



Rodney Badran
2017-2018

TABLE DES MATIERES

- I. INTRODUCTION - NOSQL**
- II. MONGODB**
- III. ARCHITECTURE NEXUS**
- IV. SHARDING**
- V. REPLICATION**
- VI. COMPARAISON ENTRE SQL ET NOSQL**
- VII. CONCLUSION**
- VIII. INSTALLATION DE MONGODB**
- IX. INSTALLATION DE MONGODB COMPASS**
- X. COMMNADES DE MONGODB**

I. INTRODUCTION - NOSQL

Les bases relationnelles ont été inventées par Edgar Codd en 1970, mais au début des années 2000, les organisations ont rencontré beaucoup de problèmes:

- ✓ Difficultés pour satisfaire les besoins des applications Web à grande échelle : bdd relationnelles (RDBMS) et l'application est écrite avec un langage orienté objet.
- ✓ Le VOLUME des données créées double tous les 2 ans.
- ✓ La VARIÉTÉ des types de données créées.
- ✓ La VÉLOCITÉ avec laquelle les données changent est également très importante (comme le IOT).
- ✓ D'où la naissance de NoSql avec Google (BigTable), Amazon (Dynamo), Facebook (Cassandra), SourceForge.net (MongoDB).

I. INTRODUCTION – NOSQL TYPES (2)

Nom Type (Français)	Bdd Entrepôts Clé- Valeurs ECV	Bdd Orientées Documents	Bdd Orientées Colonnes	Bdd Orientées Graphes
(Anglais)	Key Value Stores	Document DB	Wide-Column	Graph DB
Définition	Les données sont stockées en clé-valeur: une clé plus un blob.	Ces bdd stockent des données semi-structurées: le contenu est formaté json ou XML, mais la structure n'est pas contrainte.	Ces bdd se rapprochent des bdd relationnelles comme elles permettent de remplir un nombre de colonnes variable	ces bdd sont basées sur la théorie des graphes, sont gérées par nœuds, relations et propriétés. utiles pour des données spatiales ou financières.
Exemple	<ul style="list-style-type: none"> . Redis . Oracle Berkeley DB 	<ul style="list-style-type: none"> . CouchDB . MongoDB 	<ul style="list-style-type: none"> . Cassandra . Google Bigtable 	<ul style="list-style-type: none"> . neo4j
Pour	<ul style="list-style-type: none"> . Facilement scalable . temps de réponse en écriture/ lecture très bas 	<ul style="list-style-type: none"> . Requêtage plus complet . Flexibilité . évolutif au cours du temps 	<ul style="list-style-type: none"> . Capacité de stockage accrue . Accès rapide aux données . évolutif au cours du temps 	<ul style="list-style-type: none"> . Adapté à la gestion de données relationnelles . architecture modelable
Contre	<ul style="list-style-type: none"> . Mise a jour compliquée . requêtes limitées 	<ul style="list-style-type: none"> . Cohérence pas forcément assurée . duplication des données 	<ul style="list-style-type: none"> . Efficace pour des données de même type et similaires . Requêtage limite 	<ul style="list-style-type: none"> . architecture limitée à certains cas

II. MongoDB - INTRODUCTION

MongoDB est une bdd open source qui utilise un modèle de données orienté document. L'un des nombreux types de bases de données qui se posent au milieu des années 2000 sous la bannière NOSQL.

MongoDB est programmé en C++.

Le nom de MongoDB dérive du mot « humongous » pour représenter l'idée de supporter de grandes quantités de données (Big Ideas).

Son logo, la feuille d'arbre, représente « simple et naturel ».

Merriman et Horowitz ont formé 10Gen Inc. en 2007 pour commercialiser MongoDB.

La société a été rebaptisée MongoDB Inc. en 2013 à New York.

