République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université A/Mira de Béjaïa

Faculté des Sciences Exactes

Département Informatique

Mémoire de fin de cycle

En vue de l'obtention du diplôme Master en Informatique

Option

Administration et sécurité des réseaux

Thème

Gestion de données dans le Cloud Computing

Réalisé par :

Mlle BOUAZID Fadila.

Mlle MADANI Khedidja.

présenté devant le jury :

Président	**M^r^ ABBACHE Bournane**	**U. A/Mira Béjaïa.**
Examinateur	**M^r^ AISSANI Sofiane**	**U. A/Mira Béjaïa.**
Examinateur	**M^r^ NAFI Mohamed**	**U. A/Mira Béjaïa.**
Encadreur	**M^r^ SEBAA A/Rezak**	**U. A/Mira Béjaïa.**

Année Universitaire 2012 – 2013

⁂ ⁂ ⁂⁂ *Remerciements* ⁂ ⁂ ⁂⁂

Tout travail de recherche n'est jamais totalement l'oeuvre d'une seule personne. Â cet effet nous tenons à exprimer nos sincères reconnaissances et nos vifs remerciements à tous ceux qui ont contribué de prés ou de loin à l'élaboration de ce travail en l'occurence notre famille qui n'ont jamais cessé de nous encourager.

En tout premier lieu, nous remercions grandement Monsieur "SEBAA A/Rezak" enseignant à l'université de béjaïa, d'avoir accepté d'être l'encadreur pour la réalisation de ce mémoire, nous le remercions pour la confiance qu'il nous a accordée et surtout pour ses encouragements qui nous ont accompagnés durant tout le parcours du travail. Nous n'oublierons pas ses recommandations qui nous ont beaucoup aidés pour finaliser ce mémoire.

En suite aux Monsieurs les membres de jury qui ont eu l'amabilité d'accepter d'evaluer ce travail. Qu'ils trouvent ici l'expression de nos reconnaissances.

Enfin, nous adressons nos plus sincères remerciements à tous nos proches et amis(es) qui nous ont toujours soutenus et encouragés au cours de la réalisation de ce mémoire.

Merci à Dieu de nous avoir donné la force et le courage de tenir jusqu'à la fin de ce travail.

⁂ ⁂ ⁂⁂ *Dédicaces* ⁂ ⁂ ⁂⁂

Â la mémoire de mes chers parents, que leurs âmes reposent en paix

Â mes très chers frères : Halim, A/Rezak et Fahim

Â mes charmantes soeurs : Samia et Wassila

Â mes beaux frères : Hakim et Zahir

Â mes adorables neveux et nièces : A/Rahmane, Houssem, Imad, Amir, Meriem et Nihal

Â mon fiancé Nassim qui m'a toujours soutenu dans mon travail. Sans oublier mes beaux-parents ainsi que mes deux belles soeurs (Sonia et Célia).

Et Â tout(e)s mes ami(e)s et plus particulièrement mon binôme Khedidja

Je vous dois ce que je suis aujourd'hui grâce à votre amour, à votre patience et vos innombrables sacrifices.

Que ce modeste travail, soit pour vous une petite compensation et reconnaissance envers ce que vous avez fait d'incroyable pour moi.

Que dieu, le tout puissant, vous préserve et vous procure santé et longue vie afin que je puisse à mon tour vous combler.

***F**adila*

⁂⁂⁂⁂ *Dédicaces* ⁂⁂⁂⁂

Je dédie ce modeste travail : aux deux êtres qui me sont les plus chers au monde mon père et ma mère à qui je dois le mérite d'être arrivée là, qu'ils trouvent ici l'expression de ma profonde gratitude et mon affection.

Â ma très chère Grand-mère.

Â mon très cher frère : Ammar.

Â mes très adorables soeurs : Sabah, Nabila, Alima, Mounira, Souhila, Wassila, Rima, Ferial et ma belle Cycy.

Â ma très chère nièce : Lyna.

Â mes tantes et mes oncles

Â mes cousines et mes cousins

Â mon binôme : Fadila.

Â tous mes amis (es) sans exception.

Â tous les étudiants de la promotion informatique (2013).

K*hedidja*

Table des matières

Table des figures

Liste des tableaux

Listes des abréviations

BD	**B**ase de **D**onnées
BLOB	**B**inary **L**arge **O**Bject
CPU	**C**entral **P**rocessing **U**nit
DHT	**D**istributed **H**ash **T**able
DW	**D**ata**W**arehouse
EC2	**E**lastic **C**ompute **C**loud
HOLAP	**H**ybride OLAP
IaaS	**I**nfrastructure as a **S**ervice
Id	**I**dentifiant
IT	**I**nformation **T**echnologie
MOLAP	**M**ultidimensional OLAP
OLAP	**O**n-**L**ine **A**nalytical **P**rocessing
OLTP	**O**n-**L**ine **T**ransaction **P**rocessing
PaaS	**P**latform as a **S**ervice
PC	**P**ersonnel **c**omputer
ROLAP	**R**elational OLAP
SaaS	**S**oftware as a **S**ervice
SGBD	**S**ystème de **G**estion de **B**ase de **D**onnées
VM	**V**ue **M**atérialisée

Introduction Générale

Au fur et à mesure que les systèmes informatiques évoluent, la demande en quantité d'espace de stockage, de convivialité et de simplicité dans le travail augmente. Il y a quelques années, les espaces de stockage réduits, les lignes de commandes et les systèmes complexes étaient le quotidien des employés d'entreprises.

Les entreprises modernes traitent de grandes quantités d'informations aussi nombreuses que variées. Ainsi, elles ont besoin de grandes capacités de stockage et une puissance de calcul élevée. Les ressources matérielles et logicielles nécessaires n'étant pas à la portée de toutes les entreprises, le Cloud Computing est une solution pour résoudre ce problème.

Le Cloud Computing, est un nouveau modèle informatique qui consiste à proposer des services informatiques à la demande, accessibles de n'importe où, n'importe quand et par n'importe qui. Cette nouvelle technologie permet à des entreprises d'externaliser le stockage de leurs données et de leur fournir une puissance de calcul supplémentaire pour le traitement de grosses quantités d'informations. Le sujet qui nous ait proposé s'intitule "Gestion de données dans le Cloud Computing".

Notre travail sera divisé en cinq (05) chapitres.

Le premier chapitre présente un aperçu général sur le Cloud Computing, son architecture, ses types ainsi que ses principaux acteurs.

Dans le deuxième chapitre nous définissons les architectures, les outils et les systèmes de gestion de données traditionnels dans le Cloud.

Le troisième chapitre présente les mécanismes d'optimisation de gestion des données.

Dans le quatrième chapitre nous allons proposer d'exploiter les vues matérialisées comme technique d'optimisation qui permet de garantir un bon temps de réponse et une réduction de coût de traitement en effectuant une comparaison des prix mensuels de stockage et de traitement et en limitant la taille des vues matérialisées afin de réduire tous les coûts à payer.

Dans le dernier chapitre nous allons valider notre proposition sous Oracle et vérifier son impact avec trois fournisseurs du Cloud.

En fin, notre mémoire s'achève par une conclusion générale et perspectives.

Chapitre 1

Généralités sur les Clouds Computing

Introduction

Indéniablement, la technologie de l'Internet se développe de manière exponentielle depuis sa création. Actuellement, une nouvelle "tendance" a fait son apparition dans le monde des IT (Information Technologie), il s'agit du "Cloud Computing", qui est un paradigme assez récent consistant en une communication entre un serveur et un ensemble de machines virtuelles qui hébergent une ou plusieurs applications.

Cette technologie offre aussi des occasions aux sociétés de réduire les coûts d'exploitation des logiciels par leurs utilisations directement en ligne, en effet, plusieurs acteurs du secteur informatique parmi lesquels de grands noms comme "Microsoft", "Google", "Amazon" développent et proposent des technologies orientées Cloud au public.[1]

Dans ce chapitre nous pésentons les concepts de base de Cloud.

1.1 Historique

Le mot "Cloud" est apparu au début des années 90 pour désigner des réseaux disposant d'un mode de transfert asynchrone. Le fournisseur "Salesforce.com" est le premier hébergeur de "Cloud" en 1999, suivi en 2002 par "Amazon" qui proposa un ensemble d'hébergements d'application, de stockage et d'offre d'emploi. "Amazon" développa ses services en 2005 (Amazon Web Services) et en 2006 EC2 (Elastic Cloud Computing), ce dernier est le premier service de "Cloud" réellement accessible.

C'est en 2009 que la réelle explosion du "Cloud" survint avec l'arrivée sur le marché de sociétés comme "Google" (Google App Engine), "Microsoft" (Microsoft Azure), "IBM"(IBM Smart Business Service) qui permet à ses utilisateurs de créer et de stocker des documents sur le Cloud. En 2010, "salesforce.com" lance sa base de données Cloud avec "Database.com" pour les dèveloppeurs, marquant ainsi le développement des services de Cloud Computing utilisables sur n'importe quel terminal, exécutables sur n'importe quelle plate-forme et écrits dans n'importe quel langage de programmation.[2]

1.2 Définition

Le Cloud Computing se traduit littéralement par ”informatique dans les nuages”, faisant référence aux technologies d'Internet qui sont souvent représentées schématiquement par un nuage. C'est un concept abstrait qui regroupe plusieurs technologies servant à délivrer des services. Son but est de pousser les entreprises à externaliser les ressources numériques qu'elles stockent, ces ressources offrant des capacités de stockage et de calcul, des logiciels de gestion de messagerie, et d'autres services sont mis à disposition par des sociétés tierces et accessibles, grâce à un système d'identification, via un PC et une connexion à Internet.[1]

1.3 Services du Cloud Computing

Le Cloud computing offre trois (03) services[3] (FIG.1.1), que nous allons présenter :

1.3.1 Infrastructure as a Service

L'Iaas (Infrastructure as a Service) forme le socle du Cloud, il s'agit de serveurs, systèmes de stockage et des équipements réseau fournis en tant que service. Cette infrastructure est mise à disposition de façon à gérer automatiquement la charge de travail requise par les applications.

1.3.2 Platform as a Service

Le service Paas (Platform as a Service) est une plate-forme de développement et de déploiement d'applications fournie en tant que service aux développeurs qui l'utilisent pour créer, déployer et gérer des applications. Cette plate-forme inclut généralement la base de données, les solutions middleware[1], les outils de développement et de gestion, ceux-ci étant tous fournis en tant que service via Internet.

[1]logiciel permettant à deux ou plusieurs applications réparties dans un réseau de communiquer entre elles

1.3.3 Software as a Service

Le service Saas (Software as a Service) permet à ses clients de disposer d'applications, généralement accessibles à l'aide d'un navigateur, sans avoir à se soucier ni à administrer les réseaux, les serveurs et les systèmes d'exploitation.

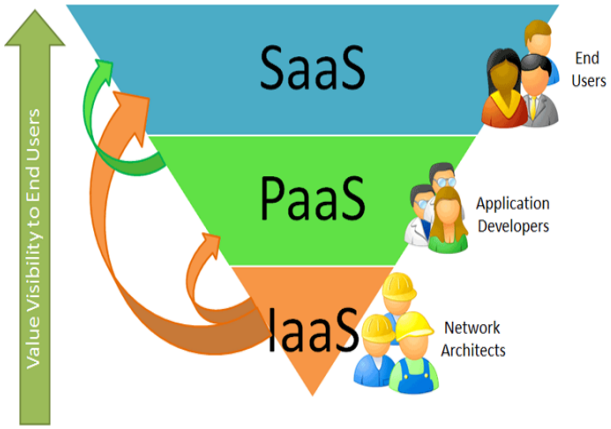

FIG. 1.1 – Les services du Cloud Computing.

