



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE IBN KHALDOUN - TIARET

MEMOIRE

Présenté à :

FACULTÉ MATHÉMATIQUES ET INFORMATIQUE
DÉPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :
MASTER

Spécialité : Réseaux et télécommunication

Par :

MEGHAZI ABDELKADER

KHIAR MUSTAPHA

Sur le thème

Traitement des requêtes OLAP lourde sur cloud

Soutenu publiquement le --/--/2022 à Tiaret devant le jury composé de :

Mr/DAOUD MEHAMED AMINE	MAA Université Tiaret	Président
Mr/CHAALAL HICHAM	MCB Université Tissemsilt	Encadreur
Mme/HAMDANNI ABDIA	MAA Université Tiaret	Examinatrice
Mr/MOSTFAOUI SID AHMED	MAA Université Tiaret	Co Encadreur

2021-2022

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

❧ *Remerciements* ❧

En premier lieu, nous remercions Dieu le tout puissant qui nous a donné le courage et la volonté de réaliser ce modeste travail.

Nous remercions nos parents qui nous ont suivis pendant nos études.

Nous remercions Mr.Challal Hicham, notre encadreur, pour son aide, son encouragement, son orientation, et pour ses précieux conseils durant la réalisation de ce travail.

Nous adressons aussi nos remerciements à tous les professeurs qui nous ont enseignés durant ce cursus universitaire.

Enfin nous tenons à saisir cette occasion et adresser nos profonds remerciements et nos profondes reconnaissances à toutes les personnes qui nous ont aidées de près ou de loin dans la réalisation de ce mémoire.

Dédicaces

Nous dédions ce projet en ce qui concerne MEGHAZI ABDELKADER :

A mon cher père ABDELKADER

Qui m'a toujours poussé et motivé dans mes études. Merci énormément papa pour ton soutien plus que précieux, merci pour ton grand cœur toutes vos qualités qui seraient trop longues à énumérer. Ma vie ne serait pas aussi magique sans ta présence et ton amour, mon papounet d'amour

A ma très chère mère,

La lumière de mes jours. Tu représente pour moi le symbole de la bonté par excellence, la source de tendresse et l'exemple du dévouement. Tu as toujours été présente à mes côtés pour me soutenir et m'encourager. Je te dédie ce travail en témoignage de mon profond amour.

Enfin, à tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Abdelkader

Dédicaces

Nous dédions ce projet en ce qui concerne KHIAR MUSTAPHA:

Je dédie ce travail aux honorables parents et mes honorable professeurs de la faculté d'informatique de université

Iben khaldoun , ainsi que comité de discussion et au superviseur , le professeur M. CHAALAL HICHAM .

Je demande à ALLAH de nous guider vers ce qu'il aime et dont il est satisfait.

Mustapha

Résumé :

Les bases de données permettent de stocker les données sur des disques dur, et sur des serveurs, ou sur le cloud, la gestion d'une base de données (SGBD) se révèle très difficile, et le nombre de requêtes de type OLAP généralement coûte utilise beaucoup de ressources, et nécessite un matériel adéquat et adapté aux besoins des utilisateurs. Souvent les requête OLAP sature le serveur, notre solution est d'exécuter les requêtes de chaque utilisateur en local, ce qui réduit énormément le coût de traitement des requêtes, et permet d'exploiter les ressources locales de chaque utilisateur et fait intervenir les appareils comme smartphone, laptop, Tablet, des objet connecté ainsi d'exploiter les ressources de ses dernières.

Mots clés: stocker, cloud , requêtes OLAP , serveur SGBD

ملخص :

تسمح قواعد البيانات بتخزين البيانات على الأقراص الصلبة ، أو على الخوادم ، أو في السحابة ، وإدارة قاعدة البيانات (SGBD) أمر صعب للغاية ، وعدد استعلامات نوع OLAP بشكل عام يستخدم الكثير من الموارد ، ويتطلب أجهزة مناسبة تتكيف مع احتياجات المستخدمين. غالبًا ما تشبع استعلامات OLAP الخادم ، يتمثل حلنا في تنفيذ استعلامات كل مستخدم محليًا ، مما يقلل بشكل كبير من تكلفة معالجة الاستعلامات ، ويسمح باستغلال الموارد المحلية لكل مستخدم ويتضمن أجهزة مثل الهاتف الذكي والكمبيوتر المحمول والكمبيوتر اللوحي والمتصل الأشياء واستغلال موارد أحدثها .

LISTE DES ABREVIATIONS :

AGL : Atelier de Génie Logiciel

BD : Big Data

SGBD : Un Système de Gestion de Base de Données

SAAS : Software as a Service

PAAS : Platform as a Service

IAAS : Infrastructure as a Service

OLTP : Online Transaction Processing

OLAP : Online Analytical Processing

ODBC : Open Database Connectivity

MOLAP : Multidimensional Online Analytical Processing

ROLAP : Relational On-Line Analytical Processing

REDIS : Serveur de dictionnaire à distance

OLTP : Traitement Transactionnel En Ligne

SGBDR : Systeme De Base De Donnée Relationnelle

JDBC : Java Data Base Connectivity

VM : Materialized View

SSD : Solide State Drive

IMDB : Base De Donnée En Mémoire

TABLE DES MATIERES

Problématique et motivations	2
------------------------------------	---

CHAPITRE I

1. Introduction :.....	9
2. Bases de données : Notions générales.....	9
2.1. Définition :.....	9
2.2. Objectifs d'une base de données :.....	9
2.2.1. Partage de l'information :.....	9
2.2.2. Organisation des données indépendamment des programmes : 9	
2.3. Rôle d'une Base de données :	10
2.4. La démarche de conception d'une base de données :.....	10
1.4.1. Principe général :	10
2.5. Types d'objets de base de données :.....	12
1.5.1 Pourquoi les bases de données sont-elles importantes ?.....	12
1.5.2 Types de bases de données :	13
2.6. Système de Gestion de Base de Données :	14

CHAPITRE II

Introduction	17
1 .Définition	17
2. Services du Cloud Computing.....	17
2.1. Infrastructure en tant que Service.....	17
2.2. Plate-forme en tant que service.....	17
2.3 .Software en tant que Service	17
3 .Types de cloud computing	18
4. Caractéristiques des données dans les nuages.....	19
4.1 Systèmes de gestion de données en nuage	20
4.2. Système de gestion des données de transaction	20
4.3 . Système de gestion des données analytiques	20
5. OLAP sur cloud :.....	21
5.1. Le rôle d'OLAP :.....	21
5.2. Restitution : Analyse OLAP multidimensionnelle.....	21
5.3. OLAP ou Online Analytical Processing : à quoi ça sert ?.....	22
5.4 Les différents types de systèmes :	23
5.5. Définition du concept de cube OLAP :.....	23
5.7. Objectif :.....	24
5.8.Opérations OLAP :	24

5.9. Stockage d'un cube de données :	24
6. La nouvelle pile Cloud Analytiques :	26
7. LEÇON D'HISTOIRE RAPIDE : TRAITEMENT ANALYTIQUE EN LIGNE OU "OLAP" :	26
7.1. CLOUD OLAP RÉUSSIT LÀ OÙ L'OLAP TRADITIONNEL ÉCHOUE :	27
7.2. Le cloud OLAP (COLAP) est l'avenir. Le futur c'est maintenant : ...	28

CHAPITRE III

1. Hibernate / JPA	28
8.1. Présentation générale	28
8.2. Différents modules :	29
8.3 . Architecture du noyau Hibernate	30
Figure 2.5 : Architecture du noyau Hibernate	31
Introduction	33
1 . Mécanisme d'optimisation et de gestion de donnée :	33
1.1 Cache :	33
1.2. Les snapshots :	33
1.3. Index :	33
1.4. Vues : [25]	34
1.5. Vue physique	34
1.5.1 Types de vues matérialisées	34
1.5.2. Problèmes avec les pensées matérialisées	35
6. Sélection des vues de matérialisées	35
7.Maintenir (mettre à jour) les vues matérialisées	36
7.1. Mises à jour automatiques :	38
7.2. Mise à jour complète :	38
7.3. Mise à jour incrémentale (rapide) :	38
Conclusion	38

CHAPITRE IV

1. Comment fonctionne une base de données en mémoire ?	3
2. Quels sont les avantages et les inconvénients des bases de données en mémoire ?	3
3. Pourquoi utiliser une base de données en mémoire dans ce projet?	4
4. Quelle est la différence entre les bases de données traditionnelles et en mémoire ?	4
5. Exemples de bases de données en mémoire	5
6 .Utilisation en mémoire dans Redis	5

7. Redis Dans l'application (IMDB)	6
7.1. Principes de base de Redis	6
7.2. Prise en charge des langages de Redis	7
7.3 Cas d'utilisation de Redis les plus répandus : [32]	8
8. Objectif d'utilisation Redis (IMDB)	10
9. Application	10
9.1. Type D'application [35]	10
9.2. outils [36]	10
9.3. Implémentation :	11
9.4. Panneau D'affichage	12
Conclusion	14
Conclusion générale :	16
BIBLIOGRAPHIE	17

Liste des figures :

Figure 1.1 : Processus de Modélisation d'une B.D	12
Figure 2.1: SGBD système de gestion de bases de données.....	14
Figure 2.2: Les services du Cloud.	18
Figure 2.3: Les types du Cloud.....	19
Figure 2.3: Exemple d'analyse des ventes d'une chaîne de magasins informatiques (vision en cube).	22
Figure 2.4 : Architecture du noyau Hibernate	31
Figure 3.1 : réplicas avec stockage en mémoire et traditionnel.....	3
Figure 4.1: Cachette avec Redis	7
Figure 4.2: Diagramme de duree quotidienne vs age.....	12
Figure 4.3: Utilisateurs par Pays Affichage Avec Redis.....	12
Figure 4.4: Cookies De Facebook Affichage Avec redis	13
Figure 4.5: Rechercher et afficher Sur Redis	13

INTRODUCTION GENERALE

Introduction générale

Une base de données en mémoire est un logiciel de stockage de données qui les conserve toutes dans la mémoire de l'hôte. La principale différence entre une base de données traditionnelle et une base de données en mémoire repose sur l'endroit où les données sont stockées. Même comparée aux disques SSD, la mémoire vive (RAM) est beaucoup plus rapide que l'accès au disque. Puisqu'une base de données en mémoire utilise ce dernier pour le stockage, l'accès aux données est beaucoup plus rapide qu'avec une base de données traditionnelle utilisant des opérations de disque.

Notre décision d'utiliser une base de données en mémoire dépendra de notre cas d'utilisation. Les bases de données en mémoire sont idéales pour l'accès aux données à volume élevé où une perte de celles-ci serait acceptable. Pensez à un grand site Web de commerce électronique. Les informations sur les produits sont cruciales et doivent être conservées sur un stockage persistant, mais les informations du panier pourraient potentiellement être conservées dans une base de données en mémoire pour un accès plus rapide.

Problématique et motivations

Qu'est-ce qu'une base de données en mémoire ?

Les principales caractéristiques des données dans le Cloud sont leur grande taille (données volumineuses) et la complexité des requêtes d'opérationnelles dues aux opérations de sélection,

de jointure (très coûteuse) et d'agrégation ont rendu le temps de réponse 'élève 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.

