé ? » et à la question pronostique « Que va-t-il se passer ? ».

1.2 Entrepôt de données

Un entrepôt de données est une collection de données [7] :

- Orientée sujet ou *business* et organisée par thème afin de permettre la réalisation d'analyses autour des sujets primordiaux et des métiers de l'entreprise. Par cette organisation on peut ainsi passer d'une vision verticale de l'entreprise à une vision transversale beaucoup plus riche en information;
- Intégrée car provenant des différents systèmes opérationnels hétérogènes ou d'origines diverses de l'organisation. Ces données doivent être standardisées, épurées, unifiées et homogénéisées pour assurer la cohérence et l'intégrité de la connaissance de l'entreprise;
- Non volatile pour être stable, en lecture seule, non modifiable afin de conserver la traçabilité des informations et des décisions prises;
- Archivée et datée pour le suivi des évolutions des valeurs des indicateurs à analyser. Les informations stockées au sein du DW ne doivent pas disparaître. Elles deviennent, de ce fait, partie prenante de l'historique de l'entreprise.

Les entrepôts de données sont utilisés dans les systèmes décisionnels. Ils sont le cœur de

l'exploitation de l'information de l'entreprise.

1.2.1 DW versus Data mart

Un entrepôt de données peut prendre la forme d'un DW, d'un magasin de données ou d'un comptoir de données (*data mart).* En général, le DW globalise toutes les données applicatives de l'entreprise, tandis que les magasins de données sont des sous-ensembles d'informations concernant un domaine particulier de l'entreprise (marketing, finances).

Les comptoirs de données sont alimentés généralement depuis les données du DW. Mais il y en a qui sont bâtis à partir des bases de données transactionnelles (systèmes sources) et qui servent ensuite à alimenter un DW. La figure 1-1 donne un aperçu d'un magasin de données.

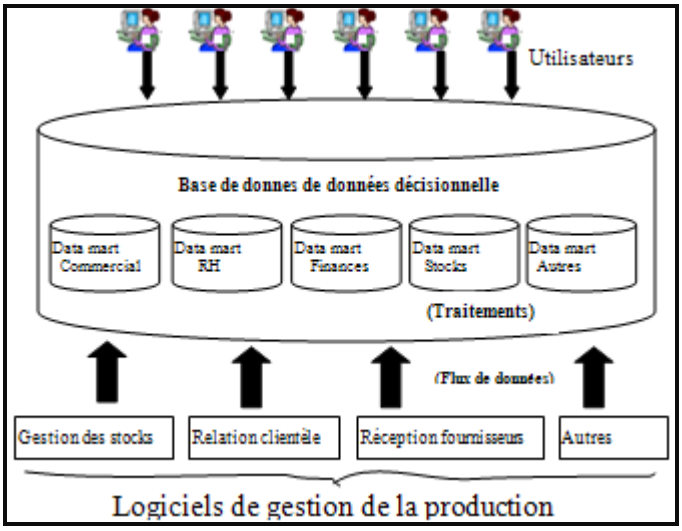

Figure 1-1: Schématisation d'un magasin de données

Quand on parle d'un entrepôt de données, il n'est pas toujours assimilé à un ou plusieurs comptoirs de données. Il se peut que l'entrepôt soit dépourvu de cette entité. Si tel est le cas, c'est un choix d'architecture dans l'entreprise.

1.3 Composantes d'un entrepôt de données

L'entrepôt de données est constitué d'un ensemble de composantes matérielles et logicielles qui forment l'environnement du DW au complet. Parmi ces composantes, certains permettent l'aboutissement à la collecte d'information définie dans le paragraphe précédent. D'autres composantes font partie de l'exploitation des informations pour le système décisionnel, principal exploitant du DW. Ces nombreuses composantes sont toutes reliées. La mise en œuvre d'un entrepôt de données ne peut pas se dissocier de ces composantes. Il y a des composantes de base communes à tout DW et plus le DW est complexe, plus il fait intervenir d'autres entités. La maintenance d'un entrepôt de données implique la gestion de tous les éléments qui le composent. Les intervenants sont donc nombreux avec des profils différents. Chaque élément de cet environnement vit avec ses propres forces et faiblesses de performance, faire fonctionner les performances de tout cet ensemble est un enjeu considérable. La figure 1-2 présente les composantes de base prenant part dans un entrepôt de données [7].

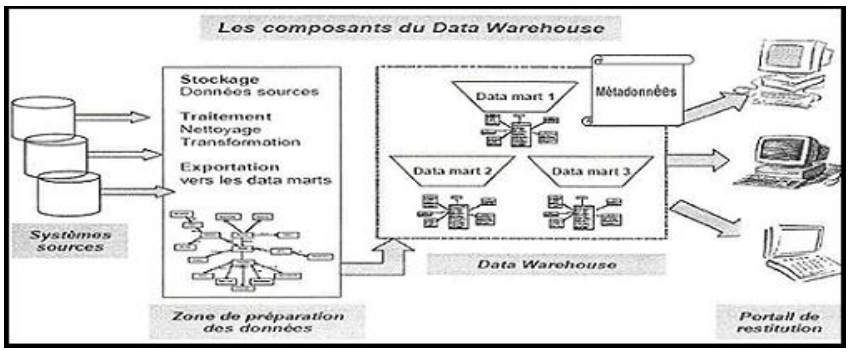

Figure 1-2: Composantes de base sommaires d'un entrepôt de données

Source : Jambu, M. (1999), p26 [7].

Dans la figure précédente, on ne peut pas détecter facilement l'intervention des opérateurs d'un DW. Il donne un aperçu de la constitution d'un entrepôt de données pour bien voir qu'il s'agit d'un environnement. La figure suivante montre ces composantes de façon détaillée pour mieux cerner où les actions peuvent être entreprises [9]. On peut voir que l'intervention est

multiple et cela dénote une équipe multidisciplinaire d'un DW.

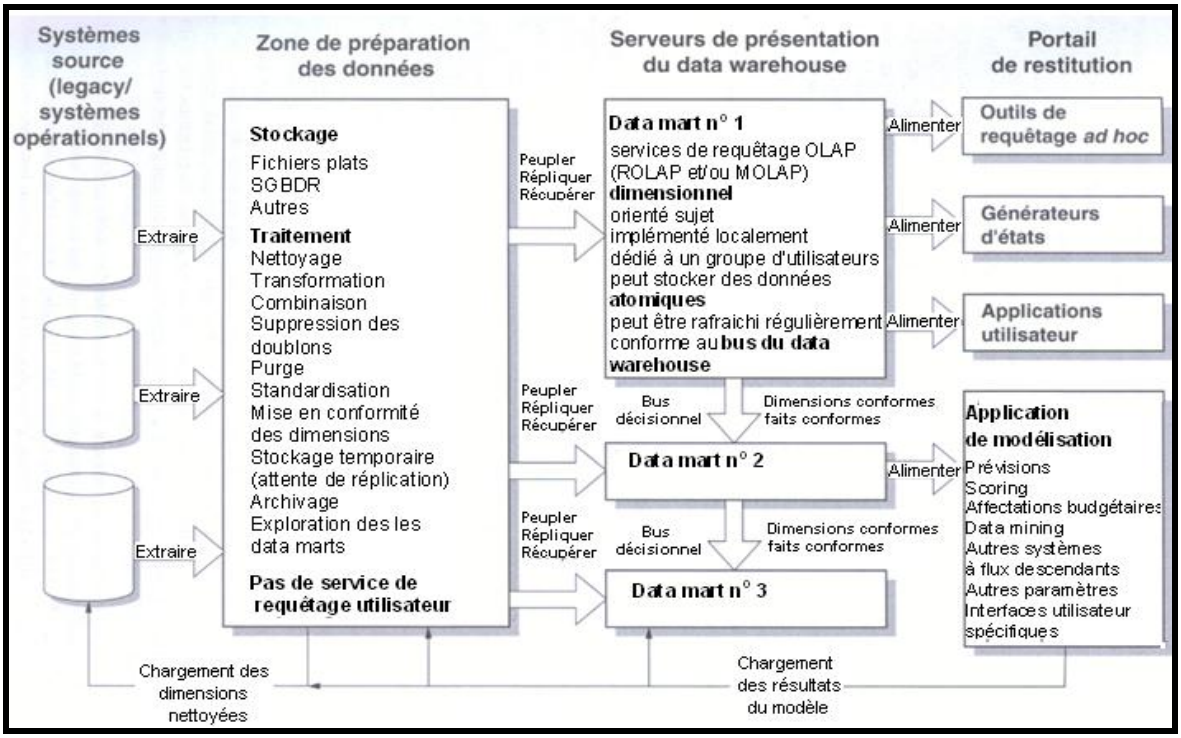

Figure 1-3: Composantes de base détaillées d'un entrepôt de données

Source : Kimbal, R. (2005), p.11 [9].

1.4 Principes d'un entrepôt de données

Plusieurs principes, normes et règles régissent un entrepôt de données. Ils sont prépondérants à son bon fonctionnement. Les paragraphes suivants en dénombrent quelques uns.

1.4.1 En amont et en aval (« *front-end* » et « *back-end* »)

En amont du DW se place toute la logistique d'alimentation des données de l'entrepôt : l'extraction des données opérationnelles, les transformations éventuelles et le chargement dans l'entrepôt : c'est l'ETC ou l'*ETL* (*Extract, Transform Load)* ou encore (*datapumping*).

En aval du DW (et/ou des data marts) se place tout l'outillage de restitution et d'analyse des données, outils de requêtes, de rapports, de tableau de bord. C'est le *BI* (*Business Intelligence*).

1.4.2 Niveau des données

Dans un DW, on retrouve les données de niveau élémentaire et les données de niveau agrégé. Les données élémentaires présentent des avantages évidents (profondeur et niveau de détail, possibilité d'appliquer de nouveaux axes d'analyse et même de revenir à posteriori sur le « passé ») mais représentent un plus grand volume et nécessitent donc des outils plus performants.

Les données agrégées présentent d'autres avantages tels que facilité d'analyse, rapidité d'accès, moindre volume mais il n'est pas toujours possible de retrouver le détail et la profondeur des indicateurs une fois ceux-ci agrégés et figés. On prend le risque de figer les données dans une certaine vue, selon les axes d'agrégation retenus. On ne peut plus revenir plus tard sur ces critères si le détail n'a pas été conservé (l'agrégation par mois ne peut pas donner une analyse par jour si le détail de la journée n'a pas été conservé). Les données agrégées sont préconçues pour répondre rapidement et efficacement à un besoin ciblé et connu avant de charger les données dans l'entrepôt. On agrège les données dans la BD par souci de performances, car il est plus rentable et efficace de les interroger déjà agrégées à travers un outil de rapport que de les calculer dans l'outil à chaque fois qu'une requête est demandée.

1.4.3 Dimensions et faits

Les notions de dimension et de fait caractérisent le DW. On retrouve ces deux notions au niveau du jargon de l'entrepôt de données. Pour les faits, il s'agit des tables de données dynamiques qui contiennent des mesures. Les tables de données plus statiques sont les dimensions. Ces dernières grossissent beaucoup moins que les tables de fait. Le modèle en étoile (*star schema*) est défini par les liens de la table de fait aux dimensions. L'annexe C

montre le modèle en étoile qui a été utilisé dans les tests. La description de la configuration de l'environnement et de la BD se trouve au paragraphe 3.3 et des annexes D et E.

1.4.4 Architecture d'un entrepôt

Un entrepôt de données est généralement construit selon une architecture en trois couches :

1) Un serveur d'entrepôt (serveur de données);
2) Un client;
3) Un serveur *OLAP* (*On-Line Analytical Processing*) qui peut être de type *X_OLAP* :
    - R_OLAP (*Relational* OLAP)
    - M_OLAP (*Multidimensional* OLAP)
    - H_OLAP (*Hybrid* OLAP)
    - D_OLAP (*Desktop* OLAP).

La différence entre ces technologies se situe au niveau du système de gestion de la base de données (SGBD). S'il est relationnel, il permet un stockage presque infini de données OLAP (R_OLAP). S'il est multidimensionnel (cube), il y a des limitations quant à la quantité de données (M_OLAP). S'il s'agit de fichiers de stockage sur le poste client, une infime quantité de données est stockée sur le poste (D_OLAP). Cette architecture sert à rendre les données disponibles en vue d'utilisation de forage, afin de permettre une escalade dans les niveaux de données. On peut ainsi passer du niveau le plus granulaire au niveau le plus sommaire.

1.4.5 Outils utilisés au niveau d'un DW

On retrouve dans un DW au minimum les outils suivants :

- Outil ETC commercial ou développé à l'interne dans l'entreprise;
- Outil pour l'exécution des requêtes;
- Outil pour l'analyse des données.

1.4.6 Structure de données

Un entrepôt de données a une structure de données représentée par un modèle de données normalisé troisième forme normale (3FN) pour les données de détail et en étoile ou en flocon pour les données agrégées. Ces données sont stockées dans un SGBD relationnel notamment pour les données élémentaires. Éventuellement, la structure peut être multidimensionnelle stockée dans un cube ou hyper cube M_OLAP, réservée aux données agrégées des magasins de données.

1.4.7 Vocation d'un entrepôt de données

L'entrepôt de données permet avant tout d'intégrer des sources de données hétérogènes diverses. Mais son objectif final n'est autre que de supporter un processus d'analyse en ligne. Il suffit de décrire les aspects d'intégration et d'analyse, l'architecture décisionnelle pour enfin présenter le DW comme centre d'info utile à la décision.

1.4.8 Résumé

Le DW fait parler plusieurs composantes : matérielles et logicielles (*hardware* et *software*). De plus, il est en constante évolution en amont et en aval. Afin qu'il puisse répondre à la vocation à laquelle il est voué, il se doit de répondre à des normes de performance fortement exigeantes. Les utilisateurs du système décisionnel de l'entreprise, qui sont, entre autres, ses grands gestionnaires, utilisent les données du DW périodiquement : plusieurs fois par jour, quotidiennement, mensuellement. Ils sont exigeants et pressés d'avoir l'information juste, fiable et diligente. Pour que le DW puisse continuer de vivre et d'évoluer dans l'entreprise, il est impératif de considérer les problèmes de performance auxquels il fait face. Un DW ne peut continuer d'exister s'il persiste dans une piètre performance.

Chapitre 2

Exploration de problèmes de performance d'un DW

Au début de l'existence des entrepôts de données (années 90), les exigences se limitaient à des demandes simples. Elles étaient exprimées pour des rapports simples, des requêtes et analyses utilisées par une petite communauté d'usagers bien identifiée par l'entreprise. De nos jours, les entreprises sont équipées d'*ERP* (*Enterprise Ressource Planning*). Ces systèmes gèrent différents types de requêtes multiples, devant être constamment disponibles pour une communauté d'usagers beaucoup plus grande, variée, diverse et exigeante. Les entreprises de toutes les industries voient leur DW intégré avec les fonctionnalités de l'ERP : finances, comptabilité, *CRM* (*Customer Relationship Management*), marketing [3]. Ceci met en évidence non seulement la masse colossale devant être absorbée, traitée et disponible par le DW, mais aussi son nombre d'usagers grandissant pour tous les domaines d'affaire de l'entreprise. Ces demandes exercent une pression sur le DW. Il doit constamment être efficace afin de répondre aux exigences actuelles de plus en plus complexes, nombreuses et surtout pressantes de l'entreprise.

Devant un contexte économique difficile, caractérisé par l'insécurité de l'emploi et l'âpreté de la concurrence, de nouveaux concepts tels que la démarche qualité, l'intelligence économique ou encore la veille technologique ont été développés pour le monde de l'entreprise. Sur le plan décisionnel, le besoin en matière d'accès à l'information est devenu pressant pour permettre aux décideurs d'anticiper sur les événements, c'est-à-dire de prévoir la demande des clients et de s'informer sur la stratégie de la concurrence. L'informatique décisionnelle avec ses entrepôts de données s'est développée pour apporter des réponses à ce besoin.
Malheureusement, les trois entrepôts de données dont l'expérience a servi dans cet essai souffrent de problèmes de performance. Plus l'entrepôt de données est gros, plus la

performance en souffre. Plus les exigences sont nombreuses et épineuses, plus la performance en subit les conséquences. Les origines de ces problèmes sont diverses et leurs causes sont abondantes. Leur présence se trouve en amont et en aval du DW.

Le présent chapitre traite la localisation de certains problèmes de performance et de leurs origines. L'exploration se fera suivant une démarche ascendante (*bottom-up*) qui permet de parcourir les éléments élémentaires pour atteindre l'exploitation des données du DW. La logique suivie offre aux différents intervenants d'un DW la possibilité d'atteindre les tâches les concernant de façon rapide. La vie quotidienne d'un entrepôt de données fait intervenir plusieurs composantes et aussi plusieurs participants. Des intervenants clés ont un rôle prépondérant à accomplir pour que le DW livre à ses usagers le service de qualité attendu. Il est essentiel de comprendre le rôle et les tâches réservées aux uns et aux autres.