II. MongoDB (2) - OBJECTIF

MongoDB est une bdd généraliste appréciée pour:

- ✓ **Assurer une Scalabilité face aux changements dynamiques et rapides des données.**
- ✓ **Assurer une haute performance et flexibilité pour les types de données.**
- ✓ **Un besoin de redondance et de continuité de service.**
- ✓ **Une répartition géographique des données (Distribution/Partition).**
- ✓ **Un besoin de système de recherche, de stockage de fichier ou d'un système information géographique.**

II. MongoDB (3) - Architecture technique

MongoDB est construit sur une architecture de collections et de documents, similaire à Json, sans schéma prédéfini.

<u><i>RDBMS</i></u>	<u><i>MongoDB</i></u>
Database	Database
Table	Collection
Tuple/Row	Document
Column	Field

II. MongoDB (4) – Json

- ✓ JavaScript Object Notation ou Json est un format de données textuelles (et pas un langage informatique) dérivé de la notation des objets du langage JavaScript qui permet de représenter de l'information structurée comme le permet XML. Son format est « .json ».
- ✓ Il représente l'information accompagnée d'étiquettes permettant d'en interpréter les divers éléments, sans aucune restriction.
- ✓ Un document JSON ne comprend que deux types d'éléments structurels :
 - Des ensembles de paires "nom" (ou clé) / "valeur« ,
 - Des listes ordonnées de valeurs,
 - {...} : les accolades définissent un objet ,
 - "clé":"valeur" : Les guillemets et les double-points définissent un couple clé/valeur (on parle de membre).
 - [...] : Les crochets définissent un tableau (array) ,
 - Les valeurs sont séparées par virgule.
- ✓ L'avantage de Json sur XML est sa vitesse de traitement et sa simplicité de mise en œuvre étant reconnu nativement par JavaScript. Comme il n'a pas beaucoup de types comme XML, il est moins compliqué que le dernier.

Json a une syntaxe courte à connaître, il est moins verbeux, donc plus léger que le XML tout en restant lisible par les machines et les ordinateurs.

II. MongoDB (5) – Bson

- ✓ **Mongo DB est décomposée en une série de fichiers BSON sur le disque dur, avec une taille croissante allant jusqu'à 2Go.**
- ✓ **BSON est construit spécifiquement par MongoDB pour stocker les données sur le disque en tant que BSON dans le répertoire de chemin de données, qui est habituellement /data/db.**
- ✓ **BSON est une extension du modèle JSON pour donner d'autres types de données, et est efficace pour codage et décodage avec des différents langages.**
- ✓ **Les types BSON sont nominalement un ensemble plus grand de types JSON (JSON n'a pas de type date, à l'exception de ne pas avoir un type « Number » universel comme le fait JSON).**
Les objets Date sont stockés sous la forme d'un entier signé 64 bits représentant le nombre de millisecondes depuis l'époque Unix (1er janvier 1970).
- ✓ **Comparé à JSON, BSON est conçu pour être efficace à la fois dans l'espace de stockage et la vitesse de numérisation (storage, space et scan-speed).**

II. MongoDB (6) – Exemple

Modèle de données basé sur des documents

Base de données relationnelle

PERSON

Pers_ID	Surname	First_Name	City
0	Miller	Paul	London
1	Ortega	Alvaro	Valencia
2	Huber	Urs	Zurich
3	Blanc	Gaston	Paris
4	Bertolini	Fabrizio	Rome

CAR

Car_ID	Model	Year	Value	Pers_ID
101	Bentley	1973	100000	0
102	Rolls Royce	1965	330000	0
103	Pugeot	1993	500	1
104	Ferrari	2005	150000	4
105	Renault	1998	2000	1
106	Renault	2001	7000	3
107	Smart	1999	2000	2

NO RELATION



MongoDB

```
{
  first_name: 'Paul',
  surname: 'Miller',
  city: 'London',
  location:
[45.123,47.232],
  cars: [
    { model: 'Bentley',
      year: 1973,
      value: 100000, ... },
    { model: 'Rolls Royce',
      year: 1965,
      value: 330000, ... }
  ]
}
```

