



HAL
open science

Compression et optimisation de l'infrastructure de stockage des bases de données relationnelles

Éric Guttin

► **To cite this version:**

Éric Guttin. Compression et optimisation de l'infrastructure de stockage des bases de données relationnelles. Base de données [cs.DB]. 2016. dumas-01538363

HAL Id: dumas-01538363

<https://dumas.ccsd.cnrs.fr/dumas-01538363>

Submitted on 13 Jun 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

**CONSERVATOIRE NATIONAL DES ARTS & MÉTIERS
PARIS**

MÉMOIRE

Présenté en vue d'obtenir le

DIPLÔME D'INGÉNIEUR CNAM

SPECIALITE : Informatique

OPTION : Systèmes d'information (ISI)

par

GUTTIN Éric

Compression et optimisation de l'infrastructure de
stockage des bases de données relationnelles

Soutenu le 21/01/2016

JURY

PRÉSIDENT : M. Michel CRUCIANU (CNAM)

MEMBRES: M. Philippe RIGAUX (CNAM)
M. Nicolas TRAVERS (CNAM)
M. Denis HOCDE (ENGIE)
M. Malik HAMICHE (ENGIE)

Résumé

En 2010, la croissance des bases relationnelles incite la DSI Gaz de ENGIE à optimiser ses coûts de stockage et de sauvegarde, en introduisant des technologies de compression pour diminuer la taille de ses bases de données, majoritairement basées sur Oracle.

Une étude préalable des différentes solutions de compression à lieu en 2011 avec les principaux éditeurs du marché des SGBDR (Oracle, IBM, Microsoft). Au final la solution Advanced Compression d'Oracle est retenue, car elle apporte des gains relativement comparables aux autres SGBDR tout en limitant l'effort de migration.

Son implémentation nécessite des opérations de réorganisation très conséquentes sur la base de données. Un des enjeux majeurs concerne les applications avec des durées d'interruptions de services très courtes qui ne peuvent être arrêtées plus de quelques heures. Pour se faire une seconde technologie Oracle nommée GoldenGate est mise en œuvre pour minimiser l'interruption des applications SAP pendant les opérations de compression.

Par l'exploitation de ces deux nouvelles technologies, les volumes des bases ont été réduits en moyenne par 2,4 sur 90% du paysage concerné, y compris les plus grosses bases. La DSI a ainsi atteint, voir dépassé, ses objectifs de ROI sur les coûts de stockage et de sauvegardes.

Mots clés : Bases de données relationnelles, compression, coûts de stockage, Oracle, DB2, SQLServer.

Abstract

In 2010, growth in relational databases encourages ENGIE IT to optimize its storage costs and backup, with introducing compression technologies to decrease the size of its databases, mainly based on Oracle.

A preliminary study of main compression solutions is made in 2011 with leading market for RDBMS (Oracle, IBM, Microsoft). Ultimately, the Advanced Compression Oracle solution, is retained because it brings relatively comparable gains to other RDBMS while limiting the migration effort.

Its implementation requires very substantial reorganization operations on the database. Business applications are facing to very short periods of service disruptions. they cannot be stopped for many hours. A second technology called Oracle GoldenGate is then implemented to minimize disruption SAP applications during compression operations.

By operating these two new technologies, databases' volumes have been reduced on average by 2.4 on 90 % of the landscape concerned, including larger databases.

Engie IT has reached or exceeded its ROI objectives on the costs of storage and backups.

Keywords : Relational database, compression, storage costs, Oracle, DB2, SQLServer.

Remerciements

« Il n'y a qu'une façon active de connaître, c'est de découvrir et corrélativement, il n'y a qu'une façon d'enseigner, c'est de faire découvrir ». Gaston BACHELARD (1884-1962).

Je tiens à remercier tout particulièrement Monsieur Michel CRUCIANU, professeur et directeur du CEDRIC du CNAM, d'avoir accepté de présider le jury de soutenance de ce mémoire.

Je remercie Monsieur Philippe RIGAUX, professeur des universités au CNAM qui a accepté de reprendre très récemment mon sujet de mémoire et m'a apporté son soutien.

Je remercie Monsieur Denis HOCDE, responsable de la cellule de performances du pôle TESAM de GDF SUEZ IT pour son aide essentielle, son soutien, et surtout pour m'avoir permis de réaliser un projet aussi ambitieux qu'instructif.

Je remercie les membres du Jury pour le temps passé à l'évaluation de ce mémoire.

Je remercie l'ensemble des professeurs du CNAM avec lesquels j'ai eu l'opportunité d'apprendre, de comprendre et qui ont su cultiver le goût de l'apprentissage.

Je remercie l'ensemble des personnes avec qui j'ai travaillé pour la réalisation de ce projet et qui ont su partager leurs connaissances et leurs expertises.

Enfin, je dédie ce mémoire à Agnès, ma femme, qui m'a toujours soutenu tout au long de ce parcours.

TABLES DES MATIERES

INTRODUCTION.....	13
CHAPITRE 1. CONCEPTS ET METHODES DE LA COMPRESSION DES BASES DE DONNEES.....	17
1.1 CONCEPTS GENERAUX DE LA COMPRESSION DES DONNEES.....	17
1.1.1 <i>Définition de la compression de données</i>	17
1.1.2 <i>Méthodes de compressions de données</i>	17
1.2 METHODES DE COMPRESSION DES BASES DE DONNEES RELATIONNELLES.....	18
1.2.1 <i>Compression avec Oracle</i>	18
1.2.2 <i>Compression avec DB2 (IBM)</i>	26
1.2.3 <i>Compression avec SQL Server (Microsoft)</i>	31
1.3 COMPARAISON DES SOLUTIONS DE COMPRESSION.....	34
1.3.1 <i>Synthèse comparative des éditeurs Oracle, DB2, SQL Server</i>	34
1.3.2 <i>Oracle versus DB2</i>	37
1.3.3 <i>Oracle versus SQL Server</i>	37
1.4 BENEFICES DE LA COMPRESSION DES BASES DE DONNEES.....	38
1.4.1 <i>La diminution des coûts</i>	38
1.4.2 <i>L'amélioration de la gestion de la disponibilité des applications</i>	39
1.4.3 <i>L'amélioration des performances</i>	40
CHAPITRE 2. ETUDE COMPARATIVE DES AUTRES SOLUTIONS SGBDR VIS-A-VIS D'ORACLE.....	41
2.1 ETAT ACTUEL DU PARC DES BASES DE DONNEES.....	41
2.2 FAISABILITE.....	43
2.2.1 <i>Faisabilité technique</i>	43
2.2.2 <i>Durées</i>	44
2.3 SCALABILITE.....	44
2.3.1 <i>Scalabilité Taille de base</i>	45
2.3.2 <i>Scalabilité Hardware</i>	45
2.4 DISPONIBILITE.....	46
2.4.1 <i>DIMA / PDMA</i>	46
2.4.2 <i>Plan de reprise d'activité (PRA)</i>	46
2.5 EXPLOITABILITE.....	47
2.5.1 <i>Administration récurrente</i>	47
2.5.2 <i>Refresh (rafraîchissement)</i>	48
2.5.3 <i>Corruptions</i>	48
2.6 SECURITE.....	48
2.6.1 <i>Règles de sécurité des mots de passe</i>	49
2.6.2 <i>Restrictions</i>	49
2.6.3 <i>Traçabilité</i>	49
2.6.4 <i>Cryptage</i>	49
2.7 FINANCIER.....	50
2.7.1 <i>TCO</i>	50
2.7.2 <i>Licences</i>	50

2.7.3	Coûts projet	50
2.8	SYNTHESE DE L'ÉVALUATION.....	52
2.9	CHOIX DE LA SOLUTION.	55
CHAPITRE 3.	ETUDE DE LA COMPRESSION DES DONNEES AVEC ORACLE.....	56
3.1	ENJEUX ET OBJECTIFS	56
3.2	SUPPORT DE LA COMPRESSION PAR SAP	56
3.3	SCENARIO DE COMPRESSION	57
3.3.1	Prévision du stockage	58
3.3.2	Montée de version d'Oracle 10 g vers Oracle 11 g	61
3.3.3	Préparation des opérations de compression	62
3.3.4	Opérations de réorganisation et de compression	63
3.3.5	Opérations de post-compression.....	72
3.3.6	Gains en volumétrie.....	75
3.3.7	Evolution des performances.....	77
3.3.8	Durées d'indisponibilité.....	77
3.3.9	Majeures difficultés rencontrées	78
3.4	BILAN DU PROOF OF CONCEPT DE COMPRESSION.....	80
CHAPITRE 4.	MISE EN ŒUVRE D'UNE SOLUTION DE MINIMISATION DES INDISPONIBILITES	81
4.1	PRESENTATION DE LA SOLUTION GOLDENGATE	81
4.1.1	Architecture générale.....	81
4.1.2	Réplication entre des environnements hétérogènes	82
4.1.3	Différentes applications	82
4.1.4	Principes de captures.....	83
4.1.5	Sélectivité et filtres	83
4.1.6	Résilience aux interruptions et aux pannes	83
4.2	PRESENTATION DU SCENARIO DE COMPRESSION AVEC INDISPONIBILITE PROCHE DE ZERO.	84
4.3	ARCHITECTURE ET CONFIGURATIONS GOLDENGATE.....	85
4.3.1	Capture en mode « Full déporté ».	85
4.3.2	Parallélisation des extract et replicat	86
4.4	MISE EN ŒUVRE AVEC GOLDENGATE.	89
4.4.1	Initialisation de la configuration GoldenGate	89
4.4.2	L'activation de la capture et lancement des Extract	91
4.4.3	Le rattrapage du delta d'activité.....	92
4.4.4	Les tests et les vérifications des données	92
4.4.5	Les résultats obtenus.....	93
4.4.6	Les majeures difficultés rencontrés	93
4.5	BILAN	94
CONCLUSION.....	95
	BILAN SUR LES DEPLOIEMENTS.....	95
	GAINS EN VOLUMETRIES.....	95
	GAINS EN PERFORMANCES.....	96
	BILAN FINANCIER	96
	BILAN PROJET	97
BIBLIOGRAPHIE	99

ANNEXE 1 PLANIFICATION DES DEPLOIEMENTS	101
ANNEXE 2 : GAINS DE VOLUMETRIES ET RETOUR SUR INVESTISSEMENT.....	102
GAINS DE VOLUMETRIE	102
RETOUR SUR INVESTISSEMENT	103
ANNEXE 3 : GAINS SUR LES PERFORMANCES.....	105
UTILISATION DE LA CPU.	105

Liste des figures :

FIGURE 1-1 POSITIONNEMENT DU POLE TESAM AU SEIN DU GROUPE GDF SUEZ	14
FIGURE 1-1 COMPRESSION D'UN BLOC DE DONNEES ORACLE [OAC12].....	19
FIGURE 1-2 MECANISME ASYNCHRONE DE L'OPTION « <i>ADVANCED COMPRESSION FOR ORACLE</i> » [OAC12].....	20
FIGURE 1-3 TRAITEMENT BATCH AVANT COMPRESSION [EPS10]	21
FIGURE 1-4 TRAITEMENT BATCH APRES COMPRESSION [EPS10]	22
FIGURE 1-5 UNITE DE COMPRESSION AVEC L'OPTION "HYBRID <i>COLUMNAR COMPRESSION</i> " [OHC12]	22
FIGURE 1-6 - STRUCTURE INTERNE D'UN INDEX DE TYPE <i>B*TREE</i> . [OWN05].....	24
FIGURE 1-7 COMPRESSION PAR TABLE AVEC DB2 [ICS12]	26
FIGURE 1-8 COMPRESSION ADAPTIVE DE DB2 [ICS12].....	28
FIGURE 1-9 COMPARAISON DE L'UTILISATION DE LA BANDE PASSANTE DES I/O AVANT ET APRES LA COMPRESSION [RAC12].	29
FIGURE 1-10 COMPRESSION PAR PAGE AVEC SQL SERVER 2014 [MTC14].....	32
FIGURE 2-1 EXEMPLE D'ARCHITECTURE DE REAL APPLICATION CLUSTER AVEC DEUX NŒUDS.....	43
FIGURE 3-1 LES GRANDES ETAPES DE LA COLLABORATION ENTRE ORACLE ET SAP [OSS12]	57
FIGURE 3-2 SCENARIO DE REORGANISATION ET DE COMPRESSION	58
FIGURE 3-3 VOLUMETRIE DES TABLESPACES ORACLE 10G DU POC AVANT COMPRESSION	75
FIGURE 3-4 VOLUMETRIE DES TABLESPACES DU POC APRES COMPRESSION	75
FIGURE 3-5 VOLUMETRIE DES FILE SYSTEM DU POC AVANT COMPRESSION	76
FIGURE 3-6 VOLUMETRIE DES FILE SYSTEM DU POC APRES COMPRESSION.....	76
FIGURE 4-1 - ARCHITECTURE GENERALE DE LA SOLUTION <i>GOLDENGATE</i>	81
FIGURE 4-2 LES MULTIPLES UTILISATIONS DE <i>GOLDENGATE</i>	82
FIGURE 4-3 - SCENARIO DE COMPRESSION UTILISANT <i>GOLDENGATE</i>	84
FIGURE 4-4 ARCHITECTURE DU POC AVEC <i>GOLDENGATE</i>	86
FIGURE 4-5 PARALLELISATION DES EXTRACTS ET REPLICATS MIXTE PAR FILTRE ET PAR RANGE	87
FIGURE 4-6 INITIALISATION DE <i>GOLDENGATE</i>	89

Liste des tableaux :

TABLEAU 1-1 RATIOS DE COMPRESSION DANS DES ENVIRONNEMENTS SAP AVEC L'ADVANCED COMPRESSION D'ORACLE [EPS10].	25
TABLEAU 1-2 DB2 BEST PRACTICES USING COMPRESSION FEATURE IN SAP ENVIRONMENT [ICS12].	30
TABLEAU 1-3 RATIO DE COMPRESSION AVEC L'OPTION ADAPTATIVE COMPRESSION DE DB2 [ICS12].	31
TABLEAU 1-4 RATIO DES TEMPS D'EXECUTION POUR RECONSTRUIRE LES INDEX AVEC SQLSERVER [UMC08].	33
TABLEAU 1-5 RATIOS DE COMPRESSION SELON LES TYPES DE COMPRESSIONS AVEC SQLSERVER (UMC08).....	33
TABLEAU 6 SYNTHESE DES DIFFERENTES SOLUTIONS DE COMPRESSION (ORACLE, DB2, SQL SERVER)	36
TABLEAU 7 SYNTHESE DE L'EVALUATION DES SOLUTIONS SGBDR	54
TABLEAU 3-1 RATIO DE COMPRESSION DES PLUS GROSSES TABLES.....	59
TABLEAU 3-2 NOUVELLE CONFIGURATION DES FILE SYSTEM POUR LA COMPRESSION.....	61
TABLEAU 3-3 OPTIONS DE LA COMMANDE BRSPACE	65
TABLEAU 3-4 REPARTITION DES TABLES ET INDEX SAP SUR LES NOUVEAUX TABLESPACES	67
TABLEAU 3-5 SCRIPTS SHELL UTILISANT L'UTILITAIRE SAP BRSPACE	68
TABLEAU 3-6 PARAMETRES DES SCRIPTS SHELL UTILISANT L'UTILITAIRE SAP BRSPACE	68
TABLEAU 3-7 ETAT DES LIEUX DE LA COMPRESSION DES TABLES DU POC	71
TABLEAU 3-8 ETAT DES LIEUX DE LA COMPRESSION DES INDEX DU POC	72
TABLEAU 3-9 DUREES D'INDISPONIBILITE DES OPERATIONS DE COMPRESSION	78
TABLEAU 3-10 TABLES PRESENTANT DU CHAINAGE DES LIGNES	79
TABLEAU 4-1 TESTS REALISES SUR LA REPLICATION GOLDENGATE	88
TABLEAU 4-2 PRINCIPAUX REPERTOIRES ET FICHIERS GOLDENGATE	91

Glossaire

Terme	Définition
Advanced Compression	Option payante de la compression avancée d'Oracle adaptée aux environnements transactionnels et aux environnements décisionnels.
Advanced LOB compression	Fonctionnalité améliorée de la compression des LOB à partir de la version d'Oracle 11g.
Advanced LOB Deduplication	Fonctionnalité Oracle 11g de compression des LOB qui élimine les contenus en doublon.
AWR (Automatic Workload Repository)	Référentiel de la base Oracle qui contient les statistiques permettant de diagnostiquer les problèmes de performances.
B*Tree	Structure de données en arbre équilibré composé de blocs branches et de blocs feuilles (appelé B-arbre par analogie au terme anglais « B*Tree »)
Batch	Traitement par lot.
Bloc	La plus petite unité de stockage au niveau des fichiers de la base de données. Sous Oracle la taille par défaut des blocs est de 8 Kilo octets.
BrTools	Ensemble de programmes et utilitaires SAP permettant d'administrer les bases de données Oracle.
Bulk Load	Chargement de données en masse par lot.
CBO (Cost Based Optimizer)	Optimiseur utilisé par le moteur SQL d'Oracle, basé sur les coûts d'accès aux données pour générer le plan d'exécution le plus efficace pour chaque requête en fonction des statistiques qui ont été calculées.
Classic Row Compression	Fonctionnalité de compression par page de DB2
Codage de Huffman	Algorithme de compression de données sans perte élaboré par David Albert Huffman et publié en 1952.
Compression Adaptive	Fonctionnalité de compression dynamique de DB2.
Datafiles	Fichiers de données Oracle. Un tablespace Oracle est composé de <i>datafiles</i> .
DataGuard	Solution Oracle de Log Shipping permettant d'alimenter un site de secours dans le cadre d'un plan de reprise d'activité.
DB2	DB2 est un SGBDR développé et commercialisé depuis 1983 par la société IBM.
Décommissionnement	Opération qui consiste à retirer des équipements ou dans le cadre de l'infrastructure de stockage à libérer de l'espace disque définitivement.
DIMA	Durée d'Indisponibilité Maximale Admissible
DML (Data Manipulation language)	Ordres du langage de manipulation des données, utilisés dans les requêtes INSERT, UPDATE, DELETE, et MERGE.
Extent	Extension des segments (Tables, index) composée de blocs contigus dont la taille dépend de l'OS.

Extract	Le processus <i>Extract</i> est un processus <i>GoldenGate</i> qui collecte les changements dans les journaux Oracle ou les <i>archive logs</i> et les envoie au processus manager.
Fichier archived log	Copie des fichiers journaux d'Oracle (Redo Log) utilisée pour les sauvegardes à chaud, ou la restauration jusqu'à une période précise.
Fichier contrôle	Fichiers contenant la description physique de la base Oracle.
Fichier redo log	Fichiers journaux qui servent à mémoriser toutes les modifications validées ou non faites sur la base à des fins de reprise en cas de panne.
Fichier trail	Fichier <i>GoldenGate</i> qui contient les informations des transactions qui ont été capturées par le processus extract.
GDF SUEZ IT	Filiale SI du groupe GDF SUEZ, devenu ENGIE.
GoldenGate	solution logicielle de type CDC (Change Data Capture) pour capturer des données transactionnelles modifiées, et les répliquer entre des environnements hétérogènes.
Hybrid Columnar compression	Technologie d'Oracle 12c combinant à la fois le mode de stockage en ligne et le mode de stockage en colonne. Il s'agit d'une « méthode hybride qui améliore le taux de compression tout en conservant les performances d'accès ».
IBM	Célèbre constructeur de Main Frame qui lançait en 1983 la première version de son SGBDR DB2.
LOB (Large Binary Object)	Type de données des SGBDR permettant le stockage des objets binaires. En français les LOB sont appelés les objets larges.
LUN (Logical Unit Number)	Désigne, dans le domaine du stockage informatique, le numéro d'unité logique d'un équipement SCSI
Microsoft	Microsoft fait <i>parti</i> du Top 3 mondial des éditeurs de logiciel avec Oracle et IBM.
NZDT (Near Zero Downtime)	Solution de l'éditeur SAP qui propose des scénarios de migration avec indisponibilité proche de zéro
OLTP	Nom anglais (Online Transaction <i>Processing</i>) donné aux systèmes transactionnels.
Oracle	Entreprise américaine spécialisée dans la technologie de l'information (IT), fondée en 1977 par Ellison, Miner, et Oates. Oracle est également le nom de son SGBDR qui a fait la renommée de la firme.
PCS	Plan de Continuité de Service
PDMA	Perte de Données Maximum Admissible
PL/SQL	Langage procédural d'Oracle
POC (Proof Of Concept)	Proof Of Concept ou preuve du concept en français est une réalisation courte d'une certaine méthode pour en démontrer la faisabilité.
PureScale	Solution d'IBM de cluster actif/passif avec DB2 via la couche cluster AIX.

RAC (Real Application Cluster)	Réal application cluster. Offre Oracle de cluster de serveurs
Ratio de compression	Ratio de compression = Taille non compressée / Taille compressée. Un ratio de compression de 2,4 équivaut à une réduction de la volumétrie de -58%.
Replicat	Processus <i>GoldenGate</i> qui récupère les changements dans les fichiers <i>trail</i> et les réplique dans la base de données cible
RMAN (Recovery MANager)	Utilitaire standard de la base de données Oracle, qui permet aux DBA de gérer les opérations de sauvegarde et de restauration.
ROI (Return On Investment)	Retour sur investissement ou ROI en anglais (Return On Investment), défini pour une période donnée comme la somme des profits actualisés du projet, c'est-à-dire les revenus moins les coûts, divisés par les fonds investis dans le projet.
Row Compression	Fonctionnalité de compression par ligne de SQL Server
SAP	Entreprise allemande qui conçoit et vend des Progiciels de gestion intégré (ERP en anglais) principalement à destination des entreprises ou des institutions dans le monde entier. SAP est le premier éditeur de logiciels en Europe et le quatrième dans le monde.
SAP BI	Système décisionnel de SAP
SAP ECC	Système ERP transactionnel de SAP
SAPS (SAP Application Performance Standard)	Unité de mesure des performances des serveurs supportés par SAP, indépendante de tout constructeur.
SGA (System Global Area)	Système de mémoire partagée du SGBDR Oracle
SGBDR	Système de gestion de base de données Relationnelle
SQL Server	SGBDR développé et commercialisé depuis 1989 par la société Microsoft
SQL*LDR	Outil Oracle de chargement de fichiers de données dans les tables
Supplemental logging	Fonctionnalité d'Oracle qui permet de compléter les informations manquantes des noms des colonnes, des tables et des schémas dans les fichiers <i>redo</i> logs.
Tablespace	Partition logique d'Oracle contenant une collection d'un ou plusieurs fichiers de données (Datafiles). Une base de données Oracle est composée de tablespaces
TCO (Total Cost of Ownership)	Coût global de possession, correspondant au coût du cycle de vie complet d'une infrastructure ou d'un logiciel.
Transact SQL	Langage procédural de SQL Server
Undo	Segment d'annulation d'Oracle anciennement nommé <i>rollback</i>

INTRODUCTION

Le groupe GDF SUEZ aujourd'hui devenu ENGIE, né en 2008 de la fusion des groupes Suez et Gaz de France, est un des leaders mondiaux de l'énergie avec 97 milliards de chiffres d'affaires en 2012 et près de 219 300 collaborateurs dans le monde. GDF SUEZ est présent sur l'ensemble de la chaîne de l'énergie, en électricité et en gaz naturel, de l'amont à l'aval et s'appuie sur un portefeuille d'approvisionnement diversifié et un parc de production électrique flexible et performant. GDF SUEZ propose des solutions énergétiques performantes et innovantes aux particuliers, aux collectivités et aux entreprises.

Au cours des dernières années, la branche Gaz du groupe a vu croître de manière très importante le volume de ses bases de données relationnelles. Cette augmentation forte des données a eu des répercussions importantes sur les coûts de l'infrastructure informatique. Pour y faire face, la filiale SI GDF SUEZ IT n'a cessé d'investir auprès de ses fournisseurs en location de serveurs et de baies de stockage toujours plus performants. En 2011, la DSI Gaz de Gdf SUEZ initie une étude de réduction des coûts portant sur l'infrastructure (serveurs, stockage, sauvegardes). Il était devenu opportun pour GDF SUEZ IT d'employer des technologies plus efficaces et moins coûteuses telle que la compression des données pour répondre à ces nouvelles exigences. Depuis plus d'une dizaine d'années les principaux éditeurs de SGBDR n'ont eu de cesse d'améliorer leurs technologies de compression et proposent des solutions qui permettent de réduire efficacement les volumes des bases de données relationnelles tout en conservant de très bonnes performances, voire parfois en les améliorant.

Objectifs du mémoire

Ce mémoire traite de la mise en œuvre des techniques de compression des bases de données relationnelles au sein de la filiale SI de GDF SUEZ.

Nous nous permettons de rappeler que la mise en œuvre d'un projet informatique doit être définie avant tout par rapport à la valeur ajoutée qu'elle apporte aux métiers de l'entreprise. Un des principaux objectifs de ce projet est d'améliorer significativement les coûts de stockage et les coûts de sauvegarde. Suite aux opérations de compression des bases de données, nous avons réalisé des actions de décommissionnement sur le matériel de stockage qui ont permis de diminuer les coûts OPEX des environnements.

Ce mémoire aborde les mécanismes de compression mis en œuvre par les SGBDR et les notions qui y sont associées ainsi que les bénéfices multiples attendus. Il présente, l'étude comparative des différentes solutions de compression avec les principaux éditeurs du marché des SGBDR (Oracle, IBM, Microsoft) et les exigences qui ont conduit GDF SUEZ IT au choix de la solution finale.

L'architecture technique y est décrite, ainsi que la mise en œuvre de la compression permettant de répondre aux exigences d'une entreprise comme GDF SUEZ.

Les applications cœurs de métiers de GDF SUEZ sont des applications très sensibles qui tolèrent difficilement les indisponibilités même planifiées. La compression des bases de données occasionne une interruption de plusieurs jours pour des bases de données volumineuses atteignant les 20 téra, ce qui n'est pas envisageable sur ce périmètre. Afin de respecter les engagements de niveaux de services contractualisés par les bénéficiaires métiers de GDF SUEZ, des technologies de réplication ont été mises en œuvre pour réduire l'indisponibilité à quelques heures. Sans la mise en œuvre de ces technologies, le projet de compression des bases de

données n'aurait pas été validé par les bénéficiaires métiers. Le mémoire décrit la mise en œuvre de la compression avec ces technologies.

Organisation de GDF SUEZ IT

Dans le groupe, la gestion du SI est gérée par branche métier. Le groupe GDF SUEZ dispose également d'une filiale SI (Système d'information) : GDF SUEZ IT, chargée des missions de production de services informatiques pour le compte des branches, des Business Unit et des filiales du groupe dans les domaines suivants :

Infrastructures IT : réseaux, environnement de travail, Datacenter, sécurité, communication ;

Etudes, développement et maintenance d'applications ;

Conseil en organisation et management autour de projets métiers à forte dimension IT.

Au sein de la GDF SUEZ IT, la **Direction des Solutions Applicatives**, assure la maintenance et le développement d'applications informatiques pour le compte des directions fonctionnelles mais aussi pour les branches du groupe GDF SUEZ, sous différentes technologies.

Organisation du pôle TESAM.

Le pôle TESAM, ainsi que démontré dans la Figure 1, appartient au département « Services SI » qui est lui-même un département de GDF SUEZ IT appartenant à la « Direction des solutions applicatives ».

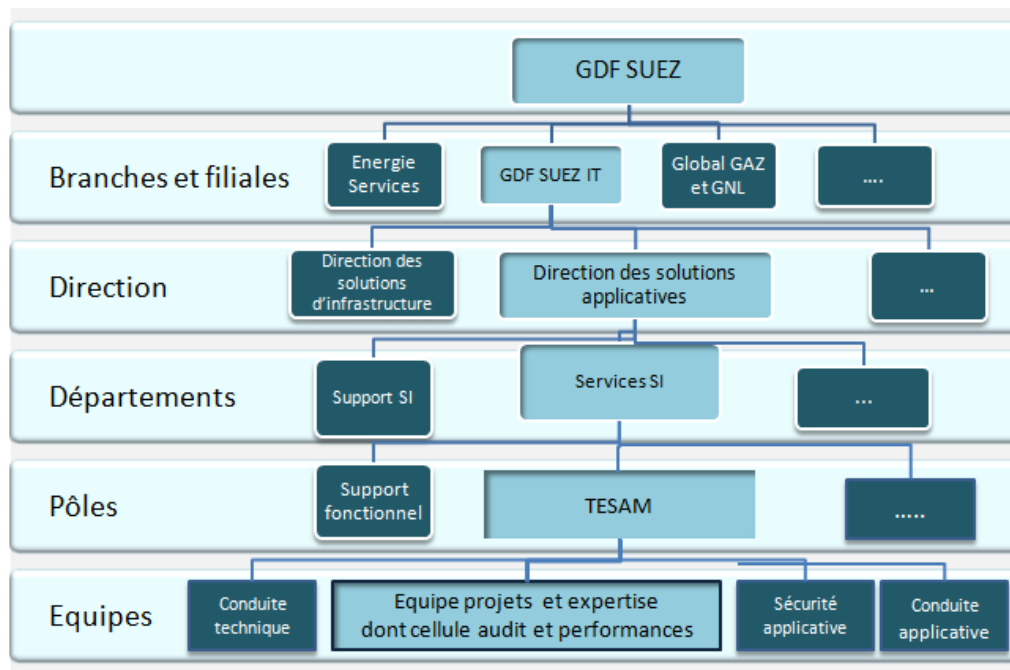


Figure 1-1 Positionnement du pôle TESAM au sein du groupe GDF SUEZ

Le pôle TESAM assure l'exploitation et la maintenance technique d'un des périmètres des applications métiers liés aux activités Gaz. Le pôle TESAM est chargé du maintien en condition opérationnelle, de la conduite technique, de l'expertise technique, de la sécurité applicative et de la conduite applicative. La majorité des applications reposent sur l'ERP SAP et le SGBDR Oracle. L'infrastructure informatique (serveurs, stockage, réseaux) est hébergée chez un fournisseur dans des Datacenter externes. La couche middleware (Oracle, composants techniques SAP,...) ainsi

que les composants logiciels (SAP, Informatica, Business Object,...) sont gérés par les équipes du pôle TESAM.

L'équipe projets et expertise est constituée de 10 personnes sous la responsabilité de 2 Pilotes opérationnels et d'un responsable d'équipe. L'équipe projet et expertise comprend une cellule d'audit et performances composée de 4 personnes dont je fais partie.

Cette cellule réalise des audits des performances et des études techniques sur les applications SAP et non SAP gérées par le pôle TESAM mais aussi de manière plus large sur tout le groupe GDF SUEZ.

Mon rôle au sein de l'équipe projets et expertise du pôle TESAM

Depuis 2009 je suis devenu le référent technique sur Oracle du pôle TESAM au sein duquel je fais partie de la cellule « audits et performances » de l'équipe « projets et expertise ». Le paysage informatique est composé essentiellement d'applications reposant sur l'ERP SAP. Les applications SAP que nous exploitons représentent environ 70 bases de données dont certaines de plus de 5 Téra Octets. J'interviens sur ces applications cœur de métier de GDF SUEZ, qui sont donc très sensibles. Dans mon rôle de référent technique Oracle, je réalise le support niveau 3 et ai en charge l'analyse des incidents sur les bases de données relationnelles. Je rédige des notes techniques Oracle pour les administrateurs SAP (ERP de l'éditeur SAP). Mon manager me délègue ponctuellement d'autres tâches concernant le planning, ou la rédaction de présentations pour des réunions avec la DSI.

J'interviens également au sein du pôle TESAM sur des projets innovants tels que le présent projet de compression des bases de données. L'ensemble des activités et des livrables que j'ai réalisés à l'occasion de ce projet comprennent :

- L'étude des différentes solutions de compression avec les principaux éditeurs du marché des SGBDR (Oracle, IBM, Microsoft) ;
- L'étude et la mise en œuvre des outils de compression de la solution finale ;
- La participation à la réalisation des outils de simulation permettant de calculer les gains attendus;
- L'étude et la mise en œuvre de la solution de réplication qui offre la possibilité d'effectuer les opérations de compression sans arrêter les applications sensibles et sans compromettre ni la sécurité et ni l'intégrité des données;
- La réalisation des opérations de compressions les plus sensibles sur les bases de données relationnelles des applications cœur de métiers (notamment les systèmes de facturation clients) ;
- L'assistance forte au chef de projet pour établir le retour sur investissement sur les gains de stockage sur chacune des lignes d'applications et d'établir le nouvel inventaire du stockage.

Pour des raisons de confidentialité, les données financières et les configurations techniques spécifiques ne figureront pas dans ce mémoire. A la place des chiffres réels nous utiliserons des hypothèses sur les coûts de stockage et les coûts des sauvegardes. Ces hypothèses permettront d'étayer la démarche exposée sur le calcul du retour sur investissement.

Organisation du mémoire

Le **chapitre 1** définit les concepts et les pratiques de la compression des données avec les bases de données relationnelles. Nous présentons différentes solutions de compression au travers des éditeurs Microsoft, IBM et Oracle.

Le **chapitre 2** présente une comparaison des différentes solutions SGBDR et met en évidence les raisons du choix de la solution proposée.

Dans le **chapitre 3** est présentée l'étude de la solution de compression au sein de GDF SUEZ IT. Nous décrivons la mise en œuvre d'un *proof of concept* permettant de valider concrètement les différentes étapes de compression avec Oracle.