2.1 Mise en contexte.

Pour mieux comprendre l'approche utilisée, il est important de préciser la situation dans laquelle la réflexion a été élaborée. Il s'agit du cas où l'entrepôt de données est déployé et est exploité par la communauté des usagers de l'entreprise. Il n'est pas mentionné le cas d'un projet de conception et d'implantation d'un DW. Cette phase est supposée, complétée et achevée. Il va de soi que si on parle de problèmes de performance, c'est que le DW a déjà été mis en place. Il va sans dire que certains concepts pourraient mieux être pris en charge dès l'étape de la conception pour contrer les faiblesses de performance de l'entrepôt.

2.2 Acteurs au niveau d'un entrepôt de données

Dans le chapitre précédent, il a été défini ce qu'est un entrepôt de données ainsi que les composantes le formant. Pour chacune des composantes, plusieurs intervenants s'y retrouvent. L'ensemble des personnes œuvrant dans un entrepôt de données se concentre dans un groupe technique et un plus grand groupe d'utilisateurs. Les usagers peuvent être répartis dans tous

les domaines d'affaires différents de la compagnie et dans toutes ses divisions géographiques. L'équipe d'un système décisionnel ne se limite donc pas aux professionnels techniques le régissant. Ses utilisateurs peuvent en faire une réussite ou un échec. Contrairement aux applications opérationnelles, l'entrepôt de données a toujours besoin d'un mentor qui a la charge de promouvoir l'entrepôt auprès de ses usagers et des parties prenantes de l'entreprise. Le mentor a la responsabilité de faire connaître aux décideurs de la compagnie la valeur ajoutée quant à l'utilisation des données de l'entrepôt. Ce ne sont pas tous les employés de l'entreprise qui connaissent les avantages de ces données. Le mentor a pour but de propager, vulgariser et populariser l'efficacité de l'entrepôt.

Outre l'équipe de développement, on retrouve les intervenants principaux suivants:

- Un administrateur ETC, dont le rôle est de surveiller tout ce qui se passe au niveau « *back-en* ». Il doit être à l'affût de tout changement impliquant les données dans toutes leurs formes. Il doit être équipé d'outils lui permettant de suivre le comportement complet de l'ETC. Ceci implique les traitements d'extraction de données des systèmes sources, le traitement de transformation et de rejets et enfin les traitements de chargement. Il joue un rôle déterminant dans la performance de l'entrepôt;
- Un administrateur de base de données (*DBA database administrator*) est nécessaire dans un DW. Quand l'entreprise en a les moyens, il peut y avoir un DBA exclusivement dédié à la BD entrepôt, autrement le ou les DBA de la compagnie s'occupe(nt) de la BD entrepôt au même titre que des autres BD opérationnelles. D'ailleurs, le profil du DBA spécialisé entrepôt de données est très recherché et est en vogue. Ce rôle est essentiel dans la performance. Le DBA doit suivre l'évolution de la BD car celle-ci grossit rapidement. Il doit parfois intervenir physiquement sur les tables lors d'un problème de performance comme celui de la table verrouillée (l*ocked table*). Il a constamment beaucoup de gestions à faire;
- Un ou plusieurs architectes techniques dont le rôle est de maintenir l'architecture de l'entrepôt dans la modélisation et le flux de données. Ce rôle permet d'assurer la

cohérence au niveau de tout ce qui est dimension, hiérarchies, faits, *lookup table*. L'évolution du modèle est toujours impliquée dans la performance;

- Un administrateur outils d'exploitation (BI), comme par exemple administrateur Microstrategy, administrateur *Report Studio* ou *Powerplay Studio* de Cognos. Ce rôle est important face à la fragilité de la performance d'un DW. Il demande une maîtrise de l'outil utilisé au niveau « *front-end* », car il est lui-même exposé à ses propres limites. Plus tôt les limites sont connues, plus tôt les problèmes de performance seront détectés et maîtrisés;
- Un ou plusieurs experts de requêtes SQL. Ce profil est nécessaire dans l'équipe technique, car chaque requête SQL doit être soumise à un examen de performance avant qu'elle ne devienne opérationnelle en production;
- Un ou plusieurs analystes d'affaire (*Business Analyst - BA)* sont souvent présents dans l'équipe de l'entrepôt de données. Le BA a pour rôle de présenter le besoin de l'usager à l'équipe technique et d'en faire une analyse exhaustive. Dans l'entrepôt de données, il a le souci d'encadrer les usagers avec les outils de restitution utilisés au sein du DW. Face à la performance, il peut être la première personne contactée pour véhiculer l'existence d'un problème de performance à l'équipe et aux responsables techniques. Le BA peut éventuellement aussi être le mentor de l'entrepôt;
- Plusieurs usagers principaux *(Power users*) font partie de la communauté utilisatrice du DW. Ils sont les premiers répondants du DW. Ils peuvent être aussi les premiers à subir les méfaits de la performance;
- Plusieurs usagers ordinaires qui peuvent être les différents responsables, les directeurs de département, les pilotes de projet, les analystes financiers, de marketing représentent les clients du DW. Ils expérimentent tout ce qui a trait à la performance d'un entrepôt de données.

Plusieurs acteurs peuvent identifier les problèmes de performance d'un DW. Ils peuvent être les personnes techniques le prenant en charge quotidiennement. Mais le plus souvent ce sont les différentes communautés d'utilisateurs qui soulèvent ce genre d'incidents. Les usagers sont

confrontés quotidiennement avec la réalité du DW. Dans certains cas, ils sont les premiers à détecter tout ce qui va et tout ce qui ne va pas. Il est dommage que ce soit les usagers qui soulèvent cette lacune. Cela diminue la crédibilité du DW. Quand on arrive à ce genre de situation, des mesures doivent permettre d'améliorer rapidement la performance de l'entrepôt.

2.3 Problèmes de performance : en amont et en aval du DW

On commence à s'inquiéter des performances d'un DW quand la durée des traitements dépasse les délais permis ou encore les délais raisonnables. Les délais sont évalués en fonction du temps maximal accordé à un traitement. Ils sont définis en fonction du temps moyen mis par chaque traitement et de la criticité du traitement lui-même. Si par exemple, un tableau de bord est demandé pour présenter des chiffres mensuels très importants aux grands responsables de l'entreprise, le délai d'exécution tolérable est seulement de quelques minutes. Il s'agit d'un exemple de délai de traitement en aval du DW.

Si les données doivent être disponibles à compter de 7h30 le matin pour l'ensemble des usagers afin qu'ils puissent commencer à produire leurs rapports quotidiens, le délai peut être de quelques minutes à quelques heures sans trop d'impacts majeurs. Ce deuxième exemple illustre le cas de traitement en amont du DW.

Au-delà du délai permis, l'entrepôt commence à signaler des signes de faiblesse de performance. Plus ces situations se répètent, plus la performance peut être considérée comme inquiétante et problématique et doit être soumise à une sérieuse analyse et à un diagnostic. Idéalement, la performance de l'entrepôt de données doit être une préoccupation quotidienne de l'équipe technique, afin que cela ne devienne pas alarmant pour les utilisateurs ou encore moins pour les dirigeants de l'entreprise.

Dans le présent document, il sera fait souvent référence à un chargement quotidien de données

pour l'unique raison que l'expérience actuelle vécue avec les entrepôts de données utilise cette fréquence. Cela ne veut pas dire que tous les entrepôts de données sont obligatoirement alimentés avec des données journalières. En fait, il existe des chargements dont la fréquence d'itération de chargement peut être calculée en minutes et d'autres en mois. Les exemples sont nombreux et faciles à comprendre. À vrai dire, la fréquence d'extraction de données est dictée par les besoins d'affaires. Les exemples suivants illustrent très bien la raison de la fréquence dans chaque cas.

1) Dans le cas de ventes de détail, on a besoin de récupérer toutes les ventes et les mouvements d'inventaire qui ont lieu chaque jour. Les chiffres sont soumis à des analyses journalières, hebdomadaires, mensuelles. Ces analyses permettent par exemple de trouver quel est le jour où tel produit est le plus vendu dans tel magasin de telle région. Elles peuvent aussi montrer si les promotions annoncées en circulaire, en magasin ou à la télévision, permettent d'avoir plus de ventes. On pourra aussi détecter quel est le produit qui est sujet à des ruptures de stocks plus fréquemment et analyser les raisons pour éviter que cela se reproduise.

2) Dans l'exploitation du réseau électrique, on a besoin d'analyser le comportement des appareils qui prennent des mesures. Plus les mesures sont rapprochées, plus la précision de l'appareil peut être analysée. L'étude de l'ensemble des mesures permet de détecter les pannes et de les prévenir. On a besoin de récolter les mesures aux cinq minutes. L'extraction doit obligatoirement avoir lieu aux cinq minutes.

Les mécanismes liés à l'ETC ne permettent malheureusement pas tous de faire les traitements de manière saine, sécuritaire à 100 %, sans embûche et aucun problème. Tout ce qui se passe du côté « *front-end* » est lié à la donnée comme telle et à son support. De plus, pratiquement tout problème qui se produit du côté « *back-end* » a une incidence sur le « *front-end* ». La figure 2-1 donne un aperçu de quelques scénarios de problèmes de performance en amont et en aval du DW.

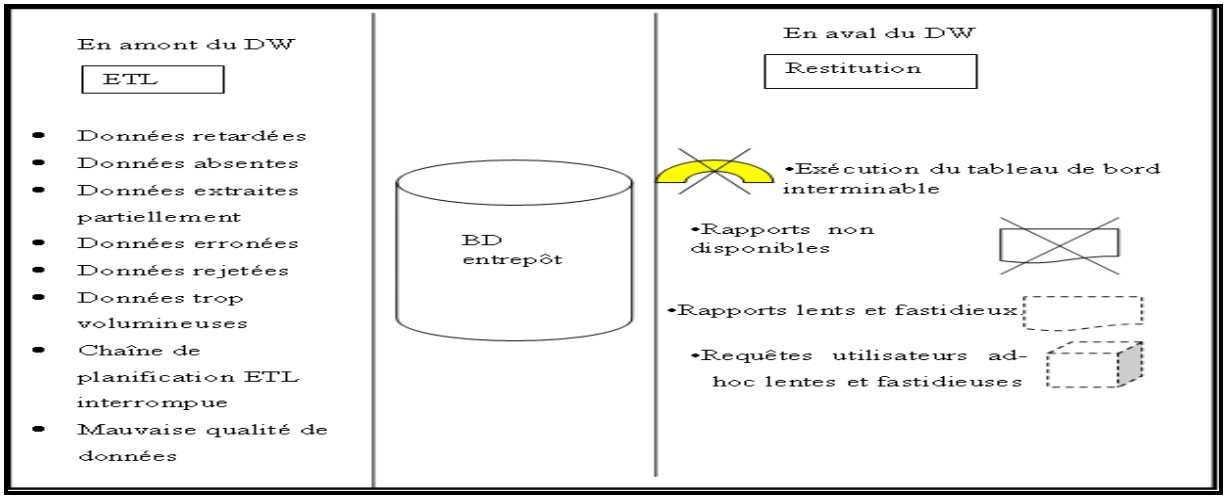

Figure 2-1: Identification de problèmes de performance en amont et en aval d'un DW

On peut voir à la figure 2-1 que l'intervention du DBA est requise pour les volets amont et aval de l'entrepôt de données. Dans le chapitre dédié aux solutions des problèmes de performance, la participation des acteurs du DW est inhérente à la résolution du problème.

2.4 Performance « *back-end* » du DW

En amont de l'entrepôt de données se trouve l'ETC. Comme le nom l'indique : Extraction-Transformation - Chargement. Les outils ou mécanismes de l'ETC permettent d'extraire des données à partir de différentes sources. Ensuite, il s'agit de les transformer légèrement (format, dénomination) ou fortement (ajout d'une règle de gestion pour raffiner le résultat). Enfin, il ne reste qu'à les charger dans la BD entrepôt, afin de les mettre à la disposition des usagers sous une forme exploitable et adaptée à la prise de décision.

L'ETC constitue la phase essentielle d'un DW. Sans les données chargées périodiquement, il n'y aura pas d'historique, par conséquent, il n'y aura pas d'analyse possible. Il est fondamental de ne manquer aucune période de chargement pour ne pas biaiser les analyses. Celles-ci sont basées sur le comportement des données s'étendant dans le temps (jours, mois,

semaines, années, trimestre, semestre, heure, minute), ce qui permet de simuler entres autres des tendances.

Un ETC permet d'éviter la réalisation de programme en lots répétitifs, souvent semblables, dont il faudra également assurer la maintenance. Il est capable de prendre en charge différentes natures de sources de données, tant en entrée qu'en sortie, les principales étant les systèmes de gestion de base de données (SGBD) relationnelles, les flux XML, mais aussi des fichiers de format fixe ou avec séparateurs (*Comma-Separated Values - CSV*), des services Web, les connecteurs pour la connexion aux progiciels. Le principe est que l'intégration d'un nouveau flux de données ne requière aucun développement. Il s'opère par une simple configuration interactive : on choisit les éléments de données dans la référentielle source, on indique les transformations simples qu'ils doivent subir et on précise la destination de la donnée dans l'entrepôt.

Une fois qu'un flux d'extraction-transformation-chargement a été défini, il est déclenché de manière régulière, ceci sous le contrôle d'un outil de planification de tâches ou d'ordonnancement comme par exemple Appworx (http://www.appworx.com). Cette planification est soumise à un monitoring afin de détecter des bris de traitement.

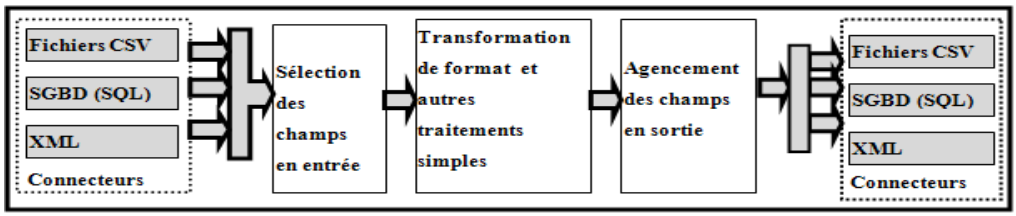

Figure 2-2: Différentes sources de données prises en charge par l'ETC

L'ETC peut être effectué par un outil commercial ou par des programmes de développement internes, c.-à-d. propres à l'entreprise. Les outils commerciaux les plus répandus sont :

Datastage d'IBM[1], Oracle *Warehouse Builder (OWB)*[2], *PowerCenter (Informatica)*[3], *Business Objects* (SAP)[4], *SSIS* (Microsoft)[5] [8]. Qu'on utilise un outil ou non, l'ETC doit être étroitement surveillé. Le temps de chargement est toujours chronométré et il doit avoir lieu quand la BD entrepôt est la moins sollicitée. Dans le cas d'un chargement quotidien, l'ETC est planifié durant la nuit quand le service de l'entrepôt est le moins requis. On suppose que les usagers sont tous déconnectés et si les usagers sont à travers le Canada par exemple, il faut considérer le décalage horaire.

Avec ou sans outil ETC, la faiblesse de performance peut apparaître aux différents stades de l'ETC. Elle est influée par plusieurs paramètres et l'utilisation d'un outil commercial n'exclut pas de problèmes de performance. Il est certes conçu pour répondre aux besoins exigés. Cependant, il demeure que c'est un produit logiciel et est donc soumis à des forces et faiblesses et il possède ses propres limites. Ces points ne sont généralement pas énumérés et documentés dans l'outil. On le découvre au fur et mesure de son utilisation au détriment de la performance du DW. Il n'en demeure pas moins que pour un gros entrepôt de données, il est préférable d'utiliser un outil commercial. Ce dernier gère de façon souple et surtout visuelle le changement d'évolution lente (*slowly changing dimension*) ainsi que les faits qui arrivent tard (*late arriving fact*). Ces deux concepts typiques aux traitements de données entrepôt sont très compliqués à mettre en pratique via des programmes ordinaires. Généralement, tous les outils ETC incorporent le module de gestion de la métadonnée (*metadata*). C'est un élément très important dans le concept d'un entrepôt de données et il est habituellement négligé dans la programmation ordinaire d'un ETC. À cet effet, il existe sur le marché plusieurs produits ETC. Le choix dépend de plusieurs paramètres et leur étude en vaut la peine avant d'en sélectionner un. L'idée peut être donnée par l'évaluation de Gartner qu'on retrouve deux fois

---

[1] http://www-01.ibm.com/software/data/infosphere/datastage/

[2] *http://www.oracle.com/technology/products/warehouse/index.html*

[3] *http://www.informatica.com/products_services/powercenter/Pages/index.aspx*

[4] http://www.sap.com/solutions/sapbusinessobjects/index.epx

[5] http://technet.microsoft.com/fr-fr/library/ms141026.aspx

par année dans la publication du quadrant magique. La figure 2.3 représente le quadrant magique de Gartner en ce qui a trait aux outils ETC et montre que les plus renommés sont ceux d'Informatica (*PowerCenter*) et d'IBM (*Datastage*).