1.4 Types du Cloud Computing

Le concept de Cloud Computing est encore en évolution. Nous pouvons toutefois dénombrer quatre (04) types de Cloud Computing [1](FIG.1.2) :

- **Le Cloud privé (ou interne) :** c'est un réseau informatique propriétaire ou un centre de données qui fournit des services hébergés pour un nombre limité d'utilisateurs.

- **Le Cloud public (ou externe) :** c'est un prestataire qui propose des services de stockage et d'applications web pour le grand public.

- **Le Cloud hybride (interne et externe) :** c'est un environnement composé de multiple prestataire interne et externe.

- **Le Cloud Communautaire :** l'infrastructure est partagée entre plusieurs organisations supportant une communauté précise et ayant des préoccupations communes. Elle peut être gérée par les organisations ou par une tierce partie.

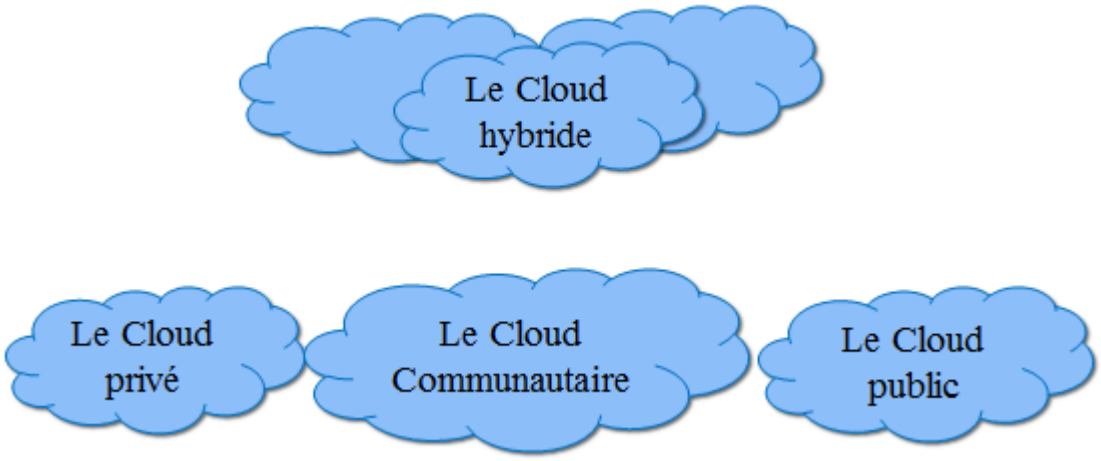

FIG. 1.2 – Les types du Cloud Computing.

1.5 Caractéristiques, avantages et inconvénients du Cloud Computing

1.5.1 Caractéristiques

Le Cloud Computing se caractéristique par :

- **Accès réseau universel :** un environnement de type Cloud Computing s'appuie obligatoirement sur le réseau (Internet) et est accessible via ce dernier, quel que soit le périphérique (PC (Personnel Computer), Mac, TV, Tablette, Smartphone).

- **Mise en commun de ressources :** dans un environnement de type Cloud le nombre de serveurs, la taille de disques ou nombre de processeurs n'est pas important alors que la puissance de calcul, la capacité totale de stockage, ainsi que la bande passante disponible sont les plus importants.

- **Elasticité :** est l'une des propriétés essentielles du Cloud, il s'agit de la capacité à monter et à descendre à l'échelle de manière dynamique sans entrainer réinitialisation ni perte de performance afin de pouvoir garantir la stabilité et la continuité du service utilisant le cloud.

- **Libre-Service :** dans un environnement de type Cloud Computing, il est possible à un utilisateur de consommer les services ou les ressources sans pour autant devoir faire une demande d'intervention auprès de son fournisseur (équipe IT ou fournisseur externe). (ex : un développeur qui souhaite tester son application sur une machine virtuelle représentative d'un poste standardisé de son entreprise peut, seul et au travers d'un portail web, provisionner et utiliser une machine sans devoir solliciter l'équipe IT).

- **Service mesurable et facturable :** dans un environnement de type Cloud le fournisseur de la solution de Cloud est capable de mesurer de façon précise la consommation des différentes ressources (CPU, Stockage, bande passante...), cette mesure lui permet ensuite de facturer le client selon l'usage.[4]

1.5.2 Avantages

En plus des caractéristiques cités en dessus nous allons présenter quelques avantages :

* **Le Cloud accélère les activités** en permettant de transformer les idées en produits et services commercialisables en moins de temps. Le Cloud ouvre une voie d'évolution quasiment illimitée en autorisant une entreprise à se développer sans accumuler les équipements informatiques gourmands en temps et en ressources.

* **Le Cloud transforme les budgets informatiques** en permettant de passer de lourds investissements à un système de paiement au fur et à mesure. Les coûts sont échelonnés et mesurés pour refléter avec précision les besoins et le taux d'utilisation.

* **Le Cloud met à portée de tous de puissantes ressources informatiques :** des organisations de toutes tailles, dans toutes les zones géographiques, peuvent accéder à des ressources informatiques qui étaient précédemment hors de leur portée. Les applications et infrastructures informatiques à l'échelle mondiale deviennent accessibles à tous sans exiger d'investissements initiaux importants.

* **La flexibilité de l'infrastructure** permet aux entreprises d'être plus agiles dans leur système d'information.

* **L'entreprise est libérée des coûts** associés aux matériels informatiques, comme celui des serveurs, de la maintenance ou du réseau.[5]

1.5.3 Inconvénients

* **Confidentialité et sécurité des données :** les données sont hébergées en dehors de l'entreprise. Cela peut donc poser un risque potentiel pour l'entreprise de voir ses données mal utilisées ou volées. Il s'agit donc de s'assurer que le fournisseur dispose d'une sécurité suffisante et qu'il propose une politique de confidentialité concernant les données de l'utilisateur par les SLA (Service Level Agreement)[2].

* **Dépendance à Internet :** en absence de connexion, on a plus accès aux services.

* **Mauvaise utilisation des ressources :** plutôt que d'utiliser la puissance répartie de millions d'ordinateur, on centralise les traitements sur quelques serveurs et on sature la bande passante d'Internet.[52,53]

[2]c'est un document ou contrat qui d´efinit la qualit´e de service requise entre un prestataire et un client

1.6 Les principaux acteurs du Cloud Computing

Les fournisseurs de services de Cloud Computing sont des hébergeurs qui mettent à disposition des infrastructures physiques proposant une plate-forme de Cloud, nous citons quelques principaux acteurs [6] :

1. **Amazon Web Services :** il s'agit d'une demi-douzaine de services, y compris l'Elastic Cloud Computing, pour la capacité de calcul et le Service de Stockage Simple (S3), pour la capacité de stockage à la demande.

   Amazon est un véritable innovateur en matière de calcul sur le web, offrant du paiement à l'usage sur des serveurs virtuels, et de l'espace de stockage. Outre ces offres de base, Amazon offre SimpleDB[3], le CloudFront[4] et le Simple Queue Service[5].

2. **Google :** Google Apps est un ensemble d'outils de productivité de bureau, comprenant messagerie email, agenda, traitement de texte et un outil simple de création de sites web. App Engine, est une "plate-forme en tant que service" qui permet aux développeurs de bâtir et d'héberger des applications sur l'infrastructure de Google.

   Le principal objectif de Google est l'exploration du web et de fournir de la publicité liée aux résultats de la recherche sur le web, l'incursion de Google dans le "logiciel en tant que service" pour les entreprises accélère le mouvement de l'industrie depuis les packages logiciels vers les services hébergés sur le web.

3. **Microsoft :** Windows Azure, une offre de "Windows en tant que plate-forme comme service" comprenant le système d'exploitation et les services pour les développeurs qui peuvent être utilisés pour construire et améliorer des applications Web hébergées.

[3] un service web de base de données

[4] un service web pour la livraison de contenu

[5] un service hébergé pour le traitement des messages entre ordinateurs

1.7 La sécurité du Cloud Computing

La sécurité du Cloud Computing devient un enjeu crucial pour les systèmes d'informations. De nombreux problèmes se posent comme la dégradation ou perte d'informations, le vol ou le transfert non autorisé d'informations, ainsi que des problèmes de qualité de service, de traçabilité et de responsabilité.

Pour qu'un service Cloud soit sécurisé il doit répondre à trois critères indispensables en matière de sécurité :

⋆ **La confidentialité des données :** la confidentialité assure que les données d'un client ne soient accessibles que par les entités autorisées. Les différentes solutions de Cloud Computing comportent des mécanismes de confidentialité comme la gestion des identités et des accès, l'isolation ou le cryptage.

⋆ **L'intégrité des données :** Les clients qui cherchent à externaliser leurs données peuvent évidemment s'attendre à être protéger contre les modifications non autorisées. Les systèmes dans les nuages fournissent un certain nombre de mécanismes de protection de l'intégrité des données.

⋆ **La disponibilité des informations :** l'un des principaux avantages fournis par des plates-formes de Cloud Computing est la disponibilité robuste basée sur la redondance réalisée avec des technologies de virtualisation..[7]

Conclusion

Dans ce chapitre nous avons donnés un aperçu général sur les concepts du Cloud, puis, nous avons défini ses services et ses types, ensuite, nous avons parlé des caractéristiques, avantages et inconvénients, enfin, nous avons cités quelques principaux acteurs du Cloud.

Dans le chapitre qui suit nous allons définir les architectures, les outils et les systèmes de gestion de données dans le Cloud.

Chapitre 2

Les données dans les Clouds Computing

Introduction

Certaines applications (réseau sociaux, sites de e-commerce...) utilisant le cloud font face au défi du BIG DATA. Cette appellation fait référence à des ensembles de données dont la taille (très grande), le taux de croissance (exponentiel) et la structure(non structurée ou semi-structurée) rend difficile et inapproprié leur traitement au moyen des systèmes et les outils de gestion de données.

Ces applications ont donc besoin de données ayant des caractéristiques spécifiques pour pouvoir assurer la disponibilité et l'allocation des ressources à la demande.[8]

Dans ce chapitre nous présentons les différents systèmes et outils de gestion des données.

2.1 Concepts

2.1.1 Cloud Computing et clusters

Le but du cloud computing est de construire un nuage de clusters qui servent à gérer l'interface entre les noeuds et l'utilisateur, c'est à dire d'interconnecter un ensemble de machines sur un réseau défini où les utilisateurs peuvent ensuite déployer des machines virtuelles dans ce nuage, ce qui leur permet d'utiliser un certain nombre de ressources(FIG.2.1). Par exemple de l'espace disque, de la mémoire vive, ou encore du CPU.[1]

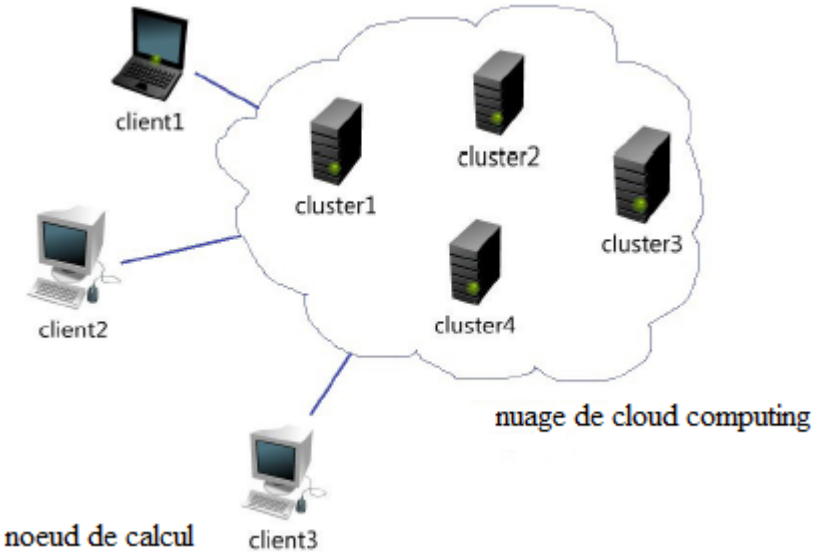

FIG. 2.1 – Machines virtuelles s'exécutant sur un noeud de calcul.