Le **chapitre 4** présente la mise en œuvre d'une solution permettant de minimiser les indisponibilités des applications durant les opérations de compression. Les opérations de compression sur les grosses bases de données nécessitent plusieurs jours d'indisponibilité des applications. L'acceptation du projet a nécessité de respecter les engagements des niveaux de services contractés pour chacune des applications.

Le chapitre **conclusion** tire le bilan du projet et élargi le champ d'application à d'autres domaines que ceux de la compression tels que la migration des bases de données vers une infrastructure de type Cloud Computing en s'appuyant sur les outils et les méthodes qui ont permis de minimiser les indisponibilités.

CHAPITRE 1.

CONCEPTS ET METHODES DE LA COMPRESSION DES BASES DE DONNEES.

Nous présentons dans ce chapitre les différents concepts utilisés dans le cadre de la compression des données dans les bases de données relationnelles, en mettant en évidence les raisons et les choix de la solution finale retenue par GDF SUEZ.

1.1 Concepts généraux de la compression des données

1.1.1 Définition de la compression de données.

Pour n'en citer que quelques applications, la compression est utilisée pour les données audio, image et vidéo dans les systèmes multimédias, pour optimiser le stockage de sauvegardes, pour améliorer les performances des entrées/sorties sur disque, pour maximiser la bande passante des réseaux tels qu'internet et bien d'autres usages encore. La compression de données a pour avantage principale de stocker les données compressées dans un espace mémoire utilisant moins d'octets.

Pour illustrer le caractère inéluctable de la compression, nous citerons le passage suivant du livre de David Salomon et Giovanni Motta « *Beaucoup de personnes aiment accumuler des données et détestent jeter quoique ce soit. Peu importe la taille du périphérique de stockage, un jour ou l'autre, il va déborder. La compression des données est utile, car elle retarde cette fatalité.* » [SMC10]

1.1.2 Méthodes de compressions de données.

Les méthodes de compression de données peuvent être réparties en deux grandes familles ; avec perte de données et sans perte de données.

1.1.2.1 Compression avec perte

Les méthodes de compression avec perte permettent d'augmenter le taux de compression en concédant une certaine perte de précision. La compression avec perte se révèle efficace lorsqu'elle est appliquée à des images graphiques (JPEG) et de la voix numérisée (MP3).

L'algorithme RLE (*Run Length Encoding*) fait parti des méthodes de compression avec perte. Il est le plus répandu pour les formats de fichier BMP, JPEG (images fixes), ou MPEG (vidéo).

1.1.2.2 Compression sans perte

La compression sans perte est constituée de techniques et algorithmes qui garantissent une restitution exacte des données après un cycle de compression-décompression. Ce type de compression est utilisé pour les bases de données, les feuilles de calcul, les documents de type texte. Pour ces applications, la perte d'un seul bit de données pourrait avoir des conséquences graves. Parmi les catégories de compression sans perte, nous pouvons citer :

- La compression par statistiques ;

- L'utilisation d'un dictionnaire.

La compression par statistiques : codage de Huffman.

Le codage de *Huffman* est un algorithme de compression des données basé sur les fréquences d'apparition des caractères apparaissant dans le flux d'origine. Les symboles les plus courants sont encodés en utilisant moins de bits que les symboles moins fréquents. Pour chaque symboles, l'algorithme établit une table de fréquence d'apparition des caractères dans le flux d'origine et associe un poids valant son nombre d'occurrences et construit un arbre où chaque caractère constitue une feuille de l'arbre.

La compression par dictionnaire

Le principe général de la compression par dictionnaire est d'établir une sélection de chaînes répétitives enregistrées dans un dictionnaire au fur et à mesure de la lecture du flux d'origine. Les chaînes répétitives et dupliquées sont encodées sous forme d'entier avec beaucoup moins de bits que les chaînes réelles. Elles sont remplacées par des pointeurs dont les valeurs de références sont stockées dans le dictionnaire. Plus les données sont redondantes, plus le taux de compression est élevé. La compression par dictionnaire utilise soit un dictionnaire statique soit un dictionnaire dynamique. En général, la compression avec un dictionnaire dynamique est préférée. Au commencement le dictionnaire est vide ou minimal et il est enrichi au fur et à mesure.

La compression par dictionnaire et la compression par statistiques peuvent être combinées dans la même méthode.

Les opérations de décompression

Les opérations de décompression sont effectuées par le décodeur

Nous allons présenter dans la suite de ce chapitre différentes solutions SGBDR qui prennent en charge la compression des données et utilisent ces techniques de compression. Selon la redondance des données, la compression par dictionnaire conduit généralement à une réduction entre 2 et 8 fois de la taille des tables et permet de diminuer nettement la consommation d'espace disque de l'ensemble des fichiers constituant les bases de données.

1.2 Méthodes de compression des bases de données relationnelles

1.2.1 Compression avec Oracle.

Depuis la version 9i de son SGBDR, Oracle propose la fonctionnalité de compression des tables qui permet d'éliminer les valeurs dupliquées. Cette fonctionnalité supporte uniquement les chargements en masse par lot (*bulk load*), comme c'est le cas avec les entrepôts de données, mais ne fonctionne pas avec les mises à jour des systèmes transactionnels où les mises à jour sont unitaires. Oracle a ensuite enrichi ses offres en apportant des innovations technologiques :

- l'option payante d'Oracle nommée « *Advanced compression* » disponible depuis la version 11g, qui permet désormais aux environnements transactionnels (OLTP) de bénéficier des fonctionnalités avancées de compression à l'instar des entrepôts de données ;
- L'option de compression hybride « *Hybrid columnar compression* » basée sur un mode de stockage en colonne d'abord disponible avec les *appliances* Exadata d'Oracle et qui a été intégrée récemment dans la version 12C sur toutes les plateformes couvertes par l'éditeur;

Nous abordons ces différentes innovations technologiques qui permettent de compresser les données de tables tout en conservant les performances des requêtes. Nous décrivons également d'autres fonctionnalités d'Oracle destinées à la compression des index et à la compression des larges objets (LOB) qui sont complémentaires de la compression des tables, et qui améliore de manière significative le taux global de compression des bases de données.

1.2.1.1 Compression des tables avec l'option « *Advanced compression* »

Cette compression introduite par Oracle dans les versions 9i, 10g, et enrichit avec la version 11g, concerne le stockage en mode ligne et repose sur les techniques de compression par dictionnaire tel que décrit au § 1.1.2.2 Compression sans perte.

L'*Advanced compression* est une option payante proposée par Oracle à partir de la version oracle 11g et qui s'applique aussi bien pour les environnements décisionnels que les environnements transactionnels (OLTP). Un bloc Oracle contient le dictionnaire de données (la définition des colonnes de la table) et les enregistrements de la table les uns à la suite des autres.

L'option *Advanced compression* utilise un algorithme de compression par dictionnaire spécialement conçu pour les applications transactionnelles (OLTP). Lorsqu'une table est définie compressée par l'option « COMPRESS FOR OLTP », les données ciblées par la compression et qui apparaissent dans les lignes du bloc sont remplacées par des pointeurs qui pointent vers les données de la table symbole.

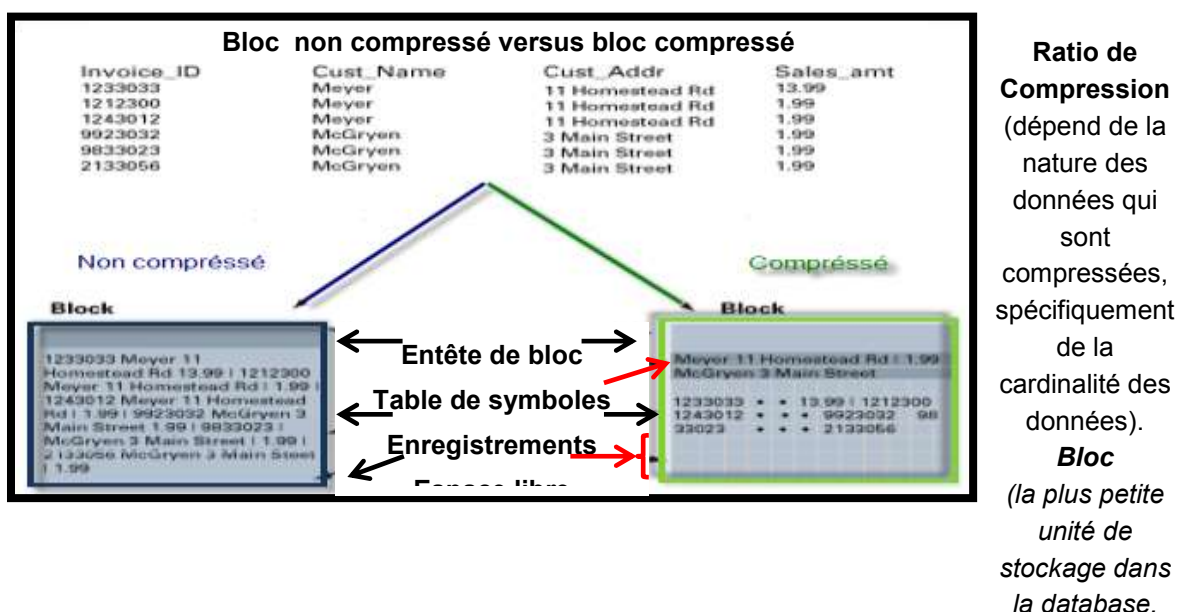


Figure 1-1 Compression d'un bloc de données Oracle [OAC12]

Compression asynchrone

L'*Advanced compression* d'Oracle s'effectue de manière asynchrone. Un bloc qui vient d'être initialisé reste non compressé jusqu'à ce que son seuil de remplissage soit atteint. Ce mécanisme permet de ne pas augmenter les attentes d'écritures sur les transactions. Lorsque le bloc atteint le seuil de déclenchement de la compression, l'ensemble des enregistrements contenus dans le bloc est compressé en une seule opération, non prioritaire par rapport aux opérations de mises à jour.

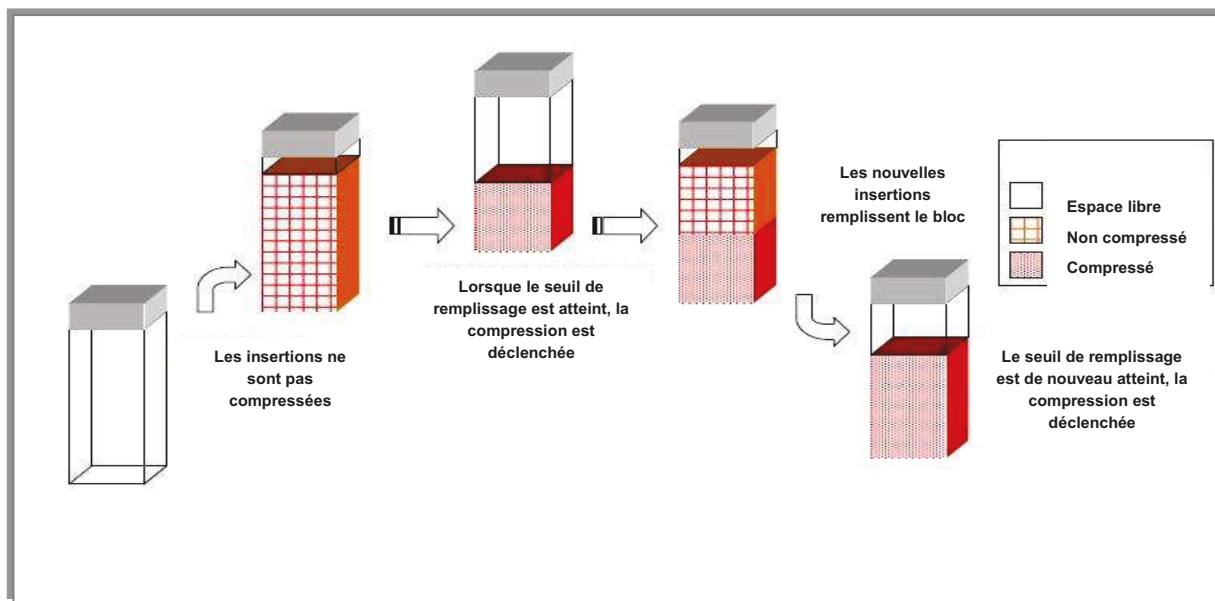


Figure 1-2 Mécanisme asynchrone de l'option « *Advanced compression for Oracle* » [OAC12]

Estimation du ratio de compression

Pour les migrations de la version Oracle 9i et de la version Oracle 10g vers la version Oracle 11g, Oracle fournit un outil permettant d'estimer les futurs taux de compression sur la version cible 11g. À partir de sa version 11g, Oracle fournit un package standard PL/SQL « DBMS_COMPRESSION » permettant d'effectuer une analyse par table en considérant un échantillon, et fournit, en sortie, le ratio de compression de la table. Il suffit ensuite de diviser la taille réelle de la table par le ratio de compression pour obtenir la taille de la table compressée. Ce package standard est très pratique, car il permet de faire des estimations sur l'ensemble des tables afin d'obtenir un ratio global de l'ensemble des tables, et ces estimations se sont révélées être proches du résultat final.

Impacts sur les mises à jour

Les insertions ne sont pas compressées à la volée et ne provoquent pas de surcoût des écritures.

Dans le cas d'une modification, Oracle ne maintient que les symboles (moins de bits) et non plus les valeurs réelles qui figurent alors dans le dictionnaire de compression. Au niveau du dictionnaire de compression, pour chaque valeur distincte un compteur indique le nombre d'occurrences au niveau du bloc portant cette valeur. Lors des modifications, les compteurs associés aux valeurs dans le dictionnaire doivent être maintenus.

Pendant les opérations de suppression, la gestion du dictionnaire entraîne un léger surcoût car il est nécessaire de maintenir l'ensemble des valeurs et de leurs compteurs dans le dictionnaire de compression.

Optimisation de la mémoire cache.

Le moteur Oracle est capable de stocker un bloc compressé tel quel dans sa mémoire cache et de le lire sans le décompresser, ce qui permet de stocker plus de données dans le cache, et donc d'améliorer son efficacité.

Optimisation des accès aux données

Afin de tirer au maximum parti de la compression, Oracle a mis en œuvre un ensemble d'optimisations. Une des principales optimisations porte sur les prédicats des requêtes. Cette optimisation consiste à n'évaluer qu'une seule fois la valeur recherchée d'une colonne compressée dans un bloc au lieu de l'évaluer sur tous les enregistrements, comme cela est le cas pour un bloc non compressé.

Dans le cadre d'une consommation de CPU ne dépassant pas 80% des ressources machines avant la compression, l'*Advanced compression* d'Oracle offre systématiquement des gains de performances sur les lectures, hors potentiellement les requêtes dont les plans d'exécution se dégradent.

La diminution des attentes liées aux lectures/écritures améliorent nettement les temps des traitements par lot (traitement batch). Dans un temps donné le nombre des lectures/écritures augmente donc et génère une augmentation de la consommation CPU. La figure ci-dessous montre la courbe de la consommation CPU avant compression durant la nuit applicative dans un environnement SAP d'un grand acteur de l'énergie :

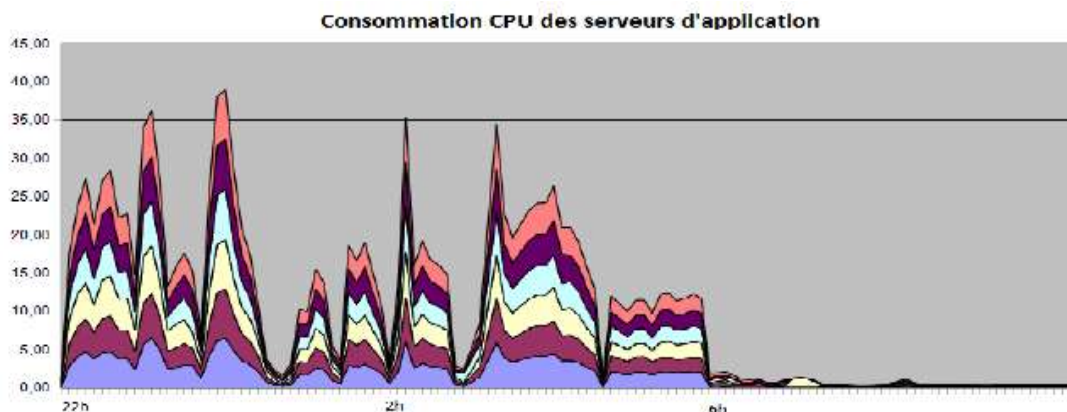


Figure 1-3 Traitement batch avant compression [EPS10]

Une fois la base de données Oracle compressée, on constate une augmentation des cycles de la CPU de l'ordre de 15%. Par ailleurs le temps des traitements batch diminue de plus de 3 heures :

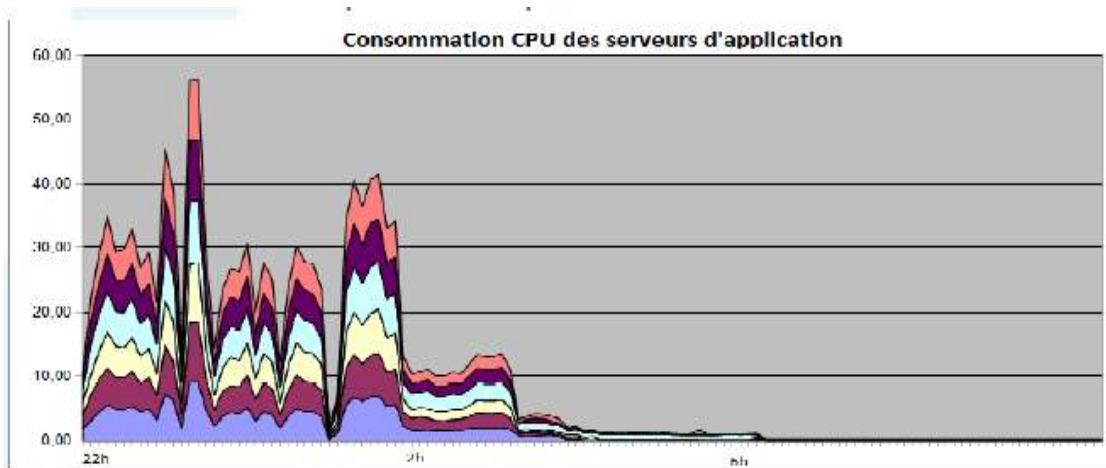


Figure 1-4 Traitement batch après compression [EPS10]

1.2.1.2 Compression des tables avec l'option « Hybrid Columnar Compression ».

Le stockage des enregistrements comportant des colonnes de différents types limite l'efficacité de la compression alors que le stockage en mode colonne offre un ratio de compression nettement supérieur. Néanmoins, ce mode de stockage présente des inconvénients majeurs pour les requêtes.

En effet un mode de stockage en colonne nécessite à minima une entrée/sortie pour chaque colonne pour accéder à l'enregistrement complet. Les tables comportant plus d'une centaine de colonnes, comme c'est le cas dans les entrepôts de données, sont un frein évident du point de vue des performances. L'accès à un seul enregistrement nécessite, dans ce cas, plus d'une centaine d'entrées/sorties.

Oracle propose depuis la version Oracle 12c ou pour les *appliances* telles que Exadata, un mode de stockage hybride pour stocker les données « Hybrid Columnar Compression ». Cette technologie combine à la fois le mode de stockage en ligne et le mode de stockage en colonne. Il s'agit d'une « méthode hybride qui améliore le taux de compression tout en conservant les performances d'accès » [OHC12]. Les lignes sont organisées par colonnes (C1, C2, C3, ...) au sein d'une unité logique nommée « Logical Compression Unit » qui est constituée de plusieurs blocs Oracle comme l'indique la figure ci-dessous :

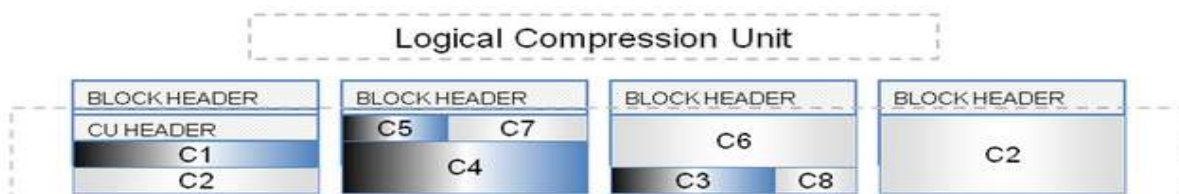


Figure 1-5 Unité de compression avec l'option "Hybrid Columnar Compression" [OHC12]

L'*hybrid columnar compression* supporte les index de type *B*Tree*, les index BITMAP et les *full text* index.

Impacts sur les mises à jour

L'*hybrid columnar compression* ne supporte pas les mises à jour unitaires comme c'est le cas dans les environnements transactionnels. Les mises à jour unitaires provoquent une dégradation de l'effet de la compression.

Les données sont en fait insérées par groupe de lignes où chacune des colonnes est compressée séparément. Lorsque que la compression par colonne est terminée, le groupe de lignes est inséré dans l'unité de compression. Au niveau des entrées/sorties Oracle n'effectue qu'un seul accès à l'unité logique de compression.

Un ensemble de contraintes techniques doit être respecté avec l'option « *Hybrid Columnar Compression* », tels que :

- Les ordres SQL INSERT doivent utiliser le tag APPEND qui indique à Oracle d'éviter les écritures en mode conventionnel (Conventional path) et de forcer les accès directs lors des insertions (direct path) ;
- Les ordres DML (Insert, update, delete) doivent utiliser les options de parallélisme ;
- Seuls les chargements de fichiers avec l'outil Oracle SQL*LDR en mode *direct path* sont supportés.

Ce système de compression est donc plutôt destiné aux entrepôts de données ou aux systèmes décisionnels. Pour les environnements transactionnels, Oracle préconise plutôt dans ce cas le mode de stockage en ligne avec son option de compression « *Oracle Advanced compression* ».

Estimation du ratio de compression

De même que pour l'*Advanced Compression*, le package *DBMS_COMPRESSION* permet d'estimer le ratio de compression par table pour l'option hybride en mode colonne.

1.2.1.3 Compression des index *B*Tree*

La compression des tables n'est pas suffisante à elle seule. En effet, les bases de données transactionnelles ou les entrepôts de données notamment avec les ERP tels que SAP comportent énormément d'index, parfois plus de 7 index par table. La compression d'index doit donc être également envisagée.

Oracle utilise deux types de structure : les index de type *Btree+* et les index de type *Bitmap*. Les index sur fonction, les reverse index, les bitmap *join* index, les *full text* index sont simplement des variations de ces deux principaux types. La compression d'index s'applique uniquement aux index de type *B*Tree+*. Les index de type *Bitmap* ne sont pas concernés.

Un index *B*tree* repose sur une structure de données en arbre équilibré composé de blocs branches et de blocs feuilles (appelé B-arbre par analogie au terme anglais « *B*Tree* ». Un index de type *B*Tree+* est un index de type *B*Tree* amélioré où toutes les données sont stockées exclusivement dans les feuilles de l'arbre et celles-ci sont reliées entre elles. Les blocs des niveaux supérieurs (blocs branches) contiennent les pointeurs des blocs de niveaux inférieurs. Le niveau le plus bas de l'arbre contient les blocs feuilles avec les valeurs indexées et, pour chacune de ces valeurs, l'identifiant de l'enregistrement associé dans la table, appelé ROWID avec Oracle.

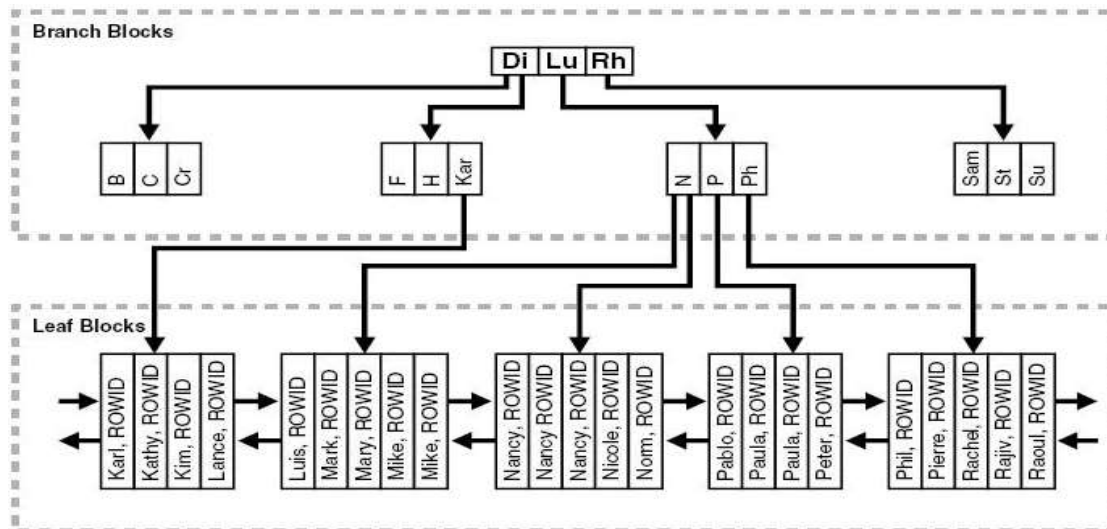


Figure 1-6 - Structure interne d'un index de type B*Tree. [OWN05]

(1) La structure de l'identifiant d'enregistrement ROWID est la suivante :

<Identifiant de Table><Numéro du fichier Oracle><Numéro du bloc dans le fichier><Numéro de la ligne dans le bloc>

La compression des index B*Tree repose également sur des techniques de compression par dictionnaire. Oracle gère, comme pour les tables, au sein de chaque bloc feuille, un dictionnaire d'encodage contenant les combinaisons des valeurs distinctes et assigne à chacune d'elle un entier.

Il apparaît judicieux de compresser uniquement les index qui comportent des clés avec beaucoup de valeurs dupliquées. Concernant un index unique, la compression d'index n'est possible que sur les N-1 premières colonnes. Pour les index non uniques, la compression peut s'appliquer sur toutes colonnes.

Oracle fournit des méthodes pour calculer le degré de compression des index. Ce degré dépend du nombre de colonnes et du taux de duplication de chaque colonne.

1.2.1.4 Compression des objets larges (LOB)

Les objets larges ou LOB correspondent à deux types sous Oracle :

- Le type CLOB permet de stocker des objets de types caractères (fichier texte, fichier XML, ..) allant jusqu'à 4 Go.
- Le type BLOB permet de stocker des objets de type binaires (photos, images, vidéo) allant jusqu'à 4 Go.

Un LOB de petite taille (< 4 Ko) est stocké automatiquement dans le même segment physique que celui de la table. Au-delà de ce seuil les données sont stockées dans un segment physique à part. La version 11g Release 1 a introduit la notion de *Securefile* pour stocker les champs de type LOB et les compresser. Elle propose plusieurs méthodes de compression.

Méthode « *Advanced LOB Compression* »

Cette méthode utilise les algorithmes standards de compression des fichiers textes, des images, des vidéos à l'instar de la commande *gzip* sous Windows ou Unix. Il existe trois niveaux de compression : *LOW*, *MEDIUM*, *HIGH*. Par défaut c'est le niveau *MEDIUM* qui est utilisé et qui offre un bon compromis entre le ratio de compression et l'augmentation de consommation CPU (un léger surcoût de la CPU allant de 3 à 5%). L'option *LOW* permet d'approcher 80% du taux de compression de l'option *MEDIUM*, tout en surconsommant trois fois moins de CPU. L'option *HIGH* permet d'atteindre le meilleur taux de compression mais elle génère par contre un surcoût significatif de consommation CPU.

Méthode « *Advanced LOB Deduplication* »

Il est très courant pour les applications de stocker plusieurs copies de pièces jointes ou plusieurs exemplaires de fichiers. Nous pouvons citer l'exemple d'une application de type CRM dont les nombreux clients reçoivent les mêmes pièces jointes. L'option « *Advanced LOB Deduplication* » élimine les copies des pièces jointes en doublon et ne conserve qu'un seul exemplaire du fichier dans un *LOB* de type *Securefile*. Les copies sont remplacées par des références qui pointent sur ce fichier.

1.2.1.5 Ratio de compression final

Dans le cadre d'un projet de compression des bases de données avec Oracle, la compression d'index et des *LOB* doit être mise en œuvre conjointement avec les options de compression des tables (*Advanced compression* ou *Hybrid Columnar compression*) et permet ainsi d'améliorer nettement le ratio de compression final. Dans les environnements ERP tels que SAP, la mise en œuvre des différentes méthodes de compression (tables, index, *LOB*) nous a permis d'atteindre un ratio de compression allant de 2,2 à 3,7 (ratio de compression = volumétrie initiale / volumétrie compressée). Le tableau ci-dessous présente les ratios de compression obtenus dans les environnements SAP d'un grand acteur de l'énergie. Les résultats obtenus avec l'*Advanced compression* d'Oracle et la compression des index et des *LOB* sont très significatifs :

	Taille de la base sans compression (GB)	Taille de la base avec compression (GB)	Taux de compression Tables+index		Economies de stockage Tables + Index GB
			%	Ratio	
SAP CRM	2000	922	53,90%	2,17	1078
SAP ECC 6 pour le secteur énergie	8600	2300	73,26%	3,74	6300
SAP BW 7.0	3700	1300	64,86%	2,85	2400
(1) Table DFKKOP	1332	165	87,61%	8,07	1167

Tableau 1-1 Ratios de compression dans des environnements SAP avec l'Advanced compression d'Oracle [EPS10].

(1) Le volume initial de la table SAP DFKKOP (qui contient les postes des pièces comptables) a été divisé par 8.

1.2.2 Compression avec DB2 (IBM)

Sur les environnements de type Unix, linux et Windows, la version DB2 se nomme LUW. Les deux types de versions sont tellement différents qu'ils requièrent chacun des compétences spécifiques pour l'administration. La version LUW 10.5 pour Unix et Windows est la dernière version du produit DB2.

Les techniques de compression de DB2 « reposent sur l'utilisation d'un dictionnaire basé sur l'algorithme LZ (Lempel-Z) » [ICS12] et tel que décrit au § 1.1.2.2 Compression sans perte. DB2 propose plusieurs types de compression :

- La compression par table ;
- La compression par page ;
- La compression adaptative ;
- La compression par colonne.

1.2.2.1 Compression par table

Cette technique de compression par dictionnaire a été introduite avec la version 9 de DB2 et a été enrichie au fur et à mesure. Les séquences répétitives peuvent concerner plusieurs colonnes à la fois ou une sous chaîne appartenant à une colonne spécifique. Les données ne sont compressées que si elles apportent une économie de stockage. Dans tous les cas, la ligne est compressée intégralement avec un jeu de symboles mais elle n'est jamais compressée partiellement. La figure ci-dessous illustre la compression par table avec DB2 :

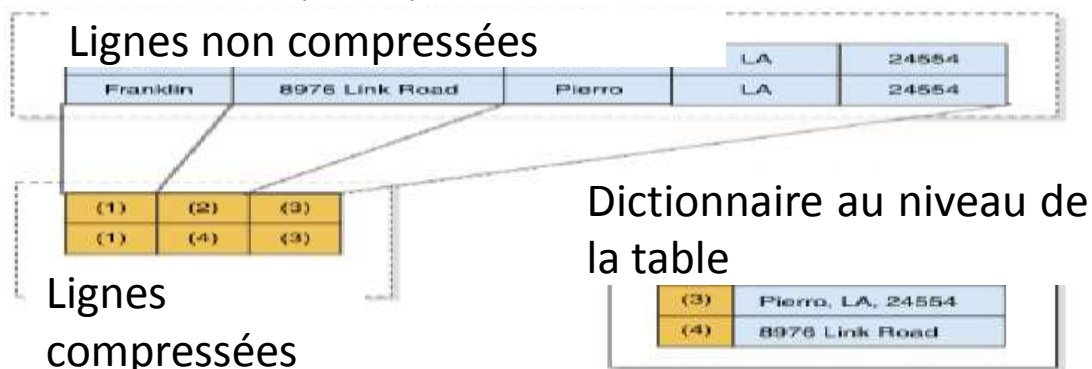


Figure 1-7 Compression par table avec DB2 [ICS12]

Limitation des effets de la compression

Au fur et à mesure des mises à jour, les dictionnaires de compression des tables qui sont fréquemment mises à jour ne reflètent plus les données, ce qui conduit à une dégradation du taux de compression. Dans de tels cas, il est nécessaire de prévoir des réorganisations régulières des tables pour reconstruire le dictionnaire de compression et maintenir le ratio de compression. Ces réorganisations des tables peuvent engendrer des indisponibilités des applications, ou des dégradations de performance.

1.2.2.2 Compression par page « *classic row compression* »

La compression par page est apparue avec la version DB2 9 et a été améliorée à partir de la version DB2 9.5, les dictionnaires de compression sont créés automatiquement au niveau de chaque page et sont enrichis au fur et à mesure si les trois conditions suivantes sont satisfaites :

- L'option COMPRESS de la table vaut 'YES' ;
- Il n'existe aucun dictionnaire de compression ;
- Le volume de la table dépasse le seuil de 2 Mo, et offre ainsi suffisamment de données pour construire le dictionnaire de compression.

DB2 lors de la création des dictionnaires des pages scan un échantillon de la table qui ne dépasse pas 2 Mo afin de limiter les impacts sur les performances.