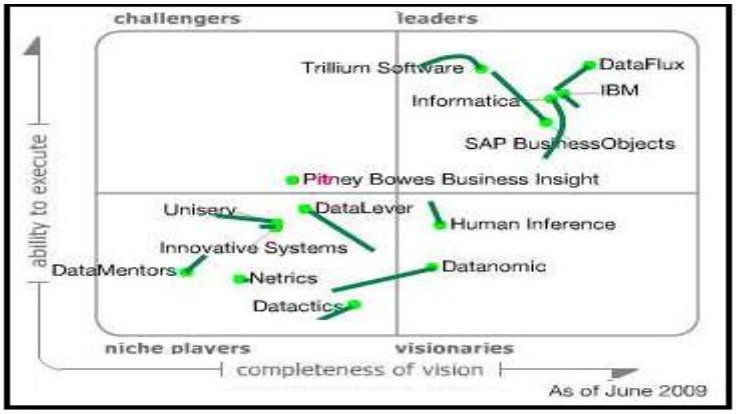

Figure 2-3: Quadrant magique Gartner- outil ETC juin-2009

Source : Gartner – http://www.ethantang.fr/2009/06/454-dataflux-mene-toujours-dans-le-gartner-magic-quadrant-for-data-quality-tools-2009/ consulté le 4-avril-2010.

Choisir d'utiliser un outil commercial spécialisé de type ETC, qui est capable de prendre en charge tout ou partie des étapes d'acquisition des données, peut s'avérer très avantageux. En effet, l'avantage augmente fortement avec la complexité de l'environnement d'un DW (hétérogénéité des systèmes sources, complexité des règles d'affaire et de transformation, traitement de volumes importants de données).

Plusieurs risques sont liés à l'absence d'un outil ETC non réglementé :

- Disposer d'un environnement non intègre en termes de qualité des données et propager de façon récurrente des données erronées empêchant l'accomplissement des objectifs finaux;
- Perdre les avantages attendus de la mise en place d'un outil d'aide à la décision;
- Perdre la confiance des utilisateurs finaux des données présentes dans l'entrepôt de données, voire de leur utilisation;
- Ne pas être en mesure de garantir la livraison de données intègres au moment

opportun lorsque requis par les utilisateurs finaux;

- Augmenter les coûts de développement (maintenance et évolution) à long terme, afin de palier l'absence d'outils spécialisés de type ETC et ne pas être en conformité avec les standards de l'industrie quant à l'utilisation de ce type d'outils (tendance croissante sur le marché de recours aux outils spécialisés ETC – 75 % des solutions d'entrepôt de données utilisent des plateformes ETC spécialisées selon une étude de Gartner http://www.gartner.com/technology/home.jsp).

2.4.1 Les défis d'un ETC

L'implantation de processus d'ETC efficaces comprend de nombreux défis. Le volume de données croît de façon exponentielle. Ces processus doivent en plus traiter des quantités importantes de données granulaires (produits vendus, transactions bancaires). Ces processus prennent en charge la mise à jour de données à chaque itération de façon incrémentale régulièrement et de façon complète (*full extract*) occasionnellement.

Les structures et applications décisionnelles incluent des data warehouse, des data marts, des applications OLAP – pour l'analyse, les rapports, les tableaux de bord, la gestion des indicateurs de performance (*scorecarding*). Toutes ces structures cibles présentent des besoins différents en terme de transformation de données, ainsi que des latences différentes.

Les transformations des processus d'ETC peuvent être très complexes. Les données doivent être agrégées, calculées et traitées statistiquement. Certaines transformations spécifiques au décisionnel sont aussi requises, comme les dimensions à évolution lente (*SCD - Slowly Chaging Dimension*).

L'ETC se doit d'être accompagné du dictionnaire de données : la métadonnée (m*etadata*) qui a pour but de définir toutes les données utilisées dans l'entrepôt donc passant par l'ETC en y indiquant : son origine (la source), sa transformation, son support la véhiculant, son format, sa destination.

Alors que le décisionnel se rapproche du temps réel, le data warehouse et data mart doivent être rafraîchis plus souvent, alors les fenêtres de chargement se raccourcissent. Un bon ETC se reconnaît dans sa complétude et dans son exhaustivité dans la prise en charge des données depuis les sources jusqu'à l'entrepôt.

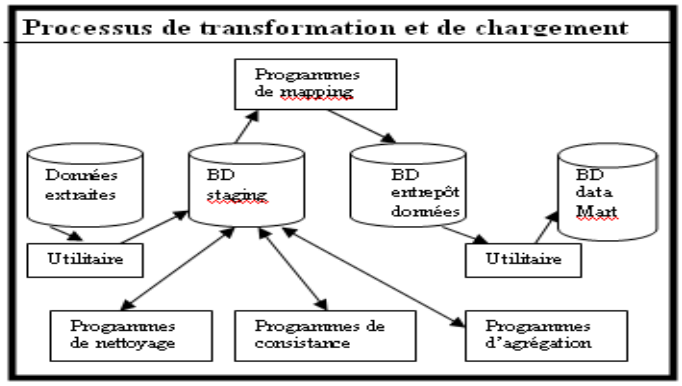

Figure 2-4: Complexité des processus ETC

Un bon ETC se doit essentiellement d'être accompagné d'un planificateur ou ordonnanceur de tâches et d'un outil de monitoring, qui aura la charge d'enregistrer tous les statuts d'exécution de chacune des tâches du planificateur. Un historique de cette planification permet une étude exhaustive, rentable et efficace sur la performance d'un ETC. Ce point sera détaillé ultérieurement dans ce document.

Les métriques disponibles permettant d'évaluer les performances de l'ETC sont liées au monitoring de la chaîne ETC. Chaque module de la chaîne est dédié à exécuter des instructions d'extraction, de transformation ou de chargement. Pour chacun on connaît le temps maximum accordé et on pourrait le céduler à même le module.

2.4.2 Métriques d'évaluation de performance d'un ETC

Il y a quelques métriques essentielles qui permettent d'évaluer la performance d'un ETC [8]. Entre autres, il s'agit de :

1) La durée d'exécution de chaque module d'extraction, ou de transformation ou de chargement, en secondes. Ce temps est égal à la différence de : l'heure de fin moins l'heure de début, converti en secondes;
2) Le nombre de fiches (*record*) traitées par seconde. Il est égal au nombre total de fiches traitées divisé par la durée de traitement en secondes;
3) Le nombre de fiches lues par seconde. Il est égal au nombre total de fiches lues depuis les systèmes sources divisé par la durée en secondes;
4) Le nombre de fiches écrites par seconde. Il est égal au nombre total de fiches écrites dans les tables de destination de l'entrepôt divisé par la durée en secondes;
5) Le débit (*throughput*) est le nombre de fiches traitées par seconde multiplié par le nombre d'octets de chaque fiche. Comme pour toutes les mesures de performance, le débit peut être considéré comme ligne de base pour l'amélioration de performance.

L'historique de ces valeurs permet de calculer un délai moyen pour chaque module et ainsi situer chaque traitement dans la moyenne. Des mécanismes automatiques peuvent être mis en place pour détecter le dépassement du délai de traitement. Par ailleurs, la surveillance du déroulement de tous les traitements permet aussi d'analyser la performance.

2.4.3 Identification des problèmes de performance « *back-end* » d'un DW

Quand on parle de l'extraction, on distingue toujours l'extraction complète (*full*) de l'incrémentale (delta). Procéder à une extraction complète veut dire ramener toutes les données présentes dans les tables opérationnelles, alors que le delta implique seulement les données qui ont subi un changement depuis la dernière itération. Suivant la conception originale, on peut trouver des dimensions avec lesquelles on procède à un chargement complet à chaque itération. Cela est souvent dû à l'absence de la date de modification ne permettant pas ainsi de garder les traces de changement. Quand on décide de mettre en place ce type de chargement pour une dimension, c'est que celle-ci ne devrait pas être volumineuse et ne devrait pas le devenir au fil du temps. Par exemple, dans le cadre de la vente au détail, on extrait quotidiennement de façon complète les dimensions de produits, magasins et régions.

Les ventes et les inventaires sont extraits pour les produits qui ont eu des transactions depuis la veille : extraction delta.

Le tableau suivant rassemble des problèmes de performance qu'on peut rencontrer au niveau de l'ETC. Chaque problème sera identifié d'un préfixe 'B' pour « *back-end* » et d'un numéro séquentiel. Cette identification sera utilisée dans la matrice problème/solution qui sera décrite dans le chapitre suivant et l'annexe H.

Tableau 2-1: Identification de problèmes de performance « *back-end* » du DW

No	Description du problème	Cause possible	Incidence sur le BI
B1	Les données n'ont pas été chargées.	Un bris réseau.	Les rapports et tableaux de bord ne peuvent pas être exécutés pour les données actualisées car elles sont manquantes.
B2	La planification ETC a été interrompue.	Un module de la chaîne est resté indéfiniment bloqué.	Les rapports et le tableau de bord ne peuvent pas être exécutés avec les données récentes.
B3	Le chargement d'une table a été fait partiellement.	Arrêt brusque d'un module de chargement sans qu'on ait repris l'exécution de la suite de la chaîne.	On pourrait avoir chargé les données détail mais pas le sommaire par exemple. On pourrait avoir des données corrompues dans les rapports et/ou tableaux de bord.

No	Description du problème	Cause possible	Incidence sur le BI
B4	Les systèmes sources n'ont pas envoyé les données.	Problème de communication via le mécanisme mis en place (FTP, connexion à la BD opérationnelle).	Les rapports et tableaux de bord ne peuvent pas être exécutés pour les données actualisées.
B5	Les systèmes sources ont envoyé des données non actualisées.	Erreur sur la date d'extraction.	Les rapports et tableaux de bord ne peuvent pas être exécutés pour les données actualisées.
B6	Les systèmes sources ont envoyé les données en retard.	L'extraction des données depuis la source a duré plus de temps que prévu.	Les rapports et tableaux de bord ne peuvent pas être exécutés pour les données actualisées.
B7	Les données sont trop volumineuses.	Lors d'un chargement pour synchronisation des données avec la source comme dans le cas de l'inventaire dans les ventes au détail.	Les données ne seront pas disponibles à la communauté usager à l'heure habituelle ; Les rapports et tableaux de bord seront tardivement disponibles.
B8	Le temps de chargement a dépassé les délais permis.	Un temps mort (*timeout*) est programmé pour éviter de laisser un chargement prendre plus de temps que la normale afin de ne pas pénaliser les autres modules de chargement.	Les données concernées par le module en question peuvent ne pas avoir été extraites ou chargées. Les rapports ne peuvent pas utiliser ces données.

No	Description du problème	Cause possible	Incidence sur le BI
B9	Il y a des erreurs dans le format de fichier envoyé par les systèmes sources.	Manque d'attention ou manque de communication du changement entre la source et l'entrepôt.	Les données ont été rejetées donc les rapports ne peuvent pas être utilisés avec ces données.
B10	Il y a une erreur dans les données.	Changement de format de données.	Données récentes non disponibles pour les rapports.
B11	Les fichiers sont rejetés.	Incompatibilité de format (un champ de trop ou de moins, un type alpha au lieu de numérique; Incompatibilité de données.	Données récentes non disponibles pour les rapports.
B12	Les données sont rejetées.	Données dont les valeurs sont inappropriées (ex. : un code CUP saisi à la place de la quantité vendue).	Données récentes non disponibles pour les rapports.
B13	Les fenêtres de temps d'extraction sont très étroites.	Peut être dû à l'indisponibilité des systèmes sources.	Le chargement peut ne pas avoir été complété dans les délais donc les rapports sont exécutés avec les données incomplètes car retardées.

No	Description du problème	Cause possible	Incidence sur le BI
B14	Chargement complet pour toutes les dimensions.	Il arrive que l'entrepôt soit désynchronisé avec la source pour certaines tables.	Généralement il faut beaucoup de temps à recharger une dimension dans sa totalité donc les données risquent de ne pas être utilisables si cette table n'est pas disponible.
B15	Absence de mécanisme de reprise après échec d'exécution de la chaîne.	C'est une lacune de la description des opérations d'enchainement des modules de la chaîne ETC.	Risque d'absence de données et corruption des données.
B16	Outil ETC ne soutient pas le parallélisme.	Lacune de l'outil ETC qu'il soit commercial ou non.	Le chargement prend plus de temps qu'il n'en faut par conséquent parfois données non disponibles.
B17	Outil ETC ne soutient pas la mise en cache.	Lacune de l'outil ETC.	Le temps de réponse à une requête réutilisée plusieurs fois prend beaucoup plus de temps que prévu.
B18	Outil de rapports ne soutient pas la mise en cache.	Lacune de l'outil de restitution.	Le temps de réponse à une requête réutilisée plusieurs fois prend beaucoup plus de temps que prévu.

No	Description du problème	Cause possible	Incidence sur le BI
B19	Calcul d'agrégat prend trop de temps.	Les jointures ne sont pas faites de façon optimale.	Temps de réponse d'une requête prend plus de temps que nécessaire.
B20	Données incorrectes.	Données plus grandes que le maximum permis ou plus petites que le minimum permis.	Données récentes incomplètes.
B21	Les dimensions ne sont pas chargées.	Module de chargement des dimensions n'a pas été complété.	Les faits ne peuvent pas être chargés.
B22	Envoi d'un fichier vide du système source.	Absence de validation au niveau du système source.	Absence de données.
B23	La BD du système source tombe.	Si L'ETC se branche sur la BD du système source, celle-ci tombe pour une raison inconnue au moment de l'extraction.	Plusieurs cas possibles : les données ne sont pas chargées car n'ont pas été extraites; Les données ont été chargées partiellement.

2.5 Performance « *front-end* » du DW

Du côté « *front-end* », on utilise des outils BI. Ils sont nombreux sur le marché et répondent tous aux besoins d'affaire. Il y en a qui incorporent le volet *reporting*, le cube, les tableaux de bord, les *scorecarding*. C'est le cas de la solution *BI Entreprise* comme la suite 'Série 8' de

Cognos[6]. Il y a aussi des outils qui n'offrent que l'outil de rapports, exemple *Crystal Reports*[7]. Tous ces outils ont des modèles de données qu'il faut considérer pour que la performance ne souffre pas. Le choix peut reposer sur plusieurs critères. Néanmoins, les objectifs visés sont communs : fournir une analyse en ligne avec les données de l'entrepôt pour permettre une prise de décision juste et éclairée. Afin d'en faire un usage efficace, il est essentiel d'étudier l'outil et de suivre les recommandations qui sont liées à la performance. Par exemple, avec le tableau de bord de *Cognos*, il est nécessaire de bâtir un modèle avec *Framework Manager*[8] qui est un outil de modélisation de l'information présentée aux utilisateurs. Sans cette modélisation, le tableau de bord ne peut se faire. Ce modèle doit être optimal afin que le tableau de bord ne souffre pas d'un manque de performance. Pour ce faire, ce modèle doit suivre certaines règles de modélisation comme le modèle en étoile, l'utilisation des clés étrangères (*surrogate key*).

Le quadrant magique de Gartner les classe comme montré dans la figure suivante :

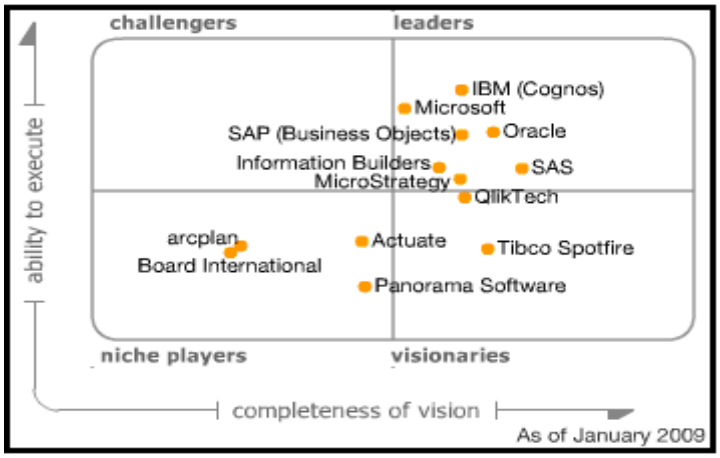

Figure 2-5: Quadrant magique Gartner - Outils BI - janvier-2009

Source : Gartner – http://mediaproducts.gartner.com/reprints/sas/vol5/article8/article8.html consulté le 4-avril-2009.