2.1.2 Entrepôt de données

Un entrepôt de données est une collection de données [9] :

⋆ Orientée sujet ou business et organisée par thème afin de permettre la réalisation d'analyses autour des sujets primordiaux et des métiers de l'entreprise ;

⋆ Integrée car provenant des différents systèmes opérationnels hétérogènes ou d'origines diverses de l'organisation. Ces données doivent être standardisées, épurées, unifiées et homogénéisées pour assurer la cohérence et l'intégrité de la connaissance de l'entreprise ;

⋆ Non volatile pour être stable, en lecture seule, non modifiable afin de conserver la traçabilite des informations et des décisions prises ;

⋆ Archivée et datée pour le suivi des évolutions des valeurs des indicateurs à analyser. Les informations stockées au sein du DW (DataWarehouse) ne doivent pas disparaître.

- **Les classes de données :**

Un entrepôt de données peut se structurer en quatre (04) classes de données [10] :

* **Les données agrégées :** elles correspondent à des éléments d'analyse représentant les besoins des utilisateurs, constituent déjà un résultat d'analyse et une synthèse de l'information contenue dans le système décisionnel, et doivent être facilement accessibles et compréhensibles.

* **Les données détaillées :** elles reflètent les événements les plus récents et les intégrations régulières des données issues des systèmes de production qui vont être réalisées à ce niveau.

* **Les métadonnées :** elles constituent l'ensemble des données qui décrivent des règles ou processus attachés à d'autres données, ces dernières constituent la finalité du système d'information.

* **Les données historiées :** chaque nouvelle insertion de données provenant du système de production ne détruit pas les anciennes valeurs, mais crée une nouvelle occurrence de la donnée.

2.1.3 Système de gestion de données

Un système de gestion de données est un logiciel qui permet d'accéder à l'information, intégrer les données et gérer différentes sources d'information dans des environnements souvent hétérogènes : autant de soucis auxquels sont confrontées les directions informatiques. Les données et les systèmes qui les gèrent doivent être disponibles à tout moment et accessibles à tous : clients, employés et partenaires.

Le système de gestion de données joue le rôle de médiateur entre des utilisateurs intelligents et des objets qui stockent l'information.[11]

Pour une gestion sûre et cohérente des données, en présence de multiples processus effectuant des mises à jour de manière concurrente, un système de gestion de données doit respecter les propriétés dites "ACID"[1][12] :

* **Atomicité :** dans une séquence d'opérations liées et dans une transaction, nous devons avoir l'assurance que toutes les opérations ont été exécutées , ou qu'aucune n'a été exécutée.

* **Cohérence :** les données sont toujours dans un état cohérent, il n'y a pas d'états transitoires incohérents qui soient visibles.

* **Isolation :** des processus ne voient que l'état avant et l'état après d'une transaction, ils sont isolés des états intermédiaires.

* **Durabilité :** une fois la transaction terminée avec succès, elle est irréversible.

2.2 Caractéristiques des données dans les Clouds

Parmi les caractéristiques de données de Cloud nous citons :

- **stockage distribué :** les données du cloud doivent pouvoir être partagées entre plusieurs noeuds d'un cluster tout en assurant :
  - Le partage automatique de charge entre les différents noeuds : lorsqu'un nouveau noeud est ajouté à l'ensemble, cette propriété doit être conservée.

  - La distribution automatique de la charge d'un noeud défaillant sur les autres noeuds.
- **Intégration native :** de la capacité essentiel de traiter des requêtes sur de très grands ensembles de données.

- **Traitement distribué :** lorsqu'un client demande un traitement impliquant

[1] Atomicité, Cohérence, Isolation, Durabilité

plusieurs noeuds du cluster, il est préférable que le traitement soit effectué au niveau de chaque noeud impliqué et les résultats remontés vers le client, plutôt que de déplacer l'ensemble des données pour un traitement local.[8]

- **Elasticité des données :** il est possible que le fournisseur demande à l'entreprise une estimation de la charge de consommation, mais en aucun cas les délais ne doivent être allongés si le besoin croît.
  Si le fournisseur réclame des délais pour commander des ressources (serveurs, disques,...) en fonction de la croissance de l'utilisation du service, c'est que son système ne respecte pas le principe d'élasticité propre au cloud computing.

- **Scalabilité des données :** c'est-à-dire la capacité du système à s'adapter aux dimensions du problème qu'il a à traiter, le mécanisme de scalabilité des ressources informatiques est par définition possible uniquement grâce au principe d'élasticité du cloud.[14]

2.3 systèmes de gestion de données des Clouds

Les systèmes traditionnels de gestion et d'exploitation des données spatiales sont du type transactionnel ou OLTP (On-Line Transaction Processing), et Les nouveaux outils d'exploitation des données spatiales sont de type analytique ou OLAP.

1. **Système de gestion de données transactionnelles**

   OLTP est une tâche majeure des BD (Base de Données) relationnelles traditionnelles et une opération quotidienne enregistrée, il réfère à un mode d'exploitation de données tourné vers la saisie, le stockage, la mise à jour, la sécurité et l'intégrité des données. Le système transactionnel est généralement une base de données, développée par application, stockant les données courantes d'une organisation, c'est-à-dire qu'il n'y a pas de données d'archives dans les systèmes transactionnels.[15]

2. **Système de gestion de données analytiques**

OLAP (On-Line analytical Processing) est une tâche majeure des systèmes de data warehouse et une analyse de données et décision. L'analyse en ligne constitue un autre aspect du processus d'entreposage des données. Codd (1993) a défini l'OLAP comme "l'analyse dynamique d'une entreprise qui est requise pour créer, manipuler, animer et synthétiser l'information des modèles d'analyse de données, cela inclut la capacité à discerner des relations nouvelles ou non anticipées entre les variables, la capacité à identifier les paramètres nécessaires pour traiter des grosses quantités de données, la création d'un nombre illimité de dimensions".

Un système OLAP est un dispositif muni d'opérateurs spécifiques permettant l'analyse en ligne des données. Il est également considéré comme un serveur d'applications pouvant traiter directement les données d'un entrepôt ou pouvant être utilisé comme un outil d'exploration de données grâce à une navigation interactive.

Les applications OLAP permettent entre autres de travailler sur des données historiques pour étudier les tendances ou les prévisions d'une activité, ou de travailler sur des données récapitulatives pour créer de l'information stratégique pour la prise de décision. L'analyse en ligne peut aussi bien s'appliquer aux données de l'entrepôt qu'à celles d'un magasin de données. Généralement, elle est plutôt effectuée sur une collection de données encore plus fine appelée cube de données.

- **Cubes de données :** le modèle multidimensionnel permet d'organiser les données selon des axes représentant des éléments essentiels de l'activité d'une entreprise.

  Trois niveaux de représentation des données sont définis dans le processus décisionnel : l'entrepôt qui regroupe des données transversales à l'ensemble des métiers de l'entreprise, le magasin de données qui est une représentation verticale des données portant sur un métier particulier et enfin le cube de données (ou hypercube). Le cube correspond à une vue métier où l'analyste choisit les mesures à

observer selon certaines dimensions. Un cube est une collection de données agrégées et consolidées pour résumer l'information et expliquer la pertinence d'une observation. Le cube de données est exploré à l'aide de nombreuses opérations qui permettent sa manipulation.[16]

Le marché de l'OLAP se divise en trois grands courants qui se différencient quant à la façon dont sont sauvées physiquement les dimensions [17] :

⋆ **ROLAP (Relational OLAP) :** les dimensions sont stockées au sein d'une base de données relationnelles, alors que le schéma logique reste multidimensionnel.

⋆ **MOLAP (Multidimensional OLAP) :** les données sont physiquement stockées sous forme de dimensions.

⋆ **HOLAP (Hybride OLAP) :** le moteur d'analyse embarque les deux approches.

- **Les caractéristiques d'OLAP sont :**

  ◦ Priorité à la sécurité et l'intégrité des données.

  ◦ Optimisation du rapport "espace de stockage et quantité de données" (non-redondance des données).

  ◦ Base de données mise à jour fréquemment (transactions).

  ◦ Outil de requête tributaire de la structure de données (un usager doit connaître la structure de la base de données pour l'interroger efficacement).

  ◦ Requêtes "non-agrégatives" c'est à dire visitent peu d'enregistrements, mais mettent à contribution les techniques d'indexation pour retourner un nombre relativement restreint d'enregistrements répondant à certains critères.[15]

* **La comparaison entre OLAP et OLTP**

	OLTP	OLAP
Utilisateurs	employé, professionnel	Analyste connaissance
Fonction	Opérations au jour le jour	Aide à la décision
Conception de la BDD	Orientée application	Orientée sujet
Donnée	Courante, détaillée	Historique, résumé
Usage	répétitif	ad-hoc
Accés	Read/write	Multiple
Unité de travail	Court, transaction simple	Requête complexe
Enregistrements accés	Dizaines	Millions
Nbre d'utilisateurs	Milliers	Centaines
Taille de la BDD	100MB-GB	100GB-TB
Métrique	Transaction	Requête

TAB. 2.1 – Comparaison entre OLAP et OLTP.

2.4 Modèle d'architecture de données dans les Clouds

Parmi les architectures de données dans les Clouds [19] nous citons :

2.4.1 L'architecture Shared-Everything

Dans cette architecture l'ensemble des processeurs du système fonctionne sous le contrôle d'un seul système d'exploitation (FIG.2.2).

◇ Les clients sont placés dans une file d'attente commune, il y a un équilibrage de charge naturel, les clients se répartissent automatiquement sur les guichets[2] en fonction de leur disponibilité.

[2]processeurs ou systèmes

◇ Tous les agents aux guichets disposent de possibilités identiques d'accès aux dossiers.

◇ L'ensemble des dossiers est partagé par l'ensemble des agents.

◇ La synchronisation entre les agents pour la mise à jour des dossiers s'effectuent par un dialogue direct (les processeurs partagent la même mémoire).

◇ La limite du système se trouve dans le nombre maximal d'agents qui peuvent être mis en parallèle.
◇ Et aussi dans le débit de l'accès aux données.

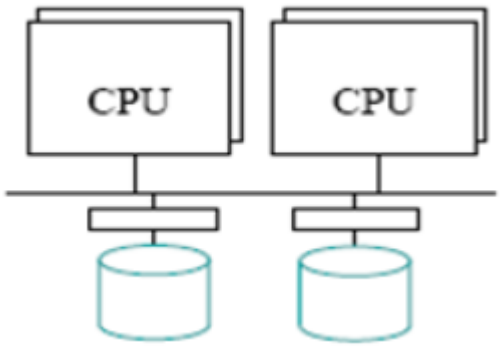

FIG. 2.2 – L'architecture Shared-Everything.

2.4.2 L'architecture Shared-Nothing

Dans cette architecture chacun des noeuds qui composent le système fonctionne sous le contrôle de sa propre copie du système d'exploitation et a un accès exclusif aux disques qui lui sont attachés (FIG.2.3).

◇ La répartition des clients vers les différents guichets ne se fait pas naturellement, il est nécessaire de prévoir un agent en amont chargé d'équilibrer les flux de clients vers les différents guichets.

◇ Dans les systèmes informatiques, ce rôle est dévolue à l'un des noeuds du cluster

qui recueille les informations sur la charge des noeuds et aiguille la requête vers les noeuds les moins chargés.

◇ Chacun des agents a accès à l'ensemble des dossiers.

◇ La synchronisation entre les agents pour la mise à jour des données nécessitent un dialogue entre les différents guichets.