Augmentation de la CPU

En moyenne, on observe avec DB2 dans les environnements tels que SAP une augmentation de la CPU comprise entre 1 et 10%. Dans la plus part des cas, l'augmentation des cycles CPU due à la compression peut être compensée par des gains sur les opérations de lectures/écritures. Il peut être nécessaire d'augmenter la capacité de la CPU afin de maintenir les performances.

1.2.2.3 Compression adaptative « *compression dynamique* »

Depuis la version 10.1 nommée Galileo, DB2 offre une nouvelle méthode de compression « Compression adaptative » qui enrichit les fonctionnalités de la compression par page et évite les réorganisations. C'est une option de la version « Enterprise Edition » de DB2 qui utilise à la fois la compression par table et la compression par page vues précédemment. La compression par table élimine les valeurs répétitives au niveau global, alors que la compression par page tient compte des spécificités des valeurs dupliquées au niveau de chaque page. Pour activer la compression adaptative il suffit d'indiquer la clause COMPRESS YES ADAPTATIVE dans les ordres DDL « CREATE TABLE » et « ALTER TABLE ». La figure ci-dessous illustre comment la compression par table et la compression par page se complète mutuellement :

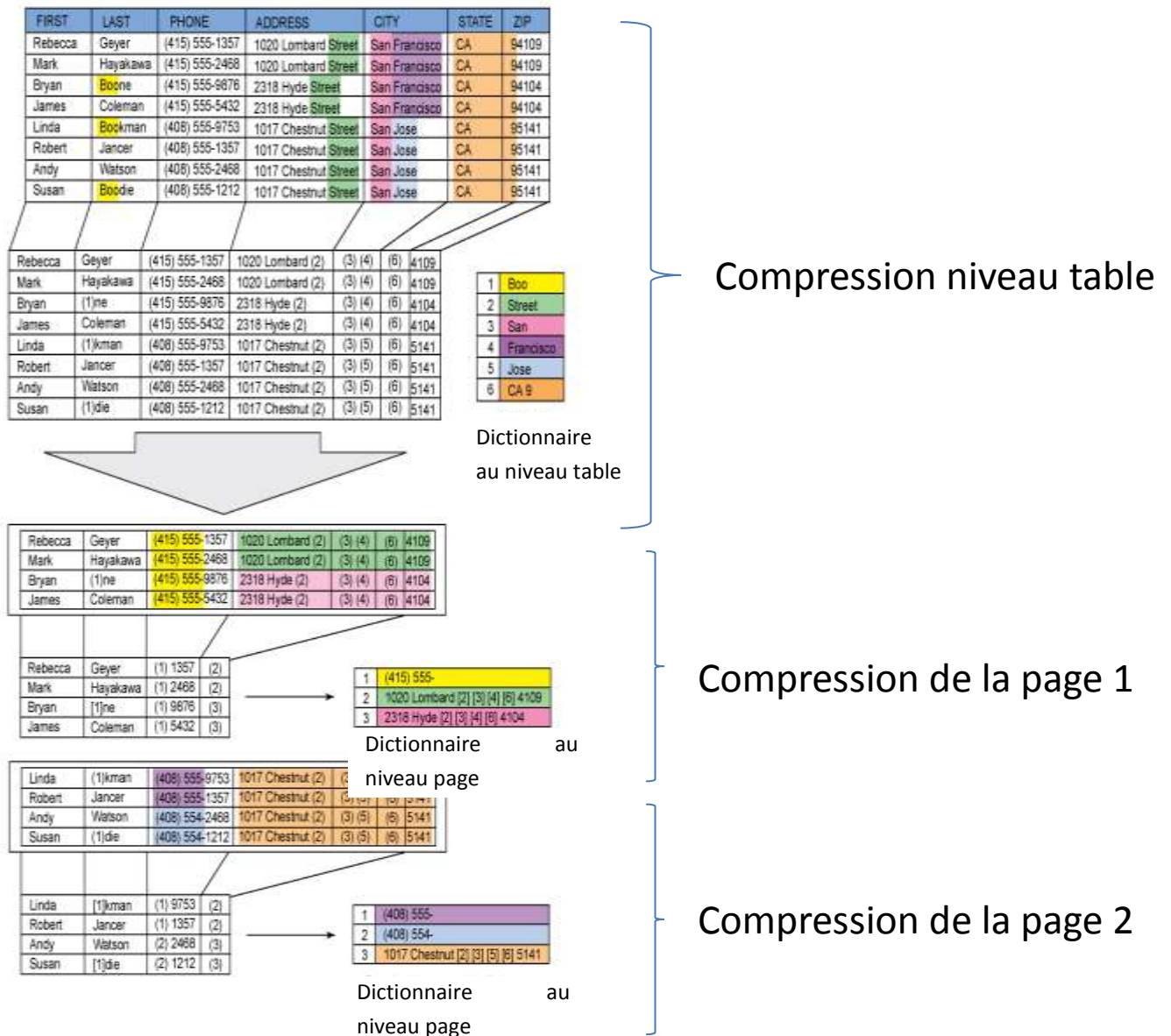


Figure 1-8 Compression adaptative de DB2 [ICS12].

Compression dynamique

Les dictionnaires au niveau des pages sont mis à jour automatiquement. Quand une page est remplie, un processus DB2 spécifique, crée le dictionnaire de compression pour les données de la page. Au fil du temps, le processus DB2 détermine lui-même quelles sont les pages qui nécessitent une reconstruction de leur dictionnaire. La compression adaptative de DB2 est comparable à l'option de compression dynamique d'Oracle « *Advanced compression* ».

Ratio de compression

L'option « *Adaptive compression* » de DB2 permet d'économiser en général entre 75% et 85% du volume initial des tables. Elle économise en général entre 5% et 15% de plus d'espace disque que la compression par table [IRS12].

Optimisation de la mémoire cache.

Avec la compression adaptative, le cache de la mémoire partagée devient plus efficient et permet de stocker plus de données. En effet les données compressées sont stockées directement telles quelles en mémoire à l'instar de l'option « Advanced compression » d'Oracle.

Optimisation des accès aux données

La compression adaptative de DB2 Galileo apporte une meilleure utilisation de la bande passante du canal des entrées / sorties. On constate le plus souvent sur une période donnée une diminution des attentes liées aux entrées / sorties physiques et une augmentation des lectures logiques (*logical reads*) des requêtes.

Le fait que la mémoire cache contienne plus de lignes, permet d'améliorer les performances des lectures logiques. La figure ci-dessous montre une comparaison de l'utilisation de la bande passante des I/O avant et après la mise en œuvre de la compression.

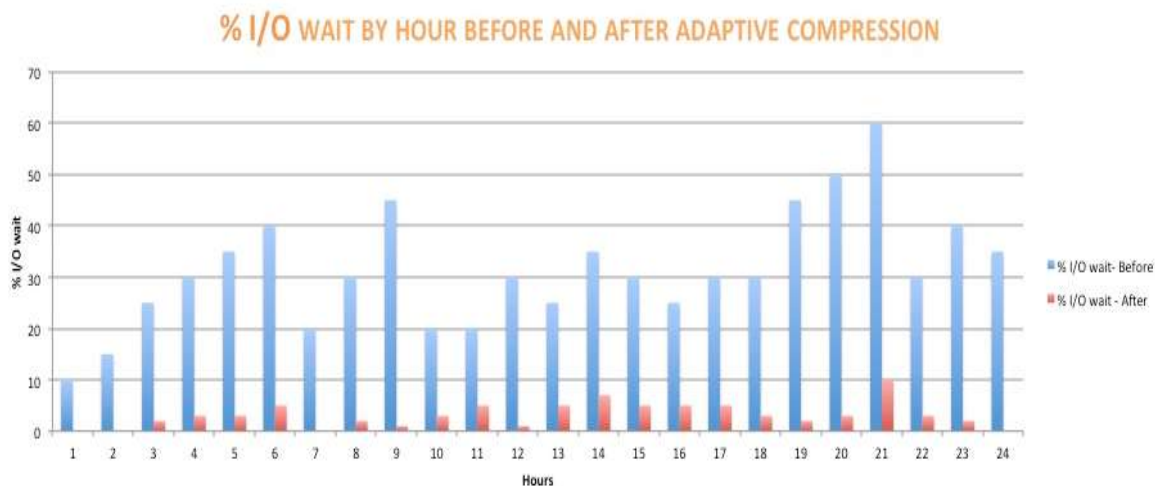


Figure 1-9 Comparaison de l'utilisation de la bande passante des I/O avant et après la compression [RAC12].

Impact sur l'utilisation de la CPU

La consommation CPU augmente en moyenne de 3% à 6% selon IBM.

1.2.2.4 Compression des index

La compression d'index est disponible depuis la version 9.7 de DB2.

Les index qui sont créés sur une table compressée et qui méritent d'être compressés sont compressés par défaut. Les index compressibles sont en général les index *B*Tree* composites (index composés de plusieurs colonnes) ou les index contenant des chaînes de caractères variables. La compression d'index est comparable à la compression adaptative. Le moteur DB2 maintient au fil des mises à jour les dictionnaires de compression de chaque page composant les feuilles de l'index. DB2 fournit une fonction qui permet d'estimer le ratio de compression des index (ADMIN_GET_INDEX_COMPRESS_INFO).

1.2.2.5 Compression des tables temporaires

Les tables temporaires sont utilisées en général par les applications pour stocker des résultats intermédiaires ou bien par le SGBDR lui-même pour effectuer des tris des requêtes, ou des jointures de type *Hash* ou lors de la création des index. La compression adaptative est automatiquement activée sur les tables temporaires.

1.2.2.6 Compression des objets larges.

Les données de type *LOB* de type « en ligne » qui sont contenues dans le segment de la table sont compressées automatiquement lors de l'activation de la compression par table. Les données *LOB* stockées en dehors du segment de la table ne sont pas compressibles avec DB2.

1.2.2.7 Compression par colonne

La version 10.5 de DB2 offre un ensemble d'innovations technologiques intitulées « *BLU Acceleration* ». Elle amène un stockage natif en colonnes, permettant d'atteindre de hauts niveaux de compression et de monter en mémoire les données les plus utilisées. Le ratio de compression obtenu est en général 10 fois supérieur comparé au mode de stockage en ligne classique.

Ce mode de compression est supporté depuis fin 2013 par SAP sous certaines conditions spécifiques (cf. SAP note 1819734).

1.2.2.8 Ratio de compression final

Les clients SAP qui mettent en œuvre la compression avec DB2 ont constaté des réductions importantes de l'espace de stockage de leurs bases de données. La compression d'index amène une économie d'espace de stockage supplémentaire de 10% à 20%. Avec l'option de compression par page classique « *Classic row Compression* », le ratio de compression final dépasse parfois 2,5.

	Taille de la base sans compression (GB)	Taille de la base avec compression (GB)	Taux de compression Tables+index		Economies de stockage Tables + Index GB
			%	Ratio	
SAP Netweaver pour le secteur bancaire	103	41	60,19%	2,51	62
SAP BW 7.0	1640	890	45,73%	1,84	750

Tableau 1-2 DB2 Best practices Using Compression Feature in SAP environment [ICS12].

Depuis la version 10, la compression adaptative permet d'améliorer le ratio de 30% par rapport à la compression par page classique « *classic row compression* ». Le tableau ci-dessous est une estimation en projetant les ratios de la « *classic compression* » présentés dans tableau précédent :

	Taille de la base sans compression (GB)	Taille de la base avec compression (GB)	Taux de compression Tables+index		Economies de stockage Tables + Index GB
			%	Ratio	
SAP Netweaver pour le secteur bancaire	103	31	60,19%	3,27	69
SAP BW 7.0	1640	686	45,73%	2,39	954

Tableau 1-3 Ratio de compression avec l'option adaptative compression de DB2 [ICS12].

1.2.3 Compression avec SQL Server (Microsoft)

SQL Server est un SGBDR développé et commercialisé depuis 1989 par la société Microsoft. Ces principaux langages de requête sont T-SQL et ANSI SQL. La dernière version est SQL Server 2014. SQL Server « offre la possibilité de compresser les tables et les index, cette fonctionnalité a été introduite dans SQL Server 2008 et elle a été améliorée dans la version 2012 » [MPS13]. Elle n'est disponible que dans l'édition Enterprise uniquement.

De manière générale, la compression des tables avec SQL Server s'opère sur les pages en utilisant les mécanismes de compression par dictionnaire décrits au § 1.1.2.2 Compression sans perte. Sous SQL Server la notion de page est équivalente à celle du bloc Oracle.

Une page est l'unité la plus petite de l'espace de stockage du SGBDR dont la taille peut varier de 4K à 64 K.

La compression SQL Server utilise comme pour Oracle et DB2 des techniques de compression par dictionnaire (cf § 1.1.2.2 Compression sans perte) et fait intervenir différents niveaux de compression :

- la compression par ligne (*Row Compression*) ;
- la compression par page

1.2.3.1 La compression par ligne « *Row compression* »

La compression au niveau ligne permet d'optimiser l'espace requis pour les types de taille fixe. Dans de nombreuses situations, SQL Server utilise plus d'espace que nécessaire pour stocker les valeurs, et sans ce type de compression, le seul contrôle disponible est d'utiliser un type de données de longueur variable.

La compression en ligne peut être mise en œuvre isolément sans la compression par page, elle permet parfois d'atteindre un ratio de compression intéressant sans trop impacter la charge CPU.

1.2.3.2 La compression par page

La compression par page de SQL Server repose sur des techniques de compression par dictionnaire comparables à celles d'Oracle et de DB2, hormis le fait qu'elle travaille, en plus, sur les préfixes des valeurs et pas seulement sur les valeurs entières. Dans certains cas, si la valeur de la colonne ne correspond pas exactement à la valeur de préfixe, une correspondance partielle est possible. A l'instar de DB2, la compression par page de SQL Server s'effectue en considérant l'ensemble des colonnes. La figure illustre un exemple de compression par page :

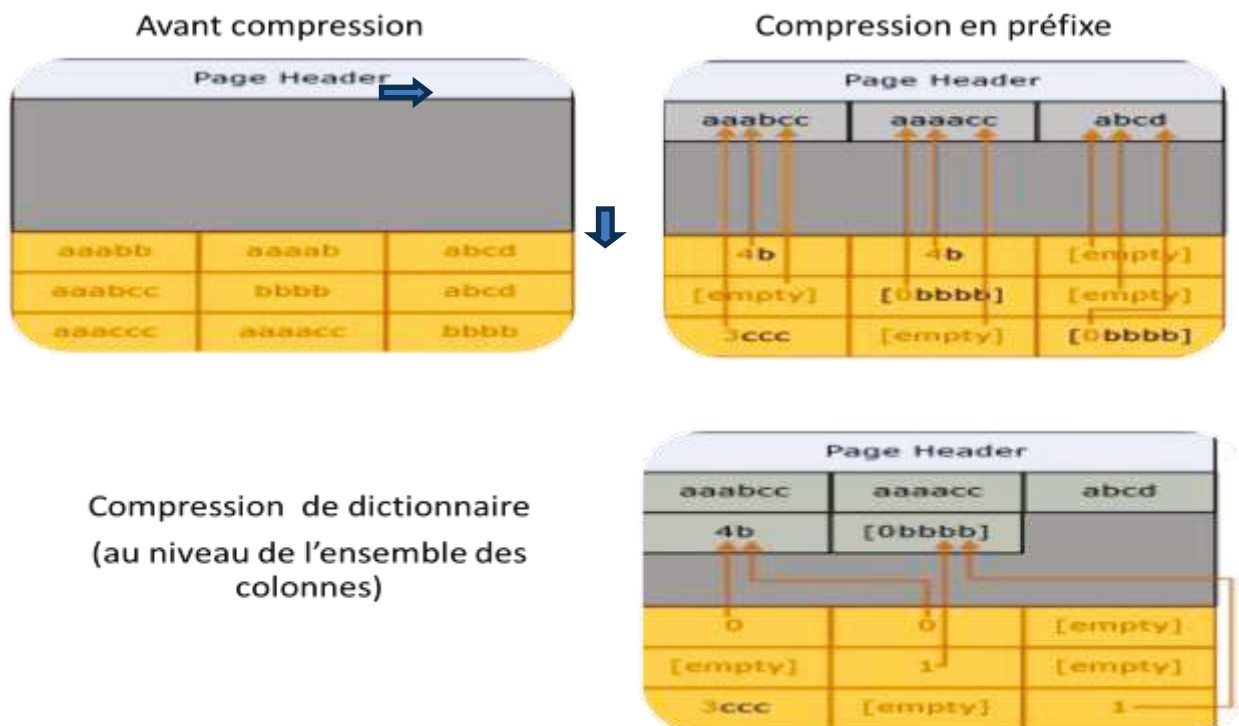


Figure 1-10 Compression par page avec SQL Server 2014 [MTC14]

La phase de compression par préfixe possède des caractéristiques similaires à celles de l'algorithme d'*Huffman* qui est une compression de type statistique (cf § 1.1.2.2 Compression sans perte). Nous sommes en présence ici d'une méthode de compression mixte comportant à la fois des règles de compression par statistiques et des règles de compression par dictionnaire.

1.2.3.3 Optimisation de la mémoire cache

Les données compressées sont stockées directement telles quelles en mémoire à l'instar de l'option « Advanced compression » d'Oracle et de la compression adaptative de DB2.

1.2.3.4 Limitation sur les performances

SQL Server n'offre pas d'optimisation sur les prédicats des requêtes avec les données compressées. Lors de l'évaluation des prédicats des requêtes (filtres), des jointures, des tris les données sont décompressées (uniquement les données nécessaires et non pas la page intégrale).

SQL Server n'offre pas d'optimisation des ordres DML Update pour les données compressées.

1.2.3.5 Augmentation de la CPU

Avec SQL Server on constate en moyenne une augmentation de la charge CPU d'environ 10% avec la compression en ligne. Depuis 2011, l'éditeur SAP préconise avec son ERP, pour les installations utilisant SQL Server, de mettre en œuvre la compression par page. Par contre la compression par page génère une augmentation de la charge CPU plus importante. Pour ne pas engendrer une surcharge CPU trop importante, Microsoft préconise de sélectionner au cas par cas les tables candidates à la compression. On trouve sur le site de Microsoft

« technet.microsoft.com » les recommandations suivantes : « *Commencez avec des tables et des index peu fréquemment utilisés pour s'assurer que les ressources systèmes répondent correctement à la compression. Si le système ne dispose pas de marge CPU suffisante, ne pas mettre en œuvre la compression de page sans effectuer des tests approfondis* » [UMC08].

1.2.3.6 Compression des index

Sous SQL Server la compression par page est similaire pour les tables et les index. Les mêmes opérations s'appliquent donc aussi bien pour la compression des tables que pour celle des index.

Lors de la compression des Index de type *B-tree*, seuls les nœuds feuilles sont compressés. Pour des raisons de performances, les niveaux des nœuds branches sont laissés non compressés.

Une étude de la société UNISYS indique que les temps d'exécution pour reconstruire des index, augmentent significativement et varient selon les types de compression comme l'indique la figure ci-dessous :

Test	Data Compression	Ratio
1	Uncompressed	1.00
2	ROW compressed	1.41
4	PAGE compressed	2.40

Tableau 1-4 ratio des temps d'exécution pour reconstruire les index avec SQLServer [UMC08].

1.2.3.7 Compression des objets larges.

Seules les données de type *LOB* de type « en ligne » qui sont contenus dans le segment de la table sont compressées automatiquement lors de l'activation de la compression par table. Les données *LOB* stockées en dehors du segment de la table ne sont pas compressibles avec SQL Server.

1.2.3.8 Ratio de compression final

Une étude en 2008 par la société UNISYS publie les ratios de compression obtenus avec SQL Server version 2008 avec les différents types de compression. Le ratio de compression est défini de la manière suivante :

Ratio de compression = Taille non compressée / Taille compressée.

Test	Data Compression	Initial Data Compression Ratio	Initial Database Space Savings
1	Uncompressed	1.00	0%
2	ROW	1.21	18%
3	PAGE&ROW	1.42	30%
4	PAGE	1.85	46%
5	PAGE-All-Tables	1.86	46%

Tableau 1-5 ratios de compression selon les types de compressions avec SQLServer (UMC08]

D'après le tableau ci-dessus, nous pouvons constater que le meilleur ratio de compression obtenu est de celui de la compression par page (PAGE et PAGE-All-Table) : 1,85 et 1,86 (compression par page de toutes les tables). Le fait de mixer la compression par ligne et la compression par page (PAGE&ROW) n'apporte pas de gains aussi importants.

1.3 Comparaison des solutions de compression

1.3.1 Synthèse comparative des éditeurs Oracle, DB2, SQL Server

Les éditeurs Oracle, IBM et Microsoft ont tendance à favoriser chacun leurs solutions. Les ratios de compression et les performances annoncés par les trois éditeurs n'ont pas pu être vérifiés. Nous récapitulons donc simplement une synthèse des caractéristiques des trois solutions de compression.

	Oracle	IBM DB2	Microsoft SQL Server
Type de compression	Compression au niveau bloc (page)	DB2 utilise à la fois la compression par table et la compression par page	Compression par page équivalente à la compression par page de DB2 hormis qu'elle travaille en plus sur le préfixe des valeurs.
Mode de compression	L'Advanced compression disponible de la version 11.1 est une compression de type dynamique et asynchrone. Lorsque le seuil de déclenchement de la compression du bloc est atteint, tous les enregistrements du bloc sont compressés en une seule opération, non prioritaire par rapport aux opérations de mises à jour	Les dictionnaires au niveau des pages sont mis à jour automatiquement. DB2 détermine lui-même quelles sont les pages qui nécessitent une reconstruction de leur dictionnaire. La compression adaptative de DB2 disponible depuis la version 10.1 est comparable à l'option de compression dynamique d'Oracle « Advanced compression ».	SQL server semble proposer une compression dynamique comme celle d'Oracle ou de DB2 mais qui présente un <i>overhead</i> sur les opérations de compression.
Ratio de compression	Oracle affiche des taux de compression entre 54% et 73%	L'option adaptive compression de DB2 permet d'économiser entre 75% et 85% du volume initial.	La compression de SQL server permet d'économiser entre 40% et 60% du volume initial.
Outil permettant d'estimer le ratio de compression	Un package PL/SQL permet d'estimer le ratio de compression pour chaque table mais pas pour les index.	DB2 fourni des fonctions pour estimer le ratio de compression des tables et des index.	<i>Pas de fonction équivalente avec SQL Server</i>
Impacts sur les mises à jour	<ul style="list-style-type: none"> - les insertions ne sont pas compressées à la volée et ne provoquent pas de surcoût des écritures ; - Oracle offre une optimisation des ordres update et ne maintient que les valeurs dans la table de symboles et non plus les valeurs réelles. 	les insertions ne sont pas compressées à la volée et ne provoquent pas de surcoût des écritures ;	SQL Server n'offre pas d'optimisation des ordres DML Update pour les données compressées

	Oracle	IBM DB2	Microsoft SQL Server
Impacts des opérations de décompression	Pas d' <i>overhead</i> lors des opérations de décompression	Pas d' <i>overhead</i> lors des opérations de décompression	<i>Information non transmise par l'éditeur</i>
Optimisation de la mémoire cache	Oracle stocke un bloc compressé tel quel dans sa mémoire cache et le lit sans le décompresser	DB2 à l'instar d'Oracle stocke aussi les blocs compressés dans sa mémoire cache.	SQL Server stocke aussi les blocs compressés dans sa mémoire cache.
Consommation de la CPU	Les cycles de la CPU augmentent de 10 à 12% selon Oracle	La consommation CPU augmente en moyenne de 3% à 6% selon IBM.	Les cycles de la CPU augmentent de plus de 15%.
Optimisation des requêtes	Les valeurs des prédicats des requêtes ne sont évalués qu'une seule fois dans la table de symboles du bloc.	<i>Information non transmise par l'éditeur</i>	SQL Server n'offre pas d'optimisation sur les prédicats des requêtes avec les données compressées
Compression des objets larges	Oracle propose une compression des objets larges (LOB/SECUREFILES) même lorsqu'ils sont stockés en dehors du segment de la table.	Les données LOB qui sont stockées en dehors du segment de la table ne sont pas compressibles avec DB2	Les données LOB stockées en dehors du segment de la table ne sont pas compressibles avec SQL Server.
Compression des index	La compression d'index est disponible depuis la version 8i d'Oracle. Elle n'est pas activée par défaut même si la compression est activée au niveau de la table.	La compression d'index est disponible depuis la version 9.7 de DB2. Elle est activée par défaut lorsque la compression de la table est active.	Sous SQL server les mêmes opérations s'appliquent aussi bien pour la compression des tables que pour celle des index
Compression des tables temporaires	<i>Pas de fonction équivalente avec Oracle</i>	La compression adaptative est automatiquement activée sur les tables temporaires	<i>Pas de fonction équivalente avec SQL Server</i>
Compression par colonne	Oracle offre cette fonctionnalité depuis la version 12C pour l'ensemble des plateformes.	DB2 offre cette fonctionnalité depuis la version 10.5	<i>Pas de fonction équivalente avec SQL Server</i>

Tableau 6 Synthèse des différentes solutions de compression (Oracle, DB2, SQL server)

1.3.2 Oracle versus DB2

D'après les informations fournis par IBM dans sa dernière version 10.5, le taux de compression de DB2 avec l'option « *Adaptive Compression* » serait 15% à 20% meilleur comparée à celui d'Oracle. En ce qui concerne les fonctionnalités de compression, DB2 et Oracle proposent des fonctionnalités similaires exceptées pour les points suivants :

- Oracle propose une compression des objets larges (LOB) lorsqu'ils dépassent une taille significative alors que DB2 n'offre pas cette fonctionnalité (et les tables LOB sont largement répandues dans les systèmes SAP tels que les CRM, alias Customer Relationship Management).
- DB2 offre la compression pour les tables temporaires alors qu'Oracle ne propose pas de fonction équivalente.

Pour DB2 comme pour Oracle, une amélioration des performances est constatée sur :

- Une meilleure utilisation du cache des requêtes du fait que les blocs compressés sont stockés tels quels en mémoire.
- Une meilleure utilisation de la bande passante des lectures séquentielles.

Au final les solutions de DB2 et d'Oracle nous semblent relativement équivalentes à charge de maintenance comparable.

1.3.3 Oracle versus SQL Server

SQL Server ne présente pas d'option de compression aussi performante qu'Oracle pour les environnements transactionnels (OLTP), à savoir :

- SQL Server n'offre pas des taux de compression aussi performant (entre 1,8 et 1,9) que l'Advanced compression d'Oracle où les taux sont compris entre 2,2 et 3,5 ;
- SQL Server propose différentes solutions de compression (en ligne, par page) qui nécessitent un arbitrage avec les capacités CPU et ne sont pas aussi faciles à mettre en œuvre que l'option « Advanced compression » d'Oracle ;
- SQL Server n'offre pas d'optimisation sur les prédicats des requêtes données compressées contrairement à Oracle qui n'évalue qu'une seule fois la valeur recherchée d'une colonne compressée dans un bloc au lieu de l'évaluer sur tous les enregistrements ;
- SQL Server à l'inverse d'Oracle ne présente pas de mécanisme d'optimisation des Ordres DML UPDATE ;

Si l'on considère ces éléments, l'option « *Advanced Compression* » d'Oracle nous paraît plus performante et plus simple à maintenir que celle de SQL Server.

1.4 Bénéfices de la compression des bases de données

La compression des bases de données a des répercussions bénéfiques sur l'ensemble de l'infrastructure informatique et selon différents aspects, tels que :

- la diminution des coûts ;
 - o de l'infrastructure de stockage,
 - o de l'infrastructure de sauvegarde,
 - o d'exploitation et d'administration,
- l'amélioration de la gestion de la disponibilité des applications ;
 - o la diminution des coûts des sites de secours,
 - o la diminution des temps d'arrêts pour la restauration,
- l'amélioration des performances.

1.4.1 La diminution des coûts

1.4.1.1 La diminution des coûts de l'infrastructure de stockage

Comme nous l'avons déjà évoqué, la compression des bases de données amène une diminution des coûts de l'infrastructure de stockage. Néanmoins cette diminution n'apparaît réellement sur le bilan financier qu'à condition que les volumétries de stockage n'étant plus utilisées soient décommissionnées définitivement.

La compression des bases de données du parc informatique de l'entreprise constitue le moment opportun pour réaliser l'inventaire de l'infrastructure du stockage. Cet inventaire sert de référence pour établir les coûts initiaux avant la compression mais il est aussi le moyen d'éliminer des composants de stockage devenus inutiles que l'entreprise continue de payer. Les bases de données devenues obsolètes et inutiles peuvent être écartées directement du projet de compression.

1.4.1.2 La diminution des coûts de l'infrastructure de sauvegarde

La croissance de la quantité de données, et la redondance supplémentaire liée au stockage des sauvegardes, ne font qu'augmenter les coûts d'infrastructure. Nous soulignons donc l'importance de réaliser également l'inventaire des sauvegardes comme pour le stockage.

Pour exemple, la filiale DSI du groupe GDF SUEZ possède de nombreuses bases de données qui comportent jusqu'à 11 sauvegardes en parallèle à un instant T (la dernière sauvegarde mensuelle, les 4 dernières sauvegardes hebdomadaires, 6 sauvegarde quotidiennes). Ces copies sont essentielles pour la fiabilité du cycle de vie de l'application, mais elles coûtent de plus en plus cher. A l'instar des économies réalisées sur l'infrastructure de stockage, la compression va permettre de diminuer les coûts de l'infrastructure de sauvegarde.

1.4.1.3 La diminution des coûts d'exploitation et d'administration

Une fois que les bases de données ont été compressées certaines tâches récurrentes effectuées régulièrement par les administrateurs des bases de données ou l'exploitant deviennent moins fréquentes et leur laisse plus de temps à consacrer sur des projets ou des études.

Les administrateurs des bases et les équipes d'exploitation passe un temps significatif à :

- Surveiller le remplissage des espaces disques ;
- Purger les données obsolètes par manque de place disque ;
- Demander régulièrement aux services gérant les infrastructures de stockage l'allocation d'espace disque supplémentaire ;
- Réaliser les copies des bases de données de production pour rafraîchir les bases de recette ou les environnements de test.

La compression permet :

- Des opérations d'administrations plus rapides liées à la diminution de la taille des objets (réorganisations, calculs de statistiques...).
- Une diminution du besoin de partitionnement pour limiter la taille des tables, donc administration plus simple et rapide.

1.4.2 L'amélioration de la gestion de la disponibilité des applications

La disponibilité d'un système indique sa capacité à remplir son rôle pendant un temps donné. Cette valeur s'exprime le plus souvent sous la forme d'un pourcentage de temps pendant lequel le système est disponible par rapport à l'horaire de service négocié lors du SLA. Les entreprises investissent des sommes importantes pour gérer la disponibilité de leurs applications.

La gestion de la disponibilité est en permanence confrontée au dilemme entre le coût de l'investissement et le coût associé au risque en cas de défaillance du système d'information. La compression des bases de données permet d'améliorer la gestion de la disponibilité en réduisant :

- la diminution des coûts des sites de secours ;
- Les temps d'arrêts pour la restauration des systèmes en cas de crash ;
- La durée des backups offline pour les systèmes avec ce mode de backup ;
- La durée des rafraîchissements des systèmes hors-production (lié au temps de restauration).

1.4.2.1 La diminution des coûts des sites de secours

Les sites de secours nécessitent des disques supplémentaires pour la mise en miroir des données sur les serveurs de basculement. Cette redondance coûte cher et n'apporte généralement de valeur ajoutée que lorsqu'un composant tombe vraiment en panne. La compression des bases de données du parc informatique de l'entreprise permet également de diminuer les coûts de stockage des sites de secours. Ces économies permettent de mettre en œuvre des sites de secours

supplémentaires pour d'autres applications qui auparavant n'auraient pas pu en bénéficier. Cela contribue au final à améliorer la disponibilité de l'ensemble des applications.

1.4.2.2 La diminution des temps d'arrêts pour la restauration

En cas d'incident majeur survenu sur une base de données, il peut être nécessaire de restaurer l'intégralité des données, ce qui nécessite un arrêt complet des applications qui utilisent cette base. La restauration d'une base de données de plusieurs Téra octets nécessite plusieurs heures voir une journée complète pendant laquelle les utilisateurs ne peuvent plus travailler. Un délai de restauration de plus d'une journée dans certains cas peut coûter plusieurs millions d'euros à l'entreprise.

Comme nous l'avons déjà décrit auparavant, la compression des bases de données amène une réduction moyenne de leur volumétrie par 2 voire 2,5. Cela va contribuer à diminuer les temps de restauration et les coûts associés.

1.4.3 L'amélioration des performances

L'amélioration des performances a déjà été décrite dans les paragraphes précédents. Nous résumerons en disant que la compression amène une amélioration des performances sur l'utilisation de la bande passante des lectures / écritures et sur l'utilisation de cache des requêtes SQL.

CHAPITRE 2.

ETUDE COMPARATIVE DES AUTRES SOLUTIONS SGBDR VIS-A-VIS D'ORACLE

L'ensemble des bases de données gérées sur le périmètre de GDF SUEZ IT repose sur le SGBDR Oracle. Nous avons effectué une pré-étude pour mesurer l'opportunité pour Gdf-Suez de migrer les bases de données Oracle vers DB2 ou vers SQL Server. Cette pré-étude a consisté à comparer les autres critères déterminants que celui de la compression et a permis de vérifier leur adéquation par rapport à l'architecture existante de GDF SUEZ IT.