Solution

Notre solution proposée est de créer des vues matérialisées pour stoker les résultats des requêtes OLAP en locale,

CHAPITRE I:

Généralités Sur Les Bases De Données

1. Introduction :

Dans ce chapitre nous présenterons les généralités des bases données, cette présentation nous permettra d'avoir une idée sur la définition de base des données, son objectif, son rôle, sa démarche de conception, ses types, et les systèmes de gestion de bases de données, pour l'exploiter dans les chapitres qui suivent.

2. Bases de données : Notions générales

2.1. Définition :

Une base de données est un ensemble d'informations structurées permettant la mise en place d'une série d'applications informatiques destinées à une grande variété d'utilisateurs.

Exemple :

Dans une entreprise, les informations concernant son fonctionnement :

- Employés
- Produits Fabriqués
- Moyens matériels (Machines, Véhicules, Magasins, etc.)

Peuvent être rassemblées et mises à la disposition de nombreux utilisateurs (cadres de l'entreprise, gestionnaires, opérateurs, etc.). [01]

2.2. Objectifs d'une base de données :

Parmi les principaux objectifs visés par une base de données, on peut citer :

2.2.1. Partage de l'information :

Une base de données permet le partage d'un ensemble unique d'informations par plusieurs utilisateurs. Cependant, il faut que cette mise en commun soit faite tout en préservant la vue particulière que chaque utilisateur peut avoir des informations, et en s'assurant que la simultanéité des traitements qui peuvent être effectués ne risque pas de dégrader l'intégrité de la base de données.

2.2.2. Organisation des données indépendamment des programmes :

Afin de construire un ensemble d'informations structurées non redondant et qui soit partageable par plusieurs utilisateurs, il est nécessaire de faire abstraction des traitements particuliers de tel ou tel utilisateur (ou programme) pour tenter

d'organiser les informations en fonction de leur nature et des liens réels qui existent entre elles. C'est de cette manière qu'on arrivera à garantir le maximum d'indépendance entre données et programmes.

2.3. Rôle d'une Base de données :

Contrairement aux approches classiques, la création d'une base de données qui soit partagée par plusieurs utilisateurs est le reflet d'une évolution dans la gestion de l'entreprise. Son rôle est de rendre possible :

La centralisation de l'information : l'information n'est plus éparpillée dans différents

Fichiers à différents endroits)

L'intégration (tout ce qui se fait dans un service est visible par d'autres services)

La diffusion de l'information archivée (si l'information est disponible à un seul endroit, elle est facile à diffuser)

Ceci a pour avantages :

D'améliorer la cohérence de l'information (une seule valeur pour une même information)

De réduire les redondances (une même information n'est stockée si possible qu'une seule fois)

De réduire les efforts de saisie et de mise à jour des informations (i.e. une information qui doit être stockée une seule fois ne sera saisie qu'une seule fois. De même que sa mise à jour ne se fera qu'une seule fois)

2.4. La démarche de conception d'une base de données :

1.4.1. Principe général :

Il est communément reconnu que la conception d'une base de données doit se faire en utilisant une méthode de conception qui définit la démarche à suivre.

Plusieurs méthodes de conception existent à cet effet et nous citerons comme exemple la méthode MERISE. Pour certaines méthodes, on dispose même d'un outil logiciel d'aide à la conception appelé aussi un Atelier de Génie Logiciel (AGL) constitué d'un ensemble de logiciels permettant l'automatisation d'un certain nombre de tâches lors des différentes phases du processus de conception (génération automatique de la structure de la B.D., de programmes d'accès et de manipulation, etc.).

Néanmoins, quelle que soit la méthode utilisée, la conception d'une base de données

passer par un processus de modélisation permettant de modéliser une certaine partie du monde réel afin de caractériser les entités qu'on manipule (étudiants, Comptes Bancaires, Ouvrages, etc.). De plus on essaye de caractériser les attributs de ces entités en fonction des problèmes que doit résoudre l'existence de la B.D. : Gestion de la scolarité, Gestion de Prêt d'ouvrages dans une bibliothèque, etc. [02]

Le cas le plus général est celui où la B.D. est partagée par plusieurs utilisateurs. Ces utilisateurs n'ont pas tous la même vue des données de la base, et n'ont pas tous à voir la base dans sa totalité car chaque utilisateur n'est concerné que par une partie de celle-ci.

Exemple :

Dans une entreprise l'ensemble des informations sur les départements, les employés, les produits, le matériel, etc. peuvent être rassemblées sous forme d'une B.D. et il est bien rare qu'un utilisateur de cette base ait besoin de toutes ces informations à la fois. [02]

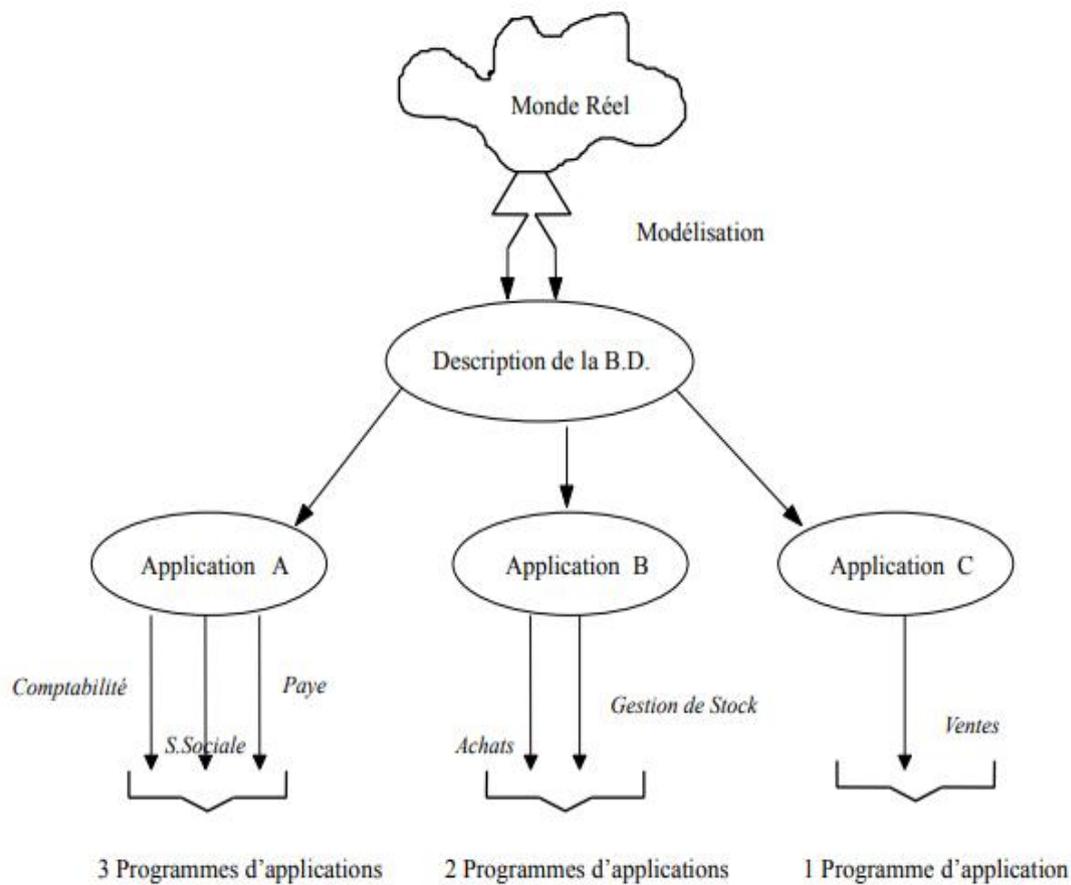


Figure 1.1 : Processus de Modélisation d'une B.D

2.5. Types d'objets de base de données :

Il existe quatre types différents d'objets de base de données qui aident les utilisateurs à compiler, saisir, stocker et analyser des données dans divers formats :

1. Tables
2. Requêtes
3. Formulaires
4. Rapports

1.5.1 Pourquoi les bases de données sont-elles importantes ?

Les bases de données sont utilisées pour conserver des montagnes de données collectées de manière ordonnée et facilement accessibles à un utilisateur autorisé. Chaque entreprise utilise une base de données différente, selon la nature de ses données. Les bases de données sont importantes pour la croissance d'une entreprise à bien des égards :

- Permet à une entreprise de prendre des décisions commerciales éclairées.

- Stockez et récupérez efficacement les informations associées.
- Aide à analyser et à aggraver les données commerciales.
- Collectez et stockez des données clients cruciaux à partir de différentes applications.
- Fournit des applications opportunes et personnalisées basées sur les données et des analyses détaillées.
- Assure un accès immédiat aux données commerciales cruciales qui peuvent être utilisées par différents services commerciaux pour comprendre les modèles de données, générer des rapports et prédire les tendances futures.
- Souvent les données sont mappées des bases de données hiérarchiques utilisées par les systèmes hérités aux bases de données relationnelles utilisées dans les entrepôts de données. [03]

1.5.2 Types de bases de données :

Il existe de nombreux types de bases de données. La meilleure base de données pour une organisation spécifique dépend de la manière dont l'organisation souhaite utiliser les données.

- Bases de données relationnelles. Les bases de données relationnelles ont dominé les années 80. Les éléments d'une base de données relationnelle sont organisés en un ensemble de tables comportant des lignes et des colonnes. La technologie de base de données relationnelle offre le moyen le plus efficace et flexible d'accéder à des informations structurées.
- Bases de données orientées objet. Les informations d'une base de données orientée objet sont représentées sous forme d'objets, comme dans la programmation orientée objet.

Bases de données distribuées. Une base de données distribuée se compose de deux fichiers ou plus, situés dans différents sites. La base de données peut être stockée sur plusieurs ordinateurs, situés au même endroit physique, ou répartis sur différents réseaux.

- Data Warehouse. Un data Warehouse, référentiel central de données, est un type de base de données spécifiquement conçu pour permettre une interrogation et une analyse rapides.
- Bases de données NoSQL. Une base de données NoSQL, ou non relationnelle, permet de stocker et de manipuler des données non structurées et semi-structurées (à la différence d'une base de données relationnelle, qui définit la manière dont doivent être composées toutes les données insérées dans la base de données). Les bases de données NoSQL ont gagné en popularité à mesure que les applications Web sont devenues plus courantes et complexes.

- Bases de données orientées graphe. Une base de données orientée graphe stocke des données en termes d'entités et de relations entre les entités.
- Bases de données OLTP. Une base de données OLTP est une base de données rapide et analytique, conçue pour accepter un grand nombre de transactions effectuées par plusieurs utilisateurs. [04]

2.6. Système de Gestion de Base de Données :

Un Système de Gestion de Base de Données (SGBD) est un logiciel qui permet de stocker des informations dans une base de données. Un tel système permet de lire, écrire, modifier, trier, transformer ou même imprimer les données qui sont contenus dans la base de données. [05]

Parmi les logiciels les plus connus il est possible de citer : MySQL, PostgreSQL, SQLite, Oracle Database, Microsoft SQL Server, Firebird ou Ingres.