---

[6] *http://www-01.ibm.com/software/data/cognos/products/cognos-8-business-intelligence*

[7] *www.crystalreports.com*

[8] *http://www.cognos-bi.info/tutorial/content_store-framework_manager.htm*

Les problèmes de performances au niveau BI sont nombreux. Dans la présente étude, ils ne font pas l'objet d'une présentation exhaustive. Ils sont abordés de façon globale et il ne sera cité que quelques cas majeurs qui semblent être les plus répandus. Le tableau 2-2 montre quelques situations connues. Chaque problème est identifié d'un préfixe 'F' pour « *front-end* » et d'un numéro séquentiel. Cette identification est utilisée dans la matrice problème/solution qui est décrite dans le chapitre suivant et dans l'annexe H.

Tableau 2-2: Identification de problèmes de performance « *front-end* » du DW

No	Description du problème	Cause possible
F1	Sélection de données supérieure au besoin.	Requête mal exprimée.
F2	Durée d'attente d'exécution de rapport prolongée par rapport à la normale.	Trafic réseau élevé.
F3	Expiration de délai d'attente de la requête.	Requête dépassant le temps permis.
F4	Utilisation de matrice (*lookup table*) complexe.	Matrice contenant des données trop volumineuses et la requête l'utilisant n'est pas performante.
F5	Choix de tri dans le rapport ne répond pas, donc pas de rapport.	Espace mémoire insuffisant.
F6	Export d'un rapport ne fonctionne pas.	Le format de l'export ne supporte pas un nombre élevé de lignes dans le rapport : Exemple Excel au-delà de 65000 lignes n'accepte pas l'export.
F7	Drill-down / drill up ne fonctionnent pas.	Niveau de détail corrompu; Index corrompu;

No	Description du problème	Cause possible
		Espace mémoire insuffisant.
F8	Données incohérentes détails versus sommaire.	Tables d'agrégat et détail désynchronisées.
F9	Problème avec le cube, il ne se génère pas.	Espace disque insuffisant; Problème dans les données qui font que la génération du cube est corrompue.
F10	Problème avec le tableau de bord- exécution dure trop longtemps.	Le modèle de données n'est pas performant.
F11	Rapport déjà exécuté, ne s'exécute pas rapidement.	La cache de l'outil de reporting est pleine.

Le tableau précédent est très loin d'énumérer tous les cas possibles de problèmes de performance au niveau « *front-end* ». Il ne faut pas oublier que pour la restitution, tous les entrepôts de données sont équipés d'outils BI. Ceux-ci ont leurs propres limites, par conséquent, leurs propres problèmes de performance. Ces points seront considérés dans les solutions envisagées à résoudre certains problèmes énumérés et illustrés dans le chapitre suivant.

2.6 Résumé

Les problèmes de performance d'un DW sont très nombreux. L'étude détaillée est élaborée seulement pour ceux se trouvant du côté « *back-end* ». Ce n'est pas par manque d'intérêt, bien au contraire. Il faut insister sur le fait que la résolution de plusieurs de ces problèmes, solutionnent en même temps certains problèmes de performance rencontrés en aval du DW et

que le bon fonctionnement de l’ETC demeure essentiel au bon fonctionnement d’un DW.

Il est évident qu’un ETC bien géré, bien administré, bien surveillé, bien orchestré donnera des résultats plus performants, plus fiables, plus rapides et de bien meilleure qualité. De ce fait, les problèmes de performances du DW seront beaucoup moins fréquents mais surtout ils seront détectés, maîtrisés et même solutionnés avant qu’ils ne soient décelés par les usagers.

Chapitre 3

Solutions aux problèmes de performance d'un DW

Pour résoudre n'importe quel problème dans une situation donnée, il faut avant tout être « maître à bord ». Tout en étant le gouverneur du site, il faut aussi posséder des outils nécessaires pour appliquer la ou les solution(s) possible(s) envisagée(s).

En entrepôt de données, trois maîtres sont majeurs dans l'application des solutions visant à résoudre les problèmes de performance. Il s'agit du DBA, de l'administrateur ETC et de l'administrateur BI. Pour la plupart des problèmes de performance indiqués dans le chapitre précédent, on retrouve les solutions au niveau des processus de l'ETC (programme d'extraction, de chargement, de transformation, de calculs d'agrégats, des scripts de mise à jour) et d'autres au niveau de la BD de l'entrepôt. Il y a aussi des solutions au niveau des outils BI qu'il ne faut pas négliger. Il y a beaucoup d'initiatives qui doivent être prises au niveau des tables : gestion d'espace alloué, gestion du choix des index, choix des partitions, gestion de l'utilisation de la cache des requêtes (BD et outil BI), gestion de choix des vues matérialisées, gestion du suivi de l'évolution en termes de volume particulièrement pour les tables de fait, gestion de l'archivage ou compression des vieilles données, gestion de rafraichissement des environnements avec les données de la production. Par surcroît, les requêtes SQL sont toujours sujettes à une profonde analyse de performance. Le rôle et les responsabilisées du DBA sont capitaux. La collaboration, la communication et le professionnalisme sont déterminants dans la gestion de tous ces problèmes de performance.
L'outil ETC est vital pour un DW. Son administrateur a un rôle décisif dans l'application de solutions pour résoudre les problèmes de performance. Il est nécessaire qu'il soit doté des outils de monitoring automatique de l'ETC, lui facilitant les tâches et la supervision du comportement de l'ETC. En revanche, son implication, son expertise, son professionnalisme,

son sens de rigueur et sa collaboration demeurent primordiaux et indispensables, devant les problèmes de performance d'un DW. Généralement il a le plein pouvoir au niveau des programmes d'ETC et des modules de l'ordonnanceur des traitements de l'ETC, mais il n'a pas systématiquement les droits administrateurs de la BD lui octroyant le plein pouvoir d'agir.

Les outils BI représentent l'interface de communication entre les usagers et les données de l'entrepôt. C'est à travers ces outils que la restitution est possible pour les usagers de l'entreprise. Les données de plusieurs années et de différents domaines de l'entreprise sont questionnées et mises sous forme de rapports, tableaux de bord, requêtes ad hoc et cubes, à des fins d'analyse et d'aide à la décision. Évidement, ils sont substantiels et cruciaux pour un DW. Leur administrateur veille à leur utilisation optimale et performante.

Ce nouveau chapitre a pour objet de présenter certaines solutions aux problèmes identifiés précédemment. Plusieurs d'entre eux peuvent être résolus avec la même solution. Certaines solutions ont été testées. Les résultats sont remarquables. Les tests ont été faits sur des tables très volumineuses dont la volumétrie est donnée en annexe E. Par contre, d'autres solutions n'ont pas été mises en pratique et n'ont donc pas été testées, faute de moyens. Il s'agit en effet de solutions très coûteuses qui ne sont pas encore très répandues dans les entreprises. Elles auront le mérite d'être soulevées et analysées, afin de voir leur portée et les considérer ultérieurement si l'intérêt est suscité.

3.1 Facteurs clé de performance d'un DW

Certains facteurs sont déterminants pour l'étude de la performance d'un DW [8]. Il s'agit :

1) du chargement des données;
2) de la disponibilité des données;
3) du volume de données;
4) des utilisateurs d'affaires;
5) du volume des requêtes;

6) de la complexité des requêtes.

Ces facteurs servent d'indicateurs à l'évaluation de problèmes de performance de l'entrepôt. De cette façon, le gain en performance peut être mesuré en temps, espace ou nombre de fiches.

3.2 Caractéristiques de la BD pour l'amélioration de la performance

La gestion des entités de la BD comme index, vues, trigger, clé primaire, clé étrangère et les contraintes de colonnes, permet de gagner de la performance [8]. Ces points sont considérés dans certaines solutions.

Il existe plusieurs solutions pour les problèmes de performance d'un entrepôt de données. Certaines solutions qui ont fait l'objet d'étude dans cet essai ont été choisies par rapport aux différentes lectures qui ont servi de base. D'autres solutions proviennent de l'expérience sur le terrain.

Certaines solutions ont été soulevées dans le cadre des tests. Elles ont été testées et ont donné des résultats non négligeables. Les résultats des tests sont décrits dans les paragraphes suivants, les détails sont reportés dans les annexes A, D et E se trouvant à la fin du présent document. Il est à noter que certaines solutions n'ont pas fait l'objet de tests, car elles sont trop coûteuses. Mais elles seront certainement explorées dans le futur. Chaque solution majeure sera identifiée par le préfixe «S» pour solution, suivie d'un numéro séquentiel. Cette identification servira dans la matrice problème/solution qui est donnée à la fin de ce chapitre.

3.3 Description de la configuration de l'environnement des tests

Les tests ont été élaborés dans une BD Oracle 10g ayant la configuration suivante :

- Oracle Database 10g Enterprise Edition Release 10.2.0.3.0 - 64bi
- PL/SQL Release 10.2.0.3.0 - Production

- CORE 10.2.0.3.0 Production
- TNS for IBM/AIX RISC System/6000: Version 10.2.0.3.0 - Production
- NLSRTL Version 10.2.0.3.0 – Production

L'annexe D donne plus de détails au niveau du serveur, CPU, mémoire.

3.4 Création de tables de fait avec l'option NOLOGGING - Solution S1

Dans un entrepôt de données, les tables de gros volume sont très courantes. Il faut les créer avec l'option NOLOGGING. Ceci est très utile par exemple lors de la création d'une table à partir d'une autre table avec la commande : CREATE TABLE AS SELECT (CTAS), commande très utilisée au niveau de la création de vues matérialisées. Lorsque cette clause est fixée pour une table ou un index, il n'y a pas de génération de journal d'*undo* ou de *redo* (ce qui ne permet pas de retrouver les opérations effectuées sur la table). Les processus vont plus vite avec cette option puisque rien n'est journalisé. Cette option génère les gains suivants :

- Gain d'espace pour les fichiers de journalisation *redo* (*Redo Log Files*) qui sont des fichiers du SGBD permettant à la BD de garder une trace de toutes les altérations de données. Ainsi en cas de crash de la BD, ils permettent de rejouer les modifications qui lui ont été apportées. Ces fichiers doivent être au moins au nombre de deux et nécessitent une attention toute particulière, tant au niveau de la sauvegarde que de l'optimisation des accès;
- Gain de temps de création de la table surtout quand il y a plusieurs tables à créer en même temps;
- Amélioration de performance pour la création du parallèle de grosses tables, car le parallélisme utilise plusieurs processus en même temps. Sans cette option, toutes les opérations simultanées sont transcrites dans le fichier « redo log ».

L'exemple suivant montre le gain de temps dans la création des tables. Plus la table est

volumineuse, plus le gain de temps est considérable. Il est très intéressant de considérer ces exemples dans un cadre récurrent et ensuite de tenir compte de l'ensemble du temps gagné.

```
Ex.   CREATE TABLE SALES_LOGGING AS SELECT * FROM SALES.
        Table created. Elapsed: 00:00:25.24
Ex.   CREATE TABLE SALES_NOLOGGING AS SELECT * FROM SALES.
        Table created. Elapsed : 00:00:06.59
Pour seulement quelques 900 000 enregistrements,  la différence de temps est
18secondes.
```

Figure 3-1: Résultat de test option NOLOGING

3.5 Amélioration des requêtes SQL – Solution S2

Que la requête SQL soit utilisée dans un programme ETC, un script simple ou complexe, une requête ad hoc d'extraction de données entrepôt, une requête d'interrogation de données ou dans un outil BI, elle doit obligatoirement être analysée à des fins de performance. On n'insistera jamais assez sur cette solution très peu coûteuse mais très avantageuse. Une commande aussi simple qu'un SELECT, pourrait provoquer une lenteur considérable. Ce ne sont pas toutes les commandes SQL qui ont été relevées et testées. Il est clair qu'on ne peut pas toutes les considérer. Néanmoins, celles présentées dans les paragraphes suivants ont donné des résultats très intéressants. Il y en a certainement d'autres qui pourraient susciter le même intérêt.

3.6 Utilisation du plan de la requête – Solution S3

Pour chaque requête soumise, le SGBD définit un plan de la requête. Ce plan définit les étapes de l'exécution de la requête. Il trace comme une carte le chemin d'exécution de la requête avec toutes les étapes nécessaires évaluant le coût de chacune d'elles. Il permet d'avoir une vue d'ensemble sur la performance de la requête. Il est essentiel d'utiliser ce plan afin de détecter si la requête est performante. Il est presque exceptionnel qu'une requête moyennement complexe soit performante dès sa première version. Un plan de requête est donné en annexe B en vue d'illustrer les informations qu'il manipule.

3.7 Performance de la requête ou script SQL – Solution S4

Parmi les intervenants de l'équipe technique de l'entrepôt de données, l'expert des requêtes SQL joue un rôle fondamental dans les performances du DW. Il est très utile que ce profil fasse partie de l'équipe. Il aura entres autres, la charge de vérifier et de n'autoriser que les requêtes les plus performantes à mettre en production. Il apparaît anodin de mentionner cette facette de développement, mais l'analyse de la performance de la requête est nécessaire. Elle peut éviter bien des soucis de performance dans un futur parfois très proche. Mettre en production la requête diagnostiquée performante est une bonne pratique de développement, qui permet de diminuer les problèmes de performance du DW.

Dans ce paragraphe, on ne fait pas l'étude de toutes les requêtes ou du moins de toutes les commandes SQL afin de prouver l'amélioration de performance. Le choix est basé sur l'expérience avec les DW. Dans ce cadre, la liste suivante a été choisie.

3.7.1 Commande HINT APPEND dans la commande INSERT

En utilisant la façon d'insérer des enregistrements dans une table tel que décrit précédemment, avec la commande INSERT, on peut utiliser le HINT APPEND. Ceci va permettre une insertion à la fin de la table, d'où un gain de temps d'exécution.

3.7.2 Commande HASH JOIN

La jointure cartésienne dans des opérations de masse coûte trop chère en temps d'exécution. Elle donne des résultats mais parfois l'opération est lente. Par exemple, la jointure cartésienne avec une table vide peut durer indéfiniment. Le HASH JOIN, permet de l'éviter. Pour ce faire, on peut appliquer le HINT FULL sur les tables visées pour un HASH JOIN. Ce dernier est presque toujours utile et même inévitable quand il y a une jointure entre une table et une *Inline View* (SELECT dans un FROM).

3.7.3 Commande NOT EXISTS au lieu du NOT IN

NOT EXISTS est une commande beaucoup plus performante que la commande NOT IN. Il est plus intéressant de définir une variable qui contiendra le résultat du SELECT et l'utiliser dans le IN, car cela diminue le nombre d'accès à la BD.

3.7.4 Boucles imbriquées - NESTED LOOP

Dans le plan de la requête, si on voit que les boucles imbriquées qui sont exprimées par l'instruction NESTED LOOP sont utilisées, il faut revoir la requête pour les supprimer et les remplacer. Le NESTED LOOP est très inefficace.

3.7.5 Favorisation de l'usage BULK COLLECT dans un curseur

L'instruction BULK COLLECT permet de lire les données de la BD et de les mettre dans une collection. Pour les opérations de mise à jour (INSERT, UPDATE, DELETE), on utilise des pseudos code FORALL. Cette instruction permet d'améliorer énormément les processus de lecture et de mise à jour.

3.8 Gestion des index - Solution S5

Les index permettent un temps réponse rapide à une requête. La gestion des index intervient au niveau des processus ETC surtout au niveau des processus de chargement [11]. En entrepôt de données, il est conseillé de détruire l'index avant le chargement des données et de le rebâtir une fois le chargement complété.

Il est important de comprendre les types d'index qui existent et comment on les utilise. On sait qu'on définit un index sur un champ de la table. De ce fait, une table peut faire référence à plusieurs index. Il existe deux sortes d'index et il est important de comprendre la différente entre les deux :

1) Index Arbre B ou B-arbre *(Bee-tree* index ou *balanced tree)* : est un arbre équilibré. Il permet de stocker les données sous une forme ordonnée. Le principe est de permettre aux nœuds d'avoir plus d'une clé, ce qui minimise la taille de l'arbre et réduit le nombre opérations d'équilibrage. Mieux encore, un B-arbre grandit à partir de la racine, contrairement à un arbre binaire de recherche qui pousse à partir des feuilles. Les index stockent la valeur des clés contrairement à celle de la structure d'un arbre [11]. Ce type d'index est optimal pour les tables avec une forte cardinalité, c'est-à-dire une grande présence de valeurs distinctes d'un champ. Ces index sont efficaces dans le cadre des requêtes connues mais pas dans le cas de requête ad hoc comme dans un DW.

2) *Bitmap indexes* : cet index fonctionne complètement différemment du précédent. Il consiste en une table simple indiquant, pour chaque valeur possible du champ, la liste des enregistrements ayant la valeur du champ. Il est très efficace pour les champs dont la cardinalité est petite. Plusieurs colonnes simples d'index bitmap peuvent dynamiquement être jointes ensemble pour créer un index composé afin de supporter les requêtes ad hoc. De part leur flexibilité, c'est très pratique de créer un index bitmap simple sur chaque clé artificielle (*surrogate key*) des tables de fait dans un DW.

Cette solution comme plusieurs autres liées à la BD est du ressort du DBA. C'est le DBA qui doit surveiller et surtout accepter ou refuser la définition des index soumis à la BD.