Nous avons identifié les fonctionnalités qui nous semblaient pertinentes pour analyser la faisabilité d'une migration de nos bases de données Oracle vers les solutions DB2 ou SQL Server :

- Faisabilité ;
- Scalabilité ;
- Disponibilité ;
- Exploitabilité ;
- Fonctionnalités ;
- Sécurité ;
- Financier.

2.1 Etat actuel du parc des bases de données

Le périmètre des bases de données Oracle pouvant donner lieu à une migration était le suivant :

- **Bases de données Oracle liées à des systèmes SAP (une centaine de DB) :**
 - o Taille des bases de données :
 - Extra Large (20 To): ECC ISU.
 - Large (de 1 To à 10 To) : ECC ISU, BI, CRM.
 - Medium (de 200 Go à 1 To): BI, CRM, SolMan.
 - Small (<200 Go): Portail, SRM, MDM, BO.
 - o Puissance serveur :
 - Extra Large (20 To) :128 cœurs
 - Large (de 1 To à 10 To) : 4 à 22 cœurs.
 - Medium (de 200 Go à 1 To) : 2 à 4 cœurs.
 - Small (<20 go): 2 cœurs.
 - o OS actuel : SPARC Solaris 64 Bits
 - o Jeu de caractères : Unicode et Non Unicode.
- **Bases de données Oracle liées à des systèmes WebLogic (une dizaine de DB).**
- **Bases de données Oracle liées à des systèmes Informatica (3 DB) :**

- Tailles des bases de données : > 1 To et < 2,5 To
- Puissance serveur : 4 à 10 cœurs.
- OS actuel : SPARC Solaris

Voici un récapitulatif des bases de données Oracle dépassant 1 Téra octets de volumétrie avant compression sur le périmètre initial du projet de compression (*database* de 18To écartée dans un premier temps car trop critique):

bases de données Oracle de plus de 1 Téra octets					
Systèmes		Serveur SUN OS : SPARC Solaris 64 Bits			DB size
Sous- projet	Environnement	Modèle processeur	Nb Cores	RAM Go	Avril 2011
SAP ECC Régulé	Recette	SPARC64-VI	10	40	2 660 Go
	Projet Recette	SPARC64-VI	10	36	4 722 Go
	Pré-production	UltraSPARC-IV+	16	64	5 320 Go
	Production	SPARC64-VI	28	112	8 778 Go
	Formation	UltraSPARC-IV+	16	64	1 468 Go
Total					22 948 Go
SAP ECC Non Régulé	Recette	UltraSPARC-IV+	8	32	6 635 Go
	Formation	SPARC64-VI	4	8	6 106 Go
	Pré-production	SPARC64-VI	22	76	6 677 Go
	Production	SPARC64-VI	22	76	20 030 Go
Total					39 448 Go
BI	Pré-production	UltraSPARC-IV+	8	32	1 560 Go
	Production	UltraSPARC-IV+	8	64	2 200 Go
	Recette	UltraSPARC-IV+	4	16	top
Total					3 760 Go
Bi régulé	Pré-production	SPARC64-VI	12	48	1 256 Go
	Production	SPARC64-VI	14	48	1 256 Go
	Recette	UltraSPARC-IV+	4	32	1 256 Go
Total					3 768 Go
Informatica	Recette	UltraSPARC-IV+	4	32	2 464 Go
	Production	SPARC64-VI	10	20	1 229 Go
Total					3 693 Go
Total général					73 617 Go

La volumétrie non compressée représentait environ 73 To.

GDF SUEZ IT dispose de solution d'architecture comme *Oracle Real Application Cluster* pour gérer la haute disponibilité pour une ligne d'applications comportant la facturation des clients particuliers des offres gaz.

Oracle Real Application Cluster (RAC) a été introduit par Oracle en 2001 et permet de déployer une seule base de données au sein d'un cluster de serveurs en mode actif/actif, assurant la tolérance de pannes en cas de défaillance matérielle ou d'interruption planifiée.

Cette solution permet de partager les données disque et mémoire sur plusieurs machines physiques. L'ensemble des autres systèmes de gestion de base de données ne propose qu'un partage des données sur disque.

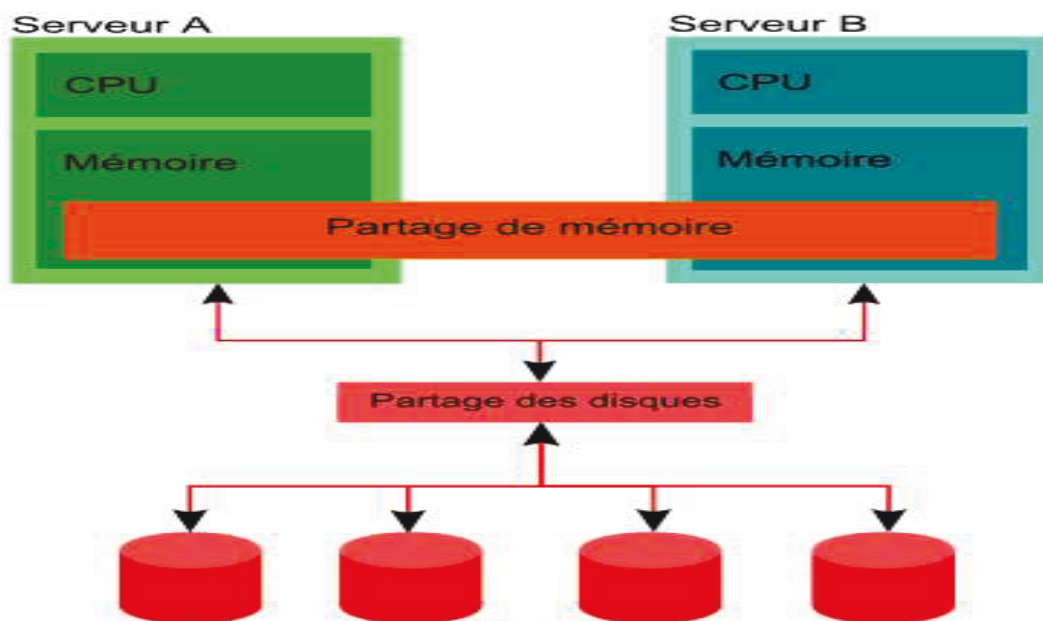


Figure 2-1 Exemple d'architecture de Real Application Cluster avec deux nœuds.

Les mécanismes de reprise des transactions respectent les exigences des architectures de haute disponibilité. Par exemple, une requête en cours d'exécution sur un nœud A peut se poursuivre sur un autre nœud B du cluster en cas de défaillance du nœud A.

On trouve différentes façons de mettre en place RAC :

- Mode répartition de charge (*Load balancing*) : tous les serveurs sont utilisés. L'équilibrage de charge est réalisé en distribuant les connexions entre les différents nœuds du cluster en fonction de la charge de ceux-ci. L'équilibrage de charge contribue à une meilleure utilisation des ressources de serveurs.
- Haute disponibilité (mode *Fail-over*) : permet à plusieurs nœuds d'un cluster d'accéder à la même base de données. Si un nœud est arrêté, les autres nœuds continuent d'assurer les services sur la base de données.
- Configuration mono (Rac avec un seul nœud) : Un serveur de secours peut être démarré en cas de crash.

2.2 Faisabilité

2.2.1 Faisabilité technique

Le premier point d'analyse a porté sur la faisabilité de ces migrations. Compte-tenu du périmètre applicatif, la faisabilité d'une migration de tout ou partie de ces bases de données a dû être démontrée, avec apport de références sur des migrations du même type. Une question s'est posée

notamment sur la possibilité ou non de rester en « Non Unicode » pour les plus grosses bases de données qui étaient dans ce cas de figure.

- ❑ **Evaluation de DB2** : lors de notre étude IBM a affiché plus de 700 migrations Oracle vers DB2 en environnement SAP, avec de nombreuses références de migrations de grands comptes (BARILLA, AUDI, PEPSY, SAP IT...). Au final IBM a présenté peu de références en France sous SAP/DB2 hormis le groupe Danone. Le maintien en non Unicode a été confirmé pour GDF SUEZ IT.

- ❑ **Evaluation de SQL Server** : Microsoft a affiché 20 ans de coopération avec SAP sur SQL-Server, et de nombreux clients migrés d'Oracle vers SQL-Server, avec quelques références de grands comptes (Microsoft, TORO, TETRAPAK, ARCELIK, GT...). Au final Microsoft a présenté peu de références de grands comptes en France sous SAP/SQL Server hormis le groupe Véolia.

2.2.2 Durées

En dehors de la faisabilité purement technique, des éléments sur la durée du projet, la durée des indisponibilités des applications SAP durant chaque migration, ont fait partis des points clés.

- ❑ **Evaluation de DB2** : IBM a proposé 4 méthodologies de migrations (hormis la solution de l'éditeur SAP NZDT **(1)**) permettant de s'adapter aux contraintes de disponibilité, dont une solution de type Capture Data Change **(2)** comparable à l'outil concurrent d'Oracle nommé *GoldenGate*. IBM a affiché une durée globale de 4 à 8 mois pour des projets de migration de même typologie (*probablement uniquement sur phase deploy*)
 - (1) La solution NZDT (*Near Zero DownTime*) de l'éditeur SAP est une solution qui propose des scénarios de migration avec indisponibilité proche de zéro.
 - (2) Cette fonctionnalité permet d'identifier les enregistrements insérés ou modifiés ou supprimés dans les tables. Le processus de capture n'a quasiment aucun impact sur la base de données source, la lecture se faisant via les fichiers logs du SGBDR ou bien les logs archivés.

- ❑ **Evaluation de SQL Server** : Microsoft n'a fait apparaitre qu'une seule méthodologie, avec arrêt le vendredi soir et remise à disposition le lundi matin. Une indisponibilité aussi longue aurait pu s'avérer complexe à gérer, notamment pour une ligne d'applications de GDF SUEZ IT. Une autre solution NZDT de l'éditeur SAP a été proposée, mais les coûts étaient à priori très importants.

2.3 Scalabilité

2.3.1 Scalabilité Taille de base

Notre plus grosse base de données Oracle était non unicode et devait avoisiner une taille de plus de 30 To sous 3 ans sans compression. Il était donc important pour nous, de connaître pour le nouveau moteur de base de données :

- La limite technique de taille des bases de données pour ce moteur ;
 - La volumétrie de la base de données maximale atteinte par un client SAP avec ce moteur.
- ❑ **Evaluation de DB2** : IBM, a affirmé qu'aucune limite technique n'existait en théorie sur la taille de la *database*. La base de données de l'entreprise Unilever qui avait été migrée d'Oracle vers DB2 faisait 30 To, donc une taille supérieure aux systèmes SAP Gdf-SUEZ.
- ❑ **Evaluation de SQL Server** : Microsoft a annoncé la limite technique de 256 *Petabytes*, et la plus grosse base hors SAP de 80 *Petabytes*. Microsoft a affiché d'autre part les références de QUANTA et NEC avec plus de 25 To en environnement SAP.

2.3.2 Scalabilité Hardware

Afin de répondre aux besoins de puissance de nos systèmes SAP à moindre coûts, nous nous sommes appuyés sur la technologie Oracle Real Application Cluster (Oracle RAC) pour assurer la scalabilité hardware de nos plus gros systèmes SAP.

Les critères de scalabilité hardware pour le nouveau moteur de base de données ont été les suivants :

- Les solutions de scalabilité hardware ;
- La puissance maximale pouvant être produite en SAPS (1) ou en nombre de cœur de CPU;
- La puissance maximale atteinte par un client SAP avec ce moteur.

(1) SAPS est l'abréviation du terme anglais : SAP Application Performance Standard. Il s'agit de l'unité de mesure des performances des serveurs SAP indépendant de tout constructeur. A titre d'exemple 100 SAPS représentent 2000 lignes de commandes traitées en une heure. Techniquement cela représente le traitement de 6000 changements d'écran SAP (*dialog steps*) ou 2400 transactions SAP.

- ❑ **Evaluation de DB2** : DB2 est compatible avec la plus part des OS. Néanmoins, la majorité du parc DB2 étant sous AIX, le risque de bug est réduit avec AIX et donc fortement conseillé. IBM, lors de l'étude en 2011, a affiché des serveurs AIX développant une puissance largement suffisante pour GDF SUEZ (688630 SAPS max), et offrant une virtualisation très efficace. IBM a proposé d'autre part un cluster DB2 actif/actif équivalent à Oracle RAC nommé *PureScale*, qui n'est supporté par SAP que sous AIX ou Linux, et avec de nombreuses restrictions (pas de réorganisation en ligne des tables, pas de sauvegarde incrémentale, nécessité de restaurer sur un serveur de même topologie...), qui l'ont rendu incompatible avec le contexte de Gdf SUEZ IT.

- ❑ **Evaluation de SQL Server** : SQL Server n'est compatible qu'avec Windows Server, donc repose évidemment sur des architectures de processeurs Intel X86 multi-cœur. Microsoft a proposé un cluster actif/actif équivalent à Oracle RAC nommé SSD, mais ne fonctionnant qu'en lecture, donc inexploitable dans les environnements transactionnels SAP de GDF SUEZ IT. Sur le site Web du support SAP, le serveur le plus puissant affichait 137.470 SAPS pour un serveur 8U de 8 CPU et 80 cœurs, ce qui était largement suffisant pour la majorité des bases de données de productions (22.000 SAPS), mais était un peu limite pour une ligne d'applications (250.000 SAPS via RAC). Aucun cluster actif/actif n'a permis d'adresser cette limitation, même si la puissance des serveurs X86 a cru de façon importante d'années en années.

2.4 Disponibilité

2.4.1 DIMA / PDMA

Les DIMA (Durée d'Indisponibilité Maximale Admissible) et PDMA (Perte de données Maximale Admissible) de nos bases de données varient en fonction des projets. L'important pour nous est de percevoir :

- Les DIMA et PDMA pouvant être proposées par le nouveau moteur de base de données ;
- Les architectures mises en œuvre pour y parvenir, en environnement transactionnel comme en décisionnel.

- ❑ **Evaluation de DB2** : IBM a proposé une solution de **cluster actif/passif** classique, via la couche cluster d'AIX. IBM ne dispose pas de solution très haute disponibilité type Oracle Real Application Cluster. Comme indiqué dans le chapitre « Scalabilité Hardware », l'utilisation du cluster actif/actif *PureScale* a présenté trop de contraintes pour être envisagée sur ce périmètre.

- ❑ **Evaluation de SQL Server** : Microsoft propose différentes solutions de haute disponibilité, via :
 - Une architecture **cluster Windows actif/passif** (toutes les données sont alors sur une même baie de disques) ;
 - **Database Mirroring**, mais avec des impacts sur les performances liées à la réplication synchrone.

SQL Server ne propose pas de solution très haute disponibilité type Oracle Real Application Cluster.

2.4.2 Plan de reprise d'activité (PRA)

Les options possibles pour le plan de reprise d'activité ont été également examinées :

- Sauvegardes et restaurations à chaud ;

- Maintien d'une copie au fil de l'eau (type Oracle *DataGuard*).

- ❑ **Evaluation de DB2** : IBM a proposé une solution de *Log Shipping* permettant d'alimenter un site de secours dans le cadre d'un plan de reprise d'activité, type Oracle *DataGuard*, avec une bascule manuelle.

- ❑ **Evaluation de SQL Server** : Microsoft a proposé une solution de *Log Shipping* permettant d'alimenter un site de secours, type Oracle *DataGuard*. SQL Server a proposé aussi une solution nommée « *AlwaysOn* » avec une bascule automatique des services *Database* qui permet ainsi au système de production SAP de Microsoft d'afficher un taux de disponibilité de 99,995%.

2.5 Exploitabilité

2.5.1 Administration récurrente

Il était important de savoir si les opérations récurrentes d'administration restaient possibles via le nouveau moteur de base de données, notamment :

- Les réorganisations base ouverte des tables et indexes ;
- Le partitionnement (les types de partitionnements certifiés par SAP) ;
- La compression base ouverte des partitions ;
- Le calcul de statistiques pour l'optimiseur ;
- Les sauvegardes à chaud (avec ou sans parallélisations).

Nos bases de données Oracle sont administrées via les outils SAP BRTOOLS, qui offrent une interface à l'ensemble des gestes d'administration des bases de données (arrêt et redémarrage, agrandissements, réorganisations, exports/imports,...).

Il était important de savoir si le nouveau moteur de base de données était compatible avec les BRTOOLS, ou offrait une interface d'administration comparable.

- ❑ **Evaluation de DB2** : On a retrouvé toutes les fonctionnalités actuelles (réorganisation en base ouverte, sauvegardes à chaud, partitionnement, statistiques...). Néanmoins, les systèmes SAP/DB2 étant peu courant en France, un risque identifié a porté sur l'obtention de consultants expérimentés.

- ❑ **Evaluation de SQL Server** : On retrouve toutes les fonctionnalités actuelles (réorganisation base ouverte, sauvegardes à chaud, partitionnement, statistiques...) mais via une interface graphique. Il est également possible d'installer Windows en mode server core sans interface graphique, pour rendre le système plus léger, et en réduisant les failles de sécurités. Le *scripting* reste possible.

2.5.2 Refresh (rafraîchissement)

L'une des opérations récurrente en environnement SAP est le rafraîchissement (*refresh* en anglais) des bases de données de recette ou de formation par les données réelles de production, que nous faisons selon les cas :

- par sauvegarde physique et restauration ;
- par des clones au niveau des baies de disques ;
- par *flashback recovery* (permettant de ramener un environnement à un point dans le passé).

Le nouveau moteur de base de données devait proposer des mécanismes de *refresh* équivalents.

- ❑ **Evaluation de DB2** : IBM a présenté des solutions de rafraîchissement des bases par sauvegarde/restauration ou clone au niveau de la baie de disque moyennant des adaptations. Par contre, IBM n'a pas proposé de solution équivalente au « *Flashback Recovery* » d'Oracle, permettant de revenir à une image antérieure cohérente des données.
- ❑ **Evaluation de SQL Server** : Microsoft a présenté des solutions de rafraîchissement des bases par sauvegarde/restauration ou clone au niveau de la baie de disque moyennant des adaptations. Microsoft a proposé également sa solution de « *Flashback Recovery* » permettant de revenir à une image antérieure cohérente des données.

2.5.3 Corruptions

Nous utilisons la solution Oracle RMAN pour réaliser les vérifications d'absence de corruptions des blocs sur nos bases de données Oracle, ou à défaut nous exécutons un programme standard de SAP. Il était donc important de connaître les opérations permettant d'identifier une corruption de bloc, ainsi que les mécanismes de corrections associés. De même, nous souhaitons connaître les mécanismes permettant de couvrir les corruptions de données liées à des gestes accidentels.

- ❑ **Evaluation de DB2** : IBM n'a pas fourni d'information lors de notre étude.
- ❑ **Evaluation de SQL Server** : SQL Server 2012 a proposé avec son SGBDR des mécanismes natifs de détection et de correction automatique des corruptions physiques de blocs via le Plan de reprise d'activité.

2.6 Sécurité

2.6.1 Règles de sécurité des mots de passe

Les mots de passe de tous nos applicatifs doivent suivre certaines règles sur leur contenu et la fréquence de renouvellement. Le nouveau moteur SGBDR doit proposer des solutions équivalentes.

- Evaluation de DB2** : IBM n'a pas fourni d'information lors de notre étude.

- Evaluation de SQL Server** : Microsoft a proposé des solutions pour définir une stratégie de gestion des mots de passes (délais d'expiration, complexité, ...)

2.6.2 Restrictions

De même, les options possibles en termes de limitation des droits selon les utilisateurs et les profils devaient être proposées par le nouveau SGBDR. (Exemple : limitation des droits en lecture sur les schémas fonctionnels pour des utilisateurs dédiés à l'administration de la base de données).

- Evaluation de DB2** : DB2 permet une définition précise des autorisations par utilisateur.

- Evaluation de SQL Server** : SQL Server permet une définition précise des autorisations par utilisateur.

2.6.3 Traçabilité

Les options possibles en termes de traçabilité des connexions ou des opérations étaient à préciser.

- Evaluation de DB2** : IBM n'a pas fourni d'information lors de notre étude.

- Evaluation de SQL Server** : Microsoft a proposé une solution pour générer des rapports d'audit sur les connexions, les accès en lecture et en écriture, de façon native. La collecte peut écartier l'utilisateur SAP, ce qui permet théoriquement une traçabilité de l'ensemble des opérations techniques sans impact sur les performances.

2.6.4 Cryptage

Les options de cryptage de tout ou partie de la base de données sont également attendues, ainsi que l'impact sur la charge CPU et l'impact sur les performances.

- Evaluation de DB2** : IBM n'a pas fourni d'information lors de notre étude.

- ❑ **Evaluation de SQL Server** : SQL Server permet de crypter la totalité d'une *database* au niveau I/O, de façon transparente pour les applications, de façon native. Cryptage fin sur une table ou un champ également possible.

2.7 Financier

Les modèles économiques suivants ont été demandés aux éditeurs DB2 et SQL Server :

- le tarif des licences;
- le tarif du support;
- dans l'hypothèse d'un changement d'infrastructure, des ordres de grandeurs ont été attendus sur les nouvelles infrastructures en fonction du *sizing* XL, *large*, *medium* et *small* décrit au début de ce chapitre.
- des éléments approximatifs de coûts ont été également attendus sur les méthodologies et les coûts projet nécessaires à la migration de tout ou partie des bases de données du périmètre.

2.7.1 TCO

- ❑ **Evaluation de DB2** : IBM a affiché un TCO réduit de 20% en moyenne sur 3 ans suite à une migration vers DB2 en intégrant les coûts projet.
- ❑ **Evaluation de SQL Server** : Microsoft a affiché un TCO réduit de 30% à 37%, mais sans intégrer l'ensemble des coûts projet. Les upgrades (montées de version) sont censés être très légers.

2.7.2 Licences

- ❑ **Evaluation de DB2** : Les licences au cœur sont selon IBM moins onéreuses qu'Oracle. Les instances hors-production sont facturées à l'utilisateur déclaré, quel que soit le nombre de systèmes dans le paysage. Les sites de secours passifs ne sont facturés qu'à un seul cœur CPU. Des licences doivent néanmoins être acquises versus Oracle où elles le sont déjà, donc le gain final n'apparaît que sur les coûts de support.
- ❑ **Evaluation de SQL Server** : Les licences au cœur CPU sont de l'ordre de 5k€. Toutes les options sont déjà incluses dans SQL-Server Entreprise (compression, audit de sécurité...). La tarification des licences au cœur CPU ne s'applique qu'aux bases de données de production et de pré-production. Les autres bases de données sont facturées au user. Chaque base de données de production peut bénéficier d'un site de secours passif non facturé.

2.7.3 Coûts projet

- ❑ **Evaluation de DB2** : Les coûts projets de IBM sont forfaitaires (généralement 20% sur site, et 80% en *remote*). Nous avons noté des coûts induits importants à prévoir sur :

- Adaptation à DB2 (formations, procédures de maintenance, scripts, adaptations des programmes spécifiques SAP écrits en langage ABAP) ;
- Changement de l'infrastructure (AIX préconisé avec DB2) ;
- Adaptation au changement d'OS vers AIX (scripts, matrices de flux...).

❑ **Evaluation de SQL Server** : Microsoft affiche un retour sur investissement sous 9 à 15 mois. Nous avons notés les coûts induits importants à prévoir sur :

- Adaptation à SQL-Server (formations, procédures de maintenance, scripts, adaptations des programmes spécifiques SAP écrits en langage ABAP) ;
- Changement de l'infrastructure (Microsoft indispensable) ;
- Adaptation au changement d'OS vers Microsoft (scripts, matrices de flux...).

2.8 Synthèse de l'évaluation

Famille critères	critère	DB2	SQLServer
Faisabilité	Faisabilité technique	IBM a affiché plus de 700 migrations Oracle vers DB2 en environnement SAP, avec de nombreuses références de migrations de grands comptes (BARILLA, AUDI, PEPSY, SAP IT...). IBM a présenté peu de références en France sous SAP/DB2 hormis le groupe Danone. Le maintien en non Unicode a été confirmé pour GDF SUEZ IT.	Affiche 20 ans de coopération avec SAP sur SQL-Server, et de nombreux clients migrés d'Oracle vers SQL-Server, avec quelques références de grands comptes (Microsoft, TORO, TETRAPAK, ARCELIK, GT...). Microsoft a présenté peu de références de grands comptes en France sous SAP/SQL Server hormis le groupe Véolia.
Faisabilité	Durée	IBM a proposé 4 méthodologies de migrations (hormis la solution de l'éditeur SAP NZDT permettant de s'adapter aux contraintes de disponibilité, dont une solution de type Capture Data Change comparable à l'outil concurrent d'Oracle nommé <i>GoldenGate</i> . IBM a affiché une durée globale de 4 à 8 mois pour des projets de migration de même typologie .	Microsoft n'a fait apparaître qu'une seule méthodologie, avec arrêt le vendredi soir et remise à disposition le lundi matin. Une indisponibilité aussi longue aurait pu s'avérer complexe à gérer, notamment pour une ligne d'applications de GDF SUEZ IT. Une autre solution NZDT de l'éditeur SAP a été proposée, mais les coûts étaient a priori très importants
Scalabilité	Taille de la base	IBM, a affirmé qu'aucune limite technique n'existait en théorie sur la taille de la <i>database</i> . La base de données de l'entreprise Unilever qui avait été migrée d'Oracle vers DB2 faisait 30 To, donc une taille supérieure aux systèmes SAP Gdf-SUEZ.	Microsoft a annoncé la limite technique de 256 <i>Petabytes</i> , et la plus grosse base hors SAP de 80 <i>Petabytes</i> . Microsoft a affiché d'autre part les références de QUANTA et NEC avec plus de 25 To en environnement SAP.
Scalabilité	Hardware	DB2 est compatible avec la plus part des OS. Néanmoins, la majorité du parc DB2 étant sous AIX, le risque de bug est réduit avec AIX et donc fortement conseillé. IBM, lors de l'étude en 2011, a affiché des serveurs AIX développant une puissance largement suffisante pour Gdf-Suez (688630 SAPS max), et offrant une virtualisation très efficace.. IBM a proposé d'autre part un cluster DB2 actif/actif équivalent à Oracle RAC nommé <i>PureScale</i> , qui n'est supporté que par SAP et uniquement sous AIX ou Linux, et avec de nombreuses restrictions qui l'ont rendu incompatible avec le contexte de Gdf SUEZ IT.	SQL Server n'est compatible qu'avec Windows Server, donc repose évidemment sur des architectures de processeurs Intel X86 multi-cœur. Microsoft a proposé un cluster actif/actif équivalent à Oracle RAC nommé SSD, mais ne fonctionnant qu'en lecture, donc inexploitable dans les environnements transactionnels SAP de GDF SUEZ IT. Sur le site Web du support SAP, le serveur le plus puissant affichait 137.470 SAPS pour un serveur 8U de 8 CPU et 80 cœurs, ce qui était largement suffisant pour la majorité des bases de données de productions (22.000 SAPS),

Famille critères	critère	DB2	SQLServer
Disponibilité	DIMA /PDMA	DB2 a proposé une solution de cluster actif/passif classique, via la couche cluster d'AIX. IBM ne dispose pas de solution très haute disponibilité type Oracle Real Application Cluster. Comme indiqué dans le chapitre « Scalabilité <i>Hardware</i> »,	SQL Server a proposé différentes solutions de haute disponibilité, via : - Une architecture cluster Windows actif/passif (toutes les données sont alors sur une même baie de disques) ; SQL Server n'a pas proposé de solution très haute disponibilité type Oracle Real Application Cluster.
Disponibilité	Plan de reprise d'activité	DB2 a proposé d'autre part une solution de Log Shipping permettant d'alimenter un site de secours dans le cadre d'un plan de reprise d'activité, type Oracle <i>DataGuard</i> , avec une bascule manuelle	SQL Server a proposé une solution de <i>Log Shipping</i> permettant d'alimenter un site de secours, type Oracle <i>DataGuard</i> . SQL Server a proposé aussi une solution nommée « <i>AlwaysOn</i> » avec une bascule automatique des services <i>database</i> qui permet ainsi au système de production SAP de Microsoft d'afficher un taux de disponibilité de 99,995%
Exploitabilité	Administration récurrente	On a retrouvé toutes les fonctionnalités actuelles (réorganisation en base ouverte, sauvegardes à chaud, partitionnement, statistiques...). Les systèmes SAP/DB2 étant peu courant en France, un risque identifié a porté sur l'obtention de consultants expérimentés	On retrouve toutes les fonctionnalités actuelles (réorganisation base ouverte, sauvegardes à chaud, partitionnement, statistiques...) via une interface graphique. Il est également possible d'installer Windows en mode <i>server core</i> sans interface graphique, pour rendre le système plus léger. Le <i>Scripting</i> reste possible
Exploitabilité	<i>Refresh</i> (rafraîchissement)	IBM a présenté des solutions de rafraîchissement des bases par sauvegarde/restauration ou clone au niveau de la baie de disque moyennant des adaptations. Par contre, IBM n'a pas proposé de solution équivalente au « <i>Flashback Recovery</i> » d'Oracle, permettant de revenir à une image antérieure cohérente des données.	Microsoft a présenté des solutions de rafraîchissement des bases par sauvegarde/restauration ou clone au niveau de la baie de disque moyennant des adaptations. Microsoft a proposé également sa solution de « <i>Flashback Recovery</i> » permettant de revenir à une image antérieure cohérente des données.
Exploitabilité	Corruptions	IBM n'a pas fourni d'information lors de notre étude	SQL Server 2012 a proposé avec son SGBDR des mécanismes natifs de détection et de correction automatique des corruptions physiques de blocs via le Plan de reprise d'activité.
Sécurité	Mots de passe	IBM n'a pas fourni d'information lors de notre étude	SQL Server a proposé des solutions pour définir une stratégie de gestion des mots de passes (délais d'expiration, complexité, ...)

Famille critères	critère	DB2	SQLServer
Sécurité	Restrictions	DB2 permet une définition précise des autorisations par utilisateur.	SQL Server permet une définition précise des autorisations par utilisateur.
Sécurité	Traçabilité	IBM n'a pas fourni d'information lors de notre étude	SQL Server a proposé une solution pour générer des rapports d'audit sur les connexions, les accès en lecture et en écriture, de façon native. La collecte peut écarter l'utilisateur SAP, ce qui permet théoriquement une traçabilité de l'ensemble des opérations techniques sans impact sur les performances.
Sécurité	cryptage	IBM n'a pas fourni d'information lors de notre étude	SQL Server permet de crypter la totalité d'une <i>database</i> au niveau I/O, de façon transparente pour les applications, de façon native. Cryptage fin sur une table ou un champ également possible.
Financier	Licences	Les licences au cœur sont selon IBM sont moins onéreuses qu'Oracle. Les instances hors-production sont facturées à l'utilisateur déclaré, quel que soit le nombre de systèmes dans le paysage. Les sites de secours passifs ne sont facturés qu'à un seul cœur CPU. Des licences doivent néanmoins être acquises versus Oracle où elles le sont déjà, donc le gain final n'apparaît que sur les coûts de support.	Les licences au cœur CPU sont de l'ordre de 5k€. Toutes les options sont déjà incluses dans SQL-Server Entreprise (compression, audit de sécurité...). La tarification des licences au cœur CPU ne s'applique qu'aux bases de données de production et de pré-production. Les autres bases de données sont facturées au user. Chaque base de données de production peut bénéficier d'un site de secours passif non facturé.
Financier	Coûts projets	Les coûts projets de IBM sont forfaitaires (généralement 20% sur site, et 80% en <i>remote</i>)). Nous avons noté des coûts induits importants à prévoir sur : <ul style="list-style-type: none"> - Adaptation à DB2 (formations, procédures de maintenance, scripts, adaptations des programmes spécifiques SAP écrits en langage ABAP) - Changement de l'infrastructure (AIX préconisé avec DB2) ; - Adaptation au changement d'OS vers AIX (scripts, flux...). 	Microsoft affiche un retour sur investissement sous 9 à 15 mois. Nous avons noté les coûts induits importants à prévoir sur : <ul style="list-style-type: none"> - Adaptation à SQL-Server (formations, procédures de maintenance, scripts, adaptations des programmes spécifiques SAP écrits en langage ABAP) ; - Changement de l'infrastructure (Microsoft indispensable) ; - Adaptation au changement d'OS vers Microsoft.

Tableau 7 Synthèse de l'évaluation des solutions SGBDR

2.9 Choix de la solution.

Les deux éditeurs DB2 et SQL Server ont mis en avant dans leur réponse une réduction du coût total de possession (*Total Cost Of Ownership*) par rapport à Oracle, via :

- La réduction des coûts de licences (par la tarification au cœur et le mode de calcul) ;
- La réduction des infrastructures pour SQL Server ainsi que DB2 via la virtualisation ;
- La réduction des coûts de stockage et de sauvegarde pour IBM via les taux de compression.