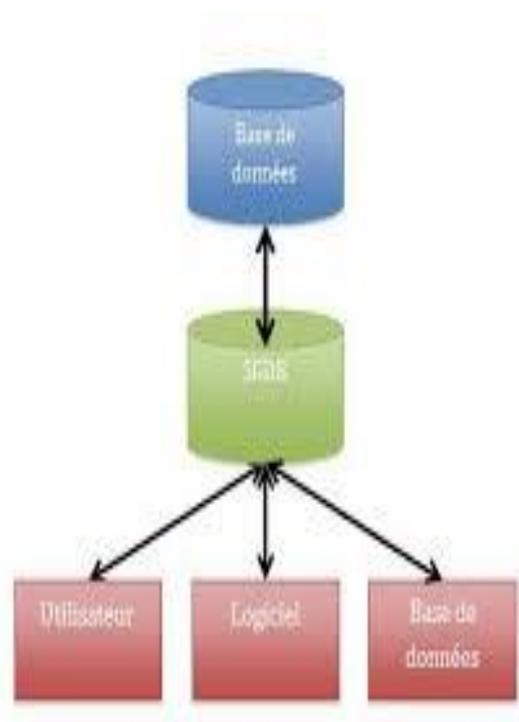


Figure 1.2 SGBD système de gestion de bases de données

CHAPITRE II:

les concepts du Cloud

Introduction

Dans ce chapitre Nous allons proposer une idée de stockage des requêtes OLAP, en traitant ces dernières dans cloud computing où nous présenterons les types et les service de cloud computing, puis les systèmes de gestions des données et requêtes OLAP sur cloud

1 .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. [06]

2. Services du Cloud Computing

2.1. Infrastructure en tant que Service

IaaS (Infrastructure as a Service) constitue la base du cloud, composé de serveurs, de systèmes de stockage et d'équipements réseau fournis en tant que service. Cette infrastructure est fournie de manière à gérer automatiquement la charge de travail requise par les applications. [07]

2.2. Plate-forme en tant que 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. erreurs d'application. Cette plate-forme comprend généralement la base de données, les solutions middleware¹, les outils de développement et de gestion, tous fournis en tant que service via Internet. [08]

2.3 .Software en tant que Service

SaaS (Software as a Service) fournit à ses clients des applications, accessibles via un navigateur, sans avoir à se soucier ni à gérer les relations, les serveurs et les systèmes d'exploitation. [09]

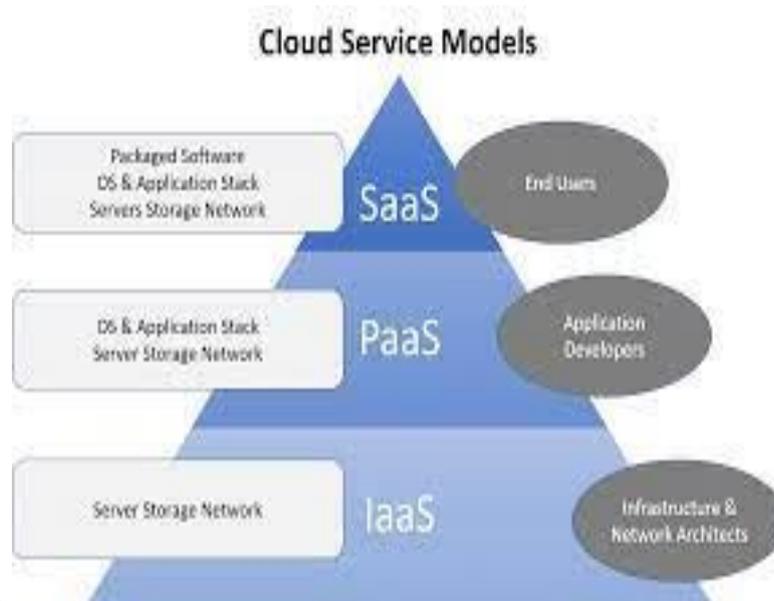


Figure 2.1 : Les services du Cloud.

3 .Types de cloud computing

Aucun concept de cloud computing n'évolue. On peut créer quatre types de cloud computing .

- Le cloud privé (ou in-house)
- Cloud public (ou externe) : Il s'agit d'un service convivial qui fournit des services de stockage et des applications Web au grand public.
- Cloud hybride (interne et externe) : Environnement composé de multiples prestataires de services internes et externes.
- Community Cloud : L'infrastructure est partagée par plusieurs organisations qui soutiennent une communauté particulière et ont des intérêts communs. Il peut être géré par des organisations ou par un tiers. [10]

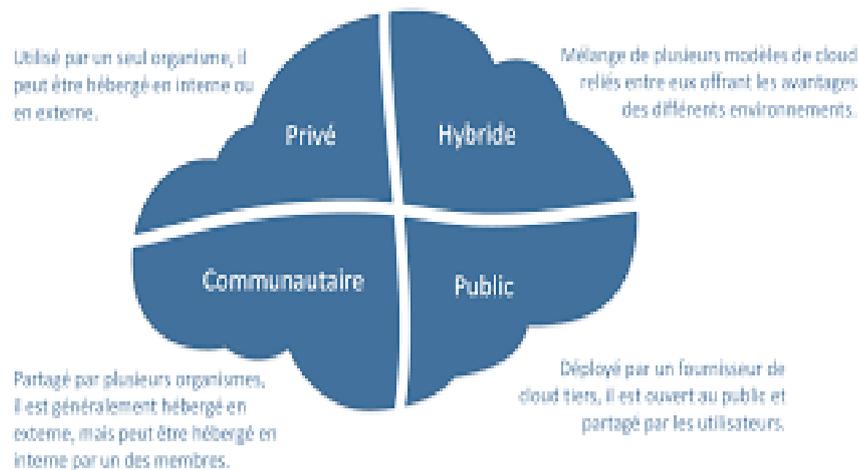


Figure 2.2: Les types du Cloud

4. Caractéristiques des données dans les nuages

Parmi les caractéristiques des données cloud nous citons :

- **Le stockage distribué** : Les données cloud doivent être partageables entre plusieurs nœuds du cluster avec une garantie :

- **Partage de charge automatique entre différents nœuds** : lorsqu'un nouveau nœud est ajouté au cluster, celui-ci doit conserver la Propriétés.

Distribution automatique de la charge du nœud défectueux aux autres nœuds.

- **Intégration native** : capacité de base à gérer des requêtes dans de très grands ensembles de données.

- **Traitement distribué** : lorsqu'un client demande un traitement qui inclut

Plusieurs nœuds du cluster, il est préférable de traiter au niveau de chaque nœud respectif et d'envoyer les résultats au client, plutôt que de transférer toutes les données pour un traitement local.

- **Flexibilité des données** : il est possible pour le fournisseur de demander à l'entreprise d'estimer la charge de consommation, mais en aucun cas les délais ne doivent être prolongés si le besoin augmente.

Si une ressource demande un délai de demande de ressources (serveurs, disques, etc.) en fonction de la croissance de l'utilisation des services, c'est que son système ne respecte pas le principe d'élasticité du cloud computing.

- **La scalabilité des données** : c'est-à-dire la capacité du système à s'adapter aux dimensions du

problème qu'il doit traiter, le mécanisme de scalabilité du calcul des ressources est possible en sélectionnant uniquement grâce au principe de résilience cloud. [11]

4.1 Systèmes de gestion de données en nuage

Les systèmes traditionnels de gestion et d'exploitation des données spatiales sont de type transactionnel ou OLTP (Online Transaction Processing), et les nouveaux outils d'exploitation des données spatiales sont de type analytique ou OLAP. [12]

4.2. Système de gestion des données de transaction

OLTP est une tâche majeure des bases de données relationnelles traditionnelles (bases de données) et un processus enregistré quotidiennement de stockage, de mise à jour, de sécurité et d'intégrité des données. Un système de transaction est généralement une base de données, développée par une application, qui stocke les données actuelles de l'organisation, c'est-à-dire qu'il n'y a pas de données d'archives dans les systèmes de transaction.

4.3 . Système de gestion des données analytiques

OLAP (Online Analytical Processing) est une tâche majeure des systèmes d'entreposage de données, d'analyse de données et de prise de décision. L'analyse en ligne est un autre aspect du processus d'entreposage de données. Codd (1993) définit OLAP comme "l'analyse commerciale dynamique requise pour générer, manipuler, déplacer et synthétiser des informations à partir de modèles d'analyse de données, et cela inclut la capacité de discerner des relations nouvelles ou inattendues entre les variables, la capacité de spécifier les paramètres nécessaires pour traiter de grandes quantités de données et la création d'un nombre illimité de dimensions"

Un système OLAP est un dispositif avec des opérateurs spécifiques qui permet l'analyse de données sur Internet. Il est également considéré comme un serveur d'application qui peut traiter les données directement à partir du référentiel ou il peut être utilisé comme un outil d'exploration des données via une navigation interactive.

Entre autres choses, les applications OLAP vous permettent de travailler avec des données historiques pour étudier les tendances ou les prévisions d'une activité, ou de travailler avec des données récapitulatives pour créer une stratégie d'information pour la prise de décision. L'analyse en ligne peut être appliquée aux données du référentiel ainsi qu'aux données du magasin de données. Habituellement, il est plutôt effectué sur un

ensemble de données plus précis appelé un cube de données.

5. OLAP sur cloud :

OLAP : définition d'une technologie d'analyse multidimensionnelle

OLAP, acronyme de Online Analytical Processing, est une technologie permettant d'effectuer des analyses de données multidimensionnelles au sein de bases de données créées à cet effet.

Découvrez la définition de l'OLAP informatique, son fonctionnement, son utilité, ses avantages, Les différentes solutions du marché et leur prix.

5.1. Le rôle d'OLAP :

1. Traitement analytique interactif (Codd) typique dans les systèmes informationnels
2. Catégorie de traitements dédiés à l'aide à la décision
3. Analyses diverses (multidimensionnelles)
4. Information : surtout dérivée et sommaire
5. Aide à la prise de décision

5.2. Restitution : Analyse OLAP multidimensionnelle

Les magasins de données reposent sur une modélisation multidimensionnelle des données extraites de l'entrepôt. Ceci permet de représenter les données d'un magasin sous la forme de points dans un espace à plusieurs dimensions avec la métaphore de *cube* ou d'*hyper cube* de données. Cette modélisation permet l'expression d'analyses en ligne (OLAP) multidimensionnelles.

Représente les données en cube multi-dimensionnel où chaque côté est une dimension d'analyse et chaque case une métrique

Opérations sur le cube :

1. Rotate : sélection du pivot d'analyse en faisant tourner le cube

2. Slice : extrait une tranche du cube, c'est-à-dire, un sous-ensemble des valeurs du cube correspondant à une certaine valeur d'une des dimensions non-découpées
3. Drill-down : détaille les données en descendant le niveau hiérarchique d'une dimension
4. Roll-up : agrège les données en augmentant le niveau hiérarchique d'une dimension

Exemple. La Figure 4 présente un exemple de cube, qui permet l'analyse des activités d'une chaîne de revendeurs informatiques. Il modélise les activités de ventes de cette chaîne. Les *Quantités* et les *Montants* des ventes représentent les indicateurs d'analyse disponibles du sujet *Ventes*. Ces indicateurs peuvent être analysées en fonction de trois dimensions : les *Magasins* où ont été effectuées les ve

ntes, les *Dates* correspondantes aux dates de ventes et les *Produits* vendus. Chacune de ces dimensions dispose de plusieurs niveaux de détail (ville, pays, continent...) et permettent d'obtenir une vision plus ou moins détaillée lors des analyses.