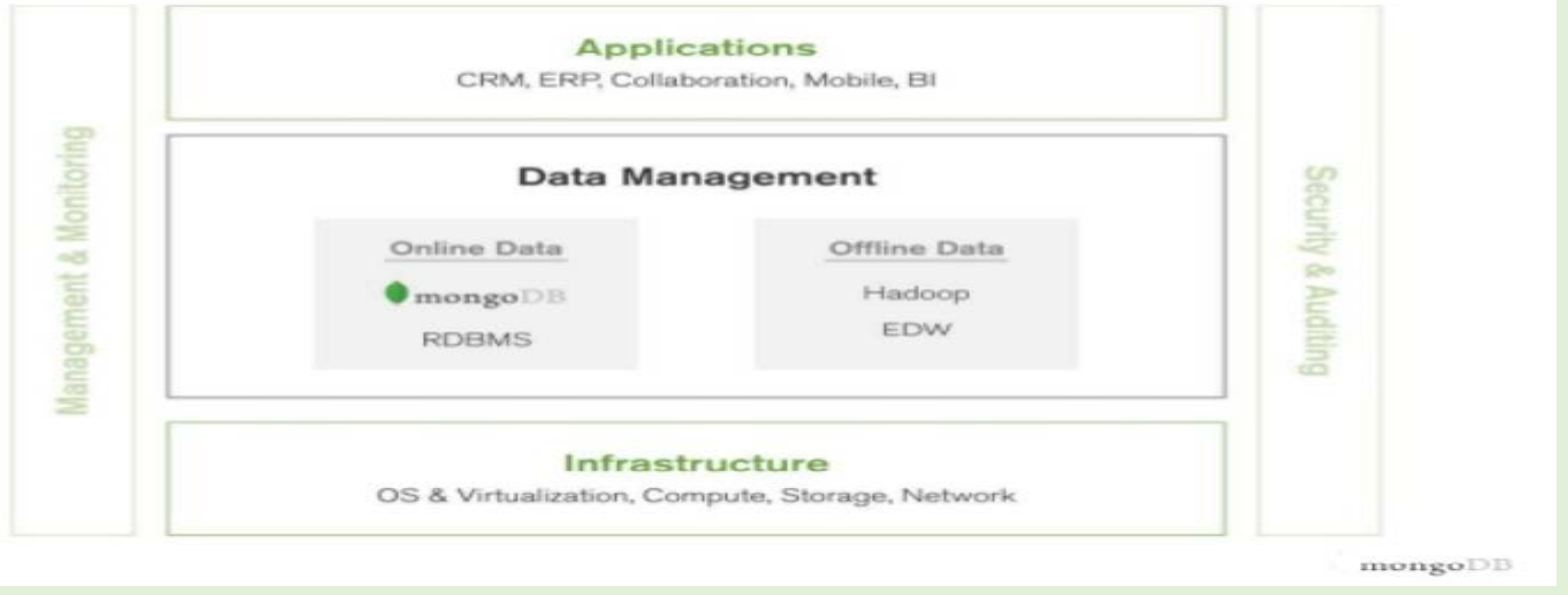
II. MongoDB (6) – N'est Pas

MongoDB n'est pas :

- ✓ **Une suite analytique**
- ✓ **Une technologie d'entrepôt de données**
- ✓ **Un outil de BI**
- ✓ **Traitement des transactions de service comptable**
- ✓ **Serveur principal d'un système de facturation ou d'un système de comptabilité générale**
- ✓ **Un moteur de recherche**