Une étude détaillée aurait été nécessaire pour valider les gains de type TCO (*Total Cost Of Ownership*). Toutes les opérations d'administration qui sont réalisées sur les bases de données Oracle ont semblé avoir leur équivalent, mais auraient induit de très profonds changements sur les modes opératoires. Les DIMA (Durée d'Indisponibilité Maximale Admissible) et les PDMA (Perte de données Maximale Admissible) proposées par Microsoft et IBM auraient pu être maintenues, via de nouvelles architectures. Cependant les clusters actifs/actifs (type Oracle Real Application Cluster) offrant *scalabilité* et très haute disponibilité n'ont pas eu d'équivalent dans les solutions proposées par IBM ou Microsoft ; d'où la nécessité d'investir dans des serveurs devant rester passifs et dédiés à la haute disponibilité.

Au-delà des coûts de projet de migration technique à proprement parler, ces migrations auraient induits des changements importants, notamment :

- des coûts d'infrastructure ;
- des couts d'adaptation de l'ensemble des procédures d'exploitation ;
- des coûts d'adaptation des développements spécifiques SAP entraînant de nombreuses modifications dans les sources des programmes ABAP ;
- des coûts des tests de non régression.

Ces profonds changements et l'absence de gains *flagrants* ont été les principaux freins à la migration des bases de données Oracle vers DB2 ou SQL Server. Pour le projet de compression des données, GDF SUEZ a donc choisi de conserver le SGBDR Oracle présent sur l'ensemble du paysage informatique et a décidé de mettre en œuvre la solution de compression « Advanced Compression » de l'éditeur Oracle.

CHAPITRE 3.

ETUDE DE LA COMPRESSION DES DONNEES AVEC ORACLE.

Nous présentons dans ce chapitre l'étude de la solution de compression au sein de GDF SUEZ IT avec ses contraintes, ses impacts et ses bénéfices.

3.1 Enjeux et objectifs

Dans le cadre du projet de compression des bases de données de GDF SUEZ IT, un premier proof of concept a été mis en œuvre pour valider les objectifs visés. Lors de ce proof of concept, une copie d'une des bases de production les plus volumineuses a été utilisée (plus de 5 Téra octets) pour réaliser les opérations de compression dans les conditions les plus proches de la réalité.

Nous rappelons que l'objectif principal est une réduction de la consommation d'espace disque des systèmes grâce à l'utilisation de l'*Advanced compression* d'Oracle. Les estimations de compression fournies par Oracle et SAP permettent de viser une diminution de ces consommations par un facteur de 2,4 sur les systèmes SAP ECC6. L'un des principaux enjeux est donc de vérifier que les ratios de compression annoncés par SAP/Oracle soient bien atteints.

Les autres objectifs attendus de ce POC concernent :

- L'estimation des indisponibilités nécessaires à la compression. Compte tenu des volumétries des bases de données de GDF SUEZ IT (la plus petite des bases de données était supérieure à un Téra octets, les opérations de réorganisation et de compression devaient se réaliser hors ligne, et nécessitaient l'arrêt des applications SAP. Les fenêtres d'indisponibilités des applications devaient donc être évaluées, optimisées, puis négociées avec les bénéficiaires. ;
- L'amélioration des performances, par la réduction du nombre d'accès disque, et aussi l'optimiser de l'utilisation des buffers de la mémoire, puisque les données occupent moins de volumétrie disque et mémoire ;
- L'amélioration de la maintenabilité des systèmes. En effet les opérations comme les calculs des statistiques de l'optimiseur SQL devraient être plus rapides ;
- La rédaction d'une feuille de route détaillée avec les durées des opérations de compression et la procédure technique d'exploitation.

3.2 Support de la compression par SAP

Oracle et SAP collaborent depuis de nombreuses années au développement de services et de solutions pour leurs clients communs. Oracle a des équipes dédiées qui collaborent avec SAP sur divers aspects comme les développements conjoints, l'avant-vente et la vulgarisation technologique, le support technique aux clients, et le conseil. Ce partenariat a débuté en 1988, avec le développement de SAP R/3.

La figure ci-dessous retrace les grandes étapes de la collaboration entre Oracle et SAP.

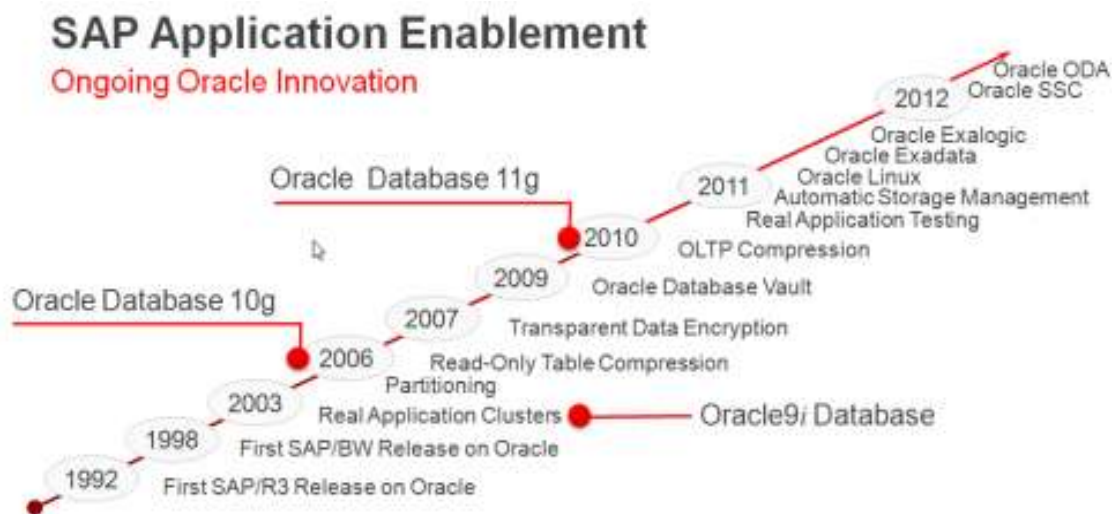


Figure 3-1 Les grandes étapes de la collaboration entre Oracle et SAP [OSS12]

L'équipe de développeurs Oracle basée au siège de SAP, à Walldorf, en Allemagne, assiste SAP dans les opérations suivantes :

- Test de performance sur les nouvelles versions des applications SAP en environnement Oracle, pour éliminer tout risque de dégradation des temps de réponses entre deux versions ;
- Correction des bugs de la base de données détectés par les tests fonctionnels de SAP et prise en compte des demandes d'évolution SAP dans la feuille de route d'Oracle ;
- Certification des nouvelles fonctionnalités d'Oracle par SAP ;
- Optimisation conjointe des nouvelles versions de la base de données Oracle et des applications SAP.

Nous soulignons que l'option Advanced compression d'Oracle a été certifiée par SAP depuis la version Oracle 11gR2 (11.2.0.2) dans les environnements ECC (ERP transactionnel de SAP) et BI (Business Intelligent)

3.3 Scénario de compression

Ce paragraphe décrit le scénario de compression qui a été réalisé sur le serveur du *Proof of concept* avec une configuration s'approchant de celle de l'environnement de production.

Pour bénéficier des effets de la compression sur les données existantes, il est nécessaire de réorganiser les bases de données Oracle. De nouveaux volumes disques doivent ainsi être créés

pour accueillir la nouvelle enveloppe de la base de données compressée. Les tables et les index de l'application SAP sont réorganisés et compressés vers cette nouvelle enveloppe. Une fois les opérations terminées, l'ancienne enveloppe est supprimée et la place disque associée est libérée.

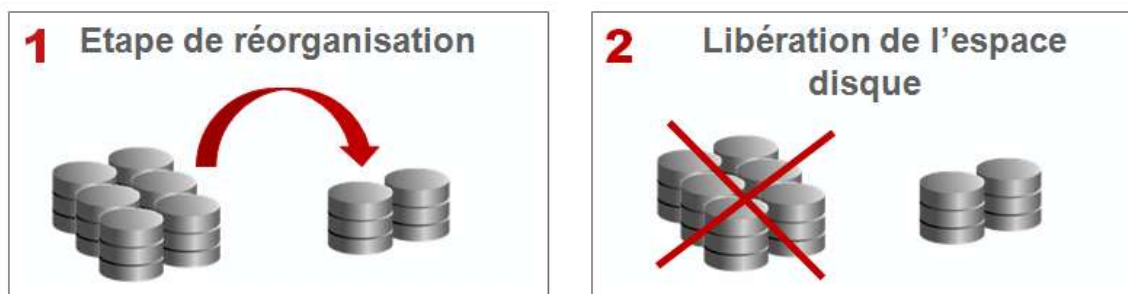


Figure 3-2 scénario de réorganisation et de compression

Le scénario de compression comprend plusieurs étapes :

- Prévoir le nouveau stockage pour la base de données compressée ;
- Réaliser la montée de version de la version Oracle 10g vers la version Oracle 11g. L'implémentation de l'*Advanced compression* d'Oracle est une fonctionnalité de la version 11G qui suppose dans notre cas une montée de version des bases Oracle 10g;
- Réaliser les opérations de préparation de la compression telle que la création des nouvelles enveloppes pour accueillir les données compressées,
- Réorganiser et compresser les données ;
- Effectuer opérations de post compression telles que la suppression des anciens tablespaces applicatifs où ne figurent plus aucunes données (ni table, ni index) et libérer les anciens disques non compressés

3.3.1 Prévision du stockage

La prévision du nouveau stockage de la base de données compressée permet de définir les besoins d'allocation disques

3.3.1.1 Estimation de la volumétrie de la base de données compressée

Oracle fournit un outil qui permet d'estimer la taille des tables après compression et permet ainsi d'estimer la taille des nouvelles enveloppes (*tablespaces*) de la base de données. Cette étape doit être effectuée des semaines, voire des mois avant les opérations de compression pour commander au gestionnaire de l'infrastructure de stockage l'espace disque nécessaire.

Compression des tables

L'estimation de la volumétrie cible de la base oracle 11g est réalisée à l'aide d'un package « *DBMS_COMP_ADVISOR* » fourni par Oracle. Ce package doit être installé au préalable sur la

base de données source, qui dans le cadre de ce POC, fonctionne sous la version Oracle 10g (10.2.0.2)

Le package Oracle *DBMS_COMP_ADVISOR* cible par défaut les tables qui possèdent plus de 1 000 000 de lignes. Il se base sur un échantillon de lignes de la table (par défaut 1%) pour estimer le taux de compression :

Voici un extrait d'un log contenant les ratios de compression des plus grosses tables du schéma SAP de la copie de la base de données de production utilisée pour le *Proof of concept* et les gains estimés en Mo :

Table	Taille de la table en MO	% de l'échantillon utilisé pour le calcul	Ratio de compression	Taille estimée de la table Compressée en Mo	Gains en MO
DBERCHZ1	1 116 726	0,01%	5,11	218 537	898 189
DBERCHZ3	279 076	0,01%	8,94	31 217	247 859
DBERDLB	230 557	0,01%	5,79	39 820	190 737
DBERDL	180 852	0,01%	8,04	22 494	158 358
BALDAT	81 113	0,20%	1,18	68 740	12 373
DFKKOP	79 615	0,01%	6,24	12 759	66 856
DPAYP	73 934	0,01%	3,41	21 682	52 252
Total	2 041 873			415 248	1 626 625
Ratio Global			4,92		

Tableau 3-1 ratio de compression des plus grosses tables

L'estimation globale de la volumétrie compressée représente 415 Go environ au lieu de 2041 Go. Le ratio global moyen est de 4,92. Il est le résultat de la division :

« Taille de la table non compressée » / « Taille de la table compressée »

Compression des index

Les index ne bénéficient pas de l'option *Advanced Compression* OLTP des tables, or ils représentent généralement autant de volumétrie que les tables en environnement SAP. Pour maintenir un taux de compression avantageux, il est nécessaire de mettre en œuvre la compression des index. Cette fonctionnalité était déjà disponible depuis la 10.2.0.2. Dans le cas de ce POC, nous avons mis en œuvre la compression sur les plus gros index qui concernent en l'occurrence ceux des grosses tables. Les index qui ont été compressés ont concerné uniquement les tables de plus de 20 Go. Nous nous sommes reportés à la note OSS SAP 1109743 "*Use of Index Key Compression for Oracle.pdf*"

Aucun des éditeurs Oracle ou SAP ne fournissent un outil pour estimer la taille des index compressés. Néanmoins l'éditeur SAP fourni un package qui permet de générer des scripts de création des index candidats à la compression. Les scripts des index n'ont compressibles ne sont pas générés.

Besoins de stockage

Les besoins de stockage des nouveaux volumes doivent prendre en compte les différentes futures ressources Oracle telles que :

- Les nouveaux *tablespaces* qui accueilleront les objets compressés (avec une marge d'espace libre allant de 5 à 10%);
- les autres *tablespaces* internes d'Oracle (*SYSTEM*, *SYSAUX*, *PSAPUNDO*, *PSAPTEMP*) qui contiennent des objets non compressés.
- Les *Redo logs* et les control files ;
- Les *archive logs* ;
- Les binaires de la version Oracle 11Gr2.

Nous avons considéré 5% de marge d'espace libre pour estimer la taille des nouveaux tablespaces à laquelle est venu s'ajouter l'évolution de la base de données.

3.3.1.2 Préparation des nouveaux volumes groupes et des nouveaux File System

Un des objectifs de ce *POC*, était de démontrer la faisabilité technique de récupérer la place disque des anciens volumes. Lors de ce *POC*, une copie d'un système existant avec la base de données ont été restaurés à l'identique avec les même définitions de volumes et les mêmes file system.

De nouveaux volumes groupes cibles ont été configurés pour accueillir la base de données compressée ainsi que les binaires Oracle 11g.

A la fin des opérations, plus aucune ressource Oracle (fichiers ou binaires) ne doit subsister sur les anciens volumes. Ceux-ci doivent être décommissionnés et récupérés par le gestionnaire d'infrastructure du stockage.

Les nouveaux volumes groupes sont configurés différemment car ils tiennent compte du type de données. Dans la configuration avant compression, il n'y a qu'un seul volume où sont rassemblés les binaires Oracle, les ressources critiques telles que les archives log et la base de données elle-même. La réorganisation et la compression de la base de données est le moment opportun pour optimiser la répartition des nouveaux File System. Le tableau comparatif ci-dessous présente une configuration cible des nouvelles files system qui tient compte de la typologie des fichiers Oracle :

Anciens Volumes et anciens File System	Type de fichiers	Taille en Go
vg01		5 842,1
/oracle	inventory et logs oracle	10,0
/oracle/stage	Zone de travail pour installation Oracle	10,0

Nouveaux volumes et nouveaux File system	Type de fichiers	Taille en Go
DG_arch_S04		200,0
/oracle/S04/oraarch	Archive log	200,0
DG_data_S04		2 550,5

Anciens Volumes et anciens File System	Type de fichiers	Taille en Go	Nouveaux volumes et nouveaux File system	Type de fichiers	Taille en Go
/oracle/S04	Binaires oracle 10g	15,0	/oracle/S04/sapbacku p	Fichiers trace SAP	0,5
/oracle/S04/mirrlogA	Redo log	4,4	/oracle/S04/sapdata1	fichier de données	425,0
/oracle/S04/mirrlogB	Redo log	4,4	/oracle/S04/sapdata2	fichier de données	425,0
/oracle/S04/oraarch	Archive log	100,0	/oracle/S04/sapdata3	fichier de données	425,0
/oracle/S04/origlogA	Redo log	4,4	/oracle/S04/sapdata4	fichier de données	425,0
/oracle/S04/origlogB	Redo log	4,4	/oracle/S04/sapdata5	fichier de données	425,0
/oracle/S04/sapbacku p	Fichiers traces SAP	0,5	/oracle/S04/sapdata6	fichier de données	425,0
/oracle/S04/sapdata1	Fichiers de données	675,0	DG_OraBin_S04		49,0
/oracle/S04/sapdata2	Fichiers de données	675,0	/oracle	inventory et logs oracle	10,0
/oracle/S04/sapdata3	Fichiers de données	675,0	/oracle/stage	Zone de travail pour installation Oracle	20,0
/oracle/S04/sapdata4	Fichiers de données	675,0	/oracle/S04	Binaires oracle 11g	15,0
/oracle/S04/sapdata5	Fichiers de données	675,0	/oracle/S04/sapreorg	Fichiers traces SAP	4,0
/oracle/S04/sapdata6	Fichiers de données	830,0	DG_redo_S04		17,6
/oracle/S04/sapdata7	Fichiers de données	675,0	/oracle/S04/mirrlogA	redo log	4,4
/oracle/S04/sapdata8	Fichiers de données	805,0	/oracle/S04/mirrlogB	redo log	4,4
/oracle/S04/sapreorg	Fichiers traces SAP	4,0	/oracle/S04/origlogA	redo log	4,4
			/oracle/S04/origlogB	redo log	4,4
Total anciens File system		5 842,1	Total nouveaux File System		2 817,1

Tableau 3-2 Nouvelle configuration des File System pour la compression

3.3.2 Montée de version d'Oracle 10 g vers Oracle 11 g.

Les bases de données Oracle étaient déployées dans la version 10g sur l'ensemble du paysage (10.2.0.2 ou 10.2.0.4.) dont le support n'était plus assuré par Oracle. Cette montée de version Oracle suppose elle-même en pré requis une montée de version OS vers Solaris 10 update 9, actuellement en update 5.

La version cible 11G au moment de ce PoC était la version 11.2.0.2 avec le niveau de patchset.

3.3.3 Préparation des opérations de compression

Les opérations de préparations consistent à adapter certaines ressources Oracle pour pouvoir supporter l'activité intensive durant les opérations de réorganisation et de compression, à savoir

- Augmenter la taille du tablespace temporaire;
- Réduire la durée de rétention des UNDO ;
- Augmenter la taille des *redo log* pour éviter des *switchs* trop fréquents pendant les opérations de réorganisation des tables ;
- Désactiver les archives log pour ne pas pénaliser les opérations de *switchs*.

3.3.3.1 Augmenter le tablespace temporaire PSAPTEMP

Le tablespace temporaire doit supporter l'activité intensive lors de la reconstruction d'index qui sont réalisées en parallèle. La taille du tablespace temporaire est estimée en fonction de la taille des N plus gros index qui seront reconstruits en parallèle simultanément. Le chiffre N constitue le degré de parallélisme.

3.3.3.2 Réduire la durée de rétention des undo

Avant la version Oracle 9i, les informations d'annulation des transactions (*rollback*) étaient conservées jusqu'au moment du commit ou du *rollback* avant d'être supprimés.

La gestion automatique des segments d'annulation, introduite depuis la version 9i, permet de définir la durée de rétention des informations d'annulation après le commit.

Cela évite notamment les erreurs de « *snapshot too old* » (clichés trop anciens) suite aux requêtes trop longues. Pendant les opérations de réorganisation, il est inutile voir contreproductif de conserver une durée de rétention longue. Au contraire les informations d'annulations doivent être libérées le plus vite possible pour laisser la place aux autres transactions.

La valeur de la durée de rétention est configurée à l'aide du paramètre UNDO_RETENTION. Pour les opérations de réorganisations et de compression, elle est réduite à 900 secondes au lieu de 43200 :

```
# sqlplus / as sysdba
SQL> ALTER SYSTEM SET UNDO_RETENTION = 900 scope=BOTH ;
```

3.3.3.3 Augmenter la taille des redo logs.

Pendant les opérations de réorganisation, les *redo log* sont très sollicités et la fréquence des *switchs* augmente, ce qui augmente le risque de contentions. Il est possible de diminuer la fréquence des *switchs* de plusieurs manières :

- Utiliser les options de NOLOGGING lors de la reconstruction des index que nous avons mis en œuvre dans les scripts de ce POC et avec l'outil BRSPACE;
- Augmenter la taille des *redo log* (à réaliser en avance de phase pendant une période de très faible activité)

3.3.3.4 Désactivation du mode archive log

Le mode archive log doit être désactivé avant de lancer les opérations de compression, afin d'accélérer la compression et réduire le risque que le File System contenant les archives log soit plein.

Nous vérifions si l'instance est en mode archive log à l'aide de la commande « *archive log list* » :

```
# sqlplus / as sysdba
SQL>archive log list
Database log mode           Archive Mode
Automatic archival         Enabled
Archive destination        /oracle/<SID>/oraarch
Oldest online log sequence 2588
Next log sequence to archive 2591
Current log sequence       2591
```

La désactivation du mode archive log nécessite un arrêt relance de l'instance Oracle.

Voici les commandes SQL Oracle :

```
# sqlplus / as sysdba
SQL> shutdown immediate;
SQL> startup mount;
SQL> alter database noarchivelog;
SQL> alter database open;
```

3.3.4 Opérations de réorganisation et de compression

Nous présentons dans ce paragraphe :

- L'outil SAP BRSPACE qui permet de réorganiser, de compresser et de créer les nouveaux tablespaces destinés aux tables et index qui sont déplacés;
- La nouvelle répartition des tables et index sur les nouveaux tablespaces qui permet de faciliter l'administration des grosses tables ;
- Les scripts Shell de compression qui mettent en œuvre les commandes de l'outil BRSPACE, les scripts SQL de compression des gros index ;
 - o Les outils de suivi des opérations ;
 - o Les options de reprise cas d'échec ;
 - o Les scripts de vérifications.

3.3.4.1 Les outils d'administration SAP Brtools.

BRTools est le nom donné par SAP à un ensemble de programmes et d'utilitaires permettant d'administrer les bases de données Oracle. Ces outils facilitent l'administration et l'exploitation des

bases de données Oracle et sécurisent les opérations des DBA Oracle. Les outils BRTOOLS génèrent les commandes SQL Oracle nécessaires et de ce fait simplifient les gestes d'administration des DBA ou des administrateurs SAP. A partir des menus, les commandes Oracle qui sont générées sont affichées à l'écran avant d'être exécutées.

L'ensemble des gestes et des opérations réalisés sur la base de données sont tracés et horodatés dans des fichiers log. Les BRTOOLS ont été spécialement conçus pour Oracle uniquement et ne fonctionnent que sur les environnements SAP. Les utilitaires sont les suivants :

- BRBACKUP sauvegarde les fichiers de données, les fichiers de contrôles, les journaux *redo log* ;
- BRARCHIVE sauvegarde les archives log Oracle (copie des *redo log*) ;
- BRRESTORE Restaure les fichiers de données, les fichiers contrôles, les fichiers *redo logs* ;
- BRRECOVER effectue les opérations de reprises en cas de crash de la base de données ou d'inconsistance sur les fichiers ;
- BRSPACE gère les opérations de maintenance de l'instance Oracle, la gestion des tablespaces et des fichiers de données, les réorganisations des segments (tables et index), la compression des tables ;
- BRCONNECT effectue les tâches d'administration Oracle telle que le calcul des statistiques pour l'optimiseur SQL, les vérifications sur l'intégrité des fichiers de données (checksum), la purge des fichiers log d'administration ;
- BRTOOLS est l'interface en mode caractères ou en mode graphique qui gère les menus de ces différents outils. Chaque menu possède un aide en ligne.

Les outils *BrTools* et plus particulièrement l'outil BRSPACE a été utilisé en mode ligne de commande dans ce Proof of Concept pour réaliser les opérations de réorganisation et de compression que nous décrivons plus loin dans ce chapitre.

3.3.4.2 Compression et réorganisation avec l'outil BRSPACE

Les réorganisations et les compressions ont été effectuées à l'aide de l'outil *BRSPACE* qui fait partis de la suite des utilitaires SAP *BRTOOLS* spécialement conçus pour les bases de données Oracle :

- Compresser les tables et reconstruire leurs index associés
- Compresser les tables comportant des champs de type LONG en convertissant leurs champs LONG dans le nouveau format Lob *SecureFile* de la version Oracle 11g.
- Compresser les tables comportant des champs avec l'ancien type LOB en les convertissant dans le nouveau format Lob *SecureFile*
- Déplacer et réorganiser le reste des tables d'un tablespace en particulier sans les compresser. Certaines tables SAP ne doivent pas être compressées conformément à la note SAP OSS 1431296.

Les principales options de l'outil BRSPACE pour la compression des tables Oracle et la conversion des objets larges son décrites dans le tableau ci-dessous

Options de la commande brspace		
Description de l'option	option <i>brspace</i>	Commentaires
Mode de réorganisation	-mode [offline online]	Le mode offline est normalement plus performant que mode online mais il entraine un lock exclusif sur toute la table.
Indique le nom du schéma Oracle du propriétaire des objets SAP. sap<sid>	-o sap<sid>	
Sélectionne les tables à réorganiser et compresser depuis le tablespace source <src_tbs>	-s <src_tbs>	
Déplace les tables sélectionnées vers le tablespace cible <new_tbs>	-n<new_tbs>	
Active la compression en ciblant les tables standards (sans LOB)	-compress ctab	
Active la compression en ciblant les tables de type LOB uniquement	-compress ctalob	
Converti les colonnes de type LOB en <i>Securefile</i>	-lob2lob	
Niveau de compression des colonnes <i>Securefile</i>	-lobcompr	Les options LOW, MEDIUM, HIGH d'Oracle correspondent à un degré de compression des Securefiles.
Désactive le logging (c'est-à-dire l'écriture des transactions dans les fichiers <i>redo log</i>) lors de la reconstruction des index	-NBR	
Ecarte de la compression les tables définies par SAP comme non compressibles	-SCT	
Nombre de process « brspace »	-parallel <n>	Augmenter ce paramètre en fonction du nombre de tables à traiter. Chaque processus brspace traitera un sous ensemble de tables.
Nombre de thread Oracle	-degree <n>	Augmenter en fonction de la volumétrie des tables. Plus les tables sont volumineuses plus le nombre de thread Oracle doit être élevé)
Initial <i>extent</i> de la table cible	-initial <n>	Les valeurs suivantes sont possibles : o 1: 16 KB o 2: 64 KB o 3: 1 MB o 4: 8 MB o 5: 64 MB o 6: 1 GB
Liste des tables à compresser et liste des tables à exclure	-p ReorgInclude.tab -p ReorgExclude.tab	

Tableau 3-3 Options de la commande BRSPACE

3.3.4.3 Créer les nouveaux tablespaces applicatifs SAP

La création des nouveaux tablespaces cible devant contenir les objets réorganisés a été effectuée à l'aide de l'outil BRSPACE. Les gros tablespaces ont été créés en parallélisant la création des différents fichiers de données.

La création d'un tablespace se compose de deux phases :

- La création du tablespace qui nécessite un premier fichier de données ;
- L'extension du tablespace par ajouts successifs de fichiers de données ;

La création d'un tablespace nécessite les paramètres suivants :

- Le mot clé « *tscreate* »
- Le nom du tablespace <TbsName> ;
- La taille en Mo du nouveau Data File <SizeMo> ;
- Le nom du répertoire sapdata[n] correspondant au Data File à ajouter <FsName>

```
brspace -c force -f tscreate -t <TbsName> -s <SizeMo> -f <FsName>
```

```
brspace -c force -f tscreate -t PSAPSR3_NEW -s 30720 -m 30720 -f /oracle/SU3/sapdata10
```

L'exemple suivant montre la commande brspace qui ajoute 3 nouveaux fichiers de données de 30720 Mo au tablespace PSAPSR3_NEW qui sont répartis sur les répertoires sapdata11, sapdata12, sapdata13

```
brspace -c force -f tsextend -t PSAPSR3_NEW -s 30720 -f1 11 -f2 12 -f3 13
```

Les commandes *BRSPACE* d'ajouts de fichier comme celle qui figure ci-dessus peuvent être exécutées en parallèle dès lors que le tablespace est déjà créé.

Après la création, il est important de positionner l'option « COMPRESS FOR OLTP » au niveau de la clause de stockage des nouveaux tablespaces. Ainsi les tables qui seront créées porteront par défaut la clause de compression et les données seront donc compressées. Néanmoins il reste toujours possible de créer des tables non compressées en précisant la clause NOCOMPRESS au moment de leur création.

3.3.4.4 Répartition des tables et index SAP sur les nouveaux tablespaces

Les tablespaces 11G ont été créés avec l'option « AUTOEXTEND » d'Oracle qui permet aux fichiers de données de s'étendre automatiquement. Le tableau ci-dessous présente la correspondance entre les tablespaces 10G et les nouveaux tablespaces contenant les objets compressés. Les volumétries sont celles obtenues à la fin des opérations du proof of concept.

Database 10G (SU3)		Database 11g + compression (POC)	
Tablespaces	Taille réelle Go	Tablespaces	Taille Estimée en Go
(1) PSAPSR3	4022	a) PSAPSR3_XXL	273
		b) PSAPSR3_BIG	433
		c) PSAPSR3_NEW	604
(2) PSAPSR3USR	28	PSAPSR3USR_NEW	13
(3) PSAPSR3701	90	PSAPSR3701_NEW	57
Total 10g	4140	Total 11g	(4) 1380

Tableau 3-4 Répartition des tables et index SAP sur les nouveaux tablespaces

- (1) Le principal *tablespace* PSAPSR3 contient la majorité des tables et index du schéma SAP et atteint une taille de 4022 Go. Les objets de ce *tablespace* une fois réorganisés et compressés sont répartis sur de nouveaux tablespaces comme suit :
- a. La plus grosse table du schéma SAP (il s'agit de la table contenant les lignes de factures) est réorganisée et compressée sur le nouveau *tablespace* PSAPSR3_XXL.
 - b. Les tables volumineuses de plus de 20 Go sont réorganisées sur un deuxième *tablespace* PSAPSR3_BIG, pour lesquelles les indexes compressibles seront compressés.
 - c. Le reste des autres tables (plus de 70000 tables) sont réorganisées et compressées sur le nouveau *tablespace* PSAPSR3_NEW.
- (2) Les tables et index du *tablespace* PSAPSR3USR contenant les tables spécifiques et leurs index sont réorganisés et compressés sur un *tablespace* homologue plus petit PSAPSR3USR_NEW de 13 Go.
- (3) De même le *tablespace* PSAPSR3701 contenant le code SAP compilé est réorganisé et compressé sur le *tablespace* homologue plus petit PSAPSR3701_NEW.
- (4) La volumétrie totale des tablespaces SAP compressée représente 1380 Go.

3.3.4.5 Scripts de compression des tables et de réorganisation des index

Les scripts shell de réorganisation et de compression des tables du Poc utilisent l'outil brspace des brtools SAP. Chaque script shell cible une volumétrie et une typologie définie des tables du schéma SAP en fonction de la présence ou non de champ de type LOB. Le tableau ci-dessous indique le rôle de chaque script et la cible des tables traitées.

Tablespace source	Rôle du script en fonction de la volumétrie et de la typologie de tables	Script	Tablespace cible	Durée de compression minutes (**)
PSAPSR3	Réorganise et compresse la table de plus de 1 To en SQL	ReorgHugeTables.BRSPACE.ksh	PSAPSR3_XXL	110
	Réorganise et compresse les tables de plus de 20 Go via brtools. (hors tables comprenant des champs LOB).	ReorgBigTables.BRSPACE.ksh	PSAPSR3_BIG	419
	Réorganise et compresse les tables de plus de 20 Go avec des champs LOB.	ReorgBigLobTables.BRSPACE.ksh	PSAPSR3_NEW	82
	Réorganise et compresse les tables de moins de 20 Go avec des champs LOB.	ReorgLobTables.BRSPACE.ksh	PSAPSR3_NEW	33
	Réorganise et compresse les tables restantes du schéma SAP	ReorgTables.BRSPACE.ksh	PSAPSR3_NEW	192
(hors tables non compressibles)	PSAPSR3USR_NEW		8	
	PSAPSR3701_NEW		198	
PSAPSR3***	Réorganise les tables non compressibles.	ReorgOthers.BRSPACE.ksh	PSAPSR3_NEW	79
			Total durée	79

Tableau 3-5 Scripts Shell utilisant l'utilitaire SAP Brspace

(***) Les durées de compression ont été obtenues lors du POC et ont permis d'estimer les temps d'indisponibilité des opérations de compression. La durée totale est de 1087 mn soit un peu plus de 23H50 pour réorganiser les tablespaces applicatifs SAP, ce qui représente une volumétrie totale de 4140 Go.

Le tableau suivant indique les valeurs utilisées pour les options de parallélisme. Les valeurs de ces paramètres peuvent être adaptées à la volumétrie des tables.

Scripts shell utilisant BRSPACE				
Script	Fichier Liste des tables à inclure ou à exclure	Mode réorganisation	Nombre De session Oracle	Nombre de thread Oracle
ReorgBigTables.BRSPACE.ksh	ReorgInclude.tab	-mode offline	-parallel 2 (***)	-degree 12 (***)
ReorgBigLobTables.BRSPACE.ksh	ReorgIncludeLOB.tab	La conversion des LOB en <i>SecureFile</i> nécessite le mode online	-parallel 2 (***)	-degree 12 (***)
ReorgLobTables.BRSPACE.ksh	ReorgExclude.tab	-mode online	-parallel 16	-degree 1
ReorgTables.BRSPACE.ksh	ReorgExclude.tab	-mode offline	-parallel 6 (***)	-degree 4 (***)
ReorgOthers.BRSPACE.ksh	N/A	-mode offline	-parallel 4	-degree 8

Tableau 3-6 Paramètres des scripts Shell utilisant l'utilitaire SAP Brspace

(***) L'outil BRSPACE dispose de deux options de parallélisme :

- Le paramètre « -parallel » qui définit le nombre de sessions Oracle utilisées ou chacune des sessions traite un sous ensemble de tables.
- Le paramètre « -degree » pour le parallélisme des ordres de réorganisation par table pour chacune des sessions Oracle.

Pour réorganiser les grosses tables de plus de 20 Go, nous privilégions donc plutôt le parallélisme par table avec le paramètre « -degree » à 12 en ne prenant que 2 session Oracle.