◇ Pour les systèmes informatiques, ce dialogue est réalisé au moyen d'un réseau d'interconnexions, il est bien moins efficace que le dialogue au travers d'une mémoire commune.

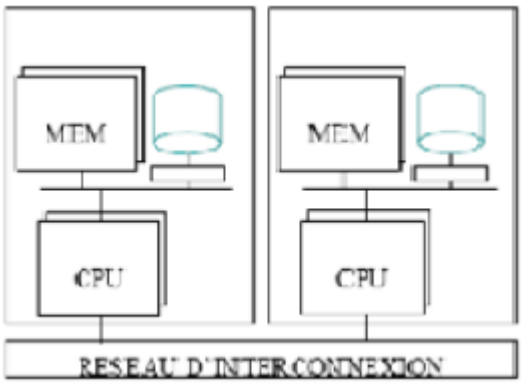

FIG. 2.3 – L'architecture Shared-Nothing.

2.4.3 L'architecture Shared-Disk

Dans cette architecture chacun des noeuds qui composent le système fonctionne sous le contrôle de sa propre copie du système d'exploitation mais peut accéder directement aux disques qui sont partagés entre les différents noeuds(FIG.2.4).

◇ Chaque ensemble de guichets accède en propre à un sous ensemble de dossiers.

◇ Un guichet désirant accéder à des dossiers gérés par des guichets appartenant à un autre sous ensemble doit s'adresser à ce sous ensemble.

◇ Ce modèle implique que la répartition des clients sur les sous ensembles de guichets, soit faite en fonction du sous ensemble de dossiers auquel le client souhaite accéder.

◇ L'équilibrage de charge est donc étroitement lié à la répartition des dossiers entre les différents sous ensembles des guichets et à la distribution des demandes des clients vis-à-vis de ces dossiers.

◇ L'accès d'un sous ensemble de guichets donné à des données gérées par d'autres sous ensembles de guichets nécessite des échanges entre ces sous ensembles.

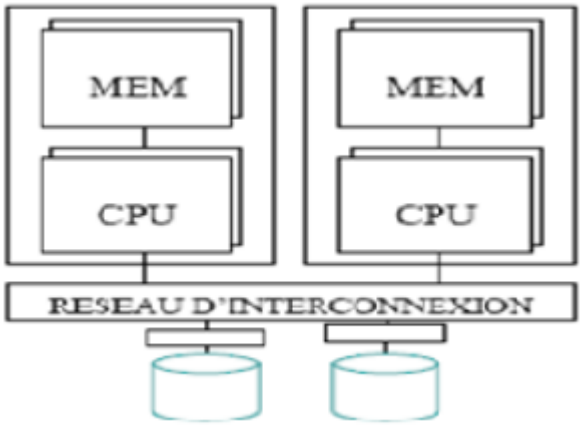

FIG. 2.4 – L'architecture Shared-Disk.

2.5 Outils de gestion de données des Clouds

2.5.1 MapReduce

1. **Définition**

"MapReduce" est un modèle de programmation proposé par "Google". L'objectif est de traiter de manière parallèle de grandes quantités de données suivant un paradigme inspiré des langages de programmation fonctionnels.[20]

2. **Principe de MapReduce**

Le principe du paradigme MapReduce est de diviser les données à traiter en partitions indépendantes, traiter ces partitions en parallèle et finalement combiner les résultats des traitements parallèles.

Ce traitement s'effectue en deux phases Map et Reduce :

- **Map :**consiste en l'étape de découpage puis de distribution des différentes partitions de données constituées. La mise en oeuvre se réalise habituellement à l'échelle d'une grappe de serveurs. Le Map est alors réalisé par un noeud qui distribue les données à d'autres noeuds. Chaque noeud en réception se charge alors du traitement sur les données reçues.
- **Reduce :**est l'opération inverse du Map qui consiste à récolter tous les résultats calculés en parallèle et les fusionner (reduce) en un seul résultat global. Chaque partition de données distribuée se structure comme un couple clée/valeur. La clée est utilisée lors de la fusion pour regrouper les valeurs qui vont ensemble. Toutes les valeurs associées à une clée sont donc réunies à la fin du Reduce.[21]

Le schéma ci-dessous résume le principe général :

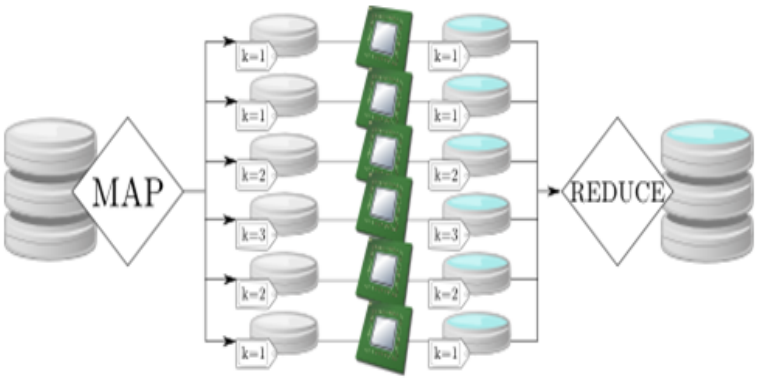

FIG. 2.5 – Principe de MapReduce.

2.5.2 Blobseer

1. **Définition**

   Un BLOB (Binary Large OBject) peut être vu comme une chaîne binaire de taille potentiellement grande (quelques Mo à quelques To). BlobSeer, est un service de stockage et de gestion de données distribuées sous la forme de blobs.

   Un Blob permet un accès rapide aux données et une forte concurrence dans les opérations de lecture, écriture et ajout (read, write, append), de plus, BlobSeer gère le versioning et utilise des politiques de réplication de données pour la tolérance aux fautes.

   Au sein de BlobSeer, chaque blob est identifié par un Id (Identifiant) unique. Les blobs sont divisés en pages de taille donnée (quelques Mo) qui peut être choisie en fonction de l'application considérée. Les requêtes de lecture et d'écriture sont basées sur des quadruplets de la forme (id, version, offset, size) désignant, une certaine version d'un segment commençant à offset et allant jusqu'à (offset + size)-1.[20]

2. **Architecture générale**

   BlobSeer est constitué de quatre types d'agents indépendants, généralement lancés sur des machines distinctes. La figure (FIG.2.6) résume l'architecture de BlobSeer ainsi que les connexions entre les différents agents.

   * **Le provider-manager :** il se charge de gérer les connexions et déconnexions de providers et de noeuds DHT (Distributed Hash Table). Il se charge également d'indiquer, lors de la création d'une page, quel provider est le plus apte à le stocker (équilibrage de charge).

   * **Le version-manager :** il gère la publication des versions. Toute la cohérence du protocole est assurée par cet agent, ainsi que l'atomicité des opérations read, write et append.

* **Les providers :** ils fournissent l'espace de stockage, ils stockent les pages en mémoire vive ou dans des fichiers (si la persistence est activée) sur le système de fichier local.

* **Les sdht :** ils forment les noeuds d'une DHT. Les données sont gérées par un ensemble d'ordinateurs interconnectés, chacun se charge du stockage des donnéesqui stocke et ainsi localise les pages au sein du service.
  Cette table de hachage distribuée utilise le principe du segment tree pour associer efficacement un couple (offset,size) à un ensemble de providers stockant les pages concernées par ce segment.[20]

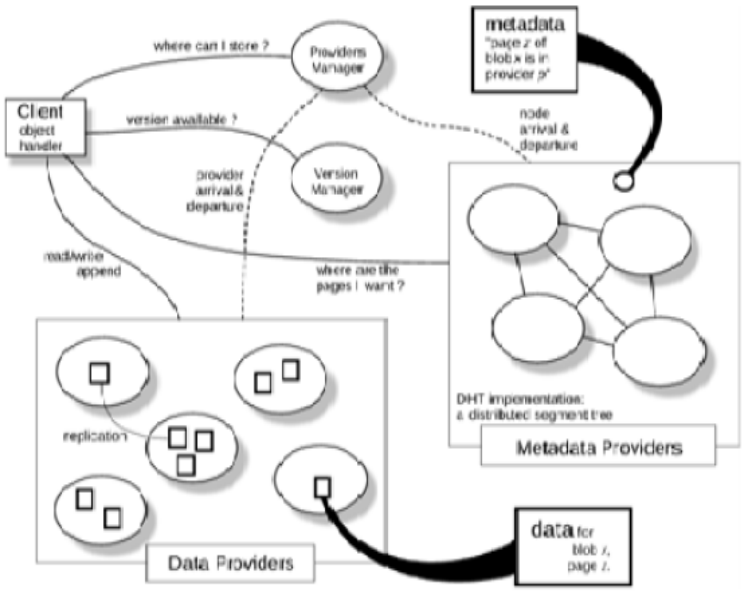

FIG. 2.6 – Architecture de BlobSeer.

Conclusion

Dans ce chapitre nous avons défini beaucoup d'architectures (shared_Everything, shared_Nothing et l'architecture shared_Disk) et des systèmes (OLTP et OLAP) de gestion de données.

Dans le chapitre qui suit nous allons cités quelques mécanismes d'optimisation de gestion de données.

Chapitre 3

Mécanismes d'optimisation de gestion de données

Introduction

Les données sont au coeur des activités de chaque entreprise, leurs importance et leurs volume augmentent de façon exponentielle, il est de plus en plus important de pouvoir se reposer sur des mécanismes d'optimisation de gestion de données qui permettent le progrés de performances des requêtes en offrant un accés rapide aux données.

3.1 Les caches

Un cache est un espace de stockage utilisé pour conserver des données à forte localité spatiale ou temporelle. La mémoire utilisée pour le cache est plus rapide que l'espace de stockage principal, ce qui permet au processeur de travailler avec des copies des données dans le cache plutôt que dans l'espace de stockage principal.

Les caches répartis sont des caches particulièrement bien adaptés aux environnements répartis et ont été largement étudiés par le passé. Si des solutions de caches répartis ont été proposées, elles sont soit intégrées au sein même d'un système de fichiers réparti, soit d'un plus haut niveau et nécessitent que les applications soient adaptées pour les utiliser.[22]

Plusieurs outils et implémentations existent [23], citant :

1. **MemCached**

   MemCached est un système de cache, qui peut être distribué, sur plusieurs serveurs MemCached. C'est un cache générique, c'est à dire qu'il peut stocker n'importe quels objets.

   Memcached fonctionne comme un serveur, au sens applicatif du terme, c'est à dire que, à la manière d'une base de données, les applications adressent des requêtes (sur TCP/IP) à memcached, qui leur répond.

2. **EhCache**

   Ehcache est un système qui n'est pas distribué, mais plutôt répliqué. Avec Ehcache, chaque objet en mémoire est répliqué sur les différentes instances Ehcache, contrairement au principe de memcached.[23]

3.2 Les snapshots

Un snapshot, est une image instantanée d'un disque, à un moment précis, qui permettra donc de revenir en arrière sur l'état de disque en cas de problème. C'est donc un moyen d'effectuer des points de restauration sur un disque.[24]

3.3 Les Indexes

Les indexes sont des structures physiques qui permettent de réduire le temps d'exécution des requêtes en précalculant les jointures et en offrant un accès direct aux données. Cependant, lors du rafraîchissement de l'entrepôt de données, ces structures doivent également être mises à jour, ce qui engendre une surcharge pour le système.

Les indexes de jointures sont des indexes qui permettent de précalculer la jointure mais en gardant seulement les clés des deux relations. Un index est une structure composée généralement d'une clé et une adresse où se trouve l'enregistrement ayant cette clé. Il est organisé généralement en B_arbre(B-tree).[25]

Les caractéristiques des indexes [26] sont les suivants :

1. Mis à jour automatique lors de modifications de la table.
2. Possibilité de créer plusieurs indexes sur une même table.
3. Peut concerner plusieurs attributs d'une même table (index composé).