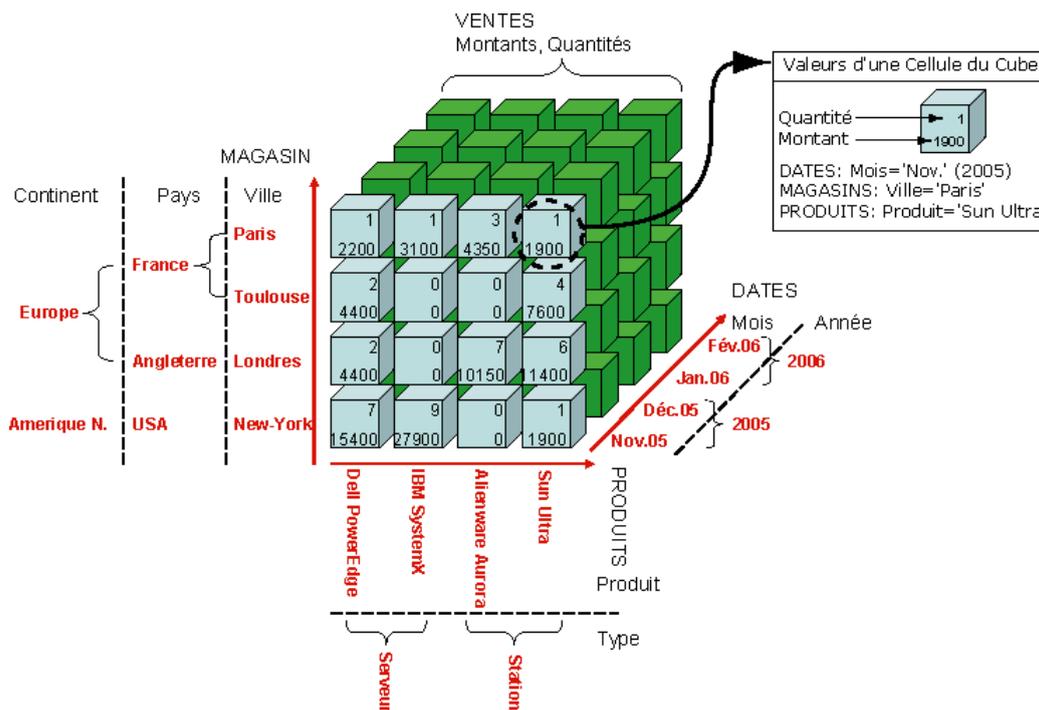


Figure 2.3: Exemple d'analyse des ventes d'une chaîne de magasins informatiques (vision en cube).

5.3. OLAP ou Online Analytical Processing : à quoi ça sert ?

L'OLAP peut être utilisé pour le data mining (forage de données) ou encore pour la découverte de relations entre les données. Une base de données OLAP n'a pas besoin d'être aussi

Large qu'une Data Warehouse, (entrepôts de données en Français) car les analyses de tendances ne requièrent pas toutes les données transactionnelles.

En utilisant ODBC (Open Database Connectivity), il est possible d'importer des données en provenance de bases de données relationnelles existantes afin de créer une base de données multidimensionnelle ou database pour l'OLAP. Ainsi, il permet de comparer des données en provenance de plusieurs bases de données. C'est particulièrement utile pour les professionnels du Business Intelligence. [13]

5.4 Les différents types de systèmes :

On dénombre différents types de systèmes. Les principaux sont MOLAP et ROLAP. MOLAP : multidimensionnel Online Analytiques Processing les produits MOLAP permettent aux utilisateurs de modéliser les données au sein d'un environnement multidimensionnel, plutôt que de fournir une vue multidimensionnelle de données relationnelles comme le font les produits ROLAP.

5.5. Définition du concept de cube OLAP :

Il faut savoir que tout comme un système OLTP s'appuie sur un SGBDR, le système décisionnel OLAP s'appuie sur le Data Warehouse. Le Data Warehouse est le socle indispensable pour obtenir les réponses aux questions essentielles à la prise de décision et au pilotage de l'entreprise. Si vous prêtez attention, vous vous rendrez compte que conceptuellement, un Data Warehouse est un modèle multidimensionnel. C'est un modèle relationnel dit dénormalisé.

Or, quitter d'un modèle de base de données opérationnelle à un modèle de base de données décisionnelle revient à transformer les éléments du modèle de base de données opérationnelle en « dimensions » et en « faits ». Nous y reviendrons dans le point suivant. Cette transformation est nécessaire car comme dit plus haut, le modèle OLTP fourni uniquement une vue aplatie et statique des données dans un point de l'espace-temps. Cette vue aplatie et statique qu'on appelle une base de données, est nécessaire pour assurer la performance des transactions opérationnelles, et ainsi garantir la disponibilité 24h/24 et la haute performance du SGBD. [14]

5.6 OLAP et cube de données

OLAP est l'une des technologies clés d'un cadre d'intelligence d'affaires. Il s'agit d'un outil puissant qui permet aux utilisateurs d'accéder rapidement et de manière interactive à plusieurs vues d'informations. Par exemple, les utilisateurs peuvent demander des rapports sur tous les revenus des ventes de produits en Europe au cours d'une année donnée, mais les utilisateurs peuvent également explorer les ventes de téléviseurs en septembre, calculer les ventes depuis le début de l'année ou comparer les revenus avec ceux des mêmes produits vendus. En janvier.

OLAP stocke les données dans des modèles de cubes de données à plusieurs dimensions. Par exemple, les ventes par région, les ventes par commercial, les ventes par catégorie, les ventes par trimestre, etc. Le cube OLAP se compose de dimensions et de faits numériques appelés mesures. Ces dimensions et mesures sont stockées sous forme d'étoile ou de flocon de neige (voir figure 5) . Les mesures (par exemple, chiffre d'affaires, unité vendue, nombre de clients) sont dérivées des enregistrements de la table de faits et les dimensions (par exemple, client, produit, temps) sont dérivées des tables de dimensions. Normalement, les cubes OLAP ont plus de 3 dimensions et chaque dimension contient plusieurs attributs. Par exemple, une dimension de magasin peut avoir des attributs comme le pays, la région ou la ville ; une dimension de produit peut avoir des attributs tels que la marque, la catégorie ou le département. Chaque cellule d'un cube contient des mesures qui sont correctement agrégées selon ses dimensions. Les opérations d'agrégation peuvent être SUM, AVG, MAX, MIN, etc. [15]

5.7. Objectif :

Obtenir des informations déjà agrégées selon les besoins des utilisateurs.

Représentation de l'information dans un hyper cube à N dimensions.

5.8. Opérations OLAP :

Fonctionnalités qui servent à faciliter l'analyse

Multidimensionnelles : opérations réalisables sur l'hyper cube

5.9. Stockage d'un cube de données :

1. ROLAP: Relational On-Line Analytical Processing

Les données sont stockées comme des tables relationnelles : une table de faits et des

tables de dimension

2. MOLAP : **M**ultidimensional **O**n-**L**ine **A**nalytical **P**rocessing

Le cube de données est stocké sous forme d'un tableau multi- dimensionnel.

Les recherches de Ventana montre que 86 % des organisations interrogées s'attendent à ce que la majorité de leurs données soient dans le cloud et 99 % s'attendent à effectuer leurs analyses dans le cloud. L'une des raisons impérieuses pour lesquelles les organisations migrent massivement vers le cloud est de simplifier leur pile d'analyse. Dans le cloud, les utilisateurs professionnels sont libérés du fardeau de la gestion de leurs propres clusters de plateformes de données et peuvent étendre et contracter leurs ressources de données pour répondre à la demande sans les frictions liées au provisionnement traditionnel des centres de données. [16]

6. La nouvelle pile Cloud Analytiques :

Hadoop a été le début du concept de lac de données et a remis en question la domination de l'entrepôt de données d'entreprise traditionnel hautement structuré (EDW). Désormais, chacun des fournisseurs de cloud public possède sa propre version d'un lac de données. Leurs systèmes de fichiers distribués respectifs (AWS S3, Azure Synapse et Google Cloud Storage) sont devenus la zone d'atterrissage des données dans le cloud. La renaissance de l'entrepôt de données dans le cloud avec Snowflake, AWS Redshift, Azure SQL DW et Google BigQuery, offre le confort de l'EDW mais avec l'élasticité et la facilité opérationnelle du cloud. Cependant, même s'il n'existe pas de référentiel unique pour les données dans le cloud, les entreprises s'efforcent toujours d'assurer la cohérence sémantique, la gouvernance des données et la capacité de gérer leurs données dans un emplacement central. Pour atteindre cet objectif, AtScale propose une couche de virtualisation intelligente des données qui connecte différentes sources de données et les rend accessibles à partir d'un point d'accès logique commun aux données pour l'analyse Cloud. [17]

7. LEÇON D'HISTOIRE RAPIDE : TRAITEMENT ANALYTIQUE EN LIGNE OU "OLAP" :

OLAP est l'acronyme de Online Analytical Processing. Le chercheur de bases de données, E. F. Codd, a inventé le terme « traitement analytique en ligne » (OLAP) dans un livre blanc publié en 1993.

Lorsque OLAP a été créé, les bases de données étaient essentiellement bidimensionnelles - enregistrements et champs - et nécessitaient un langage de requête (SQL) pour récupérer les données. Avec OLAP, les utilisateurs pouvaient utiliser des "tableaux croisés" et des tableaux croisés dynamiques pour formuler leurs requêtes. Comme une feuille de calcul, OLAP permet des calculs et des requêtes basées sur des cellules, présentant les données de manière multidimensionnelle au lieu de simplement des lignes et des colonnes.

Les utilisateurs professionnels peuvent simplement utiliser un outil de BI ou Excel pour explorer, faire pivoter et échanger des dimensions et des mesures. Il est rapidement devenu la langue par défaut des affaires. Au lieu de regarder des enregistrements, des champs et des faits, n'importe quel utilisateur pourrait regarder "Ventes par région", "Ventes par produit", "Ventes par canal", etc. Ces "par" définissent la multi dimensionnalité réelle d'OLAP et la capacité de il suffit de glisser-déposer, d'explorer vers le bas, d'explorer vers le haut, etc. avec le "clic d'une souris" au lieu d'écrire une requête SQL. [18]

7.1. CLOUD OLAP RÉUSSIT LÀ OÙ L'OLAP TRADITIONNEL ÉCHOUE :

Selon Gil Press, Senior Contributor à Forbes, dans son « A Very Short History of Big Data », le terme « Big Data » a commencé à être utilisé en 1998. En 2001, Doug Laney, analyste pour le Meta Group, commençait pour écrire sur les trois V : volume, vitesse et variété de Big Data. En 2002, Doug Cutting et Mike Cafarella travaillaient sur un projet Apache Nutch qui consistait à créer un moteur de recherche Web capable d'explorer et d'indexer les sites Web. Cela s'est transformé en un projet open source appelé "Hadoop". En 2008, Yahoo et Facebook ont commencé à utiliser Hadoop et la technologie Hadoop a commencé à vaincre les superordinateurs pour devenir le système le plus rapide de la planète en triant un téraoctet entier de données.

Les co-fondateurs d'AtScale, Dave Mariani et Sarah Gerweck, étaient chez Yahoo à cette époque et ils utilisaient OLAP avec Hadoop pour analyser les données de navigation et de publicité de Yahoo. Bien qu'ils aient atteint leurs objectifs commerciaux d'offrir une valeur réelle aux consommateurs et de promouvoir un degré incroyable de libre-service, l'architecture s'est avérée trop rigide et trop fragile, n'était pas évolutive, n'était pas sécurisable et n'était pas durable.