Pour les tables de moins de 20 Go qui représente plus de 70000 tables, nous privilégions un nombre de sessions élevé de 6 (-parallel) avec un parallélisme par session de 4 (-degree).

3.3.4.6 Scripts de compression des gros index

Pour compresser les index des grosses tables de plus de 20 Go, nous avons utilisé des scripts SQL que nous avons exécuté à la suite des scripts *shell* qui réorganisent les tables. Pour les plus petites tables, les index n'ont pas été compressés.

Concernant la compression des gros index, l'utilisation de script SQL a été privilégiée car l'outil BRSPACE présente quelques contraintes techniques : Lors de la compression des tables, les index sont forcément reconstruits de façon non compressés, puis eux-mêmes compressés, d'où une double reconstruction générant une perte de temps.

Nous nous sommes donc limités à l'utilisation de BRSPACE pour le calcul du niveau de compression, en nous reportant à la **note SAP 1109743** "Use of Index Key Compression for Oracle.pdf".

Cette note contient le script d'un package PL/SQL [*ind_comp3p.sql*] qu'il faut créer sous le schéma SAP. L'appel à la procédure « *ind_comp.get_column* » permet de déterminer pour une table donnée les index compressibles, leurs degré de compression et génère en sortie un script SQL portant le nom de la table et contenant les commandes de reconstruction des index compressés de la table.

Nom de la table dont les index doivent être compressés	Mode de reconstruction des index	Nombre de threads oracle pour le parallélisme	Tablespace cible	
DBERDL	true	'offline'	16	'PSAPSR3_BIG'
EABL	true	'offline'	16	'PSAPSR3_BIG'
DBERCHZ3	true	'offline'	16	'PSAPSR3_BIG'
DFKKOP	true	'offline'	16	'PSAPSR3_BIG'
ETTIFN	true	'offline'	16	'PSAPSR3_BIG'
DFKKOPK	true	'offline'	16	'PSAPSR3_BIG'
DBERDTAX	true	'offline'	16	'PSAPSR3_BIG'
DPAYH	true	'offline'	16	'PSAPSR3_BIG'
DPAYP	true	'offline'	16	'PSAPSR3_BIG'
DBERCHZ2	true	'offline'	16	'PSAPSR3_BIG'
DBERCHZ1	true	'offline'	16	'PSAPSR3_XXL'

```

exec ind_comp.get_column('DBERDL',true,'offline', 16, 'PSAPSR3_BIG' );
exec ind_comp.get_column('EABL',true,'offline', 16, 'PSAPSR3_BIG' );
exec ind_comp.get_column('DBERCHZ3',true,'offline', 16, 'PSAPSR3_BIG' );
exec ind_comp.get_column('DFKKOP',true,'offline', 16, 'PSAPSR3_BIG' );
exec ind_comp.get_column('ETTIFN',true,'offline', 16, 'PSAPSR3_BIG' );
exec ind_comp.get_column('DFKKOPK',true,'offline', 16, 'PSAPSR3_BIG' );
exec ind_comp.get_column('DBERDTAX',true,'offline', 16, 'PSAPSR3_BIG' );
exec ind_comp.get_column('DPAYH',true,'offline', 16, 'PSAPSR3_BIG' );
exec ind_comp.get_column('DPAYP',true,'offline', 16, 'PSAPSR3_BIG' );
exec ind_comp.get_column('DBERCHZ2',true,'offline', 16, 'PSAPSR3_BIG' );
exec ind_comp.get_column('DBERCHZ1',true,'offline', 16, 'PSAPSR3_XXL' );

```

(***) Les scripts SQL de compression des index peuvent être générés avec des options de parallélisme. Dans l'exemple ci-dessus comme le degré 16 de parallélisme a été choisi, Oracle utilisera 16 Thread pour reconstruire chaque index.

Les scripts de compression d'index utilise l'option « COMPRESS » avec le degré de compression qui est calculé par le package SAP. Voici un ordre type de reconstruction d'index avec compression qui utilise un degré de parallélisme de 16 pour réduire le temps des opérations:

```
ALTER INDEX "<index_name>" REBUILD ONLINE COMPRESS 2 TABLESPACE PSAPSR3_NEW
PARALLEL 16 NOLOGGING ;
```

Notons l'option NOLOGGING qui évite d'écrire dans les *redo log* les informations de reconstruction de l'index et optimise ainsi le temps d'exécution de l'opération.

3.3.4.7 Suivi des opérations de compression (brspace)

Le suivi de l'avancement des opérations de réorganisation et de compression est effectué à l'aide des *logs* de l'outil BRSPACE présents dans le sous répertoire « *./sapreorg* ». L'entête du fichier contient le rappel de la commande brspace utilisée.

Les messages dans les logs sont horodatés. L'outil BRSPACE affiche dans ses *logs* à intervalle régulier la date et l'heure de fin estimée du traitement.

Les scripts *Shell* vus précédemment ont été configurés pour envoyer en fin de traitement un mail de suivi aux administrateurs avec comme pièces jointes les logs de Brspace.

3.3.4.8 Vérification de la compression des tables SAP

Suite à la mise en œuvre de la compression, le nombre de tables compressées avec l'option Advanced compression est 30 273 soit 99,4% des tables du schéma SAP. Le cumul de la volumétrie des tables compressées représente 796 423 Mo soit 79% de la volumétrie de l'ensemble des tables du schéma SAP.

Le tableau ci-dessous donne **par tranche de nombre de lignes** les indicateurs suivants :

1. La colonne [ACO-Nb] représente le nombre de tables compressées, la colonne [Nb] représente le nombre de tables non compressées dans la tranche.
2. La colonne [ACO-SUM] représente la somme total en Mo des tables compressées, la colonne [SUM] représente la somme totale en Mo des tables non compressées dans la tranche.
3. La colonne [ACO-MAX] représente la taille en Mo de la plus grosse table compressée, la colonne [MAX] représente la taille en Mo de la plus grosse table non compressée dans la tranche.
4. La colonne [ACO-MIN] représente la taille en Mo de la plus petite table compressée, la colonne [MIN] celle de la plus petite table non compressée dans la tranche.

Nombre de lignes	Nombre de tables du schéma SAP (1)		Somme totale en Mo des table du schéma SAP (2)		Taille en Mo de la plus grosse table (3)		Taille en Mo de la plus petite table (4)	
	ACO-Nb compression	Nb	ACO-SUM compression	SUM	ACO-MAX compression	MAX	ACO-MIN compression	MIN
(1)1,000,000,000 - 9,999,999,999	5	0	475 722	0	311 280	0	25 113	0

100,000,000 - 999,999,999	9	1	122 387	61 556	33 687	61 556	3 671	61 556
10,000,000 - 99,999,999	55	3	112 497	85 415	11 410	42 159	218	3 472
1,000,000 - 9,999,999	246	10	60 164	54 853	9 946	22 451	18	417
100,000 - 999,999	457	13	18 261	7 249	3 573	1 984	2	31
10,000 - 99,999	981	23	3 278	1 357	315	217	0	3
(2) 0 - 9,999	28 520	139	4 114	953	112	623	0	0
Total	30 273	(3) 189	796 423	(4) 211 383				

Tableau 3-7 Etat des lieux de la compression des tables du POC

- (1) Dans la catégorie des tables de plus de 1Mds de lignes, 5 tables compressées représentent une volumétrie de 475 Go soit plus de 59% de la volumétrie totale des tables compressées (796 Go).
- (2) La catégorie des tables compressées de moins de 1000 lignes est la plus importante en nombre avec 28520 tables mais ne représente qu'une volumétrie de 4 Go soit ½ % de la volumétrie totale des tables compressées.
- (3) Le nombre de tables non compressées est de 189 avec une volumétrie totale de 211 Go.

3.3.4.9 Vérification de la compression des index

Le nombre d'index compressés est 37 soit 0,1 % des index du schéma SAP. Le cumul de la volumétrie des index compressés représente 390 697 Mo soit 47% de la volumétrie de l'ensemble des index du schéma SAP

Le tableau ci-dessous donne **par tranche de nombre de lignes** les indicateurs suivants :

1. La colonne [CPR-Nb] représente le nombre d'index compressés, la colonne [Nb] représente le nombre d'index non compressés dans la tranche.
2. La colonne [CPR-SUM] représente la somme total en Mo des index compressés, la colonne [SUM] représente la somme totale en Mo des index non compressés dans la tranche.
3. La colonne [CPR-MAX] représente la taille en Mo du plus gros index compressé, la colonne [MAX] représente la taille en Mo du plus gros index non compressé dans la tranche.
4. La colonne [CPR-MIN] représente la taille en Mo du plus petit index compressé, la colonne [MIN] celui du plus petit index non compressé dans la tranche.

Nombre de lignes	Nombre indexes du schéma SAP (1)		Somme totale en Mo des indexes du schéma SAP (2)		Taille en Mo du plus gros Index (3)		Taille en Mo du plus petit index (4)	
	CPR-Nb compression	Nb	CPR-SUM compression	SUM	CPR-MAX compression	MAX	CPR-MIN compression	MIN
1, 000,000,000 - 9,999,999,999	8	0	276 978	0	58 607	0	13 381	0
100,000,000 -	24	7	105 221	103 038	14 255	53 129	1 309	3 625

999,999,999								
10,000,000 - 99,999,999	5	152	8 498	227 603	2 831	6 411	1 005	205
1,000,000 - 9,999,999	0	558	0	80 767	0	855	0	20
100,000 - 999,999	0	895	0	17 079	0	345	0	2
10,000 - 99,999	0	1 567	0	3 724	0	38	0	0
0 - 9,999	0	30 117	0	2 798	0	42	0	0
Total	37	33 296	390 697	435 009				

Tableau 3-8 Etat des lieux de la compression des index du POC

3.3.5 Opérations de post-compression

3.3.5.1 Déplacement des tablespaces SYSTEM, SYSAUX, PSAPUNDO

Les segments présents dans *tablespace* techniques Oracle (SYSTEM, SYSAUX) et les segments d'annulations (segments *Undo*) ne sont ni compressés ni réorganisés. Néanmoins les fichiers de données de ces *tablespaces* doivent être déplacés vers les nouveaux *File System* sur les nouveaux volumes. Les changements des chemins d'accès doivent être répercutés dans les définitions du dictionnaire Oracle et les fichiers de contrôles doivent être régénérés. Les opérations de déplacements utilisent principalement des commandes *Unix* et *SQL*. Elles se décomposent en plusieurs étapes comme suit :

- Préparer la liste des fichiers de données et créer les nouveaux répertoires sur les nouveaux *File System* ;
- Générer la trace de création des fichiers de contrôle sur la base compressée ;
- Arrêter l'instance Oracle ;
- Recopier les fichiers de données conservés vers les nouveaux répertoires ;
- Adapter le fichier trace en remplaçant les noms initiaux; par les nouveaux noms des *File System* ;
- Adapter le fichier de configuration `init<SID>.ora` pour indiquer les nouveaux emplacements des fichiers contrôles ;
- Redémarrer l'instance Oracle en mode *NOMOUNT* puis réinitialiser les *redo* logs ;
- Recréation des fichiers temporaires du tablespace *PSAPTEMP* ;

3.3.5.2 Suppression des anciens tablespaces PSAPSR3*

Une fois que les tables et les index sont déplacés sur les nouveaux tablespaces, les anciens tablespaces sont détruits.

Au préalable nous nous assurons que le contenu des tablespaces à supprimer est bien vide. Il ne doit rester aucun objet sur ces tablespaces. La requête ci-dessous permet de vérifier qu'il ne reste plus aucun segment

```
SQL> SELECT DBA.tablespace_name, COUNT (seg.tablespace_name)
FROM dba_tablespaces DBA, dba_segments seg
WHERE DBA.tablespace_name IN ('PSAPSR3', 'PSAPSR3700', 'PSAPSR3USR')
AND DBA.tablespace_name = seg.tablespace_name(+)
GROUP BY DBA.tablespace_name;
```

TABLESPACE_NAME	COUNT (SEG.TABLESPACE_NAME)
PSAPSR3700	0
PSAPSR3USR	0
PSAPSR3	0

Les suppressions des anciens tablespaces sont effectuées à l'aide de l'outil *BRSPACE* comme indiqué ci-dessous :

```
# brspace -u / -c force -f tsdrop -t PSAPSR3USR
# brspace -u / -c force -f tsdrop -t PSAPSR3700
# brspace -u / -c force -f tsdrop -t PSAPSR3
```

La commande SQL Oracle pour supprimer un *tablespace* est la suivante :

```
Drop tablespace <Nom tablespace> including contents and datafiles;
```

3.3.5.3 Renommage des nouveaux *tablespaces* SAP en remplacement des *tablespaces* existants

Les nouveaux *tablespaces* suffixés par le mot clé NEW sont renommés à l'aide de l'outil *BRSPACE* avec le même nom que leur homologue.

```
# brspace -u / -c force -f tsalter -a rename -t <nouveau_tsp> -n <original_tsp>
# brspace -u / -c force -f tsalter -a rename -t PSAPSR3USR_NEW -n PSAPSR3USR
# brspace -u / -c force -f tsalter -a rename -t PSAPSR3701_NEW -n PSAPSR3700
# brspace -u / -c force -f tsalter -a rename -t PSAPSR3_NEW -n PSAPSR3
```

La commande SQL Oracle équivalente est la suivante :

```
alter tablespace <Nouveau tablespace> rename to <Nom initial du Tablespace>;
```

3.3.5.4 Réinitialisation de la durée de rétention

La durée de rétention des undo ayant été minimisée à 900 secondes, il s'agit de restaurer la durée de rétention initiale (ici 43200 secondes).

```
# sqlplus / as sysdba
SQL> alter system set undo_retention=43200 scope=both ;
```

3.3.5.5 Recalcul des statistiques avec la 11G

Le recalcul des statistiques Oracle est nécessaire pour mettre en phase les informations de volumétrie (Nombre de blocs, taille de segment) et les coûts I/O des tables et index qui ont été réorganisés et compressés avec l'optimiseur SQL.

Calcul des statistiques des tables et index du schéma SAP

Le calcul des statistiques de l'ensemble des tables et index du schéma SAP dure plusieurs heures. Ce calcul doit donc être parallélisé.

Pour paralléliser les traitements l'outil *BRCONNECT*, tout comme l'outil *BRSPACE* dispose des deux paramètres « -parallel » et « -degree ».

Pour forcer la collecte totale et intégrale des statistiques Oracle, nous utilisons une méthode de parallélisation privilégiant la volumétrie des grosses tables de plus de 20 Go. Nous ne cibons donc que 4 sessions Oracle avec un parallélisme élevé de 8 plus adapté aux grosses tables :

```
# brconnect -c -u / -f stats -t <Listetables> -f collect -parallel 4 -degree 8
```

Une seconde méthode qui privilégie plutôt le nombre de tables que leur volumétrie. C'est pourquoi le nombre de sessions Oracle est élevé (8 sessions). La liste des grosses tables est exclue à l'aide du fichier « ReorgExclude.tab ».

```
# brconnect -c -u / -p ReorgExclude.tab -f stats -t all -f collect -parallel 8 -degree 4
```

Calcul des statistiques du dictionnaire avec BRCONNECT

```
brconnect -u / -c -f stats -t oradict_stats -f collect -degree 8
```

Calcul des statistiques de charge système

```
brconnect -u / -c -f stats -t system_stats
```

3.3.5.6 Renommage des points de montage des nouveaux File System

Les nouveaux *File System* sont montés sur des points de montage portant des noms différents. Les opérations de renommage des points de montage des nouveaux *File System* permettent de conserver les anciens noms pour minimiser l'impact sur les procédures d'administration.

Les définitions des fichiers de données Oracle doivent être mises à jours et les fichiers de contrôles Oracle doivent être régénérés. Les opérations de renommage des points de montage étant similaires aux opérations de déplacement des fichiers de données vues précédemment, nous nous contenterons simplement de les énoncer.

- Générer la trace de création des fichiers de contrôle sur la base compressée ;
- Arrêter l'instance Oracle ;
- Sauvegarder les fichiers de contrôles ;
- Renommer les *File System* ;

- Adapter le fichier trace en remplaçant les nouveaux noms des *File System* par les noms initiaux ;
- Adapter le fichier de configuration `init<SID>.ora` pour indiquer les nouveaux emplacements des fichiers contrôles ;
- Redémarrer l'instance Oracle en mode *NOMOUNT* puis réinitialiser les redo logs ;
- Recréation des fichiers temporaires du tablespace *PSAPTEMP* ;

3.3.6 Gains en volumétrie

3.3.6.1 Volumétrie de la base de données

La taille des nouveaux tablespaces qui accueillent l'ensemble des objets SAP compressés représente 1380 Go au lieu de 4140 Go. Soit un ratio de compression de de 2,97 (4140/1390). Le ratio total de la base données sera moins élevé si l'on tient comptes des autres tablespaces où les objets n'ont pas été compressés tels que :

- Les objets du dictionnaire Oracle présent sur le tablespace SYSTEM ;
- Les objets du repository AWR dans le tablespace SYSAUX ;
- Les segments d'annulation gérés sur le tablespace PSAPUNDO ;
- Les segments temporaires gérés dans le tablespace PSAPTEMP ;

La volumétrie globale de la base de données non compressée représente 4771 Go :

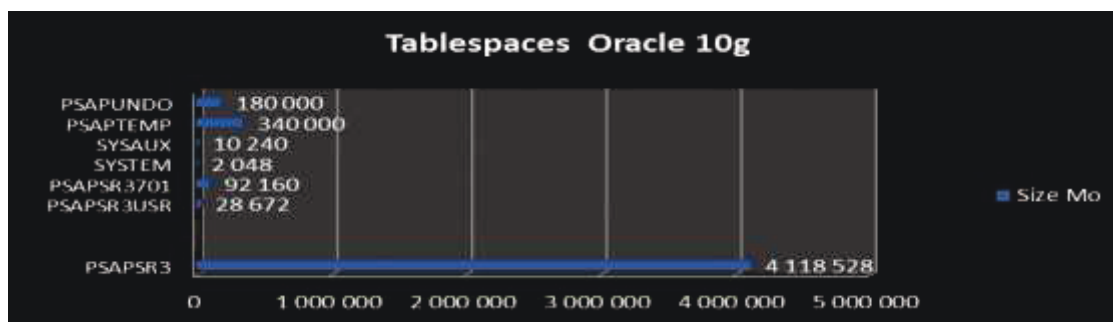


Figure 3-3 Volumétrie des tablespaces Oracle 10g du POC avant compression

La volumétrie de la base de données compressée représente 1835 Go :

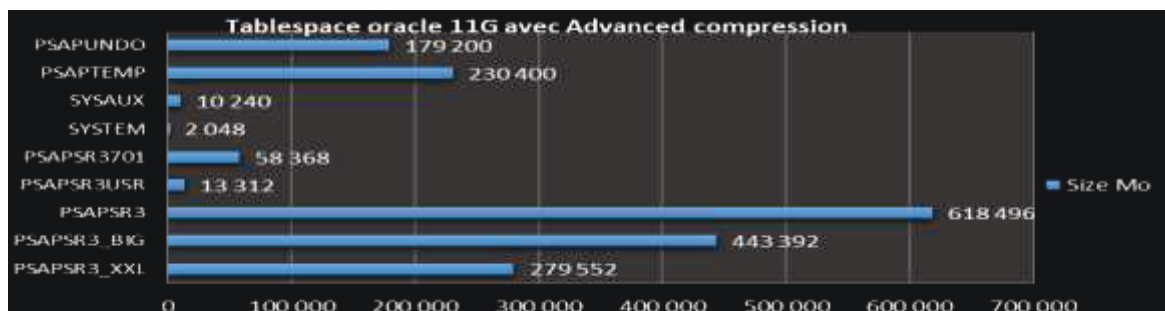


Figure 3-4 Volumétrie des tablespaces du POC après compression

Le ratio global de compression de la base de données est 4771 Go / 1835 Go soit un ratio de compression de 2,6. Le tablespace des UNDO doit être maintenu avec une volumétrie identique car les segments d'annulations ne sont pas compressés. Le ratio obtenu est supérieur aux estimations fournies par les études de SAP (ratio de 2,4). Sans la compression des index, le ratio de compression n'aurait pas été aussi élevé.

3.3.6.2 Volumétrie des File System

L'ancienne configuration comprenait 8 File System représentant un total de 5685 Go avec une place libre restante totale de 491 Go.

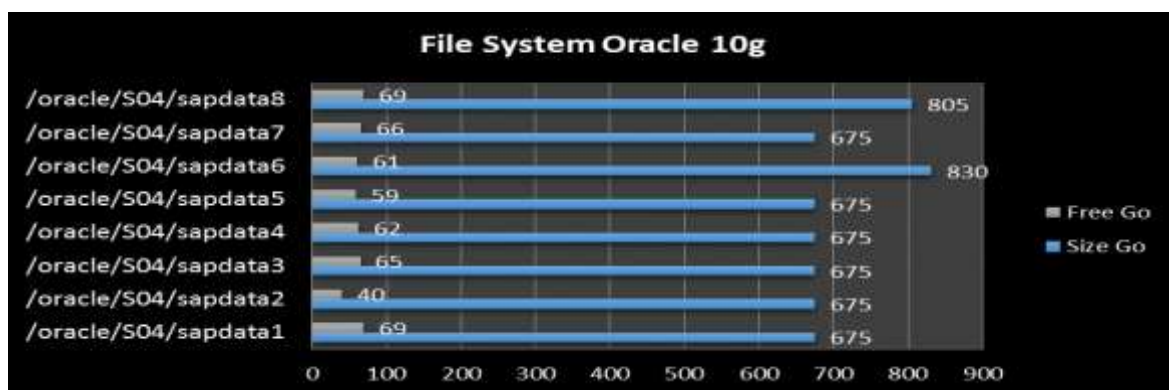


Figure 3-5 Volumétrie des files system du POC avant compression

La nouvelle configuration comprend 6 File System représentant un total de 2550 Go avec une place libre restante de 347 Go. Ces File System portent le même nom que leur homologue avant compression car les points de montage ont été renommés. La place libre restante représente 70% de l'ancienne place libre avant compression. Nous avons pris une marge d'environ 20% lors de ce *Proof Of concept* car nous ne maîtrisons pas le ratio de compression final.

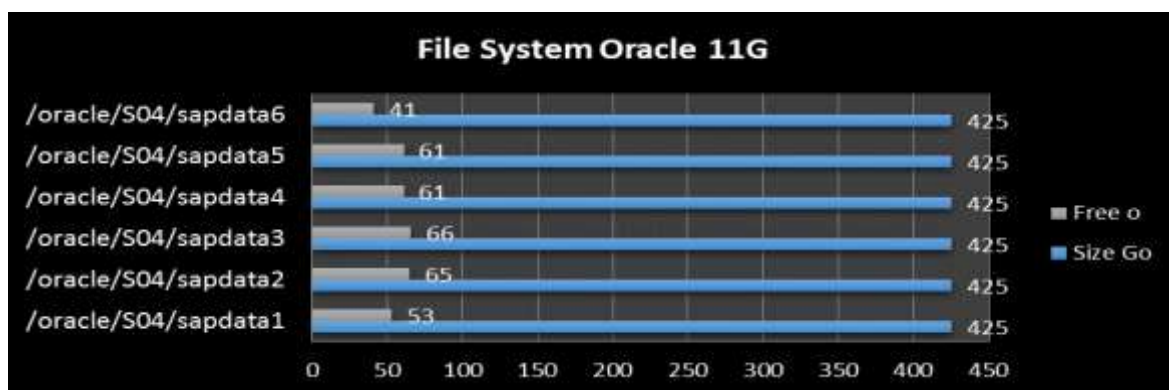
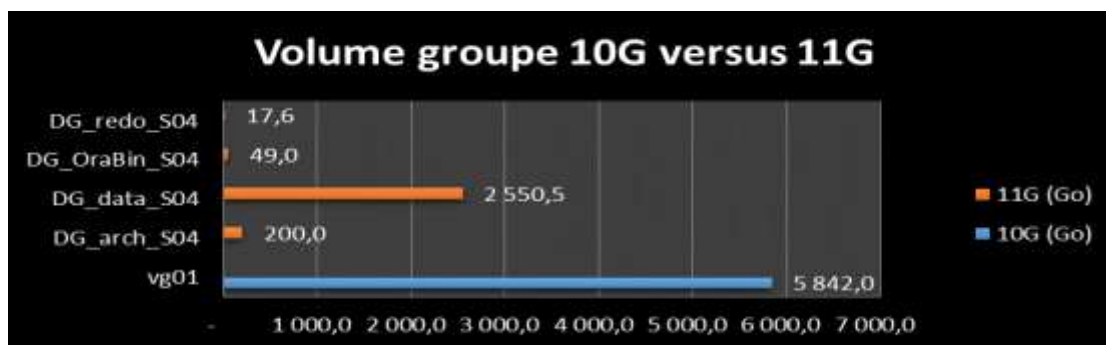


Figure 3-6 Volumétrie des File System du POC après compression

Le ratio de compression des File System de la base de données est de 2,23 (5685 Go / 2550 Go).

3.3.6.3 Volumétrie des Disk-Groups

Dans cette nouvelle configuration les *disks groups* occupent 2817 Go versus 5842 Go, ce qui représente une économie de 3025 Go soit environ presque 3 Téra.



Le ratio de compression global au niveau des *disks groups* est de 2,06. Ce ratio de compression est moins élevé que celui de la base de données ceci est dû essentiellement à la marge de sécurité que nous avons prise lors de ce Proof Of concept.

3.3.7 Evolution des performances.

Nous avons mesuré les temps d'exécution de différents traitements avant la compression et après la compression. Ces traitements concernent :

- Le calcul des statistiques de l'optimiseur SQL ;
- La transaction SAP de génération des traitements ABAP ;

L'exécution des calculs des statistiques Oracle constitue un indicateur sur les performances des accès aux données. Nous avons effectué deux tirs avant et après compression et nous avons constaté des gains significatifs de 35%. 1H55 minutes avec la base de données compressée au lieu 2H59 avant la compression.

Un second test d'évaluation des performances a été effectué à l'aide de la transaction SAP SGEN qui permet de régénérer le code source ABAP, dont les tables étaient essentiellement LONG ou LOB avant compression et SECUREFILE ensuite. Nous avons constaté un relatif maintien des performances sur ce traitement (très légère dégradation de 2%). Le traitement avant compression durait 47mn 43 secondes contre 48 mn 48 secondes après la compression.

Comme nous le verrons dans le chapitre 5 des déploiements, d'autres types d'améliorations des temps de traitement ont pu être constatés comme la diminution des temps de sauvegarde et de restauration, l'amélioration de l'utilisation de la mémoire cache, l'efficacité des lectures séquentielles.

3.3.8 Durées d'indisponibilité.

Avant la compression, nous n'avions aucune information sur la durée d'indisponibilité exacte occasionnée par les opérations. Les tests réalisés avec la copie d'une grosse base de données ont permis d'établir la feuille de route des opérations de compression avec les durées comme l'indique le tableau ci-dessous :

Opérations	Durée HH :MM :SS
Prérequis	1 :40:00
Upgrade de la version Oracle 10g vers la version Oracle 11.2.0.2	1 :40 :00

Préparation de la compression	5:25:00
Création des nouveaux tablespaces avec BRSPACE	4:20:00
Augmentation du tablespace temporaire PSAPTEMP	0:15:00
réduction de la rétention des <i>undos</i> : 43200 => 900 sec	0:10:00
Augmentation de la taille des <i>redologs</i> à 2 Go	0:30:00
Désactivation du mode <i>archive log</i>	0:10:00
Compression des tables SAP	23:52:00
Compression de la table de plus de 1 Téra	1:50:00
Compression des tables de plus de 20 Go	6:59:00
Compression des tables de plus de 20 Go avec des champs LOB	1:22:00
Compression des tables de moins de 20 Go avec des champs LOB	0:33:00
Compression des gros index de plus de 20 Go	8:20:00
compression des autres tables	6:38:00
Post- compression	12:40:00
Vérifier qu'il ne reste plus d'objets dans les anciens tbs PSAP*	0:10:00
Suppression des anciens tablespaces PSAPSR3* 10G	0:20:00
Renommage des nouveaux tablespaces	0:10:00
Réinitialiser la durée de rétention des <i>undo</i> à 43200 comme sur S04	0:10:00
Suppression et recréation du tablespace des <i>undo</i> PSAPUNDO	1:00:00
Déplacement des <i>datafiles</i> SYSTEM, SYSAUX, PATROL ..., etc ...	2:00:00
Recréation du tablespace PSAPTEMP à partir du script <i>controlfile</i>	1:00:00
Recalculer les statistiques (dictionnaire, system & tables)	2:10:00
Renommage des points de montage des nouveaux File system	0:40:00
Adaptation du <i>controlfile</i> et <i>restart</i> Oracle	0:30:00
Tests fonctionnels SAP	4:30:00
Total	43:37:00

Tableau 3-9 Durées d'indisponibilité des opérations de compression

Les opérations de compression ont duré plus de 41 Heures sans compter les opérations des *switchs* des disques sur la baie SAN qui n'ont pas eu lieu pendant le Proof Of Concept.

3.3.9 Majeures difficultés rencontrées

Les principales difficultés rencontrées concernent :

- Les clauses *Next* des tables surdimensionnées;
- Le chaînage des lignes pour certaines tables ;
- L'accentuation des contentions liées aux verrous ;
- La compression par erreur de certaines tables SAP à ne pas compresser.

3.3.9.1 Clause *Next* des tables surdimensionnées

Les clauses *Next* des petites tables avaient été mal définies et étaient trop grandes après la réorganisation et la compression. 75605 Tables d'une taille de moins de 1 Mo avaient une clause *Next* à 1 Mo. D'après la note OSS « [Note 821687 - FAQ: Space utilization and fragmentation in Oracle](#) », SAP recommande de réinitialiser la clause *Next* des tables à 64 K pour éviter de gaspiller inutilement l'espace libre.

Pour éviter ce problème, les clauses *Next* des tables ont été réinitialisées à l'aide de de l'outil *BRSPACE* et du paramètre « *-f next* ».

3.3.9.2 Chainage de lignes

En règle générale, une ligne d'une table est stockée en totalité à l'intérieur d'un bloc. Si la ligne est trop grande pour tenir dans un seul bloc, Oracle la stocke dans plusieurs blocs chaînés par des pointeurs. Le phénomène de chaînage de ligne pénalise les performances de lecture puisque pour lire une ligne Oracle a besoin de lire plusieurs blocs chaînés.

Après les opérations de compression nous avons relevés des tables qui présentaient du chaînage de lignes. L'*Advanced Compression Oracle* ne doit pas être considérée comme une cause de chaînage à proprement parlé.

La liste des tables présentant du chaînage après les opérations de compression :

TABLE_NAME	CHAIN_CNT	SIZEMO	COMPRESS_FOR
STXH	15 318	609,8	OLTP
TMENU01	6 365	7,9	OLTP
ESH_OM_CDATATYFT	33	6,6	OLTP
WDY_CONFIG_DATA	65	6,0	OLTP
ESH_OM_CDATATYT	141	5,8	OLTP
PUTTB_SHD	88	3,0	OLTP
E071C	418	2,3	OLTP
RSZWMDPARAMTEXT	14	2,0	OLTP
RSTT_TDATA	211	0,8	OLTP
/OSP/T_GD_VALUET	20	0,5	OLTP
SIDL_TEST_DATA	6	0,1	OLTP
T5CB6	37	0,1	OLTP
BSPC_DL_XMLSTORX	4	0,1	OLTP

Tableau 3-10 Tables présentant du chaînage des lignes

D'après le tableau ci-dessus, la table STXH comporte au moins 15318 lignes chaînées et occupe 609 Mo.

3.3.9.3 Contentions liés aux verrous

Nous avons observé parfois des contentions lorsque plusieurs transactions tentent d'accéder simultanément au même bloc. Ce type de contention est appelée attente ITL et oblige à réorganiser la table en fixant une valeur plus élevée pour le paramètre *INITRANS*. Le paramètre *INITRANS* indique le nombre initial d'entrées de transactions concurrentes créées dans chaque bloc de données. Avec la compression le nombre de lignes par bloc augmente et la probabilité de

la contention ITL augmente aussi. Il n'existe pas de solution idéale car le fait d'augmenter le paramètre INITRANS fait grossir la table et cela pénalise le taux de compression. Néanmoins ce phénomène n'a été observé que sur 2 tables du schéma SAP (y compris post MEP).

3.4 Bilan du Proof of concept de compression

Le *Proof Of Concept* a permis de valider les objectifs fixés :

- La réduction de la volumétrie de la copie de la base de données de production d'un système transactionnel SAP ECC 6 avec un ratio de 2,6 (1) ;
- Test de la démarche de compression et tests des outils SAP et des scripts associés ;
- La vérification du maintien des performances voir de leur amélioration ;
- L'estimation des durées de chaque opération qui a permis d'établir une feuille de route et de connaître la durée d'indisponibilité globale (41 H);
- La connaissance des difficultés majeures et des corrections pour les prochains tirs à blanc.

(1) D'autres ratios de compression ont été obtenus lors des déploiements tels que sur les systèmes décisionnels et sont décrits dans le chapitre 5.