3.4 Les vues

Une vue, c'est tout simplement une requête d'interrogation à laquelle on attribut un nom. Lors de sélection des données à partir d'une vue, en réalité la requête SQL de la vue qui est éxecutée, par conséquent, les vues ne permettent pas de gagner en performance.

En fait, dans certains cas, les requêtes sur des vues peuvent même être moins rapides que si la requête se fait directement sur les tables.[27]

3.5 Les vues matérialisées

Une vue matérialisée est un objet qui permet de stocker le résultat d'une requête d'interrogation, là où une vue se contente de stocker la requête, la vue matérialisée va stocker directement les résultats (elle va donc les matérialiser), plutôt que la requête. Lorsque l'on fait une requête sur une vue matérialisée, on va donc chercher directement des données dans celle-ci, sans passer par les tables d'origine et/ou une table temporaire intermédiaire.

Une VM (vue matérialisée) est un moyen simple de créer une vue physique d'une table, elle correspond à une photo instantanée des données au moment de l'exécution de la requête. À la différence d'une vue standard, le résultat de la requête est physiquement stocké dans la base de données.[28]

Les objectifs des vues matérialisées [29] les plus importants sont :

- L'amélioration des performances d'accès et la réduction du trafic sur le réseau, elles sont mises à jour périodiquement, ce qui les rendent très efficaces.

- Stocker les résultats d'un ordre SQL dans une table.

- Fournir à l'optimiseur un autre chemin d'accès (habituellement plus rapide) aux données.

3.5.1 Les types de vues matérialisées

Il existe plusieurs types de vues, nous citons les suivants [30] :

1. **Vues matérialisées avec agrégation :** en général, les vues matérialisées contiennent des regroupements. Pour que le rafraîchissement rapide soit possible, la liste de sélection doit contenir toutes les colonnes de la clause GROUP BY, et également un COUNT(*) et un COUNT(colonne) de chaque colonne agrégée. Pour la maintenance, un journal de vue matérialisée doit être présent pour chaque table référencée par la vue matérialisée.

2. **Vues matérialisées avec groupe d'agrégation multiple :** une vue matérialisée peut contenir plusieurs niveaux d'agrégation, typique des requêtes OLAP (On Line Analytical Processing). Par exemple une même requête peut comparer des agrégations à différents niveaux de granularité. Ces requêtes sont évidemment très gourmandes et un prétraitement des résultats est très efficace.

3. **Vues matérialisées avec jointure :** certaines vues matérialisées contiennent seulement des jointures et pas de regroupement. Elles contiennent le résultat de jointures souvent coûteuses, entre les tables.

4. **Vues matérialisées emboîtées :** une vue matérialisée emboîtée est une vue matérialisée dont la définition est basée sur une vue matérialisée. L'utilisation de ce type de vues est souvent utilisée si l'on souhaite garder les caractéristiques de rafraîchissement rapide sur une vue matérialisée complexe. Si l'on désire avoir une vue matérialisée qui contient des jointures et des agrégations en gardant la fonctionnalité de rafraîchissement rapide, elle sera nécessairement imbriquée.

3.5.2 Problèmes des vues matérialisées

De nombreux travaux traitent des problématiques concernant les vues matérialisées dans différents contextes. Nous pouvons distingué deux principaux aspects :

⋆ le problème de sélection des vues matérialisées qui consiste à déterminer l'ensemble de vues à matérialiser de telle sorte que la contrainte prise en compte soit optimal.

⋆ le problème de maintenance des vues matérialisées qui se propose de répercuter immédiatement les mises à jour survenues au niveau des sources de données.

1. **La sélection des vues matérialisées**

Le problème de sélection des vues matérialisées consiste à construire une configuration de vues optimisant le coût d'exécution d'une charge donnée. Cette optimisation peut être réalisée sous certaines contraintes telles que l'espace de stockage alloué aux vues sélectionnées ou une borne supérieure du coût de maintenance des vues sélectionnées [31].

Il existe trois possibilités pour sélectionner un ensemble de vues matérialisées [32] :

a) Matérialiser toutes les vues : cette approche consiste à matérialiser la totalité des vues candidates, elle donne le meilleur temps de réponse pour toutes les requêtes mais, stocker et maintenir toutes les vues est impraticable.

b) Matérialiser aucune vue : dans ce cas nous sommes obligés d'accéder aux données des relations de base. Cette solution ne fournit aucun avantage pour les performances des requêtes.

c) **Matérialiser seulement une partie des vues :** dans ce cas, il existe une certaine dépendance entre les vues, c'est à dire que la valeur de certaines vues peut être calculée à partir des valeurs d'autres vues. Il est alors souhaitable de matérialiser les parties partagées par plusieurs requêtes. Cette solution semble la plus intéressante par rapport aux deux approches précédentes.

Quel que soit le type de problème de sélection des vues matérialisées, ce dernier peut être défini de la manière suivante :

Étant donné une contrainte de ressource S (capacité de stockage, par exemple), le problème de sélection des vues consiste à sélectionner un ensemble de vues {V1, V2,. . ., Vk} minimisant une fonction objectif (coût total d'évaluation des requêtes et/ou coût de maintenance des vues sélectionnées) et satisfaisant la contrainte (voir la Figure FIG.3.1).

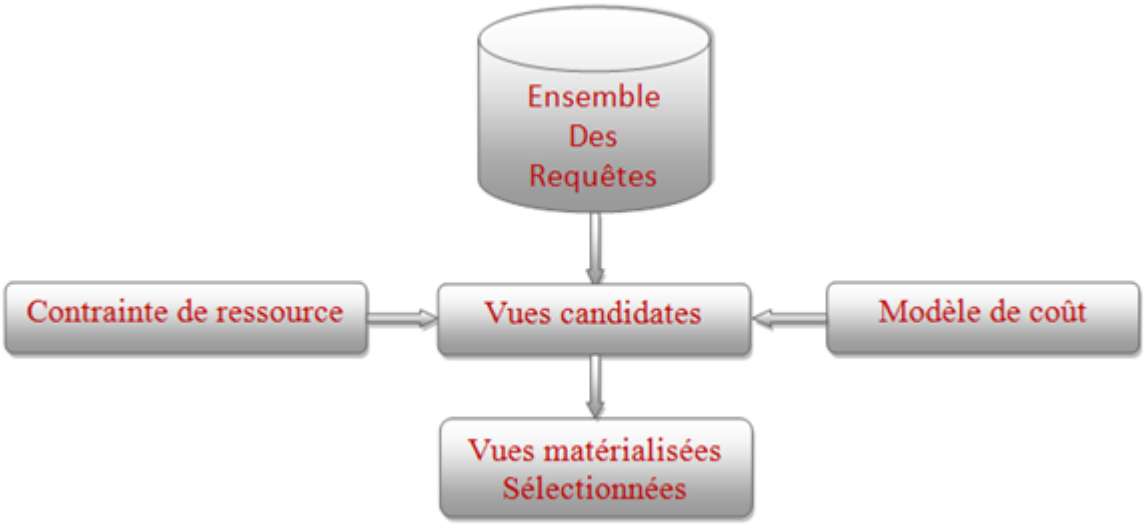

Fig. 3.1 – Le processus de sélection des vues matérialisées.

Rappelons que le rôle principal des vues matérialisées est de réduire le coût d'évaluation de certaines requêtes Q = {Q1 ; . . . ; Qk} (les plus fréquentes, par exemple) définies sur l'entrepôt.

Les algorithmes proposés pour déterminer un ensemble de vues à matérialiser peuvent être classés par le type de contraintes qu'ils utilisent :

- avec la contrainte d'espace de stockage, c'est-à-dire que la taille des vues sélectionnées ne doit pas dépasser la capacité de l'espace disponible. Un modèle de coût est proposé pour estimer le coût de stockage de chaque vue (nombre de n-uplets de la vue).
- avec la contrainte du temps de maintenance des vues. Les algorithmes proposés sont basés sur des heuristiques et utilisent des techniques de graphe et les plans d'exécution[33].

2. **Maintenance des vues matérialisées (rafraîchissement)**

La création d'une vue matérialisée à partir des données ne suffit pas, il faut donc la tenir à jour lorsque les données des tables d'origine changent.

Nous distinguons deux possibilités de mise à jour [28] :

* **Mise à jour sur demande :** afin de mettre la vue matérialisée à jour ponctuellement, une procédure stockée est nécessaire. Le plus simple sera de supprimer les données de la vue matérialisée, puis d'insérer les données à jour grâce à la même requête de sélection ayant servi à créer/initialiser la vue matérialisée.

* **Mise à jour automatique :** si à l'inverse, c'est à dire que les données soient toujours à jour par rapport aux derniers changements de la base de données, l'utilisation des triggers est importante pour mettre à jour la vue matérialisée.

- **Le type de rafraîchissement :** afin d'assurer une certaine cohérence des données, il faut mettre à jour les vues matérialisées et les tables périodiquement.
  Il existe trois façons de mises à jour qui sont, la régénération complète, rapide et forcée [29] :

- **Rafraîchissement complet :** il va ré-exécuter la requête basée sur la table de base et remplace l'ensemble des données de la VM par les données obtenues et ceci même si la table de base n'a pas été modifiée, selon le volume de données qui satisfait la requête, ce rafraîchissement peut être gourmant en ressources.

- **Rafraîchissement incrémental (rapide) :** son principe est de propager uniquement les données modifiées depuis le dernier rafraîchissement, ce type de rafraîchissement dit aussi rapide nécessite que la base de données stocke les modifications enregistrées sur les données des tables de base, on utilise pour cela un journal (fichier Log). Ce type de rafraîchissement est particulièrement efficace si les tables de base sont relativement peu modifiées.

- **Rafraîchissement forcé :** dans ce type de rafraîchissement, lorsqu'une régénération rapide n'est pas possible, alors une régénération complète est exécutée.

Conclusion

Nous avons présenté dans ce chapitre les différents mécanismes d'optimisation de gestion de données tels que : les caches qui permettant d'offrir certaines garanties en terme de performances et de cohérences à l'utilisateur, le snapshot qu' est un moyen d'effectuer des points de restauration sur un disque (système ou data), les indexes qui permettent de réduire le temps d'exécution des requêtes en précalculant les jointures et en offrant un accès rapide aux données et les vues matérialisées qui permettent d'améliorer les performances d'accès et la réduction du trafic sur le réseau.

Dans le chapitre suivant nous allons proposer d'exploiter les vues matérialisées.

Chapitre 4

Proposition

Introduction

Derrière le nuage du Cloud Computing se cache des serveurs offrant du stockage et des traitements en ligne via le réseau Internet. Le Cloud Computing est basé sur un modèle de paiement à l'usage et permet un accès simple et à la demande à un ensemble de ressources partagées.

Dans le choix d'une technologie de Cloud Computing, l'aspect financier a évidemment un rôle crucial et la tarification du Cloud Computing est un sujet assez complexe à cause de la difficulté de comparaison et d'estimation des différentes offres qui sont variées d'un fournisseur à l'autre.

Dans ce chapitre nous présentons notre solution pour l'optimisation des coûts à payer dans le Cloud Computing.

4.1 Problématique et motivations

Les principales caractéristiques des données dans le Cloud sont leur grande taille (données volumineuses) et la complexité des requêtes décisionnelles dues aux opérations de sélection, de jointure (très coûteuse) et d'agrégation ont rendu le temps de réponse élevé ce qui génère des coûts de traitement importants.

Notre objectif est d'exploiter un mécanisme d'optimisation de données qui est les vues matérialisées afin de garantir un bon temps de réponse aux requêtes et une réduction de coût de traitement dans le Cloud Computing.

Et pour cela nous étudions les coûts de stockage et de traitement des données dans le Cloud Computing et les paramètres qui influent sur ces derniers.