Pendant l'ère Hadoop, les cubes OLAP étaient encore largement utilisés et « explosaient » de

données. Pour certains utilisateurs de Big Data, OLAP et « Cube » étaient devenus difficiles. Les administrateurs OLAP et les utilisateurs expérimentés ont commencé à ressentir la douleur alors que leurs données augmentaient de manière exponentielle tandis que leurs utilisateurs professionnels s'attendaient toujours aux mêmes temps de réponse de « vitesse de pensée ». Les co-fondateurs d'AtScale ressentaient également la douleur et ont décidé de créer un nouveau type d'OLAP pour les architectures de données modernes dans le Cloud.

7.2. Le cloud OLAP (COLAP) est l'avenir. Le futur c'est maintenant :

La renaissance de l'entrepôt de données dans le cloud avec Snowflake, AWS Redshift et Google BigQuery, offre le confort de l'EDW avec l'élasticité et la facilité opérationnelle du cloud. Cloud OLAP (COLAP) fait fonctionner OLAP à la vitesse et à l'échelle du cloud pour analyser de grandes quantités de données sans les déplacer hors de l'entrepôt de données cloud ou du lac de données. En exploitant le moteur Cloud OLAP d'AtScale, les utilisateurs professionnels ont :

1. Compatibilité OLAP avec Excel et tout outil de BI qui parle MDX ou SQL
2. Requêtes multidimensionnelles rapides, cohérentes et peu coûteuses
3. Outils de modélisation simples pour ajouter instantanément de nouveaux éléments de données et sources de données
4. Accès direct à tout entrepôt de données ou données lac -- y compris les entrepôts de données cloud et les données imbriquées
5. Contrôles de sécurité et de gouvernance pour gérer l'accès aux données en un seul endroit
6. Une source unique de vérité pour les métriques commerciales critiques, définies côté serveur [19]

1. Hibernate / JPA

8.1. Présentation générale

Nous avons donc un premier aperçu de JDBC, et la démarche logique que nous avons suivie en cherchant à respecter les principes d'indépendance dictés par notre approche MVC nous a mené à « encapsuler » les lignes d'une table relationnelle sous forme d'objet avec d'obtenir une intégration naturelle à l'application. Cela revient à isoler les accès à la base dans la couche modèle, de manière à ce que ces accès deviennent transparents pour le reste de l'application. À ce stade les avantages devraient déjà vous être perceptibles:

- Possibilité de faire évoluer la couche d'accès aux données indépendamment du reste de l'application;
- Possibilité de tester séparément chaque couche, et même la couche modèle sans avoir à se connecter à la base (par création d'entités non persistantes).

Si ce n'est pas encore clair, j'espère que cela le deviendra. Au passage, on peut souligner qu'il est tout à fait possible de développer une application valide sans recourir à ces règles de conception qui peuvent sembler lourdes et peu justifiées. C'est possible mais cela devient de moins en moins facile au fur et à mesure que l'application grossit ainsi que les équipes en charge de la développer.

- Outil ORM ou Cadre (Framework) de persistance libre (open source) gérant la persistance des objets Java/J2EE en base de données relationnelle [Wikipédia, Pat05]
- Version ORM 4.3.7 (2010: 3.6) : implémentation du standard de persistance EJB 3.0 Java Persistence API (JPA)
- Possibilité d'être utilisé aussi bien dans un développement client lourd, que dans un environnement web léger de type Apache Tomcat ou dans un environnement J2EE complet [Wikipédia]
- Code SQL généré à l'exécution via des informations fournies dans un document de correspondance (mapping) XML ou des annotations[20]

8.2. Différents modules :

1. Hibernate Core : API native implémentant les services de base pour la persistance
 - Métadonnées au format XML (+ annotations depuis la version 3.6)
 - Langage HQL et interface pour écrire des requêtes
2. Hibernate Annotations (inclus dans Hibernate Core 3.6) : Remplacement des fichiers XML par des annotations JDK 5.0 implémentant les annotations du standard JPA +

annotations spécifiques à Hibernate

3. Hibernate Entity Manager : Implémentation de la partie des spécifications JPA concernant

- Les interfaces de programmation,
- Les règles de cycle de vie des objets persistants
- Les fonctionnalités d'interrogation

Hibernate Entity Manager = wrapper au-dessus du noyau Hibernate implémentant une solution complète de persistance JPA (cf. documentation Hibernate)

8.3 . Architecture du noyau Hibernate

Hibernate. properties : Fichier de configuration [21]

- Version XML : hibernate.cfg.xml permettant un paramétrage plus fin
- Configuration par programmation

1. XML mapping ou annotations :

- Méta-données (paramètres) décrites sous la forme de fichiers de correspondance XML ou sous forme d'annotation
- Utilisées pour mettre en correspondance les classes Java et le modèle relationnel

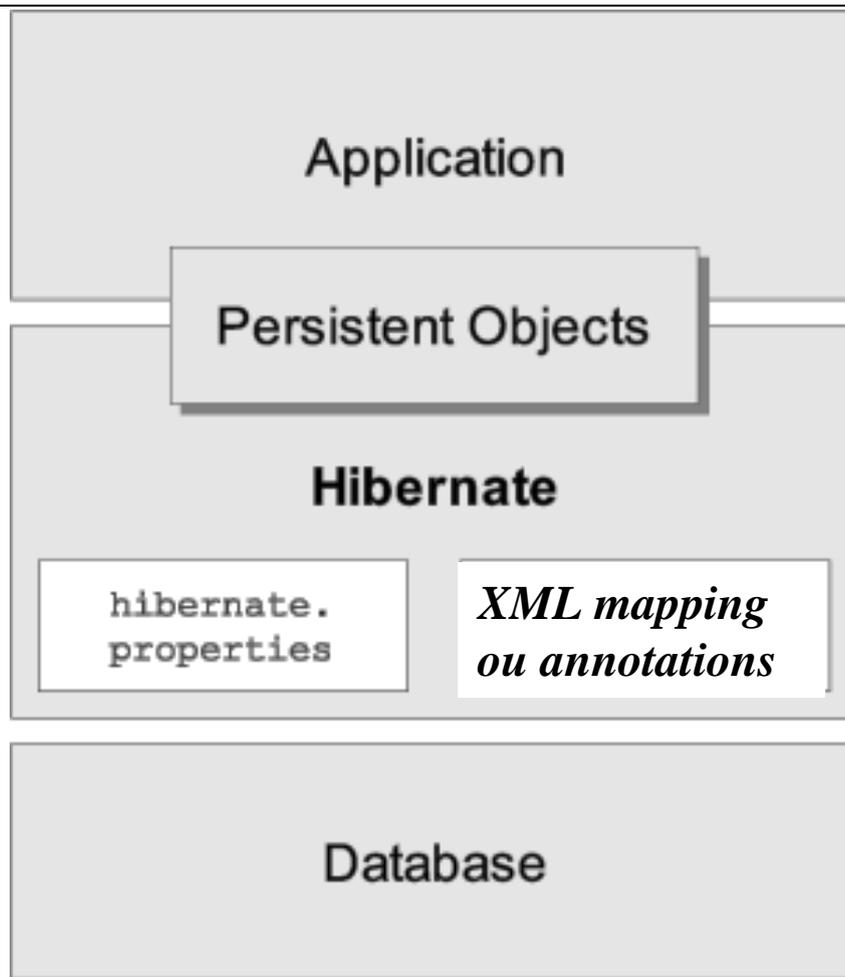


Figure 2.5 : Architecture du noyau Hibernate

Conclusion

Dans ce chapitre nous avons donné 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 Mécanisme d'optimisation de données, les outils et les systèmes de architecture OLAP dans le Cloud.

CHAPITRE III :

L'optimisation De Système De Gestion Des Données

Introduction

La donnée est au cœur des activités de chaque entreprise, son importance et son volume augmentent rapidement, pouvoir s'appuyer sur les mécanismes d'optimisation de la gestion des données devient de plus en plus important. Ce qui permet de meilleures performances de requête OLAP en fournissant un accès rapide aux données.

1 . Mécanisme d'optimisation et de gestion de donnée :

1.1 Cache :

Le cache est un emplacement de stockage utilisé pour stocker des données avec une forte localité spatiale ou temporelle. Mémoire utilisée pour le cache par rapport à l'espace mémoire principal ; Cela permet au processeur de travailler avec des copies des données dans le cache plutôt que dans l'emplacement de la mémoire principale.

Les caches distribués sont des caches particulièrement appropriés pour les environnements distribués et ont été largement étudiés dans le passé. Lorsque des solutions de cache distribué ont été proposées, elles sont soit intégrées dans un système de fichiers distribué, soit à un niveau supérieur et ne nécessitent aucune optimisation des applications pour les utiliser. [21]

1.2. Les snapshots :

Un snapshot est une image instantanée du disque à un instant précis, permettant ainsi de revenir en arrière sur l'état du disque en cas de problème. C'est donc un moyen de faire un point de restauration sur le disque dur. [22]

1.3. Index :

Les index sont des structures physiques qui réduisent le temps d'exécution des requêtes en précalculant les jointures et fournissent un accès direct aux données. Cependant, lors de la mise à jour d'un entrepôt de données, ces structures doivent également être mises à jour, ce qui crée une surcharge pour le système.

Les index de jointure sont des index qui permettent de précalculer les jointures mais uniquement la clé de deux relations. Un index est une structure qui contient généralement une clé et une adresse où les enregistrements avec cette clé peuvent être trouvés. Il est généralement organisé en arbre B. [23]

Les caractéristiques des indices sont les suivantes :

1. Mise à jour automatique pour les changements de table.
2. Possibilité de créer plusieurs index sur la même table.
3. Peut faire référence à plusieurs attributs d'une même table (index composite). [24]

1.4. Vues : [25]

Une scène est simplement une question à laquelle nous donnons un nom. La sélection de données à partir d'une vue exécute en fait la requête SQL de la vue, de sorte que les vues n'offrent aucun avantage en termes de performances.

En fait, dans certains cas, l'interrogation des vues peut être plus lente que l'interrogation directe des tables.

1.5. Vue physique

Une vue matérialisée est un objet qui permet de stocker le résultat d'une requête de requête, là où une vue ne stocke que la requête, la vue matérialisée stocke directement les résultats (afin qu'elle puisse les exécuter) et non la requête. Ainsi, lorsque nous interrogeons une vue matérialisée, nous regardons directement les données qu'elle contient, sans traverser les tables mères et/ou les tables temporaires intermédiaires.

Une VM (Materialized View) est un moyen simple de créer une vue matérialisée d'une table, c'est un instantané des données au moment de l'exécution de la requête. Contrairement à une vue standard, le résultat de la requête est stocké physiquement dans la base de données.

Les objectifs les plus importants de la pensée matérialiste sont :

- Pour améliorer les performances d'accès et réduire le trafic réseau, ils sont mis à jour régulièrement, ce qui les rend très performants.
- Stocker les résultats des commandes SQL dans une table.
- Fournit à l'optimiseur un chemin alternatif (généralement plus rapide) vers les données

1.5.1 Types de vues matérialisées

Il existe plusieurs types de visualisations, nous citons les suivantes :

1. Vues matérialisées avec agrégation : en général, les vues matérialisées contiennent des regroupements. Pour une mise à jour rapide, la liste de sélection doit contenir toutes les colonnes de la clause GROUP BY et un COUNT(*) et un COUNT(column) pour chaque colonne agrégée. Pour la maintenance, un journal de vue matérialisée doit exister pour chaque table référencée par la vue matérialisée.