II. MongoDB (8) – Est

MongoDB et le stack informatique d'entreprise

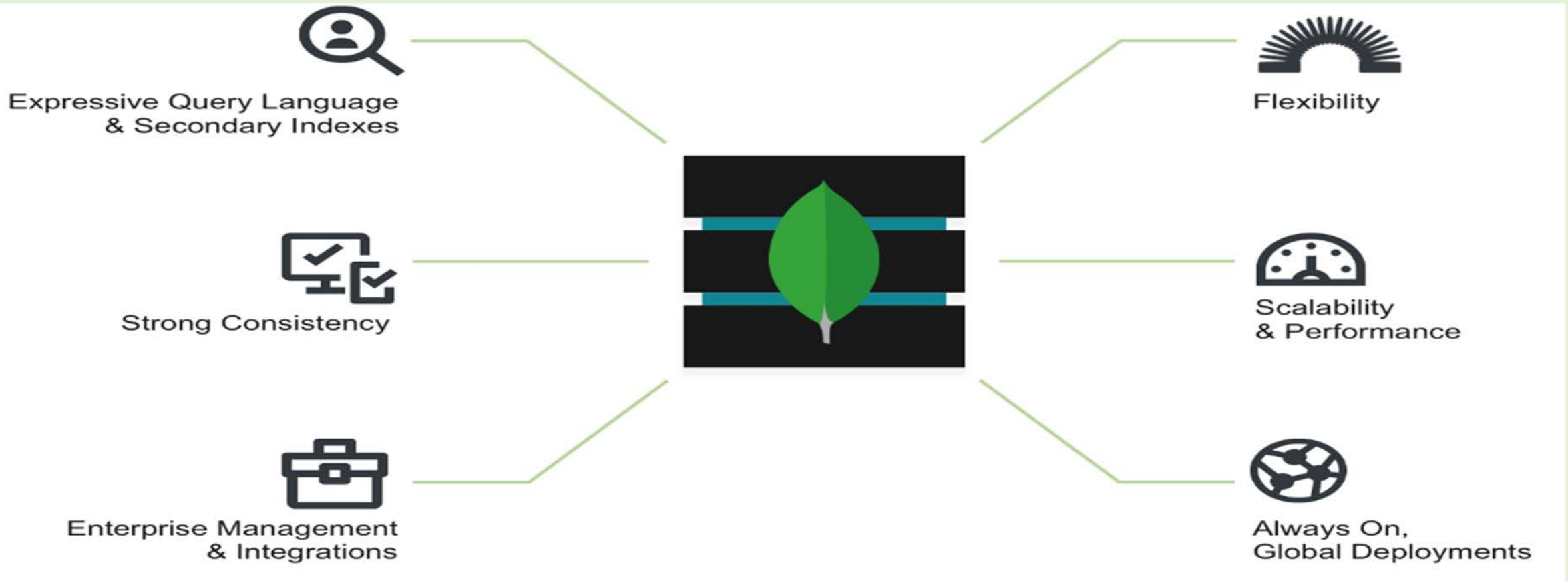


III. ARCHITECTURE NEXUS

RDBMS Advantages

NOSQL Advantages

MongoDB



III. ARCHITECTURE NEXUS (2) - RDBMS

- ✓ Les utilisateurs peuvent accéder et manipuler leurs données de manière sophistiquée pour prendre en charge les applications opérationnelles et analytiques. Et les index jouent un rôle essentiel dans la fourniture d'un accès efficace aux données supportés par la bdd plutôt que maintenus dans le code de l'application.
- ✓ Les applications doivent pouvoir lire immédiatement ce qui a été écrit dans la bdd.
- ✓ Les bdd font partie de l'infrastructure de l'application, et doivent s'intégrer parfaitement dans l'organisation informatique de l'entreprise. Les entreprises ont besoin d'une bdd qui peut être sécurisée, surveillée, automatisée et intégrée à leur infrastructure technologique à leurs processus et à leur personnel.