3.9 Gestion des partitions des tables – Solution S6

Partitionner les tables de gros volume est une nécessité en entrepôt de données. Les partitions servent à constituer les clés de recherche de telles sortes à accélérer les recherches. Une démonstration très simple permet d'identifier un gain appréciable avec l'utilisation de la table partitionnée. Le partitionnement permet de découper une table et/ou un index sur un ou

plusieurs critères logiques. Ce(s) critère(s) définissent la clé pour la recherche d'information. Avec ce découpage la table se comporte comme plusieurs tables de plus petites tailles. Les avantages principaux du partitionnement sont :

- le pouvoir de définir des critères de stockage différents pour chacune des partitions (*tablespace*, *initial*);
- le découpage logique de la table permettant un accès plus rapide aux informations (moins de lectures disques à effectuer en particulier).

Il existe quatre grands types de partitionnement :

1) *By range*: on définit les partitions par tranche supérieure exclusive;
2) *By list* : on définit une valeur par partition;
3) *By hash* : la partition de stockage est calculée dynamiquement par un calcul de type hash code;
4) *Composite* : méthode hybride dont les données sont d'abord partitionnées *by range*, puis chaque partition est partitionnée *by hash* ou *by list*.

Le partitionnement d'une BD améliore les performances et simplifie sa maintenance surtout si elle est très volumineuse. Les requêtes qui n'accèdent qu'à une fraction des données peuvent être exécutées plus rapidement car la quantité de données à analyser est moindre.

3.10 Utilisation appropriée des tables partitionnées - Solution S6

On définit les partitions beaucoup plus sur des tables de fait. En général, les tables partitionnées font toujours partie des requêtes. Et dans presque tous les cas, un critère sur la date est utilisé, ce qui permet de voir les valeurs de l'identifiant (ID) de la date sur lesquelles les partitions sont fondées. Si les partitions ne sont pas bien utilisées, le SGBD va parcourir toute la table combien même des partitions ont été définies sur la table en question. Il faut imposer dans la requête les partitions pour que l'optimiseur du SGBD n'aille lire que les parties concernées. On peut clairement voir le gain de performance avec l'utilisation des partitions.

Le contrôle de ces partitions est une tâche qui relève du DBA. La définition des partitions évolue avec le temps. Par exemple, on définit des partitions par rapport à une donnée temporelle : année et mois. Ainsi en 2009 on va définir les partitions qui commencent de 200901 (janvier) au 200912 (décembre). En 2010, on aura besoin de définir les partitions relatives à l'année 2010. Ces partitions se retrouvent dans les tables de fait. Il est à noter que souvent toutes ces tables ne contiennent pas plus d'un mois de données par partition et parfois moins. Si les commandes SQL peuvent contenir un critère de sélection qui spécifie le champ de base de la partition, alors la commande SQL ne concerne que ces partitions sans toucher au reste des données. Si les scripts ne prennent pas avantage de la clef du partionnement, cela a pour effet de provoquer la lecture de toutes les partitions alors que seulement certaines partitions sont requises pour exécuter la requête. La lecture complète des partitions prend du temps et réduit la performance.

3.11 Utilisation du parallélisme - Solution S7

Les temps de réponse pour les opérations sur les grandes bases de données peuvent être drastiquement réduits grâce à l'utilisation de l'exécution d'opérations en parallèle. (*Parallel execution* ou *PX*). Ce mode d'exécution est notamment avantageux dans les systèmes dédiés aux entrepôts de données. Il permet un gain de temps de traitement. Il peut être utilisé à l'intérieur d'un programme ou tout simplement dans une requête ad hoc. À tout instant, il est utile de voir les possibilités d'utilisation de cette technique, vu son bénéfice. Un traitement d'opérations en parallèle, entraîne plusieurs opérations qui s'exécutent simultanément. Par conséquent, il faut une architecture multiprocesseur. En effet, dans une architecture monoprocesseur, on ne peut exécuter qu'une opération à la fois, contrairement à l'architecture multiprocesseur, comme montré dans la figure 3-2, dans laquelle on voit que l'opération 2 et l'opération 2' s'effectuent en parallèle. Plusieurs processeurs peuvent travailler ensemble pour optimiser l'enchaînement d'un ensemble d'opérations éventuellement interdépendantes dans le cadre d'une même requête. Du fait de l'interdépendance, les opérations ne peuvent pas s'exécuter totalement en parallèle et elles sont synchronisées entre elles par un mécanisme de

pipelining. Ce mécanisme consiste en une méthode de transfert des résultats d'une opération à une autre, aussitôt que ceux-ci sont disponibles et non pas seulement à la fin de la première opération. De cette manière, un processeur peut travailler sur des données qui lui sont transmises par un autre processeur.

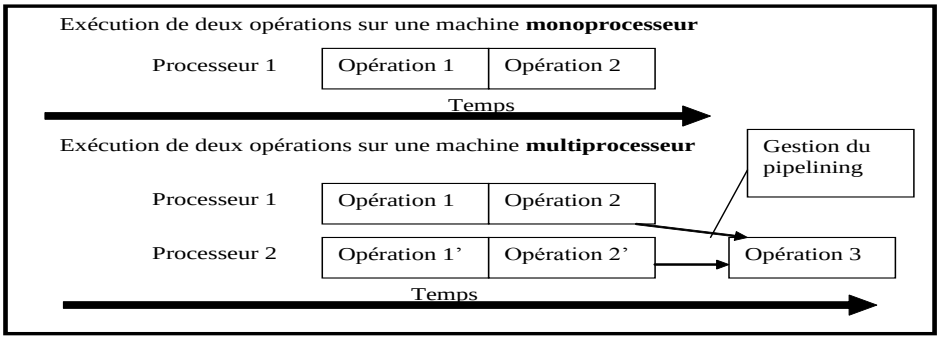

Figure 3-2: Traitement parallèle versus non parallèle et traitement intra-requête

3.11.1 Description de l'exécution en parallèle

Les systèmes bénéficiant de traitement d'exécution en parallèle devront en outre avoir une ou plusieurs des caractéristiques suivantes :

- Avoir suffisamment de bande passante (Entrée/Sortie);
- Être exposé à une sous utilisation des unités centrales de traitement *CPU* (*Central Processing Unit*) (typiquement dans les systèmes où l'utilisation est moins de 30 %);
- Posséder suffisamment de mémoire pour supporter des processus en plus, tels que des tris (*sort*), calcul d'agrégat, hachage (*hashing)* et tampons E/S.

De plus, il existe plusieurs types d'architecture de systèmes multiprocesseurs bénéficiant du principe d'exécution en parallèle : SMP (*Symmetric Multi-Processors*), MP (*Multi-Processing*), MPP (*Massively Parallel Systems*). La différence se situe essentiellement au niveau du partage de la mémoire. Il n'est pas nécessaire de décrire chacun de ces types pour démontrer le gain de performance avec le traitement en parallèle. Pour l'une ou l'autre des architectures, on peut paralléliser des requêtes grâce aux CPU mais aussi grâce aux disques.

Les SGBD dans un environnement parallèle de type décisionnel sont confrontés à différents types d'opérations :

- Requêtes complexes en accès en lecture sur de gros volumes de données;
- Opérations de calcul lourd;
- Opérations de maintenance (chargement, sauvegarde, reconstruction d'index, calcul d'agrégats).

L'objectif du SGBD est de faire en sorte que les opérations soient partagées entre les ressources du système. Pour ce faire, il doit considérer les répartitions suivantes :

1) Partager les utilisateurs sur les processeurs pour ne pas les pénaliser mutuellement;
2) Partager les données sur plusieurs processeurs pour effectuer des traitements en parallèle.

À des fins d'illustration, on considère un système simplifié où on peut affecter un processeur à une tâche et donc à un utilisateur. Dans un environnement décisionnel, si le nombre d'utilisateurs est faible, alors l'utilisation des processeurs n'est pas optimale, car il y aura des processeurs non utilisés. Pour contrer cette lacune, il s'agit de découper les tâches ou sous-tâches, ce qui revient à découper une requête en opérations plus élémentaires et à les répartir sur les différents processeurs. L'optimiseur de la BD sera chargé de cette décomposition. Les opérations, comme la commande de sélection (SELECT), sont divisées en autant de tâches parallèles que possibles, chacune traitant un fragment de la table. Les opérations plus complexes (impliquant plusieurs tables), comme les jointures, demandent des algorithmes de jointure parallèle par hachage en général, car ils favorisent le traitement en parallèle.
L'annexe F montre plus de détails sur le fonctionnement en parallèle.

3.11.2 Résultat de tests avec le parallélisme

D'après les paragraphes précédents, la mise en œuvre de l'option du parallélisme dans la BD augmenterait la performance des requêtes SQL de façon évidente. Les résultats suivants, tirés du site Internet d'InifniDB (www.infinidb.org), montrent clairement que plus le degré de

parallélisme est élevé, plus la performance est améliorée. Le temps d'exécution des requêtes diminue presque de 50 %. Les tests ont été élaborés sur des tables de fait de 6 milliards de fiches et des tables de dimensions de 30 millions de fiches avec des modules de performance de 1, 2 et 4.

On a utilisé 13 requêtes SQL dont la description des scripts se trouve en annexe A. Chaque requête a été définie avec trois degrés de parallèle (1PM, 2PMs, 4PMs). Le résultat est le temps d'exécution en secondes reporté dans le tableau 3.1.

Tableau 3-1: Données et résultats de test de parallélisme

N°	Requête	1PM	2PMs	4PMs
1	Q1.1	81	41,03	22,11
2	Q1.2	7,91	4,15	2,46
3	Q1.3	50,73	25,43	16,06
4	Q2.1	482,25	247,68	129,15
5	Q2.2	385,21	200,88	103,3
6	Q2.3	354,32	180,51	96,78
7	Q3.1	830,53	432,64	228,84
8	Q3.2	548,38	286,23	150,81
9	Q3.3	472,07	216,29	113,29
10	Q3.4	24,48	15,87	8,49
11	Q4.1	999,43	519,98	280,04
12	Q4.2	255,01	130,94	57,27
13	Q4.3	158,44	82,65	26,35

On voit d'après le tableau ci-dessus que le temps d'exécution varie beaucoup de la requête la moins performante avec 830.53 secondes à la plus performante avec 2.46 secondes. Le graphique suivant permet de mieux visualiser le gain en performance.

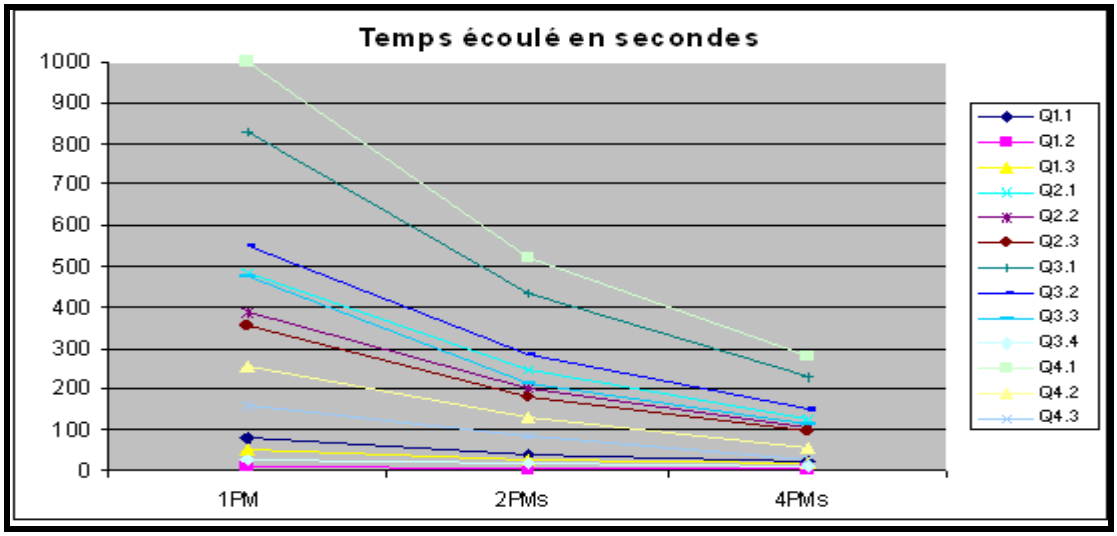

Figure 3-3: Graphique des résultats de tests de parallélisme

Traduction libre
Source : http://infinidb.org/infinidb-blog/mysql-parallel-query-processing-of-ssb-queries-via-infinidb-.html consulté le 11 février 2010.

Le graphique montre que la requête Q1.2 est la plus rapide. On voit que la même requête gagne de la performance à mesure qu'on augmente le degré de parallélisme. Ainsi, la requête Q4.1 la plus lente au départ a gagné 72 % en temps d'exécution avec le degré 4. De manière générale, plus on utilise le parallélisme, plus on gagne de la performance.

3.12 Utilisation de la cache des requêtes - Solution S8

La cache est une mémoire tampon à accès très rapide, située entre le microprocesseur et la mémoire vive (RAM). Elle sert à stocker les instructions et les données les plus fréquemment utilisées et auxquelles un système fait le plus souvent appel, permettant ainsi de réduire le temps d'attente du microprocesseur. Il est suggéré d'améliorer la performance avec l'utilisation de la cache des requêtes [13]. Celle-ci contient l'historique de récentes requêtes ad hoc et leurs réponses. La cache est maintenue au niveau du DW et contient le couple (Q, A_q) où Q représente l'énoncé de la requête et A_q sa réponse. Quand un usager soumet une requête, la cache est d'abord examinée pour détecter si une requête semblable a déjà été soumise. Ceci

permet la réutilisation de requêtes récemment évaluées et permet ainsi de gagner du temps de réévaluation. Si la requête est déjà contenue dans la cache, le temps de réponse est instantané comparativement à la requête originale.

Par exemple, on va considérer deux tables du DW. La table Produits avec les champs suivants : (CodeProd, Desc, Classe) et la table Ventes avec les champs suivants : (CodeProd, MtVte, MtTaxe). L'usager désire avoir le montant des ventes du produit 123 et soumet la requête Q1 telle que montrée dans la figure 3-4. Une fois soumise, la requête est évaluée et la réponse est transmise à l'usager, puis l'énoncé de la requête ainsi que son résultat sont enregistrés dans la cache sous forme de couple (Q1, A_{q1}). On suppose qu'un autre usager désire avoir les ventes du produit 123 pour le montant des ventes supérieur à 100. Ce dernier va soumettre la requête Q2 de la figure 3.4

Dans un DW ordinaire, la requête Q2 sera évaluée et va utiliser le même temps que pour l'évaluation de la requête Q1 déjà exécutée. Avec le système de cache, le système va parcourir la cache et identifie la requête Q1 comme étant une requête ensembliste. La requête Q2 sera modifiée en Q2' de la figure 3.4.

```
SELECT CODPROD,  DESC, MTVTE
FROM PRODUITS P
          , VENTES V
 WHERE   P.CODEPROD=V.CODEPROD
      AND P.CODPROD=123;
```

Test Q1 Cache de la requête

```
SELECT CODPROD, DESC, MTVTE
FROM PRODUITS P
          , VENTES V
 WHERE   P.CODEPROD=V.CODEPROD
      AND P.CODPROD=123
      AND P.MTVTE > 100;
```

Test Q2 Cache de la requête

```
SELECT CODPROD, DESC, MTVTE
FROM Aq1
WHERE   P.MTVTE > 100;
```

Test Q2' Cache de la requête

Figure 3-4: Tests Cache de la requête

Il pourrait y avoir d'autres requêtes qui utiliseront les résultats de la requêtes Q2' et ainsi de suite. Ce processus est géré par des algorithmes mathématiques. La performance du système de cache dépend de la taille (le nombre de requêtes), de l'organisation de la cache et de l'algorithme d'unification qui vérifie si une requête est une sous-requête d'une autre. Il n'est pas nécessaire que toutes les requêtes soient sauvegardées dans la cache. La première est toujours mise dans la cache. Les suivantes seront comparées à la première et l'algorithme

décide lesquelles garder au fur et à mesure. Le gain de performance est dans le temps de réponse.

3.13 Utilisation des vues matérialisées - Solution S9

Une vue matérialisée est une entité utilisée dans un SGBD de type relationnel. C'est une table virtuelle représentant le résultat d'une requête sur la BD. Comme son nom l'indique et à la différence d'une vue standard, dans une vue matérialisée les données sont dupliquées. Une vue matérialisée diffère d'une vue standard par le fait qu'elle a une fréquence automatique de rafraichissement. De ce fait, son exécution ne requiert pas l'interrogation des données et donc le temps d'exécution est beaucoup moindre que l'exécution de la vue standard qui exige une interrogation des données à chaque fois. On utilise la vue matérialisée essentiellement à des fins d'optimisation et de performance dans le cas où la requête associée est particulièrement complexe et lourde. La fraîcheur des données de cette vue dépend des options choisies lors de sa création. Le décalage entre les données de la table maître et la vue matérialisée peut être nul (rafraichissement synchrone) ou d'une durée planifiée : heure, jour. Suivant le contexte, il existe différent types de vues matérialisées : sur clé primaire, *rowid* étant l'identifiant unique (numéro interne au SGBD) de l'élément dans la table, avec fonction d'agrégation, sous requêtes, jointures.