1.5.1.1 Vue réalisée avec plusieurs groupes d'agrégation : la vue réelle peut avoir

plusieurs niveaux d'agrégation, ce qui est typique pour les requêtes OLAP (traitement analytique en ligne). Par exemple, la même requête peut comparer des agrégations avec différents niveaux de précision. Ces requêtes sont évidemment très gourmandes et le prétraitement des résultats est très efficace.

1.5.1.2. Vues physiques avec liens : certaines vues physiques n'ont que des liens et aucun regroupement. Ils conduisent souvent à des jointures coûteuses entre les tables.

1.5.1.3. Vue matérialisée imbriquée : une vue matérialisée imbriquée est une vue matérialisée dont la définition est basée sur la vue matérialisée. Ce type de vue est souvent utilisé lorsque l'on veut maintenir rapidement à jour les propriétés d'une vue matérialisée complexe. Si l'on veut une approche matérialisée qui inclut des jointures et des agrégations tout en conservant une fonctionnalité de mise à jour rapide, elle sera nécessairement imbriquée.

1.5.2. Problèmes avec les pensées matérialisées

De nombreux travaux traitent de questions d'approche physique dans des contextes variés. On peut distinguer deux aspects principaux : ?

Le problème du choix des idées matérielles, qui consiste à matérialiser l'ensemble des idées de manière à ce que la contrainte à considérer soit optimale. , Le problème de la maintenance des vues matérialisées, qui propose de se propager rapidement aux mises à jour qui interviennent au niveau de la source de données.

6. Sélection des vues de matérialisées

Le problème de sélection de vues matérialisées consiste à créer une configuration de vues qui optimise le coût d'exécution d'une charge donnée. Cette optimisation peut être obtenue sous certaines conditions aux limites, par ex. B. l'espace de stockage alloué aux scènes sélectionnées, ou une limite supérieure des coûts de maintenance des scènes sélectionnées.

Il existe trois manières de choisir un ensemble de considérations matérielles :

a) Mettre en œuvre toutes les idées : Cette approche consiste à incarner les idées de tous les candidats, elle donne le meilleur temps de réponse pour toutes les demandes, mais il est peu pratique de stocker et de maintenir toutes les idées.

b) Ne pas implémenter une vue : Dans ce cas, nous sommes obligés d'accéder aux données

relationnelles sous-jacentes. Cette solution n'offre aucun avantage en termes de performances de requête

c) l'incarnation d'une partie seulement des idées: dans ce cas, il existe une certaine dépendance entre les idées, c'est-à-dire que la valeur de certaines idées peut être calculée à partir des valeurs d'autres idées. Par conséquent, il est souhaitable de pratiquer les parties communes à travers plusieurs questions. Cette solution semble plus intéressante que les deux méthodes précédentes.

Quel que soit le type de problème de sélection d'offres réalisé, il peut être défini comme suit : Compte tenu des limitations des S-ressources (par exemple la mémoire), le problème de sélection de vues consiste à définir un ensemble de vues $\{V_1, V_2, \dots, V_k\}$ pour réduire une fonction objectif (coût global d'évaluation de la requête et/ou coût de maintenance défini visuellement) et le respect de la contrainte .

N'oubliez pas que l'objectif principal des considérations matérialisées est de réduire le coût de l'évaluation de questions spécifiques. $Q = \{Q_1, \dots, Q_k\}$ (par exemple le plus courant) est défini dans l'entrepôt.

Les algorithmes proposés pour déterminer un ensemble de vues peuvent être classés selon le type de contraintes qu'ils utilisent : avec des contraintes de mémoire, i. H. dire que la taille des vues sélectionnées ne doit pas dépasser la capacité de l'espace disque disponible. Un modèle de coût est proposé pour estimer le coût de stockage (nombre de n-uplets de vue) de chaque vue. Avec des restrictions de temps de maintenance visuelles. Les algorithmes proposés sont basés sur l'inférence et utilisent des techniques de graphes et des plans d'exécution.

7.Maintenir (mettre à jour) les vues matérialisées.

La création d'une vue matérialisée à partir des données n'est pas suffisante, elle doit donc être tenue à jour au fur et à mesure que les données des tables d'origine changent.

Nous distinguons deux options de mise à jour : * Mise à jour à la demande : Une procédure stockée est nécessaire pour mettre à jour rapidement la vue matérialisée. Le plus simple serait de supprimer les données de la vue matérialisée, puis d'insérer les données mises à jour en utilisant la même requête de sélection utilisée pour créer/initialiser la vue matérialisée[26]

7.1. Mises à jour automatiques :

L'utilisation de déclencheurs pour mettre à jour la vue matérialisée est importante si, au contraire, vous souhaitez que les données soient toujours à jour avec les dernières modifications de la base de données. • Type d'actualisation : pour assurer la cohérence des données, les vues matérialisées et les tables doivent être actualisées régulièrement. Il existe trois modes de mise à jour : reconstruction complète, rapide et forcée : [26]

7.2. Mise à jour complète :

cela exécutera à nouveau la requête en fonction de la table de base et remplacera toutes les données de la VM par les données reçues, même si la table de base n'a pas changé, en fonction de la quantité de données qui satisfait la demande, cette mise à jour peut être gourmande en ressources. [26]

7.3. Mise à jour incrémentale (rapide) :

Son principe est de ne propager que les données qui ont changé depuis la dernière mise à jour. Ce type de mise à jour dite rapide nécessite que la base de données stocke les modifications enregistrées sur les données des tables de base. Un fichier journal est utilisé pour cela. Ce type de mise à jour est particulièrement efficace lorsque les tables de base sont relativement inchangées. - Mise à jour forcée: Avec ce type, quand une génération rapide n'est pas possible ; la génération complète est exécuté. [26]

Conclusion

Dans ce chapitre, nous avons présenté divers mécanismes pour améliorer la gestion des données, tels que SYSTEM ou DATA), des index qui permettent de réduire le temps d'exécution des requêtes en précalculant les connexions et d'accéder plus rapidement aux données et aux vues réelles. Cela améliore les performances d'accès et réduit le trafic réseau.

CHAPITRE IV :

Cas d'étude et implémentation

1.introduction

Le traitement des requêtes lourdes SQL sur bases de données volumineuses nécessite la mobilisation d'importantes ressources matérielles informatique du côté serveur, ce qui sature le serveur et fait chuter les performances des services web et serveur SGBD, le cloud s'avère une bonne solution, mais reste très coûteux, car il facture les tarifs du stockage des données et le tarif du traitement de données SÉPARÉMENT, ce qui gonfle la facture des usagers, dans notre travail nous voulons trouver une solution pour joindre l'utile à l'agréable, ainsi nous voulons exploiter les ressources matérielles informatique des utilisateurs afin de traiter et stocké les requêtes en local, ce qui permet de réduire la charge de travail des serveurs, et donne plus de confidentialité aux données des utilisateur

1. Comment fonctionne une base de données en mémoire ?

Une base de données en mémoire fonctionne de manière très similaire à toute autre base de données, mais les données sont conservées dans la RAM plutôt que sur un disque traditionnel. Le remplacement de l'accès au disque par des opérations de mémoire réduit considérablement la latence requise pour accéder aux données.

L'utilisation de la RAM comme support de stockage comporte un risque. En effet si une panne de serveur se produit, toutes les données seront perdues. Pour éviter cela, des jeux de répliques peuvent être créés dans des bases de données modernes telles que MongoDB avec un mélange de moteurs en mémoire et de stockage sur disque traditionnel. Ce jeu de répliques garantit le fait que certains des membres du cluster restent des données persistantes. [27]

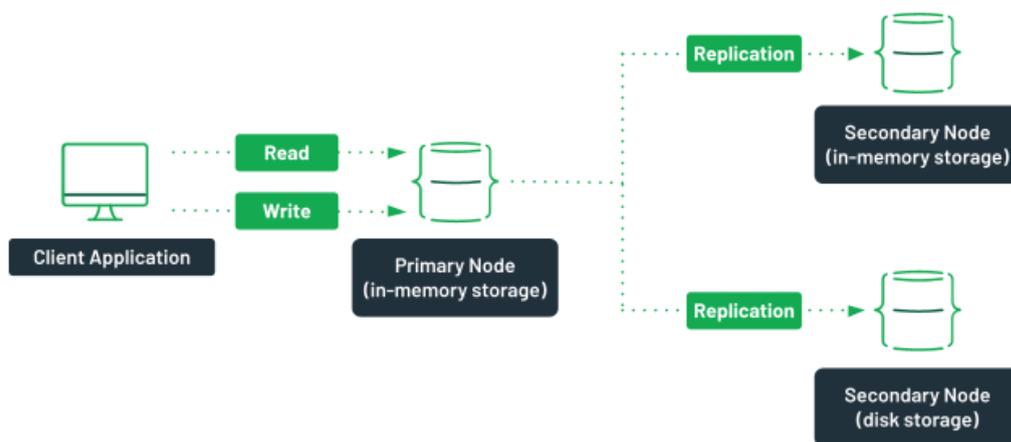


Figure 4.1 : répliques avec stockage en mémoire et traditionnel

2. Quels sont les avantages et les inconvénients des bases de données en mémoire ?

L'avantage le plus évident de l'utilisation d'une base de données en mémoire est la vitesse de récupération des données de leur base sans qu'il soit nécessaire d'effectuer des opérations sur le disque, la latence est considérablement réduite et plus cohérente. Puisqu'il n'y a plus de raisons de limiter le nombre d'opérations de lecture sur un disque, différents algorithmes peuvent être utilisés pour rechercher des données, augmentant ainsi les performances globales d'une base de

données en mémoire.

Cela peut sembler une bonne idée d'utiliser exclusivement des bases de données en mémoire pour bénéficier des gains de vitesse, mais il y a quelques inconvénients. Premièrement, le coût de la RAM est environ 80 fois plus élevé que le prix de l'espace disque traditionnel. Cela augmenterait considérablement les coûts d'exploitation de notre infrastructure, que les serveurs soient hébergés dans le cloud ou sur site.

Deuxièmement, le manque de persistance des données en cas de panne peut être un problème dans certains cas. Bien qu'il existe des moyens d'atténuer les risques associés à ces pertes de données, ces risques peuvent ne pas être acceptables pour notre analyse de rentabilisation.

Enfin, comme les serveurs ont généralement beaucoup moins de RAM que d'espace disque, il n'est pas rare de manquer d'espace. Dans ce cas, les opérations d'écriture échoueraient, perdant toutes les nouvelles données stockées.

L'utilisation d'une base de données sur disque avec un SSD NVMe peut s'avérer être une alternative solide à celles en mémoire. Ces disques offrent une bande passante de données similaire à la RAM, bien que la latence soit légèrement supérieure. Mais c'est très onéreux. [27]

3. Pourquoi utiliser une base de données en mémoire dans ce projet?

Le principal cas d'utilisation des bases de données en mémoire est lorsque des données en temps réel sont nécessaires. Avec sa très faible latence, la RAM peut fournir un accès quasi instantané aux données nécessaires. En raison des pertes potentielles, les bases de données en mémoire sans mécanisme de persistance ne doivent pas être utilisées pour les applications critiques.