Comme nous l'avons décrit, l'implémentation de l'Advanced compression est une opération conséquente sur la base de données qui nécessite des réorganisations des tables et des index. De fait, au cours de l'opération, la base n'est pas accessible pour l'application. Au total cela représente une indisponibilité de plus de 41 H sans compter les *switchs* des disques et sans prendre aucune marge de sécurité, ce qui n'est pas envisageable pour un environnement de production.

CHAPITRE 4.

MISE EN ŒUVRE D'UNE SOLUTION DE MINIMISATION DES INDISPONIBILITES

Un second Proof Of Concept a donc été réalisé pour étudier une solution logicielle nommée *GoldenGate* dont nous avons détourné l'usage standard de réplication de données afin de réduire les indisponibilités et de sécuriser les opérations de bout en bout. Ce second POC a finalement permis de mettre au point la stratégie des déploiements des compressions des bases de données qui sont décrits dans le chapitre 5.

4.1 Présentation de la solution GoldenGate.

Oracle *GoldenGate* est une solution logicielle de type CDC (Change Data Capture) pour capturer des données transactionnelles modifiées, et les répliquer entre des environnements hétérogènes, en temps réel et avec un faible impact sur le système source.

4.1.1 Architecture générale

Le cœur de la solution *GoldenGate* est la plateforme logicielle de Change Data Capture, incluant les processus de capture, de routage, de transformation et de livraison des données transactionnelles entre diverses sources et cibles en temps réel.

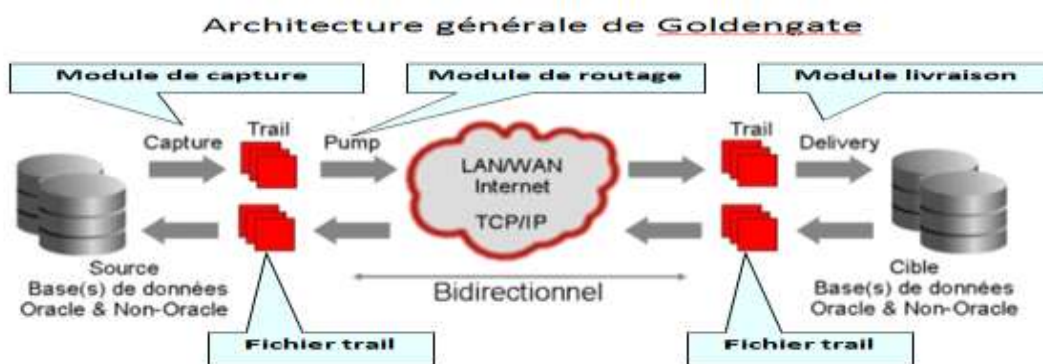


Figure 4-1 - Architecture générale de la solution GoldenGate

GoldenGate a une architecture distribuée s'exécutant sur des serveurs de types différents et interagissant avec des bases de données de différents types. Dans le cadre de notre Proof Of concept, nous utilisons deux instances *GoldenGate* pour répliquer dans un sens uniquement les données des tables de la base de données de tests source dans leurs tables correspondantes sur une copie restaurée.

GoldenGate est constitué des composants suivants:

- Le processus **Extract** collecte les changements dans les redo logs et archive logs de la base de données Oracle et les envoie au Manager. A noter que ce *process* doit se connecter à la base de données pour non seulement collecter l'emplacement des logs

mais également pour connaître les identifiants des objets dans les logs et pour extraire certaines données de la base de données ;

- Le fichier *trail* contient les informations des transactions qui ont été capturées par le processus *extract*, dans un format indépendant du type de système ou de *database*. Un fichier *trail* contient un descripteur de fichier et chaque enregistrement présent dans le fichier contient son propre descripteur d'enregistrement. Chaque processus *extract GoldenGate* conserve la trace de sa propre position dans les fichiers *Trail* en utilisant des checkpoints qui sont stockés dans des fichiers à part. Ce mécanisme comme nous le décrivons après permet à *GoldenGate* d'être résilient aux interruptions et aux pannes ;
- Le processus **Replicat** récupère les changements dans les fichiers *trail* et les réplique dans la base de données cible.
- Le processus **Pump** est un *process* d'extraction optionnel dans le système source. Si le *process pump* n'est pas utilisé, le *process Extract* doit envoyer les données directement dans un fichier *trail* distant. Dans un environnement qui inclut le processus *pump*, le processus *extract* écrit les données dans un fichier *trail* sur le système source, puis le processus *pump* lit ce fichier et envoie les données au fichier *trail* distant. Le *process pump* ajoute ainsi une flexibilité de stockage et permet d'isoler le *process Extract* de toute activité TCP/IP.
- Le **manager** est le *process* de contrôle de *GoldenGate*. Il doit être lancé sur chaque système avant le démarrage des *process Extract* et *Replicat*, et doit rester démarré tant que les autres processus sont démarrés.

4.1.2 Réplication entre des environnements hétérogènes

GoldenGate supporte des SGBDR différents en source et en cible avec une multiplicité de scénarios. Parmi les principaux SGBDR supportés nous citerons Oracle, Microsoft SQL Server, Sybase, IBM DB2 UDB, DB2 OS/390, SQL/MP, SQL/MX.

4.1.3 Différentes applications

Grace à son architecture qui se veut très simple, Oracle *GoldenGate* peut être configuré de plusieurs manières différentes pour plusieurs cas d'utilisation.

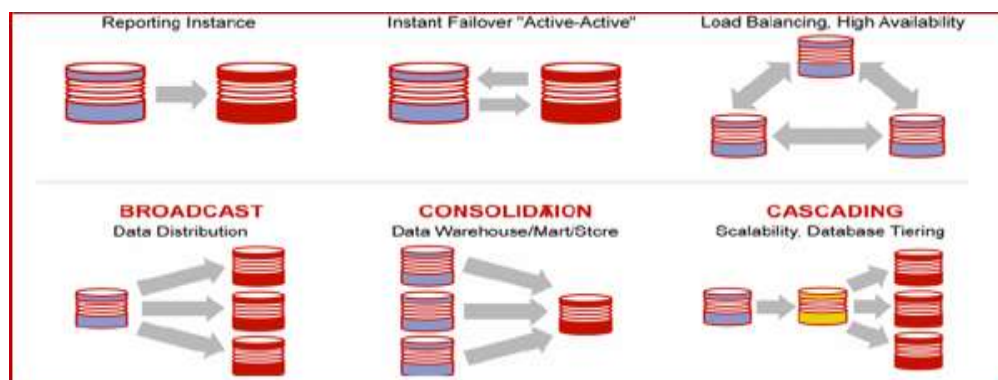


Figure 4-2 les multiples utilisations de *GoldenGate*

- **Disaster recovery**, data protection: *GoldenGate* peut être configuré pour répliquer les données dans une base de données dite de secours et ainsi protéger les données de tout incident matériel ou logiciel. L'avantage de *GoldenGate* est qu'il permet d'avoir un environnement actif – actif, donc de permettre les activités de lecture et écriture sur la base de secours ;
- **Zero downtime migration and upgrades**: *GoldenGate* va permettre également de faire de la migration ou *upgrades* de base de données, de hardware, d'applicatif sans interruption de service. Il est important pour les applications critiques de continuer à avoir l'accès à la base de données pendant les opérations de maintenance planifiées ou non ;
- **Query offloading** : Afin d'éviter les activités de *reporting* sur la base de production qui pourraient diminuer les performances, *GoldenGate* va permettre de répliquer les données sur une base de données de *reporting* et ainsi permettre les opérations de rapport sur celle-ci ;
- **Data Consolidation**: La *Data Consolidation* permet de rassembler un ensemble de données réparties sur plusieurs *databases* sources dans une seule et même *database* cible pouvant servir d'entrepôts de données ou de référentiel.

4.1.4 Principes de captures

Les changements de données sont collectés par accès direct aux logs de transaction natifs des bases de données. Les logs de transaction contiennent tous les changements appliqués à la base de données et sont automatiquement maintenus par la base de données, indépendamment de *GoldenGate*. Par conséquent l'impact sur les performances du système source s'en trouve considérablement réduit, par opposition aux techniques de collecte basée sur des triggers.

Avec Oracle les fichiers *redo logs* ne contiennent pas les ordres SQL mais des données binaires. Les noms des schémas et des objets (tables, vues) ne sont pas stockés dans les fichiers *redo logs*. Ils ne permettent donc pas de reconstituer les ordres DML les ordres DDL (CREATE, TRUNCATE, ALTER, DROP). Oracle met à disposition une fonctionnalité nommée « *Supplemental Logging* » qui permet de compléter les informations manquantes des noms des colonnes, des tables et des schémas dans les fichiers *redo logs*. La fonction *supplemental logging* ajoute également les colonnes complémentaires pour identifier de manière unique les lignes des ordres UPDATE lorsque la table concernée ne comporte pas de clé unique.

Avec *GoldenGate*, le *supplemental logging* d'Oracle doit être activé au niveau de la base de données mais aussi au niveau du schéma (dans notre cas le schéma oracle de SAP), voir de chaque table.

4.1.5 Sélectivité et filtres

GoldenGate permet aux utilisateurs de filtrer les tables selon ses propres critères. Le module de capture ignore alors simplement les entrées dans le journal de transaction qui ne sont pas éligibles aux opérations de réplication. Les caractères joker (*Wild card*) peuvent être utilisés pour remplacer la fin ou le début des noms des tables recherchées. Les utilisateurs peuvent aussi choisir d'appliquer des règles de transformation sur des colonnes spécifiques via les fonctions intégrées, via des fonctions codées par l'utilisateur ou via des procédures stockées.

4.1.6 Résilience aux interruptions et aux pannes

L'outil *GoldenGate* est résilient aux interruptions et aux pannes. *GoldenGate* utilise un numéro de séquence de commit (*Commit Sequence Number*) permettant d'identifier et conserver une trace des transactions qui ont été traitées. Le numéro CSN de *GoldenGate* est indépendant des numéros de séquence de commit qui varient d'un SGBDR à un autre. Par exemple, avec une Base de données Oracle, *GoldenGate* utilise le numéro SCN d'Oracle (*System change Number*) pour représenter son CSN.

Le processus *extract* écrit le numéro CSN à la fois dans son fichier de *checkpoint* et dans ses fichiers *trail*. En cas de *crash* ou bien d'arrêt normal du processus **Extract**, le processus reprend les opérations depuis le fichier *trail* en cours en se basant sur le numéro de CSN déjà enregistré dans son fichier *checkpoint*.

4.2 Présentation du scénario de compression avec indisponibilité proche de zéro.

Les opérations de montée de version vers Oracle 11g, de compression, de réorganisation ayant déjà été décrites dans le sous chapitre du précédent, nous ne décrivons ici que les opérations de configuration, d'exécution et d'exploitation de l'outil *GoldenGate*.

GoldenGate permet de rattraper le delta entre 2 bases de données ayant été à un instant T0 identiques, puis ayant évolué à T1. La base de données productive est répliquée sur une copie, mais reste accessible par le système : la copie de base est upgradée, réorganisée et compressée. A la fin des opérations d'upgrade Oracle 11G et une fois la copie compressée, *GoldenGate* rattrape le delta des mises à jour entre les 2 bases, et il ne reste plus qu'à transférer le système sur sa nouvelle base, compressée. L'interruption totale de service de la plus grosse base de données compressée durant le *Proof Of Concept* n'a pas dépassée plus de 4H. Voici le schéma de ce scénario étapes par étapes.

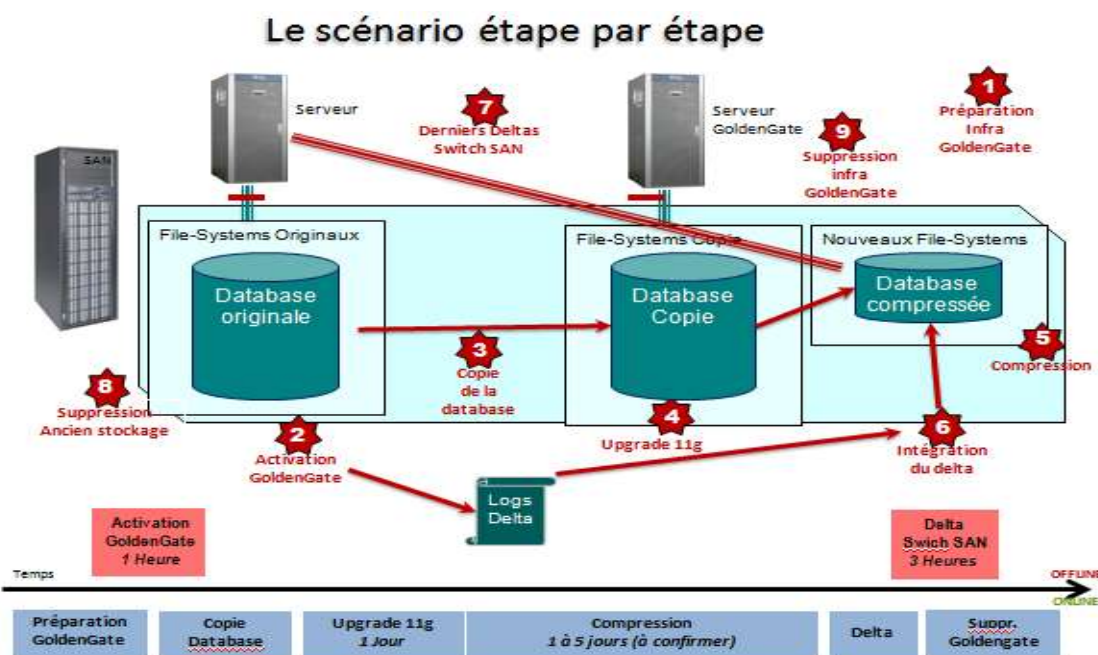


Figure 4-3 - scénario de compression utilisant *GoldenGate*.

1. La préparation de la solution *GoldenGate* est effectuée sur la base source et sur le serveur de reconstruction utilisé pour la réorganisation et la compression de la copie de la base de données.
2. Arrêt de la base de données source pour réaliser un clone de la base sur la baie de disque. La base de données est redémarrée excepté l'application SAP pour éviter toute mise à jour pendant ce laps de temps car elle ne serait pas répliquée. Les processus *extract* de *GoldenGate* sont alors initialisés sur le même numéro SCN d'Oracle. L'application SAP est redémarrée et est de nouveau disponible aux utilisateurs. L'indisponibilité de l'application SAP aura duré environ 1 heure.
3. La restauration du clone de la base de données source est effectuée sur le serveur de reconstruction. Nous soulignons que depuis l'étape 2 précédente, les transactions effectuées par les utilisateurs sont enregistrées par les processus *extract* dans les fichiers *trail*.
4. La montée de version d'Oracle de la 10g vers la 11g est effectuée. Cette étape est indispensable pour pouvoir utiliser l'option Advanced Compression d'Oracle.
5. Les opérations de compression et de réorganisation de la base de données sont effectuées vers des nouvelles ressources de stockage de la baie de disque (nouveaux file system Unix).
6. Après les opérations de compression, les processus *replicat* sont exécutés et intègrent toute l'activité transactionnelle qui a été générée depuis l'étape (2).
7. Lorsque tout le delta a été rattrapé, l'application SAP sur le serveur source est arrêtée. Les processus *replicat* terminent d'appliquer le reliquat des dernières transactions. Les opérations des *switchs* SAN sur la baie de disques sont alors effectuées. Sur le serveur Source, les anciens volumes de la base de données sont masqués et les nouveaux volumes de stockage contenant la base compressée sont présentés à la place des anciens. L'indisponibilité de l'application SAP aura duré environ 3 heures.

L'autre principal avantage de ce scénario de migration outre de minimiser l'indisponibilité est de pouvoir sécuriser les opérations de bout en bout. Si les opérations de compression ou de réorganisation ne se déroulent pas comme prévues, la base de données source n'est aucunement impactée par une restauration de données et les utilisateurs peuvent poursuivre leurs activités de manière transparente.

4.3 Architecture et configurations Goldengate

Ce paragraphe présente l'architecture et la configuration *Goldengate* utilisée dans le cadre de ce POC. Concernant les processus *GoldenGate* *extract* et *replicat* différentes configurations ont été testées.

4.3.1 Capture en mode « Full déporté ».

La configuration de *GoldenGate* qui a été évaluée au cours de ce *Proof Of Concept* repose sur le principe de déporter les processus de capture *extract* sur le serveur distant de reconstruction. Cette configuration permet de minimiser les impacts sur le système source. Les processus *extract*

utilisent comme source les *archives logs* qui sont la copie des fichiers *redo logs*. Voici un schéma qui illustre cette configuration :

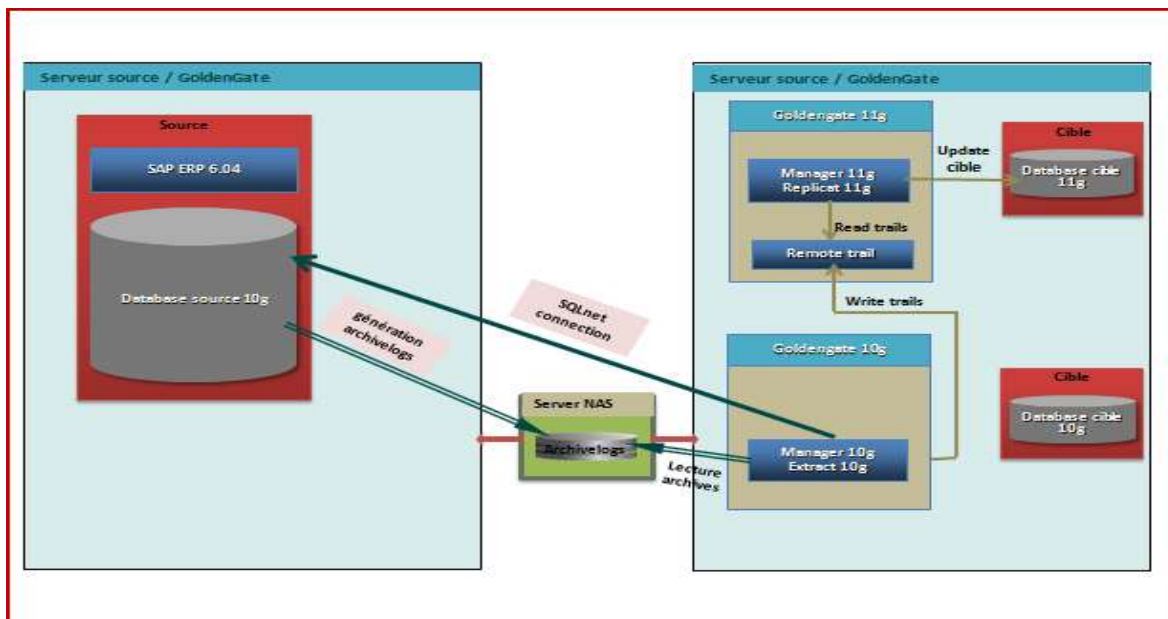


Figure 4-4 Architecture du POC avec GoldenGate

L'architecture donc est composée des éléments suivants:

- Un **NAS** pour partager les *archive-logs* entre la source et la cible (on utilise ici une destination d' *archive-logs* secondaire afin de ne pas perturber la gestion standard des *archive logs* primaire sur la source),
- Un flux **SqlNet** actif pour l'établissement de connexion entre la base source et le manager.
- Deux installations de *GoldenGate* sur le serveur cible :
 - Une installation en version **10g** intégrant *manager* et *extract*
 - Une installation en version **11g** intégrant un *manager* et *replicat*

Remarque : Du fait que la copie de la base de données source est en version oracle 10g, il est nécessaire de configurer un moteur Goldengate 10 de capture afin qu'il soit compatible avec la version Oracle de la database source. De la même manière, la répliation GoldenGate sur la base de données cible compressée en 11g sera effectuée par un moteur GoldenGate 11 compatible avec la version d'Oracle 11g.

4.3.2 Parallélisation des extract et replicat

Un travail approfondi avec un consultant sénior Oracle nous a permis de mettre au point une stratégie de répliation avec du parallélisme. La configuration retenue est une configuration avec parallélisation mixte par filtre et par *range*.

L'idée est la suivante :

- Définir un lot de tables fortement modifiées et ne subissant pas de modification de structure (car les modifications de structure ne sont pas supportées par GoldenGate dans cette configuration), sur laquelle on utilise du range pour paralléliser chaque table ;
- Définir un lot des autres tables (par exclusion des précédentes), subissant potentiellement des modifications de structures, sur laquelle est utilisée une réplcation mono intégrant à la fois DML et DDL afin d'assurer la cohérence des données.

Le diagramme ci-dessous montre une configuration de répartition de charge, elle comprend cinq groupes d'extract et cinq groupes de replicat :

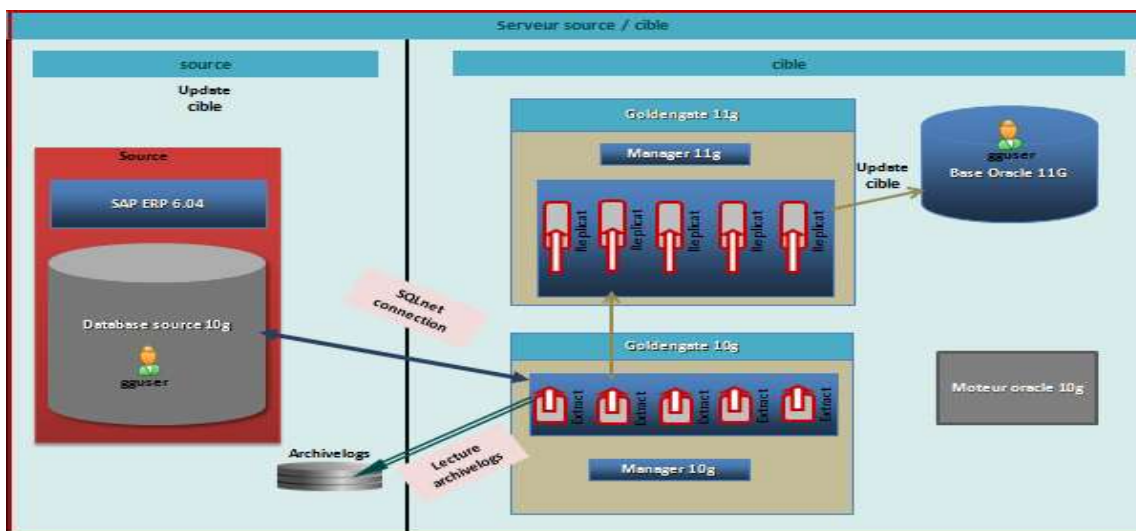


Figure 4-5 Parallélisation des extracts et replicats mixte par filtre et par range

4.3.2.1 Paramétrage technique

La configuration de la source et la cible est constituée de:

Role	Process	Groupe de processus	Fichier de paramètre	Fichier Trail	script obey
Source/Target	MGR	MGR	mgr.prm	N/A	N/A
Source	EXTRACT	E8_1P4	E8_1P4.prm	./dirdat/ea	add-e8.obey
		E8_2P4	E8_2P4.prm	./dirdat/eb	
		E8_3P4	E8_3P4.prm	./dirdat/ec	
		E8_4P4	E8_4P4.prm	./dirdat/ed	
		E8_OTHER	E6_OTHER.prm	./dirdat/ee	
Target	REPLICAT	R8_1P4	R8_1P4.prm	./dirdat/ea	add-r8.obey
		R8_2P4	R8_2P4.prm	./dirdat/eb	
		R8_3P4	R8_3P4.prm	./dirdat/ec	
		R8_4P4	R8_4P4.prm	./dirdat/ed	
		R8_OTHER	R6_OTHER.prm	./dirdat/ee	

4.3.2.2 Performances obtenues

Ci-dessous les performances d'extraction et de répliation à partir d'un volume de données ».

	Extract	Script obey	Fichier Trail	Replicat	script obey	Temps extract	redo	trail	Temps replicat
5 process	E8_1P4	add-e8.obey	./dirdat/ea	R8_1P4	add-r8.obey	00:08:30	4800 Mo 6 redo logs	1287 Mo (11 fichiers)	00:04:20
	E8_2P4		./dirdat/eb	R8_2P4					
	E8_3P4		./dirdat/ec	R8_3P4					
	E8_4P4		./dirdat/ed	R8_4P4					
	E8_OTHER		./dirdat/ee	R8_OTHER					

4.3.2.3 Tests réalisés

Les tests suivants ont été réalisés dans cette configuration :

Type d'opération	Opération sur la source	Status de la répliation sur la cible
DML	Insertion d'une ligne dans la table	Répliqué
	Mise à jour d'une ligne dans la table	Répliqué
	Suppression d'une ligne dans la table	Répliqué
	Truncate d'une table	Répliqué
DDL Table	Création d'une table	Répliqué
	Insertion d'une colonne dans la table	Répliqué
	Modification d'une colonne de la table	Répliqué
	Suppression d'une colonne dans la table	Répliqué
	Suppression d'une table	Répliqué
	Réorganisation d'une table	Répliqué
DDL Indexe	Création d'un index	Répliqué
	Modification d'un index	Répliqué
	Suppression d'un index	Répliqué
	Réorganisation d'un index	Répliqué
	Move d'un index d'un tablespace vers un autre	Répliqué
Autre	Collecte de statistique sur la source	Pas de répliation

Tableau 4-1 Tests réalisés sur la répliation Goldengate

Remarque : L'absence de répliation des collectes de statistiques est lié au fait que la répliation se limite aux données du schéma SAP, et non aux données systèmes dont fait partie les statistiques. Ce n'est pas bloquant, mais cela doit être pris en compte.

4.3.2.4 Conclusion

On note une réduction de la durée de répliation de 56% par rapport au mode « mono extract et replicat ». Les risques d'erreurs ou d'inconsistances de cette solution avec parallélisation peuvent être réduits par un test en fin de répliation afin de vérifier que les tables traitées par range n'ont pas subies de modification de structure comme attendu.

4.4 Mise en œuvre avec GoldenGate.

Nous décrivons dans la suite de ce sous chapitre les opérations de mise en œuvre du scénario GoldenGate permettant de minimiser les indisponibilités lors des opérations de compression. Ces opérations sont les suivantes :

- L'initialisation de la configuration GoldenGate ;
- L'activation de la capture et le lancement des *extract* ;
- Le rattrapage de l'activité transactionnelle et le lancement de *replicat* ;
- Les tests et les vérifications des données ;
- Les résultats obtenus ;
- Les majeures difficultés rencontrées.

4.4.1 Initialisation de la configuration GoldenGate

Les binaires des *extract* ainsi que les scripts Shell qui pilotent les processus GoldenGate sont situés sur le serveur POC. Ainsi l'activité de capture des *extract* présente moins d'adhérence avec le système source et ne consomme pas sa CPU.

La définition de la capture est effectuée sur la base de données source mais les processus *extract* sont démarrés sur le serveur GoldenGate et se connectent à distance sur la base source au travers d'une connexion SQLNet. Voici une représentation graphique de l'architecture :

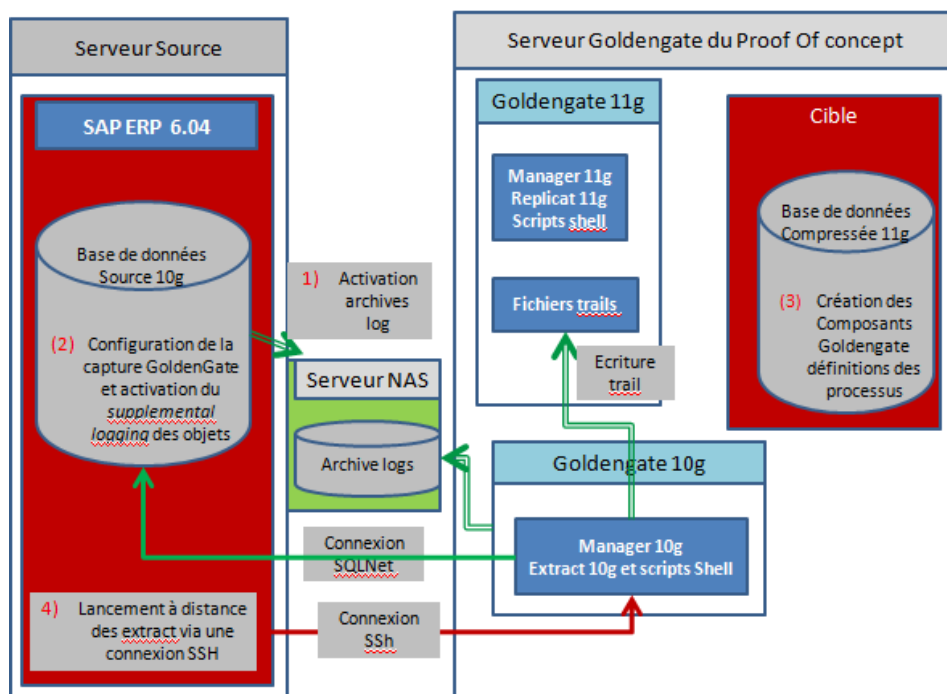


Figure 4-6 Initialisation de GoldenGate

L'initialisation de la configuration de GoldenGate pour la capture de l'activité nécessite plusieurs étapes :

- (1) L'activation du mode archive log au niveau de la base de données source..
- (2) La création de la configuration de la capture (process *extract*) sur la database source.
- (3) La création des composants Goldengate sur le base de données compressée du Proof Of concept.

4.4.1.1 Activation du mode archive log

Les archives log primaires restent stockés sur le serveur source. Seule la destination secondaire qui pointe sur un répertoire du système NAS est utilisée par GoldenGate.

Le mode « *Archived Log Only* » qui est mis en œuvre permet à GoldenGate de se limiter à la lecture des *archive logs*, sans chercher à lire le contenu des *redo logs*. Ceci peut générer un déphasage important entre source et cible en cas de longue inactivité.

Afin de s'assurer que les archives log sont switcher régulièrement, il est préférable de positionner le paramètre « ARCHIVE_LAG_TARGET », qui force un *switch* de façon régulière si l'activité n'est pas soutenue.

4.4.1.2 Configuration de la capture sur la base de données source

Cette étape consiste à créer un schéma Oracle pour GoldenGate dans la base de données source avec lequel GoldenGate va réaliser ses opérations de capture d'activité. Sous ce schéma sont installés des composants GoldenGate (tables, index, packages, *database trigger*). Pour créer les objets GoldenGate sur la source nous avons créé un tablespace spécifique dédié au schéma GoldenGate. Le rôle DBA doit être accordé au schéma GoldenGate

GoldenGate nécessite des informations supplémentaires au niveau des objets Oracle pour effectuer le *tracking* de l'activité Oracle dans ses fichiers *trail*. Ces informations sont obtenues grâce à l'activation de la fonctionnalité Oracle « *supplemental logging* » d'abord au niveau de la base de données et ensuite pour l'ensemble des tables du schéma SAP.

Le *supplemental logging* doit être activé au niveau de la *database* source à l'aide de la commande Oracle « ALTER DATABASE ADD SUPPLEMENTAL LOG DATA ; » et au niveau de chaque table de la source à l'aide de la commande GoldenGate « ADD TRANDATA <NomTable> ».

La capture des ordres DDL (CREATE, DROP, TRUNCATE, ALTER) est effectuée via un trigger (GGS_DDL_TRIGGER_BEFORE) placé au niveau de la base de données source. Les ordres DDL sont tracés avec le numéro de séquence Oracle SCN dans une table du schéma GoldenGate.

4.4.1.3 Configuration des composants GoldenGate sur la cible

Il est nécessaire de configurer deux moteurs GoldenGate sur le serveur cible :

- un moteur Goldengate 10 de capture qui soit compatible la base de données source en version Oracle 10g et qui permet de lancer les processus de Extract.
- un moteur GoldenGate 11 pour la réplication compatible avec la version Oracle 11g de la base compressée et qui permet de lancer les processus *Replicat*.

GoldenGate doit être installé dans son propre répertoire distinct, avec un utilisateur OS dédié. Cela permet de maintenir le logiciel GoldenGate plus facilement.

Les principaux répertoires et fichiers GoldenGate sont les suivants :

/dirprm	*.prm	Fichiers de configuration des <i>extract</i> , des <i>replicat</i> .
/dirdat	2lettre+n°de séquence de fichiers sur 6 digits	Fichiers <i>trail</i> produits par les <i>extract</i> . Il est possible d'indiquer à Goldengate de purger les fichiers <i>trail</i> qui ont déjà été traités par les <i>replicat</i> .
/dirchk	*.chk	Goldengate maintien des fichiers de checkpoint sur le disque afin de gérer des points de reprise en cas d'erreur.
/dirrpt	*.rpt	Fichiers report sur les <i>extract</i> et les <i>replicat</i> . Par défaut, GoldenGate génère un rapport chaque fois qu'un <i>extract</i> ou un <i>replicat</i> est démarré. En outre, le paramètre REPORT doit être inclus dans le fichier de configuration de l' <i>extract</i> ou du <i>replicat</i> pour générer des statistiques d'exécution intermédiaires au moins sur une base quotidienne. Cette information peut être précieuse pour évaluer les performances.
/dirrpt	*.dsc	Fichiers de rejets des <i>replicat</i> (discard file) GoldenGate utilise ces fichiers pour écrire tous les enregistrements qu'il ne peut pas traiter avec le code d'erreur. Ce fichier peut être exploité à des fins de débogage en cas d'échec.

Tableau 4-2 Principaux répertoires et fichiers Goldengate

De même que la source, un schéma Oracle avec tous les objets nécessaires à GoldenGate doivent être créés au préalable dans la base cible Oracle 11g. Le schéma est créé avec un user Oracle spécifique et les objets du schéma sont créés sous un tablespace spécifique.