À des fins d'amélioration de performance, l'utilisation du *fast refresh,* est préconisé plutôt que les rafraichissements complets. Bien que séduisants, ces objets devront être utilisés avec parcimonie et à bon escient. Il pourra être utile de faire des tests de *refresh* (*complete* et *fast*), de bien peser les volumétries, de doser les fréquences de *refresh* en fonction des besoins réels.

3.14 Compression des données - Solution S10

La vocation première d'un DW est de sauvegarder les données des différents domaines, secteurs, régions de l'entreprise. Cette sauvegarde constitue l'historique des données. Mais après des années, on constate que les usagers du DW n'interrogent que les données à partir

d'une certaine année. Par exemple, pour la vente au détail, les usagers ne font peut-être aucune analyse au-delà de sept ans. Si l'entrepôt est plus vieux, il est recommandé d'archiver les données antérieures. Ainsi, les tables de fait diminueront considérablement de volume, ce qui augmente la performance de l'entrepôt. Cette solution repose sur le besoin d'affaires. Les DBA ne peuvent pas prendre une telle décision sans qu'elle ne soit approuvée par le propriétaire de l'entrepôt. Mais c'est une solution qu'il faut considérer lors de l'étude de problèmes de performance d'un DW.

3.15 Utilisation de BD chargement en colonnes - Solution S11

Le SGBD classique est connu avec l'appellation *row-oriented* « orienté ligne ». Il s'agit du cas où l'enregistrement dans la table se fait par ligne.
Le DBMS orienté colonne (*Columnar DB*) est un système de gestion de base de données qui sauvegarde le contenu par colonne au lieu de le faire par ligne. Ceci a un avantage énorme pour les BD entrepôts de données, où les agrégations sont calculées à travers un très grand nombre de données similaires [12].

La description du fonctionnement d'un SGBD orienté colonne est donnée ci-dessous à travers un exemple très simple.

Tableau 3-2: Données pour exemple de BD orientée colonnes

IDempl	Pren	Nom	Dept	Âge
1	David	Smith	TI	35
2	Marie	St-Paul	Finances	28
3	Erik	Martin	RH	40

Le tableau 3-2 inclut un identificateur employé (IDempl), le prénom et le nom de l'employé (Pren et Nom), le département (Dept) et l'âge. Le SGBD accède aux données comme montré

dans la figure 3-5.

1, 2,3, David, Marie, Erik, Smith, St-Paul, Martin, TI, Finances, RH, 35, 28,40

Figure 3-5: Exemple accès aux données avec Columnar DB

Il s'agit donc des cellules d'une même colonne. Cela permet une meilleure compression et aussi d'aller plus vite lors d'une lecture séquentielle de la table sur une colonne (agrégation par exemple).

Les bases de données « *column-based* » exigent aussi une définition préalable des besoins des utilisateurs et une préconception physique et logique des DW RDBMS. Elles impliquent un autre compromis : l'optimisation de l'insertion de nouveaux enregistrements et la sélection/récupération des données. Cependant, elles utilisent moins d'espace de stockage et sont plus efficaces dans leurs exigences entrées/sorties par rapport aux DW RDBMS.

3.16 Technologie DW monofonctionnel - Solution S12

Le DW monofonctionnel (*Data warehouse appliances*) [5], comporte un set de composantes intégrées : serveur, mémoire, système d'exploitation, base de données et des connections spécialisées préconfigurées et réglées pour la rigueur du DW. C'est un ensemble de matériels (contrôleur, SGBD) et de logiciels dédiés particulièrement aux entrepôts de données et conçus pour être plus performants pour des analyses.

Bien que les BD relationnelles, axées sur les enregistrements, soient (pour le moment) le choix le plus courant en matière d'entrepôt de données, leur structure est loin d'être parfaite pour traiter les entrepôts. Les SGBD relationnels axés sur les enregistrements sont difficiles à concevoir, extrêmement inefficaces dans leur utilisation de l'espace de stockage et des entrées sorties (E/S), difficiles à maintenir et obligent les concepteurs à faire un compromis entre optimiser la performance des requêtes et offrir une flexibilité de requêtage maximale.

Les appliances[9] décisionnelles ont été développés pour répondre à ces carences. Ce sont des dispositifs de bases de données relationnelles (RDBMS) livrés sur du matériel à configuration personnalisée. Ils ont les mêmes points forts et faibles que les RDBMS, à la différence que, l'ensemble matériel/logiciel est optimisé pour offrir des requêtes plus rapides pour un coût total moindre. Ils nécessitent toujours toute la planification, la conception et la gestion requises par les bases de données relationnelles axées sur les enregistrements et la flexibilité des requêtes est toujours limitée par l'accès aux données SQL.

Les appliances décisionnelles se concentrent aujourd'hui sur la haute disponibilité, la facilité d'utilisation et les faibles coûts d'acquisition, mais ont encore du chemin à faire en ce qui concerne leur intégration au système d'information des entreprises et les niveaux de service fournis.

3.17 Technologie de l'accélérateur de requêtes - Solution S13

Les BD spécialisées permettent l'amélioration de la performance, car elles sont dédiées à l'entrepôt de données. À l'heure actuelle [14], il y a très peu d'entreprises qui ont déployé cette solution comparativement à la solution précédente (S12). Les *query accelerators* sont plus spécialisés que les BD spécialisées et sont donc moins connus. Pour que cette solution soit achetée par une entreprise, il faut que cette dernière soit à un degré de maturité de BI élevé. Son entrepôt de données doit être utilisé à sa pleine capacité et doit être vital à la compagnie. C'est une nouvelle solution qui n'a pas fait ses preuves et est donc coûteuse.
Les accélérateurs de requêtes peuvent adresser des défis de performance bien spécifiques. Par exemple, l'étude [14] a constaté que les plus hauts pourcentages d'organisations utilisant ou projetant de déployer cette technologie pour la planification d'affaire et la prévision (46 %), des finances (44 %) et le rapport de gestion exécutif et l'analyse (44 %).

[9] Applications mais on utilise souvent le mot appliances

3.18 Technologie BD spécialisées – Solution S14

Comme pour la solution d'accélérateur de requêtes, très peu d'entreprises utilisent la technologie de BD spécialisées car elles sont encore peu connues sur le marché. Cette technologie œuvre directement dans la performance de la BD dans sa globalité.

3.19 Utilisation d'outil de surveillance de l'ETC - Solution S15

De tout ce qu'on fait, le contrôle est la clé du succès. L'absence de contrôle amène inévitablement le risque et le risque entraîne l'erreur. Si on ne sait pas d'où provient une erreur, il est fort probable qu'elle soit du côté des éléments dont on n'a pas le contrôle. Le meilleur système est celui qui laisse le moins de place au risque. L'ETC, tel qu'il a été décrit dans les chapitres précédents, est le système critique d'un entrepôt de données. Tout ce qui le concerne est sujet à contrôle, sans oublier que le contrôle sans action n'est d'aucune utilité. Par conséquent, il faut contrôler et encore contrôler et toujours contrôler l'ETC. Un ETC qui n'a pas d'outil de monitoring est voué à l'échec et sa survie est très courte. Il est nécessaire de le doter d'un outil permettant de le surveiller. Il en ressort que la qualité de données est essentielle et la tâche primordiale à prendre en considération est : comment revenir en arrière si l'ETC plante!

D'autres tâches subséquentes sont toutes aussi importantes : comment revenir à l'état initial si les données semblent incohérentes ? Comment valider les transformations?

Tout entrepôt de données doit faire l'objet d'un suivi de la performance de la chaîne d'alimentation, car celle-ci doit illustrer de façon claire et précise les indicateurs de temps de progression des flux de données durant les phases d'acquisition de données. Via l'outil ETC, l'administrateur ETC est en mesure d'anticiper efficacement l'évolution de la performance de l'alimentation de données. De plus, il peut prévenir un éventuel goulot d'étranglement.

L’équipe entrepôt de données devrait toujours mettre en place des fonctionnalités de suivi statistique et de rapports de performance afin de disposer d’un support qui rend compte des tendances et évolutions (améliorations, dégradations) de la performance de l’entrepôt. Des indicateurs de performance clairement définis de suivi de l'évolution de la chaîne d'alimentation, tels qu’un suivi du temps de traitement, des requêtes ou des scripts consommateurs de temps et de ressources, l’identification des flux en retard, les volumes de données traitées et la latence entre les différentes étapes du processus d’acquisition, permettraient à l’équipe de cibler et de prioriser les optimisations à effectuer continuellement. Un outil simple et accessible permettant d’être proactif aux nombreux problèmes de performance d’un DW.

3.20 Utilisation d’un outil de monitoring de la BD - Solution S16

Il est inconcevable de gérer un SGBD sans outil de monitoring surtout pour une BD entrepôt de données. Son but est de pouvoir suivre le comportement de la BD et d’interpréter ou d’expliquer la présence de certains bugs ou problèmes y compris ceux de performance. Il joue un rôle essentiel dans la détection des problèmes de performance. L’outil de monitoring a la prérogative de garder l’historique du comportement quotidien de la BD. Il offre la possibilité d’analyser en tout temps la performance d’un script, requête ainsi que la volumétrie des tables, index.

Pour assurer un bon fonctionnement d'une base de données en général et un entrepôt en particulier, il est nécessaire d'utiliser des outils de surveillance du comportement du système afin d'effectuer d'éventuelles corrections proactives. L’annexe I donne un aperçu du fonctionnement de l’outil de monitoring de la BD et comment son utilisation peut aider à la performance d’un entrepôt de données.

3.21 Mécanisme de reprise d’une chaîne ETC - Solution S17

Il est indispensable de prévoir le mécanisme de reprise lors de l’interruption de la chaîne ETC.

Il est prévu que la chaîne ETC s'exécute à une fréquence périodique : quotidienne, hebdomadaire, horaire, aux 15 minutes. Il est aussi prévu que certains modules s'exécutent séquentiellement alors que d'autres peuvent l'être en parallèle. Cependant, lors d'une rupture de l'exécution de n'importe quel module, à n'importe quelle étape (extraction, transformation ou chargement), la reprise peut être très fastidieuse. Il est essentiel de savoir quel module redémarrer, avec quelles données et surtout à quel moment. Parfois, il est plus bénéfique de reprendre l'extraction depuis le début par exemple. À un autre moment, il est plus avantageux de ne reprendre que le module de calculs des agrégations, le tout à partir du planificateur utilisé pour les modules de la chaine ETC. Dans certaines situations, on peut même procéder à une reprise manuelle, dans le sens où on fait exécuter un module sans utiliser de planificateur. Chaque situation est propre au processus de l'ETC tel qu'il a été défini et maintenu avec le temps. Pour comprendre comment se fait ou doit se faire la reprise, il importe de comprendre parfaitement ce que fait chaque module et le lien entre les modules. Par exemple, le module de chargement des dimensions est traité avant celui des faits, ce dernier est lié avec le précédent et doit se trouver à un niveau inférieur. Idéalement, un ETC est doté d'un rapport statistique.

Il arrive parfois que la chaîne ETC ait été interrompue par un problème quelconque. Il arrive aussi qu'elle soit exécutée dans le vide en attente d'un module verrouillé pour des raisons variables. Il arrive pour des raisons particulières qu'on ait à l'interrompre volontairement de façon manuelle. Il est essentiel qu'on ait un mécanisme de reprise automatique. On doit être en mesure de reprendre la chaîne à tout niveau de son interruption sans aucune perte ni duplication de données. On peut citer l'exemple de chargement des dimensions qui a été fait à moitié. Le mécanisme doit être en mesure de ne pas entamer le chargement de faits tant que celui des dimensions n'a pas été complété.

Savoir ce qu'il faut faire et quand il faut le faire, lorsque le déroulement de la chaîne a été brisé, aide énormément pour la performance du DW. Ainsi, les données seront restituées aux usagers plus rapidement. La qualité des données utilisées est plus fiable ce qui augmente le degré de confiance des usagers quant à la manipulation des données du DW.

3.22 Mécanisme de validation des données - Solution S18

Tout entrepôt de données doit être muni d'un mécanisme de validation de données. La validation concerne l'intégrité, la cohérence, la pertinence, la «propreté» des données pour palier la faiblesse relative de l'étape de filtrage et de nettoyage des données issues des systèmes opérationnels sources. Interroger des données entièrement de qualité et valides prendrait moins de temps puisqu'elles incluent des données incorrectes. Un entrepôt de données est par définition un stockage de plusieurs années de données. Par conséquent, si les données ne sont pas correctes depuis longtemps, on peut avoir accumulé des volumes considérables de données invalides.

Un DW fait face à des données gigantesques hétérogènes quotidiennement. Un mécanisme de validation des données s'impose. Sans la présence d'un dispositif qui vérifie ce qu'on charge, les données peuvent facilement devenir insensées. Par exemple, dans le cas de la vente, on a saisi au niveau du système opérationnel, le CUP (Code Universel de Produit), sur 9 chiffres (989789560) dans le champ de la quantité vendue. Cette valeur erronée a été acheminée au DW. À partir de cette erreur, le processus de validation mis en place utilise une règle validant le maximum permis pour chaque champ. Le besoin d'affaire pourrait exiger de recharger les données après ressaisie du système opérationnel, ce qui accuserait un retard dans la disponibilité des données quotidiennes donc affecterait la performance. De plus, il y a des chargements qui ne pourront pas être repris comme celui des inventaires, car il s'agit d'une image du jour.

3.23 Analyse de performances de l'outil BI - Solution S19

Tous les outils BI ont de nouvelles versions périodiquement. La performance au niveau de l'entrepôt de données est une préoccupation majeure pour les concepteurs BI. D'une version à l'autre, plusieurs caractéristiques sont mises à jour dans le but d'améliorer la performance. Il est pertinent de rester à l'affût de nouvelles versions de l'outil BI utilisé au niveau de

l'entreprise. Le rôle de l'administrateur BI est essentiel. Il lui revient de rester informé par la lecture, la documentation à jour, les forums, les conférences, les formations. de l'évolution de l'outil BI qu'il administre. Par exemple, lorsqu'on utilise pour les rapports et tableaux de bord, l'outil Microstrategy version 8 (www.microstrategy.com), les usagers se plaignent de ne pouvoir exporter certains rapports vers Excel ou un fichier texte. Il s'agit des rapports énormes de quelques millions de lignes. Ce point est réglé dans la version 9.

Les outils BI ont leurs propres limites et sont voués aux problèmes de performance au même titre que tous les logiciels. Il est nécessaire de connaître les caractéristiques, la configuration, en fait les points fondamentaux permettant l'optimisation de la performance de l'outil. D'une version à l'autre, les producteurs de ces outils annoncent une amélioration. La performance n'est pas de mise à chaque version.

L'annexe G présente des points généraux pour les problèmes de performance d'un entrepôt.

3.24 Matrice problèmes –solutions

Il est très intéressant de pouvoir reprendre les problèmes traités dans le deuxième chapitre et de les classer avec les solutions étudiées dans ce chapitre. La matrice de l'annexe H montre clairement quel problème peut être résolu avec quelle solution.

3.25 Résumé

La même solution peut-être utilisée pour résoudre plusieurs problèmes différents. L'étude n'a pas permis de déceler une solution parfaite qui arrangerait tous les problèmes de performances en même temps. Plusieurs solutions ne sont pas mentionnées par la limitation de pages dans la rédaction de l'essai. À titre d'exemple, une bonne stratégie de statistiques de la BD est un point qui a été étudié mais non cité dans les solutions. Dans la pratique, on a vu la performance du DW affaiblie à cause des statistiques de la BD.

Même si une solution règle un problème bien identifié, dans la pratique, il revient à considérer du cas par cas. Par exemple, l'utilisation des index est une solution qui de prime abord permet

d’accélérer les requêtes, mais dans certains cas où on cherche plus de 10 % des données, le *full scan* est plus performant.

Plusieurs solutions proposées peuvent être utilisées dans l’immédiat et ne requièrent pas un coût supplémentaire. Par contre, d’autres sont coûteuses. Cependant, de bonnes pratiques de développement et de vigie sont à la portée de tous.

Conclusion

Les solutions apportées sont loin d'être exhaustives. Du point de vue général, on ne finira jamais d'améliorer la performance d'un DW. L'informatique évolue sans arrêt. La vitesse et les capacités des ordinateurs continuent de doubler tous les dix-huit mois et les infrastructures se modernisent activement pour supporter les communications à grande vitesse. Le rythme soutenu de cette révolution de l'information se répercute en profondeur sur l'évolution des entrepôts de données. C'est un environnement qui est et demeurera constamment en évolution, en termes de volumes de données à traiter, d'infrastructure, d'ajout de nouvelles entités physiques, d'ajout de nouveaux types de connexions, d'ajout de nouvelles applications. Néanmoins, si la gestion de performance demeure une préoccupation quotidienne conjointe, des responsables clé du DW l'administrateur ETC et les DBA, les problèmes de performance seront beaucoup moins apparents et certainement mieux gérés et plus rapidement pris en charge.