4.2 Les coûts et la tarification dans les Clouds

Amazon, Google et Microsoft, entre autres, se livrent une guerre de prix sur le stockage et le traitement dans le Cloud c'est pour cela il est très difficile de dire quel service est le moins cher que l'autre, alors nous avons fait une comparaison des prix mensuels de stockage et de traitement pendant l'année 2012, qui se limite à trois (03) plates-formes : Microsoft Windows Azure, Google BigQuery, et Amazon S3.

4.2.1 Tableau comparatif de prix de stockage

Les côuts de stockages des données dans le Cloud diffèrent d'un acteur à un autre selon les critères d'offres, pour cela nous avons fait un tableau (TAB.4.1) comparatif qui regroupe les différents prix mensuels des principaux fournisseurs.

Fournisseurs	Microsoft	Google	Amazon
Gratuit	20 GO	5 GO	5 GO
Inférieur à 1 TO	0,14 $/GO	0,12 $/GO	0,125 $/GO
1-50 TO	0,125 $/GO	0,12 $/GO	0,110 $/GO
50-100 TO	0.112 $/GO	0,12 $/GO	0,095 $/GO
Supérieur à 100 TO	0.085 $/GO	0,12 $/GO	0,090 $/GO

TAB. 4.1 – Tableau comparatif des prix de stockage.

4.2.2 Tableau comparatif des prix de traitement

Les côuts de traitement des données dans le Cloud diffèrent d'un acteur à un autre selon les critères d'offres, pour cela nous avons fait un tableau (TAB.4.2) comparatif qui regroupe les différents prix mensuels des principaux fournisseurs.

Fournisseurs	Microsoft	Google	Amazon
Gratuit	100 GO	-	-
Inférieur à 1 TO	0,040 $/GO	0.035 $/GO	0.01 $/GO
1-50 TO	0,032 $/GO	0.035 $/GO	0.01 $/GO
50-100 TO	0.020 $/GO	0.035 $/GO	0.01 $/GO
Supérieur à 100 TO	0.018 $/GO	0.035 $/GO	0.01 $/GO

TAB. 4.2 – Tableau comparatif de prix de traitement.

4.2.3 Critères de comparaison

Dans cette section nous présentons les principales différences entre les principaux acteurs et qui expliquent la divergence ou la convergence des prix :

1. **Amazon S3 :** Amazon S3 est le service de stockage en ligne d'Amazon qui est le seul service web de stockage qui regroupe ces caractéristiques[39],[40],[41] :
   - Une offre gratuite de 5 GO d'espace de stockage et de capacité illimitée.
   - Un service " on-demand " c'est à dire de ne payer que ce que nous consommons.
   - Le service se base sur un principe de facturation à la consommation ou bien plus nous stockons, moins nous payons.
   - Une abstraction totale de la couche physique du stockage.
   - Amazon insiste sur une solution de stockage sécurisée à moindre coût tout comme de la transaction des données.
   - Service disponible sur : Windows, Mac Os, Linux.

2. **Google BigQuery :** Google BigQuery est un nouveau service Cloud d'analyse en temps réel des grands volumes de données lancé par Google, les majeures caractéristiques [42],[43] de ce service sont les suivantes :
   - Une offre gratuite de 5 GO d'espace de stockage.
   - Stockage illimité (jusqu'à plusieurs centaines de teraoctets en ne payant que ce que nous utilisons).
   - Un service fiable et sécurisé (les données sont automatiquement répliquées sur plusieurs sites).
   - C'est un service ultra-rapide (exécution des requêtes en quelque seconde).
   - Service disponible sur : Windows, Mac Os, Windows Phone, Chrome OS, iPhone, iPad et Android.
3. **Microsoft Windows azure :** Microsoft Windows azure est une plateforme Cloud proposée par Microsoft. Cette plateforme est caractérisée [44] par :
   - Une offre gratuite de 20 GO d'espace de stockage.
   - Plateforme ouverte et flexible qui permet de générer, déployer et gérer rapidement des données.
   - Un stockage sécurisé des données avec Windows Azure.
   - Microsoft commercialise ces services azure en directe dans un modèle " pay as you go" facturable au client chaque mois ou via des abonnements prépayés d'une durée plus longue, ou encore par le bias d'accords inter-entreprises pré négociés.
   - Service disponible sur : Windows, Mac Os, Chrome OS, Windows Phone, Linux, iPhone, iPad et Android.

4.3 Les rapports entre les coûts de stockage et les coûts de traitement

Afin d'avoir une idée sur le rapport entre le prix de stockage et le prix de traitement, nous supposons :

◇ P_{us} **:** représente le prix unitaire de stockage.

◇ P_{ut} **:** représente le prix unitaire de traitement.

Etude du 1er cas : comparaison entre le prix de stockage et le prix de traitement dans Microsoft :

D'après les deux tableaux (TAB.4.1), (TAB.4.2) nous avons conclu que le prix de traitement de 1GO de données pendant 1 mois est équivalent à 2/7 multiplié par le prix de stockage de la même quantité.

D'où la formule suivante :

$$\boxed{P_{ut} = \mathbf{2/7} * P_{us}}$$

Etude du 2ème cas : comparaison entre le prix de stockage et le prix de traitement dans Google :

D'après les deux tableaux (TAB.4.1), (TAB.4.2) nous avons conclu que le prix de traitement de 1GO de données pendant 1 mois est équivalent à (1/3) multiplié par le prix de stockage de la même quantité.

D'où la formule :

$$\boxed{P_{ut} = \mathbf{1/3} * P_{us}}$$

Etude du 3ème cas : comparaison entre le prix de stockage et le prix de traitement dans Amazon :

D'après les deux tableaux (TAB.4.1), (TAB.4.2) nous avons conclu que le prix de traitement de 1GO de données pendant 1 mois est équivalent à (1/12) multiplié par le prix de stockage de la même quantité.

D'où la formule :

$$\boxed{P_{ut} = \mathbf{1/12^*}\ P_{us}}$$

D'après les cas cités en (section.4.3) nous avons conclu que le prix de traitement de 1GO de données pendant 1 mois est équivalent à un facteur "a" multiplié par le prix de stockage de la même quantité.
D'où la formule suivante :

$$\boxed{Formule_1 : P_{ut} = \mathbf{a} * P_{us} \text{.....(*)}}$$

4.4 Les rapports entre les coûts de stockage et de traitement sans et avec utilisation des vues matérialisée

Afin d'obtenir des rapports entre le prix de stockage et le prix de traitement sans et avec utilisation les vm_s, nous supposons que :

- C_s **:** représente le coût de stockage ($).
- C_t **:** représente le coût de traitement ($).
- T_d **:** la taille des données (GO).
- T_{vm} **:** la taille des vues matérialisées (TO).
- $Periode_s$ **:** la période (temps) de stockage (mois).
- $Taille_{td}$ **:** la taille de traitement des données (TO).

◦ $Taille_{tvm}$ **:** la taille de traitement des VM_s (TO).

⋆ Les formules généralisées sans utilisation des vues matérialisées :

$C_s = T_d * P_{us} * Periode_s$.....(1)

$C_t = Taille_{td} * P_{ut}$.....(2)

⋆ Les formules généralisées avec utilisation des vues matérialisées :

$C_s = (T_d + T_{vm}) * P_{us} * Periode_s$.....(3)

$C_t = Taille_{tvm} * P_{ut}$.....(4)

4.4.1 Objectif des vues matérialisées dans le Cloud :

Soient :

$[C_s + C_t]$**SVM :** le coût total sans utilisation des vues matérialisées.

$[C_s + C_t]$**AVM :** le coût total avec utilisation des vues matérialisées. Nous cherchons à réduire les coûts en utilisant les vues matérialisées :

$$\boxed{Formule_2 : [C_s + C_t] \textbf{ SVM} > [C_s + C_t] \textbf{ AVM}.....\textbf{(**)}}$$

en remplaçant (1), (2), (3) et (4) dans (**) on obtient :

$[(T_d * P_{us} * Periode_s) + (Taille_{td} * P_{ut})] > [((T_d + T_{vm}) * P_{us} * Periode_s)$
$+ (Taille_{tvm} * P_{ut})]$
$\Longrightarrow [(T_d * P_{us} * Periode_s) + (Taille_{td} * P_{us} * a)] > [((T_d + T_{vm}) * P_{us} * Periode_s)$
$+ (Taille_{tvm} * P_{us} * a)]$

$\Longrightarrow [(T_d * Periode_s) + (Taille_{td} * a)] > [((T_d + T_{vm}) * Periode_s)$
$+ (Taille_{tvm} * a)]$

$\Longrightarrow [(T_d * Periode_s) + (Taille_{td} * a)] > [(T_d * Periode_s) + (T_{vm} * Periode_s) + (Taille_{tvm} * a)]$

$\Longrightarrow (Taille_{td} * a) > [(T_{vm} * Periode_s) + (Taille_{tvm} * a)]$

Soit : $\alpha = (Taille_{tvm} / Taille_{td})$

$\Longrightarrow T_{vm} < (Taille_{td} * a) - (Taille_{tvm} * a)/Periode_s$

$\Longrightarrow T_{vm} < (Taille_{td} - Taille_{tvm}) * a / Periode_s$

$\Longrightarrow T_{vm} < [Taille_{td} - (Taille_{td} * \alpha)] * a / Periode_s$

D'où la taille maximale des vues matérialisées à exploiter pour réduire tous les coûts est :

$Formule_3$: $T_{vm} < \mathbf{a} * Taille_{td} * (\mathbf{1} - \alpha) / Periode_s$.....(***)

Nous remarquons que la taille des vm_s dépend de l'apport "α" gagné en utilisant les vm_s et aussi du facteur "a" qui est le rapport entre le prix unitaire de stockage et le prix unitaire de traitement des trois acteurs (Amazon, Google et Microsoft) du Cloud ainsi de la période de stockage qui est inversement proportionnelle à cette taille.

Conclusion

Dans ce chapitre nous avons proposer d'exploiter les vues matérialisées en effectuant une comparaison des prix mensuels de stockages et de traitement et en limitant la taille des vues matérialisées afin de réduire tous les coûts à payer.

Dans le chapitre suivant nous allons valider notre proposition sous Oracle. <

Chapitre 5

Implémentation et évaluation

Introduction

Dans le monde de la base de données, il arrive fréquemment d'avoir à exécuter de très grosses requêtes SQL sur de très gros volumes de données combinant des jointures, des aggrégats etc. Ces requêtes sont très consommatrices de ressources et peuvent ralentir la base de données pendant un petit bout de temps.

Pour valider notre solution, nous allons utiliser "ORACLE" version 10g comme un SGBD qui dispose de toutes les fonctionnalités nécessaires à la validation.

5.1 Présentation d'ORACLE

Oracle est un SGBD réparti, il a pour rôle de gérer l'accès aux bases de données qu'il stocke et restitue à volonté. "ORACLE" se démarque des autres gestionnaires de bases de données par son côté administration très développé (Gestion des utilisateurs, des profils, des rôles et privilèges, des tablespaces) et aussi de part son architecture complexe qui repose sur la notion d'instance et qui assure un traitement rapide, sécurisé, et efficace des données.[45]

5.1.1 Traitement d'une requête par ORACLE

Le traitement d'une requête par "ORACLE" est découpé en trois étapes essentielles [46] à savoir :

1. **Parcours :** le processus utilisateur envoie la requête au processus serveur afin qu'il analyse sa syntaxe puis la compile et vérifie les privilèges de l'utilisateur pour les objets référencés à accéder. La solution de réussite ou non de l'analyse est renvoyée au processus utilisateur à la fin de cette phase.

2. **Exécution :** à cette étape, le processus serveur se prépare à la récupération des données.

3. **Récupération des données :** les données sont récupérées au processus utilisateur selon la qualité de mémoire disponible, une ou plusieurs récupérations peuvent être nécessaires.