Des droits de type DBA doivent être accordés au user Oracle de Goldengate.

Une table dédiée aux opérations de checkpoint est créée sur la base cible à partir de la console Goldengate GGSCI en ligne de commande. Cette table doit être indiquée dans le fichier de configuration GLOBALS situé à la racine du répertoire d'installation de Goldengate. Cela permet au *replicat* d'effectuer une meilleure récupération qu'avec le fichier de checkpoint seule.

4.4.2 L'activation de la capture et lancement des *Extract*

Cette opération est délicate car si l'activation de la capture échoue lors d'un déploiement pour une raison quelconque, la suite des opérations est compromise et il faudra trouver une autre fenêtre de maintenance pour recommencer la compression.

L'activation de la capture consiste à initialiser les *extracts* et à les lancer avant de redémarrer toute activité coté application SAP. L'activation de la capture et le lancement des processus *extract* sont ordonnancés depuis le serveur source au moyen d'une connexion distante *ssh* pour synchroniser les opérations avec les arrêts / relance de l'application SAP.

4.4.3 Le rattrapage du delta d'activité

Pour appliquer les derniers deltas, la séquence des opérations est la suivante :

- Arrêt de l'application SAP pour qu'il n'y ait plus aucune activité sur la base de données ;
- Réaliser le switch des *redo* et repérer la dernière séquence de *redo* ;
- Vérifier que les derniers archives logs sont bien vus du serveur *GoldenGate* ;
- Vérifier que les *extracts* ont traités l'ensemble des archive logs, et sont inactifs ;
- Application des derniers *Trail* par les *Replicat* ;
- Effectuer les check des derniers deltas
- Vérifier que les *replicats* sont inactifs.

4.4.4 Les tests et les vérifications des données

Les tests consistent principalement à réaliser des comptages sur des données des bases source et cible compressée après la réplication à des fins de comparaison. A l'issue de ces tests aucun écart ne doit être signalé sinon l'intégrité des données est compromise ainsi que toutes les opérations de compression.

En cas d'écart, comme la base de données source n'est pas impactée par la compression, il n'y a pas nécessité de restaurer la sauvegarde. Les tests de comptages sont effectués à l'aide de requête SQL qui sont de deux types :

- Des comptages des lignes de toutes les tables du schéma SAP par table;
- Des requêtes de comptage sur des données d'une liste de clients prédéfinie.

Les requêtes de comptages des lignes des tables du schéma SAP sont générés dynamiquement et exécutées en parallèle à partir d'un outil que nous avons développé en script Shell et SQL. L'objectif principal a été d'effectuer les comptages de plus des 70000 tables, dont une de plus d'un Téra octet avant compression, en moins d'une heure pour impacter le moins possible le temps global d'indisponibilité.

Les comptages ont fait apparaître des deltas sur certaines tables. Ces écarts sont dus aux raisons suivantes :

- Les processus *extract* ont été arrêtés brutalement avec la commande *GoldenGate* « *Stop Forced* » suite à une erreur de configuration, ce qui a été évité lors des opérations de déploiement ;

- Les opérations de compression sur la cible avec l'outil *brspace* ont généré des mises à jour dans les tables techniques SAP des BRTOOLS qui ont donc évolué par rapport à la source.

Toutes les tables en delta ont été exportées via l'utilitaire d'Oracle *datapump* sur la base de données source après arrêt de SAP sur celle-ci, puis importées sur la base de données cible après exécution des dernières réplifications. Au final, les écarts restants ne concernent que des tables techniques SAP et n'ont pas d'impact sur l'intégrité des données.

4.4.5 Les résultats obtenus

Les différents tirs que nous avons réalisés sur le POC nous ont permis d'affiner la configuration des *extracts* et des *Replikat*. Le scénario définitif retenu est le scénario de parallélisation des *extracts* et *Replikat* utilisant à la fois les méthodes de filtres et de range.

Cette configuration nous a permis lors des déploiements en production de rattraper plus de 2 jours d'activité en moins de 24H

4.4.6 Les majeures difficultés rencontrés

Lors de la mise en œuvre de *GoldenGate* nous avons rencontrés des difficultés au niveau de sa configuration avec les éléments suivants :

- Les impacts de l'activation du *supplemental logging* ;
- La synchronisation du démarrage des processus *extract* ;

Le fait d'activer le *supplemental logging* sur la base source engendre un risque de saturation du file system des archives log, lié à l'augmentation du nombre d'archives logs générés (jusqu'à 15%). Ce risque a été diminué en augmentant la fréquence de purge des archives logs primaires par l'outil *Rman*. D'autre par les archives log secondaires ont été configurés en mode optionnel pour ne pas bloquer la base de données source en cas de saturation.

Une mauvaise synchronisation de l'activation de la capture lors de l'arrêt / démarrage de l'application SAP pouvait entraîner l'échec complet des opérations de compression. Ce risque a été maîtrisé par un appel SSH à la commande d'activation *GoldenGate* dans la chaîne de backup.

Certaines erreurs *GoldenGate* bloquantes nous ont posé des difficultés. Des erreurs Oracle ORA-00600 sont survenues sur la base de données source juste après l'activation du *supplemental logging* et avant le lancement des *extract*. Le Bug Oracle 5196175 a clairement été identifié auprès du support technique et nous avons installé le patch correctif qui a résolu le problème.

Certaines tables avec des champs de type LOB qui était fréquemment mises à jour présentaient parfois une erreur *GoldenGate* récurrente lors de la réplification « *Inconsistent Row Image* » ce qui entraînait la perte de l'enregistrement sur la cible. En fait pour les champs de type LOB comme l'information n'est pas toujours complète dans les fichiers *redo log* ou archives log, *GoldenGate* recherche alors cette information dans la base de données source et ne la trouve plus nécessairement si l'opération date un peu. Ce Bug a été clairement identifié sur le support Oracle. Il est totalement résolu avec la mise en œuvre du mode « *Integrated Capture* » de *GoldenGate*.

4.5 Bilan

La mise en œuvre de *GoldenGate* a nécessité de tester plusieurs configurations de capture et de réplication au cours du second Proof Of concept pour permettre le rattrapage de l'activité dans les meilleurs délais. Le second Proof of concept a permis de maîtriser la solution et d'avoir une configuration de réplication qui puisse absorber l'activité transactionnelle entre le début et la fin des opérations de compression. La solution *GoldenGate* présente les avantages suivants :

- elle permet de lever le risque sur la faisabilité du projet de compression : la durée d'indisponibilité engendrée;
- elle permet de gagner de 6 mois à 1 an sur la durée du projet global (car une plus faible indisponibilité permet une planification beaucoup plus soutenue), donc ROI plus rapide ;
- elle limite le HNO, et faisant donc plus recours aux effectifs assurant le récurrent (meilleure gestion des ressources) ;
- elle sécurise les opérations, car lors du *GoLive* l'upgrade 11g et la compression sont déjà réalisés, et la *database* source reste disponible (peu de risque de *Nogo*, et retour arrière assuré).

CONCLUSION

Nous avons démontré que la mise en œuvre des opérations de compression engendrait des durées d'indisponibilités conséquentes de plus de deux jours qui n'étaient pas concevables pour les applications cœur de métier du client. Ainsi l'utilisation de la solution logicielle *GoldenGate* a permis de réaliser un scénario de compression avec une durée d'indisponibilité totale de moins de 3 heures et de sécuriser les opérations de bout en bout.

Cette solution a nécessité une infrastructure temporaire qui a servi de plateforme pivot pour les opérations de compression, et l'acquisition de licences *GoldenGate* (commercialisé en fonction de la puissance CPU (*core*) des serveurs). Ces coûts ont néanmoins été largement couverts par les gains liés à la compression rendue alors possible.

Bilan sur les déploiements

Plus de 60 bases de données SAP ont été migrées en 8 mois, conformément au planning initial, et malgré les projets de montée de version SAP EHP5 et les changements de baies vers VMAX en parallèle. Ce résultat est largement lié à l'utilisation d'Oracle *GoldenGate*, qui a permis en réduisant drastiquement les fenêtres d'indisponibilité de négocier plus facilement le planning des opérations. La compression des environnements *Informatica* a été reportée de fin 2012 au premier trimestre 2013, afin d'intégrer lors du 3^{ème} trimestre de 2012 des tests de compatibilité entre Oracle 11g et *Informatica*, sans quoi une migration fin 2012 restait techniquement possible.

Pour les systèmes de recette, de pré-production et de production de chaque ligne applicative, nous avons réalisés des upgrades vers la version Oracle 11g avec la compression des bases via l'infrastructure temporaire *GoldenGate*. Les indisponibilités programmées des applications n'ont pas dépassé 0,5 jour. Les gains de volumétries dus à la compression ont été immédiats. Avant de réaliser les opérations sur les productions, nous les avons validés sur les systèmes de recette respectifs et sur les systèmes de pré-production pour mieux maîtriser les risques (qui sont déjà minimiser par l'utilisation de la solution *GoldenGate*).

Les systèmes de type Bac à sable et de formation ont été quant à eux uniquement upgradés en version Oracle 11g et ils ont été rafraîchis ensuite par les copies de production déjà compressés. Les gains de volumétries ont été obtenus après les opérations de compression des environnements de production.

Pour les systèmes de développement ainsi que les systèmes de portails avec Business Objects, la volumétrie ne dépassait pas les 300 Go et de ce fait les gains de compression ne compensaient pas l'achat des licences de l'option Advanced compression d'Oracle. Pour ces environnements, nous avons planifié uniquement un upgrade vers la version Oracle 11g et une indisponibilité de 1,5 jour par environnement. (cf ANNEXE 1 planification des déploiements)

Gains en volumétries

La principale hypothèse portait sur le ratio de compression des bases de données, qui était estimé à 2,4 à partir des informations transmises par l'éditeur SAP. Les ratios de compression obtenus lors des déploiements (entre 2,0 et 3,7) sont assez proches de ceux annoncés par les équipes SAP et

Oracle. Le tableau suivant illustre les gains de volumétries réalisés sur les bases des environnements de production :

- Système SAP ECC ligne non régulé : 2,5 ;
- Système SAP ECC ligne régulé: 2,0 (lié à la présence d'indexés déjà compressés) ;
- Système décisionnel SAP BI régulé : 2,1 ;
- Autre système décisionnel SAP BI : 3,7 (lié à une purge de cubes préalable, mais dont les gains ne pouvaient être obtenus que par une réorganisation complète de la *database*);
- Système non SAP avec Informatica : 2,2 .

Sur les environnements respectifs de recette et de pré production ses gains sont équivalents car il s'agit des mêmes bases de données.

Gains en performances

Nous avons démontré que les bases de données utilisant les techniques de compression ont de meilleures performances que des bases de données ne les utilisant pas. La durée des traitements par lot diminue d'environ 15%. On constate en parallèle une augmentation de la consommation CPU de l'ordre de 10 à 12%. Les bonnes performances de l'Advanced compression d'Oracle s'expliquent par le mode asynchrone de la compression qui minimise les impacts sur les écritures et aussi par le fait que les données une fois compressées sont filtrées, sélectionnées et stockées en mémoire partagée telles quelles par le moteur SQL sans être décompressées. Des collectes de mesures ont été effectuées sur une période de deux semaines avant et après la compression sur les bases de production. Les résultats montrent que le hit ratio minimum du cache de la base compressée (au-delà de 90%) est supérieur de 5% au hit ratio avant la compression (aux environ de 85% minimum en moyenne).

Au-delà de l'amélioration des performances, nous avons constaté aussi une amélioration de l'administration récurrente des bases de données compressées. A savoir une diminution des temps de sauvegardes, une diminution des temps de calcul des statistiques de l'optimiseur, une diminution des temps de rafraîchissement des environnements de recette.

Bilan financier

Les objectifs ciblés ont été définis par rapport aux besoins d'économies des applications et non pas pour répondre à un défi technique. « *La finalité de nos projets d'investissement n'est pas de fournir une solution informatique, mais bel et bien une solution globale d'entreprise* » LRO09]. C'est pour cette raison que les bases de données ne dépassant pas un Téra n'ont pas été compressées car les gains réalisés ne couvraient pas nécessairement le prix de licences de l'Advanced compression d'Oracle et de la solution *GoldenGate* ainsi que les coûts projets. La mise en œuvre de la compression a nécessité de la part de la DSI GDF SUEZ It des investissements. Ces investissements ont été rentabilisés dans des délais courts (ROI atteint en 18 mois pour une des lignes applicatives SAP) et les ROI de l'ensemble des lignes applicatives sont désormais atteints voir dépassés car certaines des anciennes volumétries ont été décommissionnées plus tôt que prévu.

Le bilan financier a été établi à partir du calcul suivant : (Coûts d'exploitation avant) - (Coûts d'exploitation après + Coûts de mise en œuvre de la compression). Le bilan financier a été effectué sur chaque ligne applicative avant les déploiements pour définir si oui ou non si la compression apportait des gains significatifs. Les résultats attendus ont été atteints et nous avons finalement obtenu les Retour sur investissement suivants :

- Système SAP ECC ligne non régulé : ROI en 18 mois ;
- Système SAP ECC ligne régulé: ROI en 24 mois ;
- Système décisionnel SAP BI régulé : ROI en 33 mois ;
- Autre système décisionnel SAP BI : ROI en 28 mois ;
- Système non SAP avec Informatica : ROI en 26 mois ;

Au ROI est souvent associée à la notion de « délai de remboursement ». Le délai de remboursement est le moment où les recettes cumulées dépassent les dépenses cumulées. C'est une notion simple qui indique le temps nécessaire pour rembourser l'investissement réalisé.

L'apport de la compression des bases de données sur le paysage informatique montre son intérêt économique.

Bilan projet

Nous soulignons l'importance des tests qui ont contribué à la qualité du projet, et à minimiser les risques. Une fois la solution validée, les bénéficiaires des applications métiers SAP ont donné leur feu vert pour déployer la compression sur l'ensemble des bases de données car nous leur avons apporté une solution sécurisée. Nous avons rencontrés des difficultés techniques concernant la compression d'une base de données SAP BI mais nous sommes parvenus à livrer cet environnement avant l'échéance finale. A l'issue de ce projet nous avons disposé d'une méthodologie éprouvée au fil des déploiements.

L'optimisation de notre démarche et de nos d'outils nous a permis de mettre en œuvre la compression avec un second paysage informatique de GDF SUEZ IT. Il s'agit d'une autre ligne applicative qui gère la facturation des clients particuliers de GDF SUEZ et comprend des bases de données relationnelles d'environ 20 Téra octets reposant sur des architectures en cluster de serveurs avec la solution d'Oracle « Real Application Cluster assurant la tolérance de panne en cas de défaillance matérielle ou d'interruption planifiée. A ce jour ce second projet est terminé avec les ratios de compression suivants : 2,3 pour le système ERP SAP ISU, 1,8 pour le système SAP CRM et 1,6 pour le système SAP BI. Les ratios sont moins importants car une partie des tables et des index étaient déjà compressés avant avec les options classiques de compression (pas l'option Advanced compression).

La mise en place de la compression des bases de données relationnelle est un investissement quasi obligatoire dans les grosses infrastructures informatique permettant au SI des entreprises de supporter l'augmentation rapide et constante du stockage, pour maîtriser les coûts et les performances. D'autres démarches d'optimisation des coûts et de rationalisation de l'infrastructure de manière plus générale permettent d'aligner les objectifs informatiques sur les objectifs stratégiques métiers tels que le *Cloud Computing*. Le passage au Cloud privé offre aux grands comptes tels que GDF SUEZ et aussi les PME l'opportunité de maîtriser la facture toujours

croissante des services informatiques. Les technologies comme *GoldenGate* permettent aussi de migrer les systèmes existants vers des infrastructures de type Cloud en minimisant les durées d'indisponibilité.

J'espère pouvoir également continuer dans la même optique à participer à des projets innovants de rationalisation et d'optimisation de l'infrastructure, tout en m'alignant sur la stratégie métier des clients.

BIBLIOGRAPHIE

- BOD11. BRtools For Oracle DBA (Citation : 2011)
https://help.sap.com/saphelp_nwpi71/helpdata/en/46/e42438f63966c6e1000000a1553f7/content.htm
- CGF14. Oracle. Oracle Advanced Compression Helps Global Fortune 500 Company Meet Storage Savings Initiative (Novembre 2014). <http://www.oracle.com/technetwork/database/database-technologies/performance/global500acotwp0511final-396808.pdf>
- EPS10. EDF projet SIMM. Maîtrise des volumes de données : sauvegarde & compression (citation: Juin 2010).
http://www.ndclients.com/oracle/forum_oracle_sap/3.presentation_edf_sauvegarde_compression.pdf
- ICS12. IBM. SAP community. Best Practice: Using DB2 Compression Feature in SAP Environment (Citation: 19 octobre 2012) <http://scn.sap.com/docs/DOC-14442>
- IRS12. IBM. Roger E. Sanders. Optimize storage with deep compression in DB2 10. (17 mai 2012).
<http://www.ibm.com/developerworks/data/library/techarticle/dm-1205db210compression/index.html>
- LRO09 : Christophe Legrenzi. Le ROI Informatique <http://www.legrenzi.com/downloads/BP-N20-acadys.pdf>
- JDN01. Journal du net. Encyclopédie e-business.
http://www.journaldunet.com/encyclopedie/definition/283/41/21/proof_of_concept.shtml
- MPS13. Microsoft. Presstore. SQL Server 2012 Internals: Special Storage (Citation: 15 novembre 2013). <https://www.microsoftpresstore.com/articles/article.aspx?p=2225060&seqNum=4>
- MTC14. Microsoft. Technet. SQL server 2014 Page Compression Implementation (Citation: 24 novembre 2014). <http://technet.microsoft.com/en-us/library/cc280464.aspx>
- OAC12. Oracle. Advanced Compression with Oracle Database 11g (Citation: janvier 2012)
<http://www.oracle.com/technetwork/database/storage/advanced-compression-whitepaper-130502.pdf>
- OHC12. Oracle. Hybrid Columnar Compression on (HCC) Exadata (Citation: novembre 2012).
<http://www.oracle.com/technetwork/database/exadata/ehcc-twp-131254.pdf>
- OMP03. Meikel Poess et Dmitry Potapov. Data Compression in Oracle VLDB conférence 2003.
www.vldb.org/conf/2003/papers/S28P01.pdf
- OSS12. Oracle Database: The Database of Choice for Deploying SAP Solutions (juillet 2013).
<http://www.oracle.com/us/products/database/039450.pdf>
- OWN05. Oracle. Useful Notes From The 10gR2 Concepts Guide. Section 5 p26, Oracle Database Concepts 10g Release 2, Oracle B14220-02 (citation: octobre 2005). <http://www.oracle-wiki.net/startdocsusefulnotes>
- RAC12. Raju Pilai. DB2 V10 Adaptive Compression (citation: 27 décembre 2012).
<http://www.channeldb2.com/profiles/blogs/db2-v10-adaptive-compression>
- SMC10. David Salomon and Giovanni Motta. Handbook of Data compression 5ème édition. (ISBN 978-1-84882-902-2) Chapitre 6 dictionary Method (Citation : 2010)

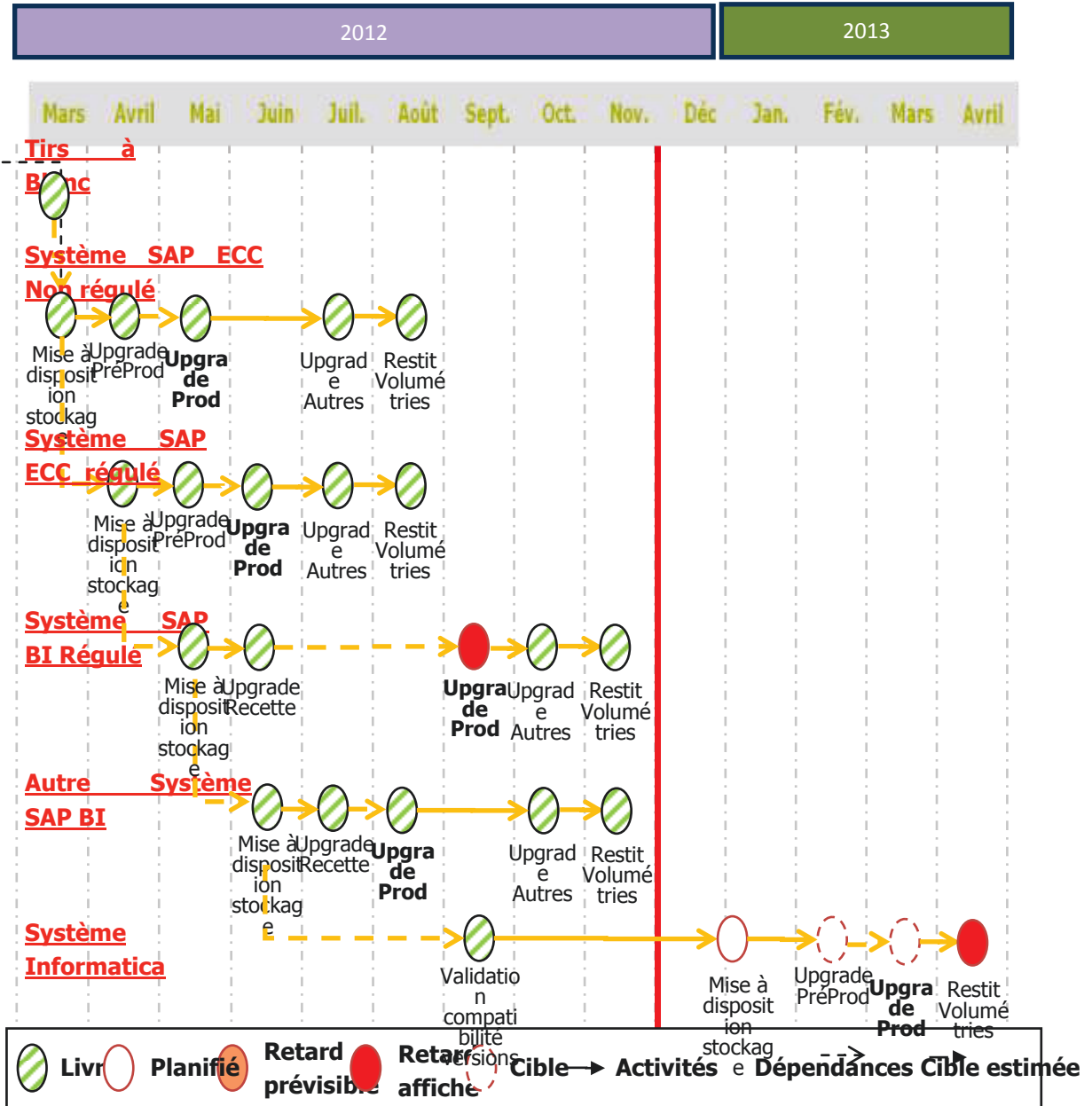
SRP12. SQL Advent. A Look Inside SQL Server Row and Page Compression (19 decembre 2012).
<http://blogs.lessthandot.com/index.php/datamgmt/dbprogramming/how-sql-server-data-compression/>

UMC08. UNISYS. Microsoft SQL Server 2008 Data and Backup Compression (citation: 19 aout 2008).
http://download.microsoft.com/download/5/2/A/52A5523A-C868-4B8D-9F23-8CE62EBBD475/Unisys_SQL_Compression_white-paper_082008.pdf

WHC09. Wikipedia. Huffman coding (Citation : Novembre 2009),
http://fr.wikipedia.org/wiki/Codage_de_Huffman

ANNEXE 1 PLANIFICATION DES DEPLOIEMENTS

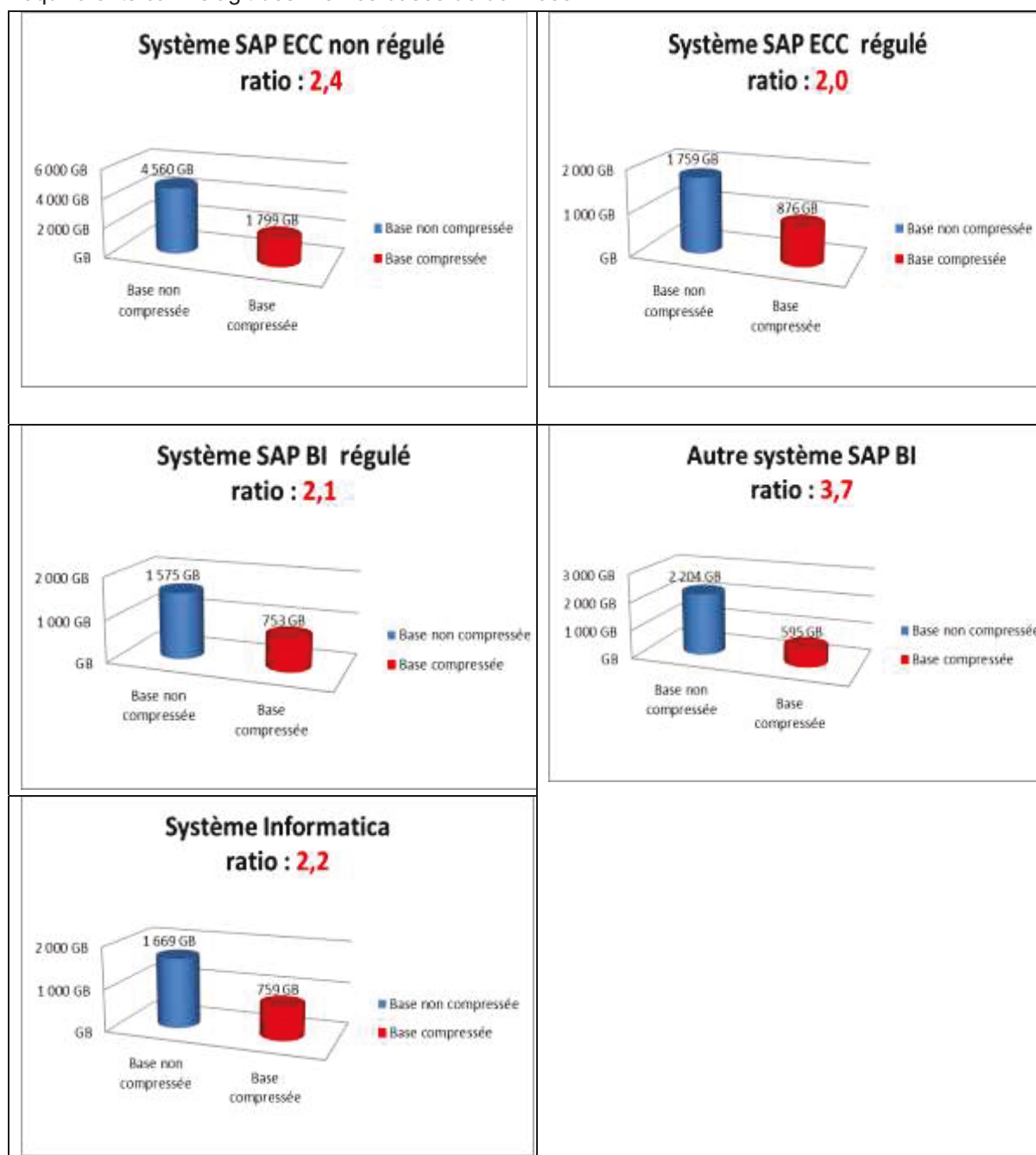
Le tableau suivant indique le planning des opérations de déploiement par ligne d'applications :



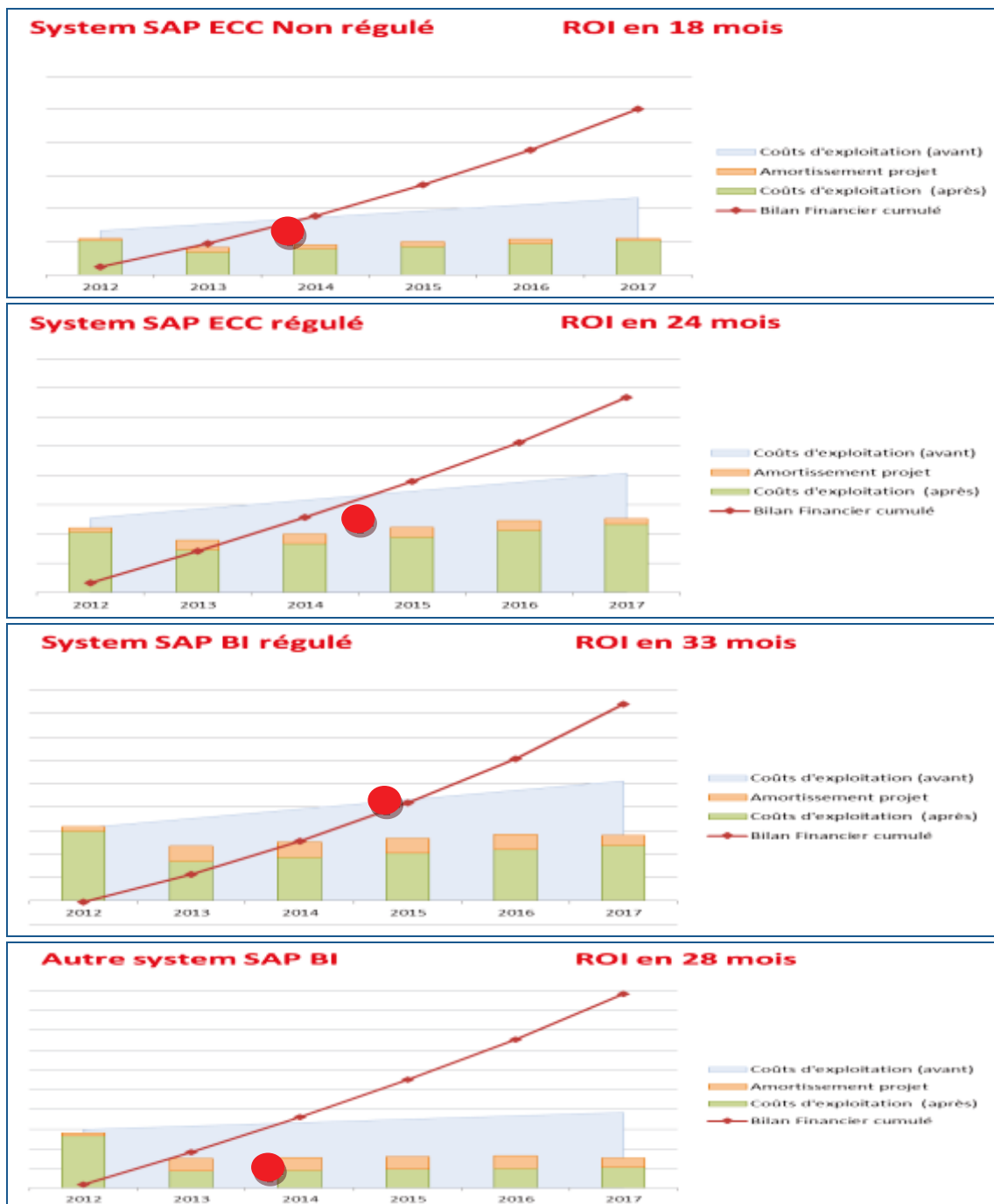
ANNEXE 2 : GAINS DE VOLUMETRIES ET RETOUR SUR INVESTISSEMENT

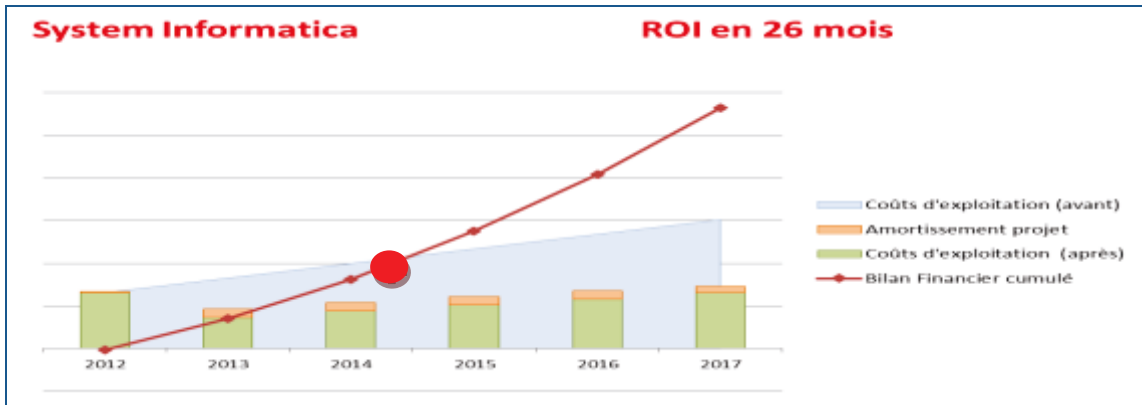
Gains de volumétrie

Le tableau suivant illustre les gains de volumétries réalisés sur les bases des environnements de production. Sur les environnements respectifs de recette et de pré production ses gains sont équivalents car il s'agit des mêmes bases de données.



Retour sur investissement





ANNEXE 3 : GAINS SUR LES PERFORMANCES

Utilisation de la CPU.

L'augmentation de CPU entre 2 et 15% (courbe en rouge) est due en partie aux opérations de compression / décompression lors des lectures mais elle est aussi due à une amélioration des lectures séquentielles des tables et des index compressés comme l'indique le graphique ci-dessous :