Il faut retenir, que l'ETC est le cœur d'un entrepôt de données. Il est régi par des normes, par la gouvernance, par la maintenance de son architecture, de son design, de son modèle et de son évolution. Le pilotage de l'ETC est l'une des clés de succès pour tout l'entrepôt. Les actions à entreprendre dans tous les cas d'échecs de l'ETC doivent être documentées au fur et à mesure que chaque problème est identifié et solutionné et ceci durant toute la vie de l'entrepôt.

La conception d'un DW nécessite une équipe technique composée de professionnels spécialistes multidisciplinaire expérimentée. Certes, ce sont des juniors qui un jour deviennent des seniors avec une expertise dans un domaine ou un autre. Cependant, dans un DW, il est essentiel d'avoir des séniors qui ont la responsabilité de gestion, de monitoring et de l'évolution du DW. Les juniors peuvent y participer mais sous encadrement.

Suite à l'expérience vécue dans trois entrepôts de données différents et à l'étude établie dans le cadre de cet essai, il en ressort qu'il est urgent de prendre les problèmes de performance du DW au sérieux. On a bien vu que certaines solutions sont d'ordre pratique et à la portée de tous et ne coûtent pas trop cher. Elles sont applicables dans l'immédiat, comme par exemple, celle de soumettre les scripts et requêtes SQL à un expert à des fins de performance (solution S4) ou celle d'utiliser des vues matérialisées (solution S9).

Le coût des solutions décrites précédemment varie énormément. Des solutions S1 à S10 et S16 sont du ressort du DBA et sont donc peu coûteuses. Le DBA fait toujours partie de l'équipe technique DW et ne coûte pas plus cher en prenant en charge ces solutions. Les solutions S15, S17 et S18 sont la charge de l'administrateur ETC, elles sont aussi accessibles en prenant la bonne pratique de les inclure dans ses tâches. En revanche, les solutions S11 à S14 sont chères car ne sont pas encore très répandues dans les entreprises.

Il reste encore beaucoup à faire pour les performances d'un DW. Des solutions directement orientées entrepôt de données commencent à apparaître et sont très prometteuses. Elles sont conçues du point de vue matériel et logiciel pour les DW, comme la solution de BD en colonnes (solution S11) ou encore celle des appliances décisionnelles (solution S12). Elles sont à vocation entrepôt données et leur conception a pris en charge les problèmes de performance. Elles restent encore trop coûteuses pour la majorité des entreprises.

Liste de références

[1] Armstrong, R., *Seven steps to optimizing data warehouse performance*, Computer, vol. 34, n° 12, December 2001, p. 76-79.

[2] Beulke, D., *Data Warehouse Performance*, 27th Annual International Conference The Computer Measurement Group Inc., 8-13 December 2002, p. 293-306.

[3] Chenoweth, T., et Corral, K., et Demirkan, H., *Seven Key Interventions for DATA WAREHOUSE SUCCESS*, Communications of the ACM, vol. 49, n° 1, January 2006, p. 114-119.

[4] Cajolet-Laganière, H., et Collinge, P., et Laganière, G., *Rédaction technique, administrative et scientifique*, 3e éd., Édition Laganière enr., Sherbrooke (Québec), 1999, 468p.

[5] Dijcks, J.P., et Krishnan, K., *Achieving High Performance in Your Data Warehouse with Data Warehouse Appliances*, http://download.101com.com/tdwi/Webinar/AchievingHighPerformance/index.html, February, 11th, 2009.

[6] Dictionnaire en ligne mediadicco, http://www.mediadico.com/dictionnaire, 02 février 2010.

[7] Jambu, M., *Introduction au Data Mining. Analyse intelligente des données*, Paris Eyrolles, Paris, 1999, 144p.

[8] Kimball, R., *The data warehouse ETL toolkit*, Willey, Indianapolis, 2004, 491p.

[9] Kimball, R., Le *data warehouse : Guide de conduite de projet*, Paris Eyrolles, Paris, 2005, 576p.

[10] Office de la langue française le grand dictionnaire de terminologie, http://granddictionnaire.com, 15 janvier 2010.

[11] Poolet, M.A., *Indexing the data warehouse: optimize performance by indexing dimensions and fact tables*, SQL Server Magazine, 10.8, 2008, p45(2).

[12] Russsom, P., *Columnar Databases - Designed for Data Warehouse High Performance*, http://download.101com.com/tdwi/Webinar/ColumnarDatabases/index.html, November, 19th, 2008.

[13] Saharia, A.N. et Babad, Y.M., *Enhancing data warehouse performance through query caching*, ACM SIGMIS Database, vol. 31, n° 3, June 2000, p. 43-63.

[14] Stodder, D., *Five steps to optimizing BI and Data Warehouse Performance*, Intelligent Enterprise, Vol. 11, Issue 10, October 2008, p6-10. *http://www.intelligententerprise.com/showArticle.jhtml;jsessionid=EBHCXHXFKBP1GQSNDLRSKHSCJUNN2JVN?articleID=211201854*, October, 20th, 2008.

Annexe A

Scripts des requêtes de test de parallélisme

Les requêtes ayant servies au test de parallélisme sont simples. Elles se concentrent sur la commande SELECT avec la fonction SUM.

Tableau 3-3: Scripts des requêtes du test de parallélisme

Nom de la requête	Scripts
Q1.1	select sum(lo_extendedprice*lo_discount) as revenue from dateinfo, lineorder where lo_orderdate = d_datekey and d_year = 1993 and lo_orderdate between 19930101 and 19931231 and lo_discount between 1 and 3 and lo_quantity < 25;
Q1.2	select sum(lo_extendedprice*lo_discount) as revenue from dateinfo, lineorder where lo_orderdate = d_datekey and d_yearmonth = 'Jan1994' and lo_orderdate between 19940101 and 19940131 and lo_discount between 4 and 6 and lo_quantity between 26 and 35;

Nom de la requête	Scripts
Q1.3	select sum(lo_extendedprice*lo_discount) as revenue from dateinfo, lineorder where lo_orderdate = d_datekey and d_weeknuminyear = 6 and d_year = 1994 and lo_orderdate between 19940101 and 19941231 and lo_discount between 5 and 7 and lo_quantity between 26 and 35;
Q2.1	select sum(lo_revenue), d_year, p_brand1 from part, supplier, dateinfo, lineorder where lo_orderdate = d_datekey and lo_partkey = p_partkey and lo_suppkey = s_suppkey and p_category = 'MFGR#12' and s_region = 'AMERICA' group by d_year, p_brand1 order by d_year, p_brand1;
Q2.2	select sum(lo_revenue), d_year, p_brand1 from part, supplier, dateinfo, lineorder where lo_orderdate = d_datekey and lo_partkey = p_partkey and lo_suppkey = s_suppkey and p_brand1 between 'MFGR#2221' and 'MFGR#2228' and s_region = 'ASIA' group by d_year, p_brand1 order by d_year, p_brand1;

Nom de la requête	Scripts
Q2.3	select sum(lo_revenue), d_year, p_brand1 from part, supplier, dateinfo, lineorder where lo_orderdate = d_datekey and lo_partkey = p_partkey and lo_suppkey = s_suppkey and p_brand1= 'MFGR#2239' and s_region = 'EUROPE' group by d_year, p_brand1 order by d_year, p_brand1;
Q3.1	select c_nation, s_nation, d_year, sum(lo_revenue) as revenue from customer, supplier, dateinfo, lineorder where lo_custkey = c_custkey and lo_suppkey = s_suppkey and lo_orderdate = d_datekey and c_region = 'ASIA' and s_region = 'ASIA' and d_year >= 1992 and d_year <= 1997 group by c_nation, s_nation, d_year order by d_year asc, revenue desc;
Q3.2	select c_city, s_city, d_year, sum(lo_revenue) as revenue from customer, supplier, dateinfo, lineorder where lo_custkey = c_custkey and lo_suppkey = s_suppkey and lo_orderdate = d_datekey and c_nation = 'UNITED STATES'

Nom de la requête	Scripts
	and s_nation = 'UNITED STATES' and d_year >= 1992 and d_year <= 1997 group by c_city, s_city, d_year order by d_year asc, revenue desc;
Q3.3	select c_city, s_city, d_year, sum(lo_revenue) as revenue from customer, supplier, dateinfo, lineorder where lo_custkey = c_custkey and lo_suppkey = s_suppkey and lo_orderdate = d_datekey and (c_city='UNITED KI1' or c_city='UNITED KI5') and (s_city='UNITED KI1' or s_city='UNITED KI5') and d_year >= 1992 and d_year <= 1997 group by c_city, s_city, d_year order by d_year asc, revenue desc;
Q3.4	select c_city, s_city, d_year, sum(lo_revenue) as revenue from dateinfo, customer,supplier, lineorder where lo_custkey = c_custkey and lo_suppkey = s_suppkey and lo_orderdate = d_datekey and (c_city='UNITED KI1' or c_city='UNITED KI5') and (s_city='UNITED KI1' or s_city='UNITED KI5')

Nom de la requête	Scripts
	and d_yearmonth = 'Dec1997' and lo_orderdate between 19971201 and 19971231 group by c_city, s_city, d_year order by d_year asc, revenue desc;
Q4.1	select d_year, c_nation, sum(lo_revenue - lo_supplycost) as profit from customer, supplier, part, dateinfo, lineorder where lo_custkey = c_custkey and lo_suppkey = s_suppkey and lo_partkey = p_partkey and lo_orderdate = d_datekey and c_region = 'AMERICA' and s_region = 'AMERICA' and (p_mfgr = 'MFGR#1' or p_mfgr = 'MFGR#2') group by d_year, c_nation order by d_year, c_nation;
Q4.2	select d_year, s_nation, p_category, sum(lo_revenue - lo_supplycost) as profit from customer, supplier, part, dateinfo, lineorder where lo_custkey = c_custkey and lo_suppkey = s_suppkey and lo_partkey = p_partkey and lo_orderdate = d_datekey and c_region = 'AMERICA' and s_region = 'AMERICA' and (d_year = 1997 or d_year = 1998)

Nom de la requête	Scripts
	and lo_orderdate between 19970101 and 19981231 and (p_mfgr = 'MFGR#1' or p_mfgr = 'MFGR#2') group by d_year, s_nation, p_category order by d_year, s_nation, p_category;
Q4.3	select d_year, s_city, p_brand1, sum(lo_revenue - lo_supplycost) as profit from supplier, dateinfo, part, customer, lineorder where lo_custkey = c_custkey and lo_suppkey = s_suppkey and lo_partkey = p_partkey and lo_orderdate = d_datekey and s_nation = 'UNITED STATES' and (d_year = 1997 or d_year = 1998) and lo_orderdate between 19970101 and 19981231 and p_category = 'MFGR#14' group by d_year, s_city, p_brand1 order by d_year, s_city, p_brand1;

Annexe B

Exemple d'un plan de requête

Le plan de requête permet de voir le calcul du chemin que le SGBD a choisi afin d'exécuter la requête de façon la plus performante. Ce calcul n'est pas toujours optimal. Il est donc nécessaire de consulter le plan de la requête et de l'analyser pour revoir le cas échéant la requête ou le script en question. En fait, ce plan donne des indications afin de déterminer si la requête ou le script sont performants.

Figure 3-6: Modèle de données de test - Inventaire

Annexe C

Modèle de données des tests

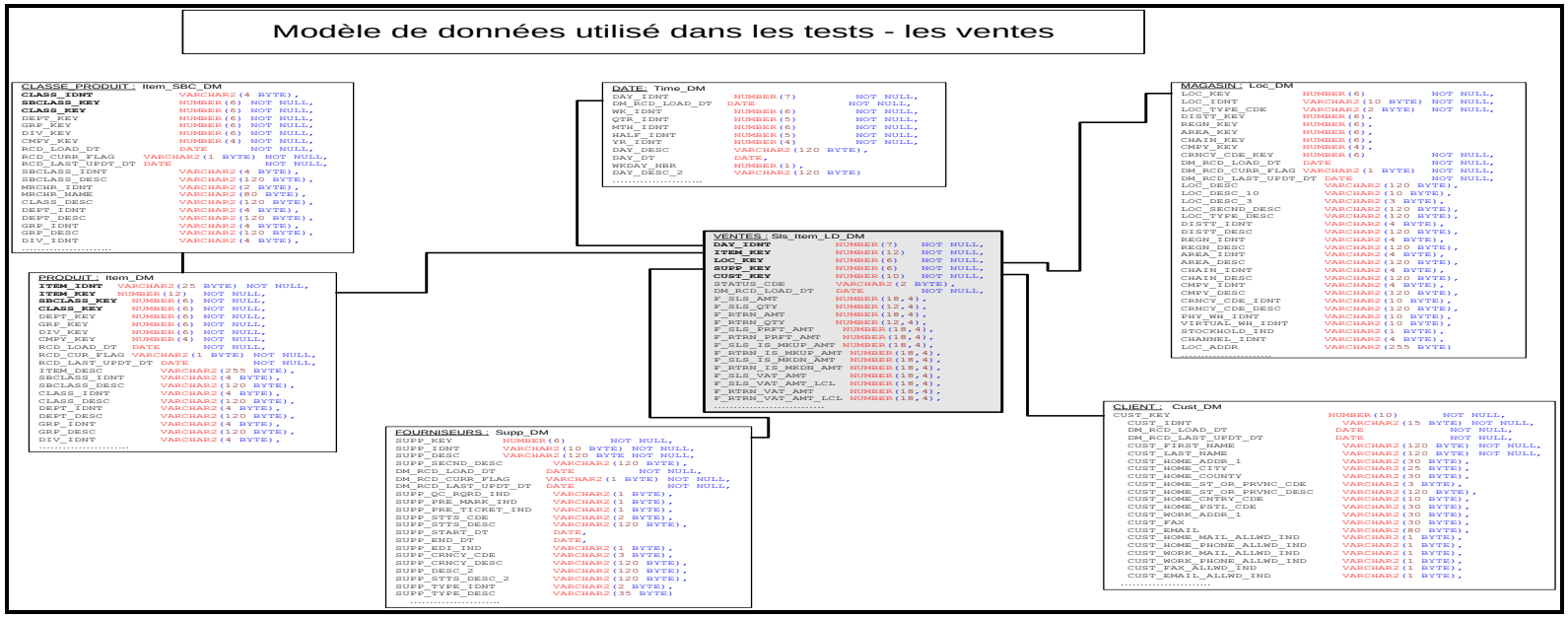

Figure 3-7: Modèle de données de test -Ventes

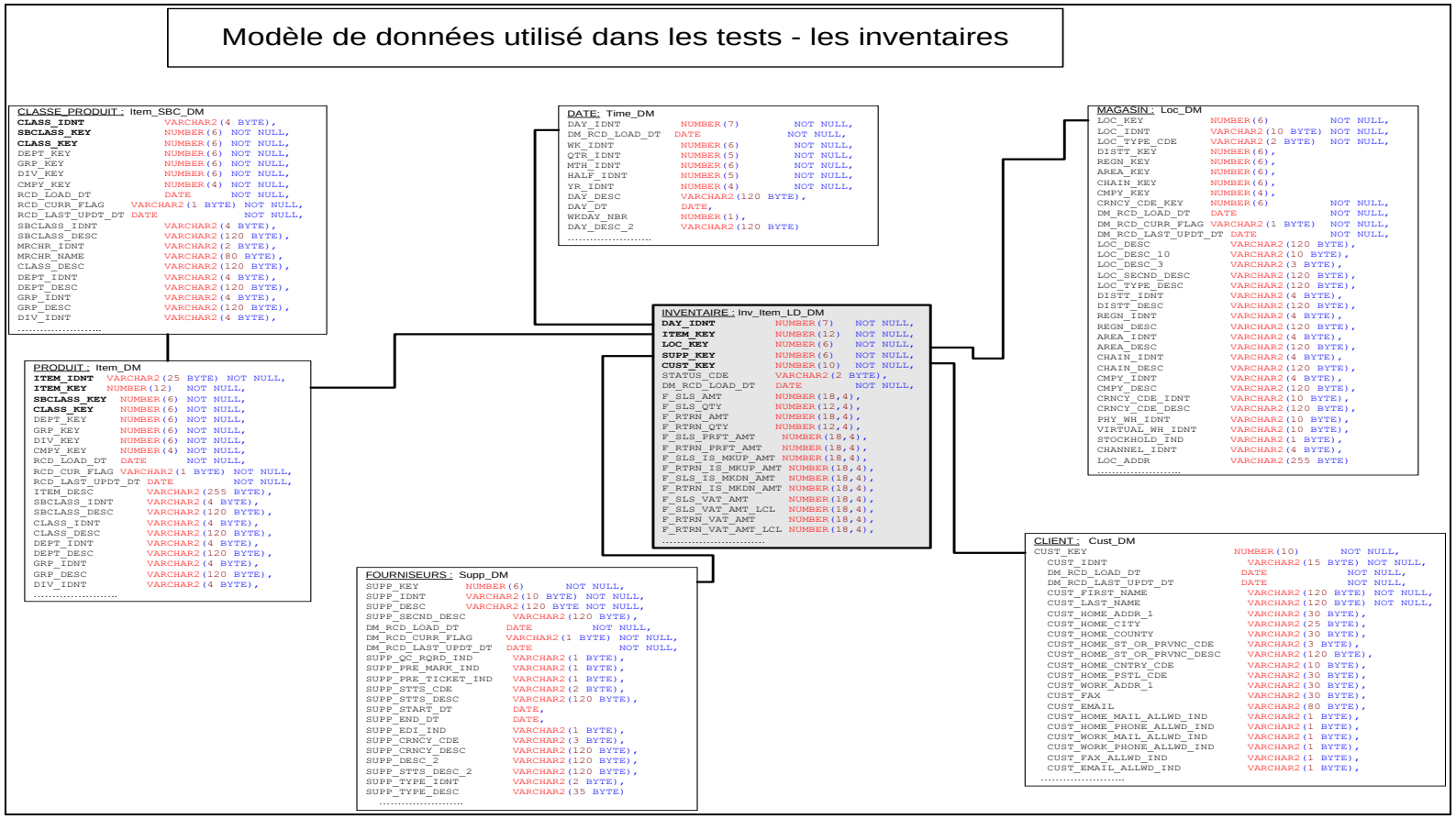

Figure 3-8: Modèle de données de test - Inventaire

Annexe D

Configuration de la BD et de l'environnement de test

Les requêtes suivantes ont permis de produire les données de la configuration de la BD :

```
Pour la version du SGBD, il suffit d'interroger la vue V$VERSION.
SELECT * FROM v$version;
Le résultat de la requête est le suivant:
```

BANNER

Oracle Database 10g Enterprise Edition Release 10.2.0.3.0 - 64bi

PL/SQL Release 10.2.0.3.0 - Production

CORE 10.2.0.3.0 Production

TNS for IBM/AIX RISC System/6000: Version 10.2.0.3.0 - Production

NLSRTL Version 10.2.0.3.0 - Production

Pour les options Oracle installées il faut interroger la vue v$option.