Toute application dans laquelle le besoin de vitesse est plus important que le besoin de durabilité peut tirer parti de la base de données en mémoire par rapport à la base traditionnelle dans l'application que nous avons construite et sa fonction principale ainsi que les valeurs d'un ensemble de statistiques sur les utilisateurs de Facebook au moment de leur utilisation de l'application;

- Dans de nombreux cas, la base de données en mémoire ne peut être utilisée que par une petite partie de l'application globale, tandis que les données les plus importantes sont stockées dans une base sur disque.

4. Quelle est la différence entre les bases de données traditionnelles et en

mémoire ?

Une base de données en mémoire est différente d'une base traditionnelle dans le sens où elle conserve ses données dans la RAM. Cela permet aux applications d'accéder plus rapidement aux données nécessaires. En revanche, les informations ne sont pas conservées en cas de panne ou d'arrêt du système

5. Exemples de bases de données en mémoire

Les bases de données en mémoire peuvent trouver leur place dans de nombreux scénarios différents. Certains cas d'utilisation typiques pourraient inclure: [28]

- **Données IoT** Les capteurs IoT peuvent fournir de grandes quantités de données. Une base en mémoire pourrait être utilisée pour stocker et calculer des données à stocker ultérieurement dans une base traditionnelle.
- **Commerce électronique** : Certaines parties des applications de commerce électronique, telles que le panier d'achat, peuvent être stockées dans une base de données en mémoire pour une récupération plus rapide à chaque page vue, tandis que le catalogue de produits pourrait être stocké dans une base traditionnelle.
- **Jeux**: Les classements nécessitent des mises à jour rapides et des lectures rapides lorsque des millions de joueurs accèdent à un jeu en même temps. Les bases de données en mémoire peuvent aider à trier les résultats plus rapidement que les bases de données traditionnelles.
- **Gestion de session** : Dans les applications Web avec état, une session est créée pour suivre l'identité d'un utilisateur et les actions récentes. Le stockage de ces informations dans une base de données en mémoire évite un aller-retour vers la base de données centrale à chaque requête Web.

6. Utilisation en mémoire dans Redis

Redis, qui signifie Remote Dictionary Server (Serveur de dictionnaire à distance), est un système de stockage de données clé-valeur en mémoire, open source et rapide. Le projet a démarré

lorsque Salvatore Sanfilippo, le développeur initial de Redis, souhaitait améliorer la capacité de mise à l'échelle de sa start-up italienne. De là, il a développé Redis, qui est maintenant utilisé comme base de données, cache, agent de messages et file d'attente.

Redis offre des temps de réponse inférieurs à la milliseconde, permettant des millions de requêtes par seconde pour les applications en temps réel dans des secteurs comme les jeux, la publicité, les services financiers, les soins de santé et l'IoT. Aujourd'hui, Redis est l'un des moteurs open source les plus populaires, désigné comme la base de données « la plus aimée » par Stack Overflow pendant cinq années consécutives. En raison de ses performances rapides, Redis est un choix populaire pour la mise en cache, la gestion de session, les jeux, les classements, l'analyse en temps réel, le géospatial, l'appel de voiture avec chauffeur, le chat/la messagerie, le streaming multimédia et les applications pub/sub.

L'un des plus gros utilisateurs de Redis company est * Amazon * [29]

7. Redis Dans l'application (IMDB)

Nous avons utilisé l'application afin d'accélérer les requêtes et récupérer les données le plus rapidement possible

L'application est un panneau pour afficher les statistiques des utilisateurs de Facebook afin d'étudier le comportement des utilisateurs et d'analyser les données à des fins scientifiques et commerciales et les plus utilisées pour Facebook dans un groupe de pays du monde

La fonction de Redis consiste à conserver les informations extraites de la base de données dans la mémoire aléatoire (RAM)

Utiliser Redis méthodes de stockage à l'aide de clés:

l'une de ses bonnes caractéristiques, c'est qu'il stocke toutes les données dans des clés ainsi qu'il peut stocker n'importe quel format de données (html,sql,code,json.. .) avec fonction horaire[30]

7.1. Principes de base de Redis

Redis est relativement simple à apprendre car il n'y a qu'une poignée de commandes qu'il suffit de connaître. Nous couvrirons quelques concepts principaux de Redis et quelques commandes courantes utiles dans les deux prochaines sections, [31]

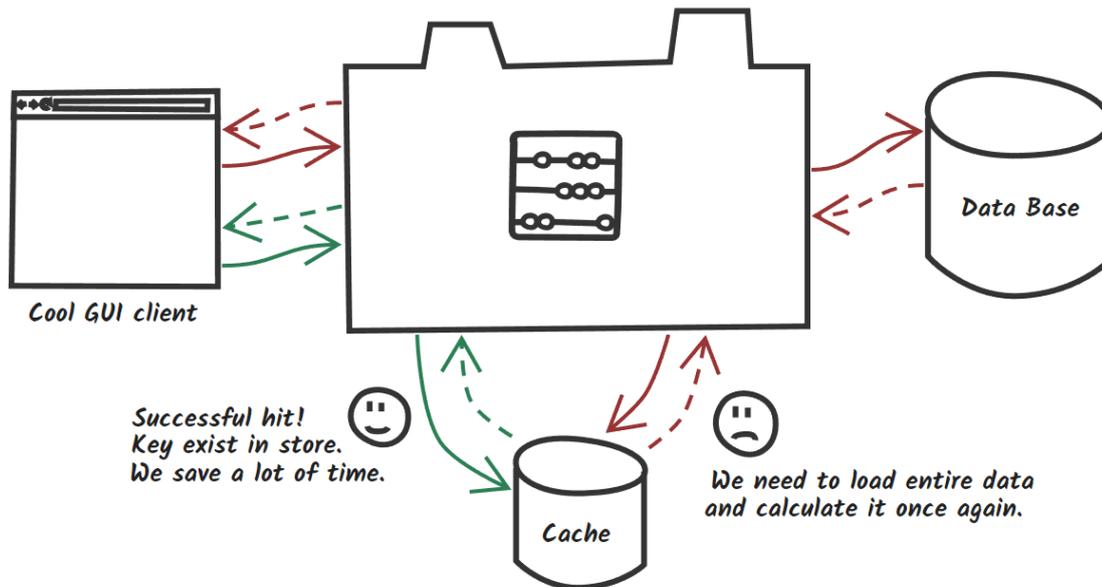


Figure 4.2: Cachette avec Redis

7.2. Prise en charge des langages de Redis

Redis prend en charge la plupart des langages et des protocoles de programmation de premier plan, notamment:

- Python
- Java
- PHP
- Perl
- Go
- Ruby
- C/C#/C++
- JavaScript
- Node.js

7.3 Cas d'utilisation de Redis les plus répandus : [32]

Mise en cache

Redis est un outil de choix pour mettre en œuvre un cache en mémoire hautement disponible afin de réduire la latence d'accès aux données, d'augmenter le débit et d'alléger la charge de notre base de données et de notre application relationnelle ou NoSQL. Redis peut traiter des éléments fréquemment demandés en offrant des temps de réponse inférieurs à la milliseconde et nous permet de mettre à l'échelle facilement pour répondre à des charges plus élevées sans accroître le backend le plus coûteux. La mise en cache de requêtes de base de données, de sessions persistantes, de pages Web et d'objets souvent utilisés comme des images, des fichiers et des métadonnées sont les exemples communs de la mise en cache avec Redis.

Messagerie, instantanée ou non, et files d'attente

Redis prend en charge le pub/sub avec la correspondance de modèles et une variété de structures de données telles que des listes, des ensembles ordonnés et des hachages. Cela permet à Redis de prendre en charge les salles de conversation ultra performantes, les flux de commentaire en temps réel, les flux des réseaux sociaux et l'intercommunication entre serveurs. La structure de données des listes Redis facilite la mise en place d'une file d'attente légère. Les listes offrent des opérations atomiques, ainsi que des capacités de blocage, ce qui les rend compatibles avec de nombreuses applications nécessitant un courtier de messages fiable ou une liste circulaire. [33]

Classements de jeux

Les développeurs de jeux choisissent souvent Redis lorsqu'ils veulent concevoir des classements en temps réel. Il suffit d'utiliser la structure de données des ensembles triés Redis qui assure l'unicité des éléments tout en veillant à ce que la liste reste triée en fonction des scores de l'utilisateur. Pour créer une liste de classement en temps réel, il suffit de mettre à jour le score d'un utilisateur chaque fois qu'il change. Vous pouvez également utiliser les ensembles triés pour gérer des données en séries chronologiques en utilisant des horodatages comme score.

Gestion de sessions

Les développeurs d'application choisissent souvent Redis comme stockage de données en mémoire avec une disponibilité et une persistance élevées pour stocker et gérer les données de session pour des applications à l'échelle d'internet. Redis offre une latence inférieure à une milliseconde, ainsi que l'évolutivité et la résilience nécessaire pour gérer des données de session

telles que des profils utilisateur, des informations d'identification, un statut de session et une personnalisation spécifique à l'utilisateur.

Streaming multimédia riche

Redis offre un stockage des données rapide et en mémoire pour alimenter les cas d'utilisation de streaming en direct. Redis peut être utilisé pour stocker des métadonnées du profil et des historiques de visionnage des utilisateurs, des tokens/informations d'authentification pour des millions d'utilisateurs et des fichiers manifestes pour permettre aux CDN de diffuser des vidéos à des millions d'utilisateurs d'ordinateur et de téléphone, en même temps. [34]

Géolocalisation

Redis offre des structures et des opérateurs de données en mémoire conçus dans un but pour gérer les données géospatiales en temps réel à grande échelle et à vitesse élevée. Les commandes comme GEOADD, GEODIST, GEORADIUS et GEORADIUSBYMEMBER pour stocker, traiter et analyser les données géospatiales en temps réel rendent la géolocalisation facile et rapide avec Redis. Vous pouvez utiliser Redis pour ajouter des fonctions basées sur la géolocalisation à vos applications, telles que le temps et la distance du trajet ainsi que les points d'intérêt.

Machine Learning

Les applications modernes orientées données nécessitent du Machine Learning pour traiter rapidement du volume, de la variété et de la vitesse importants des données et automatiser la prise de décision. Pour des cas d'utilisation comme la détection de fraude dans les jeux et les services financiers, les enchères en temps réel en technologie publicitaire et le jumelage pour les rencontres amoureuses et le covoiturage, la capacité de traiter les données en temps réel et de prendre les décisions en quelques dizaines de millisecondes est de la plus haute importance. Redis vous offre un stockage de données en mémoire et rapide qui conçoit, entraîne et déploie rapidement des modèles de Machine Learning.

Les analyses en temps réel (OLAP)

Redis peut être utilisé avec des solutions de streaming telles qu'Apache Kafka et Amazon Kinesis comme stockage de données en mémoire pour intégrer, traiter et analyser des données en temps réel avec une latence inférieure à la milliseconde. Redis est un choix idéal pour les cas d'utilisation d'analyse en temps réel, tels que les analyses des réseaux sociaux, le ciblage

publicitaire, la personnalisation et l'IoT.