5.1.2 La réplication sous ORACLE

La réplication est un processus qui consiste à copier et à mettre à jour des objets (tables, vues, indexes...) entre divers sites qui peuvent être éloignés géographiquement, la mise à jour des données peut s'enclencher de manière manuelle ou automatique.[47]

Sous ORACLE trois types de réplications [48],[49] sont possible :

1. **La commande COPY :** au niveau du serveur local, la commande "COPY" réplique régulièrement les données. Elle remplace le contenu de la table au lieu de la mettre à jour, ce qui est considéré comme un inconvénient.

2. **La réplication de clichés (Snapshots) :** un snapshot est utilisé afin de répliquer les données à partir d'un site maître vers différentes cibles.

3. **Les vues matérialisées :** une vue matérialisée est un moyen de créer une vue physique d'une table.

5.1.3 Gestion de la sécurité sous ORACLE

La plupart des entreprises sont, aujourd'hui plus qu'hier, sensibilisées au problème de la sécurité des données, pour cela ORACLE s'est accentué sur la réduction des risques internes et externes à travers l'option "ORACLE Label Security", qui permet de gérer les différents accès à la base.

Toute l'infrastructure de sécurité des données repose sur une classification des données et un contrôle d'accès basé sur des étiquettes, au niveau des cellules des tables (et non pas seulement au niveau des lignes).[50]

5.1.4 Oracle et le Cloud Computing

La stratégie Oracle consiste à proposer un vaste portefeuille de logiciels, produits matériels et services permettant de prendre en charge les Cloud publics, privés et hybrides, les clients pouvant ainsi choisir l'approche adaptée à leurs besoins. Contrairement aux concurrents ayant une approche limitée du Cloud, Oracle fournit les offres de Cloud les plus vastes, complètes et intégrées du secteur.

Oracle dans un Cloud privé constitue un point d'entrée simple et rapide dans cette technologie, nous bénéficions très rapidement d'une agilité accrue et d'une réduction des coûts informatiques.

Dans le contexte d'une stratégie de Cloud Computing à long terme, Oracle propose des offres exhaustives s'appliquant à l'ensemble de la pile technologique : applications, solutions middleware, bases de données, systèmes d'exploitation, virtualisation, serveurs, systèmes de stockage et gestion.[51]

5.1.5 Oracle et les vues matérialisées

Les vues matérialisées permettent de créer physiquement des sous-ensembles de données, généralement constitués de données difficiles à calculer et à générer, comme des jointures complexes, des sous-requêtes corrélées, ou des group by.

Les données ne sont pas extraites au moment de l'interrogation mais préparées à l'avance. Même si elles requièrent de l'espace de stockage en conséquence, elles sont très efficaces car Oracle réécrit automatiquement la requête de l'utilisateur, afin qu'elle utilise la vue matérialisée.[52]

5.2 Description générale

Afin d'exploiter les vues matérialisées pour avoir un temps de réponse des requêtes court ainsi un coût de traitement réduit nous avons utilisé la base de données "Pubs" qui contient dix(10) tables (FIG.5.1) et trois (03) vues matérialisées (mv_auteur, mv_editeur, mv_magasin).

Pour illustrer l'impact d'utilisation des VM_s dans une BD nous avons exécuté un ensemble de requêtes (TAB.5.1) sur les tables de la BD "Pubs" ensuite exécuté les mêmes requêtes sur les VM_s crées.

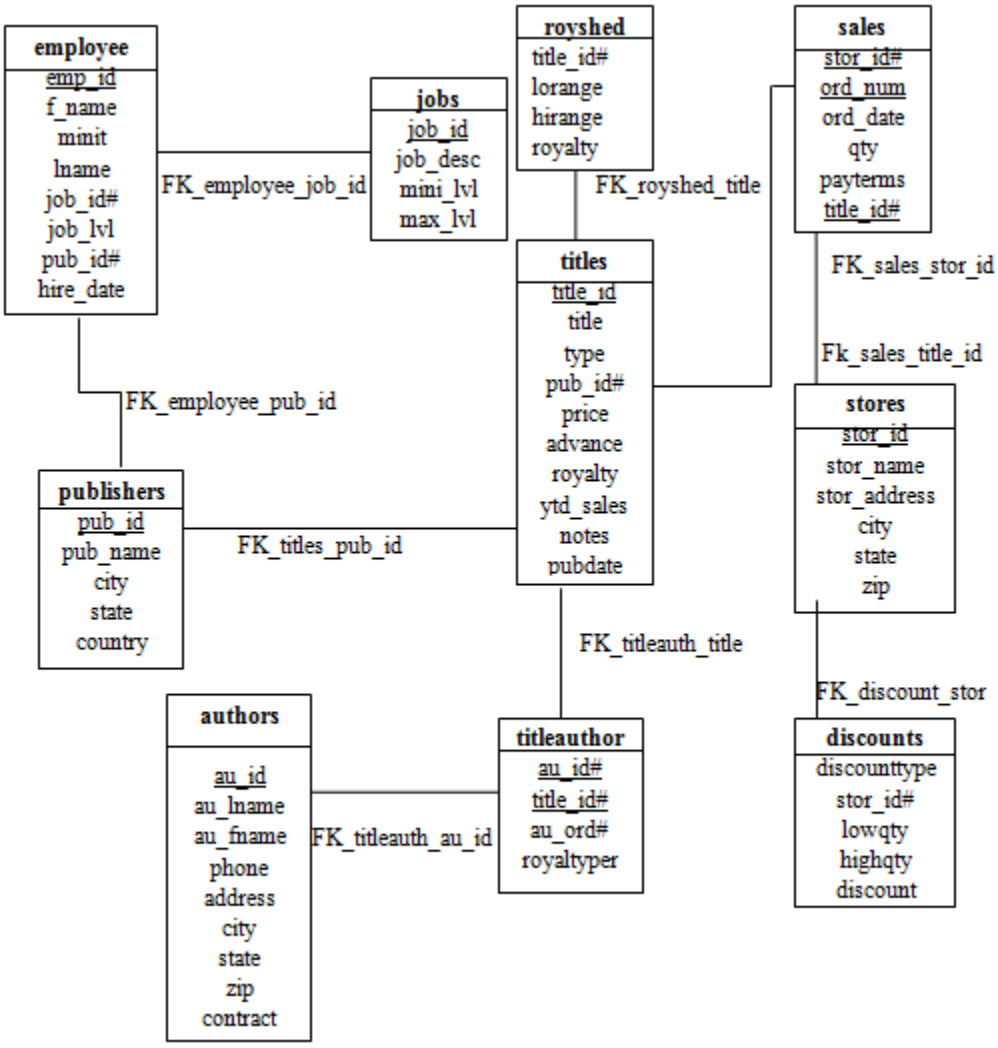

FIG. 5.1 – Le schéma de la base de données Pubs.

D'aprés ce schéma nous allons faire une description pour la table authors.

Description de la table "authors" :

La table "authors" possède neuf (09) attributs :

au_id comme clé primaire de type varchar2(11) ;
au_lname varchar2(40) ;
au_fname varchar2(20) ;
phone char(12) ;
address varchar2(40) ;
city varchar2(20) ;
state char(2) ;
zip char(5) ;
contract number.

Description des vues matérialisées implémentées :

1. **vm_auteur :** cette vue permet d'afficher les auteurs et les titres qu'ils ont publiés.

   CREATE Materialized View rabia.vm_auteur AS

   SELECT au_lname, au_fname, title, phone, address, city, state, type, price, pubdate, notes, au_ord, royaltyper

   FROM authors, titleauthor, titles

   WHERE (authors.au_id=titleauthor.au_id)AND(titles.title_id=titleauthor.title_id) ;

2. **vm_editeur :** cette vue permet d'afficher la liste des employés et leurs poste de travail.

   CREATE Materialized View rabia.vm_editeur AS

   SELECT fname, lname, job_desc, pub_id, hire_date, job_lvl

   FROM employee, jobs

   WHERE(employee.job_id=jobs.job_id) ;

3. **vm_magasin :** cette vue permet de donner la liste des magasins qui ont vendu le plus grand nombre d'exemplaire en une seule vente.

   CREATE Materialized View rabia.vm_magasin AS

   SELECT sales.stor_id, stores.stor_name, titles.title, ord_num, ord_date, qty, stor_address, city, state, type, price, pubdate, royalty

   FROM stores , sales , titles

   WHERE(stores.stor_id=sales.stor_id)AND(sales.title_id=titles.title_id) ;

Description des requêtes exécutées :

1. **1ère requête :** cette requête permet de lister les auteurs qui reçoivent des droits de 100 avec les titres qu'ils ont publiés.

   - **Exécution sur les tables de la BDD :**

     SELECT au_lname, au_fname, title, phone, address, city, state, type, price, pubdate, notes, au_ord, royaltyper

     FROM authors, titleauthor, titles

     WHERE(authors.au_id=titleauthor.au_id)AND(titles.title_id=titleauthor.title_id) AND (titleauthor.royaltyper=100) ;

   - **Exécution sur les vues matérialisées :**

     SELECT au_lname, au_fname, title, price, pubdate, royaltyper

     FROM vm_auteur

     WHERE(royaltyper=100) ;

2. **2ème requête :** cette requête permet d'afficher la liste des employés et leurs poste de travail dont la classification professionnelle est de 200 ou 35.

   - **Exécution sur les tables de la BDD :**

     SELECT fname, lname, job_desc, pub_id, hire_date, job_lvl
     FROM employee, jobs
     WHERE (employee.job_id=jobs.job_id)AND(job_lvl=200)OR(job_lvl=35) ;

   - **Exécution sur les vues matérialisées :**

     SELECT fname, pub_id, job_lvl
     FROM vm_editeur
     WHERE (job_lvl=200)OR(job_lvl=35) ;

3. **3ème requête :** cette requête donne les magasins qui ont vendu le plus grand nombre d'exemplaire en une seule vente et qui doit avoir une quantité supérieure à 20.

   - **Exécution sur les tables de la BDD :**

     SELECT sales.stor_id, stores.stor_name, titles.title, ord_num, ord_date, qty, stor_address, city, state, type, price, pubdate, royalty
     FROM stores, sales, titles
     WHERE (stores.stor_id=sales.stor_id)AND(sales.title_id=titles.title_id)
     AND(sales.qty>20) ;

- **Exécution sur les vues matérialisées :**

SELECT stor_id, title, ord_num, ord_date, qty
FROM vm_magasin
WHERE (qty>20) ;

5.3 expérimentation

Afin de voir l'impact de la technique d'optimisation que nous avons utilisé c'est-à-dire les vues matérialisées, nous l'avons éxpérimentée sur une base de données "Pubs" sous oracle 10g.

Nos expérimentations ont été réalisées sur un Pc sous Windows 7 Pro doté d'un processeur Pentium 2.20 GHz, d'une mémoire centrale de 3.00 Go et d'un disque dur de 300 Go.

Le Tableau (TAB.5.1) détaille le comparatif des temps d'exécutions des requêtes décrites précédemment avec et sans utilisation des vues matérialisées.

Reqêtes	R_1	R_2	R_3
$Temps_{svm}$	0,65ms	0,32ms	0,21ms
$Temps_{avm}$	0,04ms	0,03ms	0,04ms
L'apport α	16,25	10,66	5,25

TAB. 5.1 – Tableau comparatif des temps d'exécutions des requêtes avec et sans les vm_s.

Remarque

◇ $Temps_{svm}$ **:** c'est le temps de réponse d'une requête sans utilisation des vm_s.

◇ $Temps_{avm}$ **:** c'est le temps de réponse d'une requête avec utilisation des vm_s.

5.3.1 Exemple d'implémentation

Afin de voir l'impact de notre proposition (VM_s) nous avons fait un comparatif entre le prix total sans utilisation des vm_s et le prix total avec utilisation des vm_s pour calculer les coûts de stockage et de traitement pour les trois fournisseurs (Amazon, Google et Microsoft) du Cloud.