SELECT * FROM v$option;

Le résultat de la requête est dans le tableau suivant :

Tableau 3-4: Données de configuration de la BD de test

Parameter	value
Partitioning	TRUE
Objects	TRUE
Real Application Clusters	FALSE
Advanced replication	TRUE
Bit-mapped indexes	TRUE
Connection multiplexing	TRUE
Connection pooling	TRUE
Database queuing	TRUE

Parameter	value
Incremental backup and recovery	TRUE
Instead-of triggers	TRUE
Parallel backup and recovery	TRUE
Parallel execution	TRUE
Parallel load	TRUE
Point-in-time tablespace recovery	TRUE
Fine-grained access control	TRUE
Proxy authentication/authorization	TRUE
Change Data Capture	TRUE
Plan Stability	TRUE
Online Index Build	TRUE
Coalesce Index	TRUE
Managed Standby	TRUE
Materialized view rewrite	TRUE
Materialized view warehouse refresh	TRUE
Database resource manager	TRUE
Spatial	TRUE
Visual Information Retrieval	TRUE
Export transportable tablespaces	TRUE
Transparent Application Failover	TRUE
Fast-Start Fault Recovery	TRUE
Sample Scan	TRUE
Duplexed backups	TRUE
Java	TRUE
OLAP Window Functions	TRUE
Block Media Recovery	TRUE
Fine-grained Auditing	TRUE

Parameter	value
Application Role	TRUE
Enterprise User Security	TRUE
Oracle Data Guard	TRUE
Oracle Label Security	FALSE
OLAP	TRUE
Table compression	TRUE
Join index	TRUE
Trial Recovery	TRUE
Data Mining	TRUE
Online Redefinition	TRUE
Streams Capture	TRUE
File Mapping	TRUE
Block Change Tracking	TRUE
Flashback Table	TRUE
Flashback Database	TRUE
Data Mining Scoring Engine	FALSE
Transparent Data Encryption	TRUE
Backup Encryption	TRUE
Unused Block Compression	TRUE
Oracle Database Vault	FALSE

Pour le nombre de CPUs, de Cores et de mémoire physique on utilise la vue v$osstat

SELECT * FROM v$osstat;

Le résultat de la requête se trouve dans le tableau suivant:

Tableau 3-5: Données de configuration l'environnement de test

STAT_NAME	VALUE	OSSTAT_ID
NUM_CPUS	12	0
IDLE_TIME	585757744	1
BUSY_TIME	258296367	2
USER_TIME	219815627	3
SYS_TIME	38480740	4
IOWAIT_TIME	118858948	5
AVG_IDLE_TIME	48791542	7
AVG_BUSY_TIME	21503010	8
AVG_USER_TIME	18296295	9
AVG_SYS_TIME	3185128	10
AVG_IOWAIT_TIME	9883358	11
OS_CPU_WAIT_TIME	295191000	13
RSRC_MGR_CPU_WAIT_TIME	0	14
LOAD	0.0029296875	15
NUM_CPU_CORES	6	16
NUM_VCPUS	6	18
NUM_LCPUS	12	19
PHYSICAL_MEMORY_BYTES	25769803776	1008

.

Annexe E

Volumétrie des données de test

L'entrepôt de données utilisé lors des tests dessert au minimum sept domaines d'affaire dans l'entreprise :

- La commercialisation
- Les opérations
- Le marketing
- Les finances
- La vérification interne et amélioration continue
- La logistique
- La chaine d'approvisionnement

La BD contient 2.5 To (Téraoctets) de données dont 0.04 GO pour les dimensions (0 %), 627.8 GO pour les ventes (26 %), 1678.9 GO pour les inventaires (68 %) et 139 GO pour autres (6 %).

Le volume total est réparti comme suit : 1.6 To pour les données et 0.9 To pour les index.

Annexe F
Fonctionnement du parallélisme

F.1 Opérations pouvant être parallélisées

Les types d'opérations pouvant être parallélisées sont :

- Les méthodes d'accès comme : Table Scans, Index Full Scans et Parionned Index Range Scans;
- Les méthodes de jointure comme : Nested Loop, Sort Merge, Hash et Star Transformation;
- Les instructions DDL (Langage de Définition de Données) comme : CREATE TABLE … AS SELECT, CREATE INDEX, REBUILD INDEX, REBUILD INDEX PARTION, MOVE SPLIT COALESCE PARTITION;
- Les instructions DML (*Data Manipulation Language*) langage de manipulation de données comme : INSERTS AS SELECT, UPDATE, DELETE et MERGE;
- Les diverses instructions SQL comme : GROUPE BY, NOT IN, SELECT DISTINCT, UNION, UNION ALL, CUBE, ROLLUP;
- Les fonctions comme celles d'agrégats de table.

F.2 Paramètre d'initialisation du parallélisme

Lorsqu'une séance démarre, le SGBD crée une zone commune (*pool*) pour les opérations en parallèle. Le paramètre d'initialisation PARALLEL_MIN_SERVERS (zéro par défaut) spécifie le nombre d'exécutions en parallèle qui sont créées au démarrage de l'instance.

Lorsqu'une requête SQL est exécutée, l'optimiseur décide si l'opération doit s'exécuter en parallèle et détermine le degré de parallélisme (DOP : Degry Of Parallelism) pour chaque opération. Généralement le DOP est limité au double du nombre de processeurs et il est

associé à une seule opération à la fois. Il est mentionné dans les cas suivants :

- Au niveau d'une expression SQL : avec des HINTS ou avec la clause PARALLEL;
- Au niveau de la session avec la commande ALTER SESSION FORCE PARALLEL;
- Au niveau de la définition de la table;
- Au niveau de la définition de l'index.

Le paramètre PARALEL_MAX_SERVERS est la limite de sessions parallèles que le SGBD peut créer.

F.3 Exemples d'utilisation du parallélisme

La configuration du parallélisme doit être « par défaut », ce qui signifie que lorsqu'un processus utilise du parallélisme, il peut utiliser jusqu'à 40 processus en parallèle, déterminés par la formule : (CPU_COUNT X PARALLEL_THREADS_PER_CPU), 10 X 4 = 40.

Le maximum de processus possible en parallèle est configuré à 96. Donc, si 4 requêtes sont lancées en même temps sur le serveur, les deux premières peuvent prendre 80 processus, la troisième peut prendre les 16 derniers et la quatrième n'aura aucun processus en parallèle de disponible. Donc, le maximum du DOP configuré sur les tables ou index sera initialisé à quatre (4). Pour déterminer le nombre de processus en parallèle possible, le SGBD utilise la formule : (DOP X PARALLEL_THREADS_PER_CPU), 4 X 4 = 16.

Cela permet d'avoir (96 ÷ 16)= 6 requêtes en traitement simultané. Une requête qui « généralement » utilise du parallélisme aura beaucoup plus de chance d'en avoir de disponible et les temps d'exécution seront plus stables.

Les changements ne seront pas tous appliqués en même temps! En procédant par étape, il sera plus facile d'effectuer des vérifications sur les temps d'exécution de la chaîne ETC pour s'assurer que les temps seront respectables.

Dans la même optique, selon la recommandation du DBA, il est préférable d'utiliser le *BUFFER_POOL* « **RECYCLE** » au lieu de « DEFAULT ». Ce pool est plus rapide. Aussi, il

est plus intéressant de configurer l'attribut *logging* à « ***NOLOGGING*** », ce qui évite d'écrire dans les « REDO LOGS » d'Oracle et améliore les performances.

Le nombre de CPU alloués à une instance est un paramètre de configuration du SGBD. C'est le nombre de processus pouvant être attribués à un traitement parallèle si la table sollicitée est définie avec du parallélisme. À la fin de l'exécution d'une requête entraînant l'utilisation du parallélisme, les processus ne sont pas libérés tant que la session reste ouverte. Il est donc essentiel d'exécuter un script qui déconnecte toutes les sessions encore actives afin de permettre une utilisation optimale du parallélisme. De plus, pour mieux favoriser cette utilisation, il serait intéressant de laisser les processus prêts à intervenir s'ils sont libérés. Pour ce faire, on doit initialiser le paramètre PARALLEL _MIN_SERVER à 10.
Si une requête nécessitant du parallélisme obtient les processus requis, elle réalise un gain de performance de près de 500 % sur les tables partitionnées.

F.4 Quand faut-il implémenter l'exécution parallèle?

L'exécution des opérations en parallèle assure de bonnes performances dans les DW. Les systèmes *OLTP* (*On-Line Transaction Processing*) peuvent également en bénéficier lors de l'utilisation en lot. En général, les processus bénéficiant de l'exécution parallèle sont :

- Les requêtes requérant le scan de grosses tables, de jointure ou des index partitionnés (*partitioned index scan*);
- La création ou la reconstruction de grands index;
- La création de grandes tables incluant les vues matérialisées;
- L'insertion (INSERT), la mise à jour (UPDATE), la fusion (MERGE) et l'effacement (DELETE) de gros volumes de données;
- L'accès à de gros objets tels que les LOBs (*Large OBjects*) qui permet de stocker d'importants volumes de données comme le CLOB qui permet de sauvegarder les données de type 'chaîne de caractères'.

Annexe G
Points généraux d'amélioration de performance

Les points suivants sont communs à tous les problèmes de performance. Ils sont d'ordre général :

- Pour gagner de la performance les clés primaires doivent être des entiers;
- Supprimer ou désactiver l'intégrité relationnelle (*relationnal integrity RI*) avant le chargement puis la restaurer après que le chargement est complété;
- Supprimer ou désactiver les index avant le chargement puis les restaurer après le chargement;
- Charger uniquement les données modifiées avec l'utilisation du « delta » de traitement;
- Utiliser les technologies *RAID* (*Redundant Array of Independant/Inexpensive Disk*), qui veut dire « matrice redondante de disques indépendants/bon marché », une technologie qui tire profit de plusieurs disques durs. Elle permet de stocker des données sur de multiples disques durs afin d'améliorer, en fonction du type de RAID choisi, la tolérance aux pannes et/ou les performances de l'ensemble. RAID peut réduire les risques de pertes de données en cas de défaillance d'une unité de stockage et améliorer les performances. Par contre, il ne peut pas protéger totalement des défaillances matérielles (éventualité de pannes successives de plusieurs disques), protéger les données des erreurs humaines (suppression accidentelle de fichiers), protéger l'utilisateur des risques extérieurs au système (surcharge électrique qui grillerait l'ensemble des disques, incendie), protéger les données des virus et des *spywares* qui pourraient corrompre les données;
- L'utilisation de disques et des contrôleurs -1500 *RPM* (*Revolution ou Round Per Minute*) nombre de tours par minute;
- Les serveurs et les disques dédiés au BI ne doivent pas être partagés avec les

autres applications;

- Utilisation de plusieurs serveurs pour le BI, tels que : un serveur de base de données, un serveur d'analyse et un serveur de rapport;
- Utilisation de serveurs avec une grande mémoire (16GO et plus). Ce qui augmente la mise en cache des données physiques et réduit l'accès aux données;
- Utilisation d'un serveur avec plusieurs processeurs pour permettre un plus grand parallélisme;
- Garder les statistiques de la BD à jour;
- Prendre en considération le degré de maturité de l'entreprise en matière de BI.

Annexe H

Matrice problèmes - solutions

Il est très intéressant de pouvoir reprendre les problèmes traités dans le deuxième chapitre et de les classer avec les solutions étudiées dans le chapitre suivant. La matrice montre clairement quel problème peut être résolu avec quelle solution.

H.1 Matrice problèmes « *back-end* » – solutions

Tableau 3-6: Matrice problèmes de performance « *back-end* » - solutions

Matrice problèmes / solution du volet « *back-end* » du data warehouse																							
Problèmes ▶ / Solutions ▼	**B1**	**B2**	**B3**	**B4**	**B5**	**B6**	**B7**	**B8**	**B9**	**B10**	**B11**	**B12**	**B13**	**B14**	**B15**	**B16**	**B17**	**B18**	**B19**	**B20**	**B21**	**B22**	**B23**
S1								x					x										
S2								x											x				
S3								x											x				
S4								x											x				
S5								x											x				
S6								x											x				
S7								x											x				
S8																							
S9								x											x				

Matrice problèmes / solution du volet « *back-end* » du data warehouse

Problèmes ▶ / Solutions ▼	**B1**	**B2**	**B3**	**B4**	**B5**	**B6**	**B7**	**B8**	**B9**	**B10**	**B11**	**B12**	**B13**	**B14**	**B15**	**B16**	**B17**	**B18**	**B19**	**B20**	**B21**	**B22**	**B23**
S10								x											x				
S11								x											x				
S12								x											x				
S13								x											x				
S14								x											x				
S15	x	x	x	x	x	x	x	x			x	x	x	x	x	x	x		x	x	x	x	x
S16					x					x										x	x		
S17	x	x	x	x						x	x				x					x			
S18									x	x	x									x	x	x	
S19																		x					

H.2 Matrice problèmes « *front-end* » – solutions

Tableau 3-7: Matrice problèmes de performance « *front-end* » – solutions

Matrice problèmes / Solutions du volet « *front-end* » du data warehouse											
Problèmes ▶ / **Solutions ▼**	**F 1**	**F 2**	**F 4**	**F 4**	**F 5**	**F 6**	**F 7**	**F 8**	**F 9**	**F 1 0**	**F 1 1**
S1									x		
S2	x	x	x	x	x	x	x		x		
S3		x	x	x							
S4		x	x	x							
S5		x	x	x							
S6		x	x	x							
S7		x	x	x							
S8											x
S9		x									
S10	x										
S11											
S12				x							
S13											
S14				x							
S15											
S16											
S17								x			
S18		x						x	x	x	x

Annexe I

Outils de monitoring de la base de données

La plupart des bases de données ont un outil de monitoring, notamment pour Oracle on a DB Console. L'outil de monitoring permet de visualiser le comportement de la BD. Cet outil a plusieurs tâches qui ne seront pas toutes recensées dans ce paragraphe. Le but est de comprendre que ces mécanismes interviennent dans l'étude de performance de la BD qui a son tour joue un rôle dans la performance de l'entrepôt de données.

L'outil de monitoring permet de suivre en temps réel :

- la détection des requêtes les plus lentes;
- la détection des requêtes les plus utilisées;
- la détection des requêtes les plus coûteuses;
- le monitoring des login / connexions à la BD;
- le monitoring des sessions de la BD;
- le monitoring des instructions DDL au niveau de la BD;
- la gestion des *locks*;
- la gestion de l'espace.