8. Objectif d'utilisation Redis (IMDB)

L'utilisation de cette technologie n'a pas de limite d'utilisation, elle inclut tous les domaines, mais nous l'utilisons pour : [34]

- Possibilité de travailler dans l'environnement local
- Confidentialité Des Données
- La sécurité Des Systèmes
- Réduire le coût du stockage des données dans Cloud
- Accélérer le processus de récupération des données
- Augmenter les performances des applications

9. Application

Description de l'application, des outils utilisés pour créer l'application et des techniques

9.1. Type D'application [35]

L'application est un site qui sélectionne et sert de panneau d'affichage pour les statistiques des utilisateurs de Facebook

9.2. outils [36]

Langages De Programmation

Front End

- Javascript (chart.js)
- Bootstrap
- Html&Css
- jQuery

Back End

- Php
- Redis

9.3. Implémentation :

Dans notre projet nous avons pour objectif d'extraire les données à partir d'un Cloud, mais comme nous n'avons pas pu obtenu un accès au Cloud, nous avons installé une base de données MySQL, avec la table utilisateurs et dernière connexion et table de cookies.

Nous avons exécuter la requête 1 : `SELECT user_id, age FROM public.user_fb where age BETWEEN '$age_min' AND '$age_max' ;`

Nous avons utilisé cette requete afin d'identifier les différents groupes d'age afin de compter les groupes cibles les plus importants et de les utiliser dans l'analyse. Nous avons utilisé ce type de requete pour sélectionner les données utilisées dans l'analyse.

SELECT est utilisé pour récupérer des données en général.

L'instruction SELECT copie les données d'une seule table dans une liste de la base de données Redis pour un accès facile très rapide.

L'opérateur BETWEEN définit des valeurs dans une certaine plage pour déterminer la quantité de données copiées dans la base de données Redis, permùétant d'

En précisant l'age ou la nationalité de l'utilisateur ou le sexe de l'utilisateur.

Tout cela dans le but d'augmenter les performances et de réduire les couts de stockage des données et la vitesse de traitement des données.

Après nous avons conçu une application PHP qui se connecte au SGBD, une fois la connexion établie nous avons stocké le résultat de la requête sur la table... sur le SGBD REDIS,

A chaque rafraichissement on ajoutés le résultat antérieur au résultat actualiser, ce qui représente un stockage du résultat de la requête 1, ainsi on obtenait le résultat facilement

et un temps très rapide par rapport à l'exécution de la requête sur le serveur MySQL, les résultats sont affichés dans le paragraphe suivant.

9.4. Panneau D'affichage

Page d'accueil avec statistiques générales



Figure 4.3: Diagramme de durée quotidienne vs age

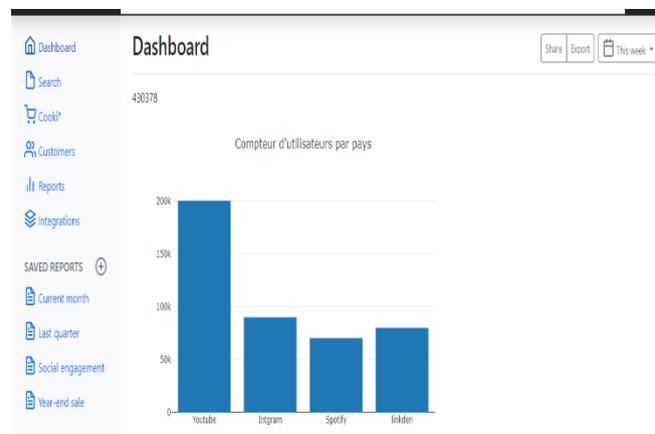


Figure 4.3: Utilisateurs par Pays Affichage Avec Redis

Deuxième page Cookies

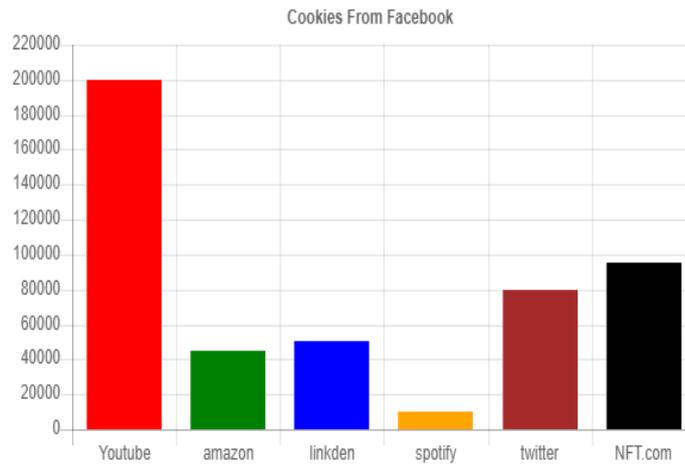


Figure 4.4: Cookies De Facebook Affichage Avec redis

Troisième page

La troisième page utilise beaucoup plus que la dernière page 11. La fonction de cette page est de rechercher et d'afficher les données des utilisateurs de Facebook au moment de l'entrée sur Facebook de l'année 2004 à 2022

Avec Deux type de Rechercher

- Rechercher Avec Select
- Rechercher Avec input

#	id	Date	Country
#	100103	2021-09-23 22:24:00.08	Saint Kitts and Nevis
#	100120	2021-08-04 23:25:00.08	Botswana
#	100470	2021-08-19 05:55:00.08	South Africa
#	100512	2021-10-12 10:50:00.08	Guam
#	100564	2021-11-18 21:23:00.08	Malaysia
#	100566	2021-09-05 06:53:00.08	New Caledonia
#	100613	2021-07-21 09:45:00.08	Solomon Islands
#	100643	2021-09-14 23:36:00.08	Bangladesh
#	100655	2021-07-18 04:15:00.08	Christmas Island
#	100830	2021-09-02 23:44:00.08	Brazil
#	100956	2021-10-14 17:34:00.08	Benin
#	101259	2021-05-18 06:47:00.08	Gabon
#	101533	2021-07-26 22:43:00.08	Malawi

Figure 4.5: Rechercher et afficher Sur Redis

Conclusion

Nous avons développé une application qui permet d'interroger une base de données extérieure et sauvegarde le résultat de la requête en local, ceci permet de réduire le temps de réponse et allège le traitement du serveur, nous avons constaté un gain important en termes d'exécution et du traitement.

CONCLUSION GENERAL

Conclusion générale :

Nous avons créé un système qui permet de sauvegarder une vue matérialisée sur mémoire locale, grâce à l'association des ressources des appareils de l'utilisateur en l'occurrence les tablettes, smartphone, laptop.

L'association des ressources des utilisateurs permet d'augmenter les performances des serveurs et de réduire le cout d'exploitation des SGBD extérieur ce qui énormément réduit la facture sur cloud et allège les traitement des serveurs.

BIBLIOGRAPHIE

- [01] [http://WWW.ASTERA.COM/FR/TYPE/BLOG/UN-APEr%c3%a7u-rapid-des-diff%c3%a9rents-types-de - base s-de -donnee%c3%a9es/](http://WWW.ASTERA.COM/FR/TYPE/BLOG/UN-APEr%c3%a7u-rapid-des-diff%c3%a9rents-types-de-base-s-de-donnee%c3%a9es/)
- [02] <http://www.emi.ac.ma./ntounsi/cours/db/polys/bd1-intro.pdf>
- [03] [http://www.oracle.com/ch-fr/database/what-is-database/ LINK5](http://www.oracle.com/ch-fr/database/what-is-database/)
- [04] <http://sql.sh/sghd>
- [05] <https://tel.archives-ouvertes.fr/tel-00348094>
- [06] https://www.tenable.com/products/tenable-cs?utm_campaign=gs-{17232829325}-{135323757046}-{597239359155}_00024154&utm_promoter=tenable-ent-00024154&utm_source=google&utm_term=cloud%20computing&utm_medium=cpc&utm_geo=emea&gclid=EAIaIQobChMI7NrU_oIQ-gIVSYODbx0gagfnEAAYASAAEgJlqPD_BwE
- [07] <https://www.netapp.com/hybrid-cloud/what-is-infrastructure-as-a-service-iaas/>
- [08] <https://www.techtarget.com/searchcloudcomputing/definition/Platform-as-a-Service-PaaS>
- [09] <https://www.techtarget.com/searchcloudcomputing/definition/Software-as-a-Service>
- [10] https://www.researchgate.net/publication/354683300_OVERVIEW_OF_CLOUD_COMPUTING_AND_ITS_TYPES
- [11] <https://www.figer.com/Publications/nuage.htm>
- [12] [https://www.europarl.europa.eu/RegData/etudes/etudes/join/2012/475104/IPOL-IMCO_ET\(2012\)475104_FR.pdf](https://www.europarl.europa.eu/RegData/etudes/etudes/join/2012/475104/IPOL-IMCO_ET(2012)475104_FR.pdf)
- [13] <https://www.lebigdata.fr/olap-online-analytical-processing>
- [14] <https://www.lebigdata.fr/olap-online-analytical-processing>
- [15] <https://www.lebigdata.fr/olap-online-analytical-processing>
- [16] <https://docs.microsoft.com/fr-fr/analysis-services/multidimensional-models-olap-logical-cube-objects/cube-storage-analysis-services-multidimensional-data?view=asallproducts-allversions>
- [17] <https://hal.archives-ouvertes.fr/tel-02519803/document>
- [18] <https://dommaster18.ru/fr/how-to-fix/v-olap-analyze-ispolzuyutsya-kursovyaya-rabota-tehnologiya-olap-olap/>
- [19] <https://www.linkedin.com/pulse/le-cloud-est-futur-mais-quel-du-%C3%A9pisode-1-laurent-allard>
- [20] <http://orm.bdpedia.fr/introjpa.html>
- [20] <http://architectures-web.smile.fr/Le-cache/Cache-de-données>
- [21] Site officiel de gandin, <http://www.gandi.net/>, documentation en ligne, 2011.
- [22] <https://www.lemagit.fr/definition/Snapshot-Stockage>

- [23] N. Mai, K. Aouiche et J. Darmont, Sélection simultanée d'index et de vues matérialisee, Laboratoire ERIC, Université Lumière Lyon2, 2006.
- [24] N. DURAND, introduction à des notions avancées (Index, Déclencheurs, Transactions), Marseille, conférence, 2013.
- [25] J. Revaille, Les vues et les indexes, livre, 2009.
- [26] L. Bella rèche, techniques d'optimisation des requêtes dans les data Warehouse, ENSMA Université de Poitiers France, 2003.
- [27] <https://support.microsoft.com/fr-fr/windows/activer-les-mises-%C3%A0-jour-automatiques-des-applications-70634d32-4657-dc76-632b-66048978e51b>
- [28] Hakim MADI, Conception et réalisation d'une base de données répartie sous oracle : cas de l'hébergement des résidences 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] <https://www.linkedin.com/pulse/redis-my-personal-imdb-feat-nodejs-part-iii-harshavardhan-deekeswar>
- [31] <https://www.ionos.fr/digitalguide/hebergement/aspects-techniques/quest-ce-que-redis/>
- [32] <https://aws.amazon.com/fr/redis/>
- [33] <https://redis.com/solutions/use-cases/caching/>
- [34] Kamel AOUCHE, Techniques de fouille de données pour l'optimisation automatique des performances des entrepôts de données, thèse, Université Lumière Lyon 2, décembre 2005.
- [35] Boly Aliou, fonctions d'oubli et résumés dans les entrepôts de données, thèse, l'école nationale supérieure des télécommunications de paris, décembre 2006
- [36] [Aws.amazon.com/fr /redis test_elasticache for raides avantage](https://aws.amazon.com/fr/redis/test_elasticache_for_raides_avantage)