Supposant que nous avons 25 GO de données à stocker et 20TO de données à traiter pendant un mois.

Les résultats sont illustrées dans le tableau ci-dessous :

Le coût total avec Microsoft			
Coût mensuel	Capacité de stockage	Données traitées	Coût total
SVM	3.5 $(0,14 $/GO)	655,36 $(0,032 $/GO)	658,86 $
AVM	576,94 $	65,536 $	642,47$
Le coût total avec Google			
Coût mensuel	Capacité de stockage	Données traitées	Coût total
SVM	3 $(0,12 $/GO)	716,8 $(0.035 $/GO)	719,8 $
AVM	617,4 $	71,68 $	689,08 $
Le coût total avec Amazon			
Coût mensuel	Capacité de stockage	Données traitées	Coût total
SVM	3,125 $(0,125 $/GO)	204,8 $(0.01 $/GO)	207,92 $
AVM	131,125 $	20,48$	151,60 $

TAB. 5.2 – Tableau comparatif des prix de stockage et de traitement sans et avec les vm_s.

Nos expériences montrent que le temps de réponse est réduit ainsi que les coûts à payer pour les utilisateurs et aussi les processeurs deviennent plus libres.

Conclusion

Nous avons implémenté dans ce chapitre les vues matérialisées sous Oracle où notre objectif de base est d'améliorer le temps de réponse des requêtes et de réduire le coût de traitement dans le Cloud Computing.

En effet, nos résultats expérimentaux en utilisant notre solution qui consiste à limiter la tailles des vues matérialisées, montrent la réduction de coût à payer pour les utilisateurs et l'utilisation efficace des ressources des fournisseurs et ceci pour les trois acteurs du Cloud que nous avons étudié.

Conclusion Générale et Perspectives

Aprés notre travail que nous avons mené autour du Cloud Computing nous pouvons sans hésitation affirmer que ce dernier est une révolution dans le domaine informatique attirant derrière lui, de nouvelles technologies ainsi que de nouvelles façons de penser et de concevoir les systèmes d'informations d'aujord'hui.

L'accroissement continu de la volumétrie des données et la compléxité des requêtes dans le Cloud rend difficile et inapproprié leur traitement au moyen des systèmes et des outils de gestion de bases de données traditionnels, et afin de remédier au problème de Big Data il est de plus en plus important de pouvoir se reposer sur d'autres solutions et outils entre autre les mécanismes d'optimisation de gestion de données qui permettent le progrès de performances des requêtes en offrant un accés rapide aux données.

Dans notre travail nous avons proposé les vues matérialisées qui permettent d'optimiser le temps de réponse ainsi de génèrer un coût de traitement réduit en donnant une étude comparative des tarifications de stockage et de traitement des données dans le Cloud entre les principaux acteurs (Microsoft, Google et Amazon).

Nous avons proposé de limiter la taille des VM_s et de l'implémentée sous le SGBD Oracle version 10g en effectuant des exécutions des requêtes sur les tables de base de données et sur les vues matérialisées, ensuite nous avons calculer l'impact de notre proposition.

L'evaluation de notre technique d'optimisation de données a montré son impact positif sur le temps d'exécution des requêtes ainsi qu'une réduction de coût de traitement.

Dans le but de compléter cette étude, il serait vraiment intéressant de se reposer sur les perspectives suivantes :

* Exploitation d'autre méthodes pour la limitation de la taille des VM_s telle que la modélisation par la méthode de simplexe.
* Implémentation d'autre techniques d'optimisation comme les indexes et la fragmentation.
* Définition d'algorithme de sélection des VM_s basé sur le modèle de facturation à l'utilisation
* Intégration des caches pour l'amélioration de la qualité des services (réduction des temps de réponses, augmentation de la disponibilité)et la réduction des coûts d'utilisation.

Bibliographie

[1] V. Kherbache, M. Moussalih, Y. Kuhn, A. Lefort, cloud computing, mémoire de licence professionnelle, IUT Nancy charlemagne, 2010.

[2] M. Audin, Etat de l'art du Cloud Computing et adaptation au Logiciel Libre,livre blanc, 2009.

[3] M. Sarazin, Saas l'usage de demain, mémoire de master2 miage,université d'évry val d'essonne, 2009.

[4] Etat de l'art de cloud computing, livre blanc, sogeti entreprise services consulting, 2009.

[5] P. Warrior, "le Cloud computing" il tire sa puissance du réseau, livre blanc, 2010

[6] Le cloud computing une nouvelle filière fortement structurante, publication, direccte ile de france, 2012.

[7] T. Chardonnens, Les enjeux du Cloud Computing en entreprise, Université de Fribourg(Suisse), thèse, 2012.

[8] G. Louise-Marie, Etude des framework distribués de grilles de données en vue d'une solution de stockage élastique dans le Cloud, 2012.

[9] M. Jambou, Introduction au Data Mining, Analyse intelligente des données, Paris Eyrolles, 1999.

[10] J-F. Desnos, Entrepôt de données, 2009.

[11] Site officiel de Microsoft, http ://www.microsoft.com/, partie gestion, 2013.

[12] René J. Chevance, Transactionnel et transactionnel réparti, livre, 2000.

[13] Site officiel du cloud magazine, http ://www.cloudmagazine.fr/, 2012.

[14] Gill, Rao, Entrepöt de données spatiales OLAP et SOLAP, 1996.

[15] Cécile Favre et all, Las entrepöts de données pour les nuls...ou pas, Université de lyon, 2011.

[16] Site officiel de bleent, http ://www.bleent.com/, 2011.

[17] G.Gardarin, XML Des bases de données aux services Web dunod, livre, Paris, 2002

[18] P. Arlaud, J. Dupire , conférence, 2010

[19] M. Dorier, BlobSeer un système de fichiers pour le calcul hautes performances sous Hadoop MapReduce, Rapport, 2009.

[20] M. Jakrse, P. Rauzy, MapReduce, conférence, 2010.

[21] M. Lorrillere, J. Sopena, S. Monnet et P. Sens, Vers un cache réparti adapté au cloud computing, Université Pierre et Marie Curie, 2013

[22] http ://architectures-web.smile.fr/Le-cache/Cache-de-donnees

[23] Site officiel de gandi, http ://www.gandi.net/, documentation en ligne, 2011.

[24] N. Maiz, K. Aouiche et J. Darmont, Sélection simultanée d'index et de vues matérialisées,Laboratoire ERIC, Université Lumière Lyon2, 2006.

[25] N. DURAND, introduction à des notions avancées(Index, Déclencheurs, Transactions), Marseille,conférence, 2013.

[26] J. Ravaille, Les vues et les indexes, livre, 2009.

[27] L. Bellatreche, techniques d'optimisation des requêtes dans les data warehouses,ENSMA Université de Poitiers France , 2003.

[28] Hakim MADI , Conception et realisation d'une base de donnees repartie sous oracle : cas de l'hebergement des residences universitaires,Université A/Mira de Bejaia - Master II 2009.

[29] Faiza GHOZZI JEDIDI, conception et manipulation de bases de données dimensionnelles à contraintes, thèse, université Toulouse III, 2004.

[30] Kamel AOUICHE, Techniques de fouille de données pour l'optimisation automatique des performances des entrepôts de données, thèse, Université Lumière Lyon 2, décembre 2005.

[31] Boly Aliou, fonctions d'oubli et résumés dans les entrepôts de données, thèse, l'école nationale supérieure des télécommunications de paris, décembre 2006

[32] Bellatrecheladjel, techniques d'optimisation des requêtes dans les data warehouses,ENSMA Université de Poitiers France , 2003.

[33] http ://blogs.msdn.com/b/windowsazurefrance/archive/microsoft-baisse-les-prix-du-stockage-azure-pour-les-blob-et-les-tables.aspx, 2012.

[34] http ://www.distributique.com/actualites/lire-bigquery-le-service-analytique-en-mode-cloud-de-google-18261.html, 2012.I

[35] http ://venturebeat.com/amazon-s3-storage-lowers-prices-hell-yeah/, 2012.

[36] Site officiel de windows Azure, http ://www.windowsazure.com, partie prix, 2012.

[37] http ://newtech.about.com/od/softwaredevelopment/ss/Using-The-Amazon-Simple-Storage-Service-S3_3.htm, 2012.

[38] Site officiel de woueb, http ://www.woueb.net,partie caractéristiques,2012.

[39] http ://www.businesswire.com/, 2012.

[40] http ://www.journaldunet.com, 2012.

[41] Site officiel de google, http ://www.google.fr, 2012.

[42] http ://www.yeswecloud.fr/cloud/google-se-lance-dans-la-cinquieme-dimension-avec-les-projet-google-glass-1148.html, 2012.

[43] Site officiel de windows azure, http ://www.windowsazure.com/, partie définitions, 2012.

[44] Djamou yikam dave odilon, mémoire de fin d'études bases de données reparties sous oracle cas de société call in out, école supérieur de management de commerce et d'informatique Maroc,2007/2008

[45] A.Marie. Gérer une instance Oracle,livre, 2003.

[46] P. Wojciechowski, la réplication sous Oracle,livre, paris 2005.

[47] R. chapuis, les bases de données oracles 8i, administration, optimisation, livre, 2001.

[48] P. Boissel, Oracle aujourdhui, le point de vue de l'expert, Learning Tree International, 2006.

[49] R. Moussa, Systèmes de Gestion de Bases de Données Réparties et Mécanismes de Répartition avec Oracle, livre, 2005.

[50] Site officiel d'Oracle,http ://www.oracle.com/, 2013.

[51] P. Marcenac, Oracle 11g :Que peut-on en attendre ?, Livre blanc,2008

[52] http ://www.partagedefichier.com/blog/cloud-computing-definition.

[53] http ://www.scriptol.fr/technologies/cloud.php.

Résumé

Le volume de données (Big Data) contenu dans les Cloud Computing est destiné à s'accroître sans cesse, augmentant ainsi la complexité des requêtes décisionnelles. Pour y remédier, l'administrateur doit exploiter les mécanismes d'optimisation (indexes, vues matérialisées ou caches), afin d'améliorer les performances d'exécution des requêtes ce qui génère une réduction des coûts de traitement.
Dans notre travail, nous définissons les concepts de base du Cloud Computing, la gestion des données dans les Clouds et les techniques d'optimisation des données. Nous proposons une solution afin de réduire le temps de réponse aux requêtes ainsi que le coût de traitement dans le Cloud Computing qui est l'utilisation des vues matérialisées qui permettent de pré-calculer des requêtes coûteuses et de stocker leurs résultats pour cela, nous avons fait une comparaison de tarification dans le Cloud Computing.
Nous terminons par une étude expérimentale et des tests comparatifs sous le SGBD oracle 10g sur une base de données réelle qui montrent l'intérêt de notre proposition.

Mots clés : Cloud Computing, Big Data, complexité des requêtes, mécanisme d'optimisation, vue matérialisée, SGBD oracle 10g

ABSTRACT

The amount of data (Big Data) contained in the Cloud Computing is intended to increase constantly, thus increasing the complex of decision-support requesting. To remedy this, the administrator must use the optimization mechanisms (indexes, materialized views, or caches), to improve performance query execution which generates a reduction the costs of treatment. In our work, we define the basic concepts of Cloud Computing, data management In the Clouds and the optimization techniques of data. We propose a solution to reduce the response time to queries as well as the cost of treatment in the Cloud Computing is the use of materialized views that allows pre-compute expensive queries and store their results for this, we made a comparison of pricing in cloud computing. We conclude with an experimental study and comparative tests in the Oracle 10g DBMS on a real database that shows the interest of our proposal.

Keywords : Cloud Computing, Big Data, complex queries, optimization mechanism, materialized view, Oracle 10g DBMS