III. ARCHITECTURE NEXUS (3) - NOSQL

- ✓ **Un modèle de données flexible permet de stocker et de combiner facilement des données de n'importe quelle structure et de permettre une modification dynamique du schéma sans interruption.**
- ✓ **Le partitionnement de NoSQL permet aux bdd d'évoluer sur du matériel standard déployé sur site ou dans le cloud permettant une croissance quasi illimitée avec un débit plus élevé et une latence plus faible que les bdd relationnelles.**
- ✓ **Les bdd NoSQL sont conçues pour les systèmes hautement disponibles qui offrent une expérience cohérente, de haute qualité aux utilisateurs avec l'exécution sur plusieurs nœuds.**

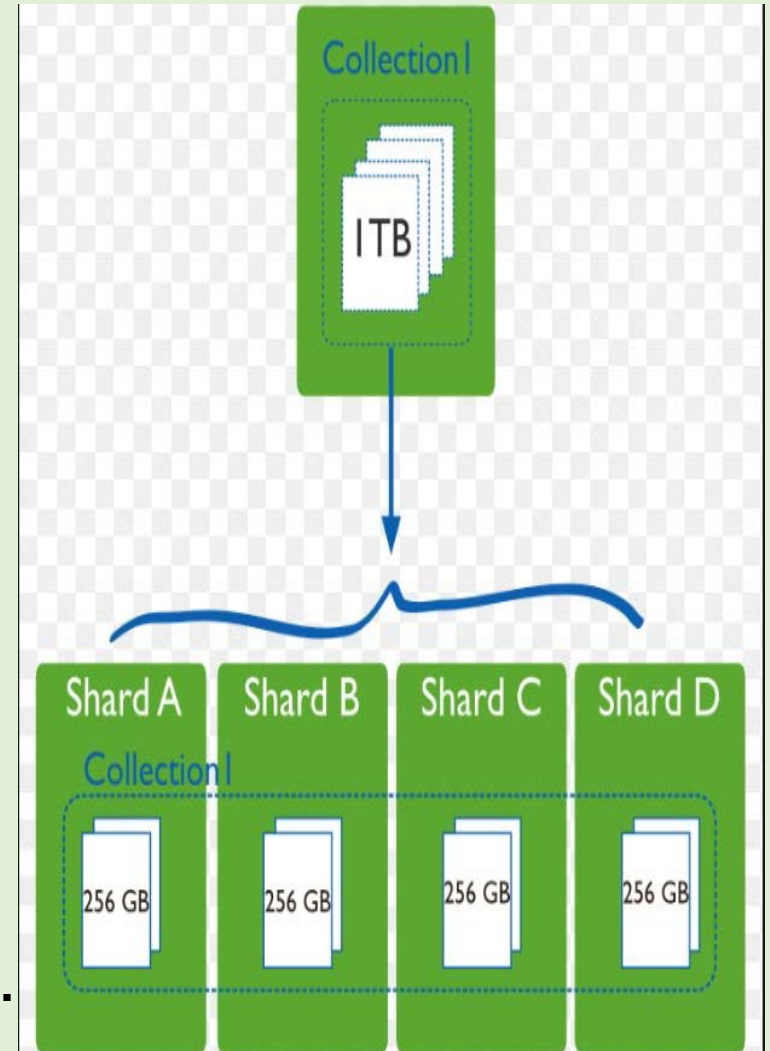
IV. SHARDING

Le mot shard signifie une petite partie d'un tout.

MongoDB divise la bdd en plusieurs bdd plus petites qui ne partagent rien et peuvent être réparties sur plusieurs serveurs.

Comme la taille des données augmente, une seule machine peut ne pas être suffisante pour stocker les données ni fournir un débit de lecture et d'écriture acceptable, alors Sharding résout le problème avec la mise à l'échelle horizontale (Horizontal Scaling) : on ajoute des machines pour soutenir la croissance des données et les exigences des opérations de lecture et d'écriture.

Le Sharding de MongoDB est synonyme de partitionnement horizontal.



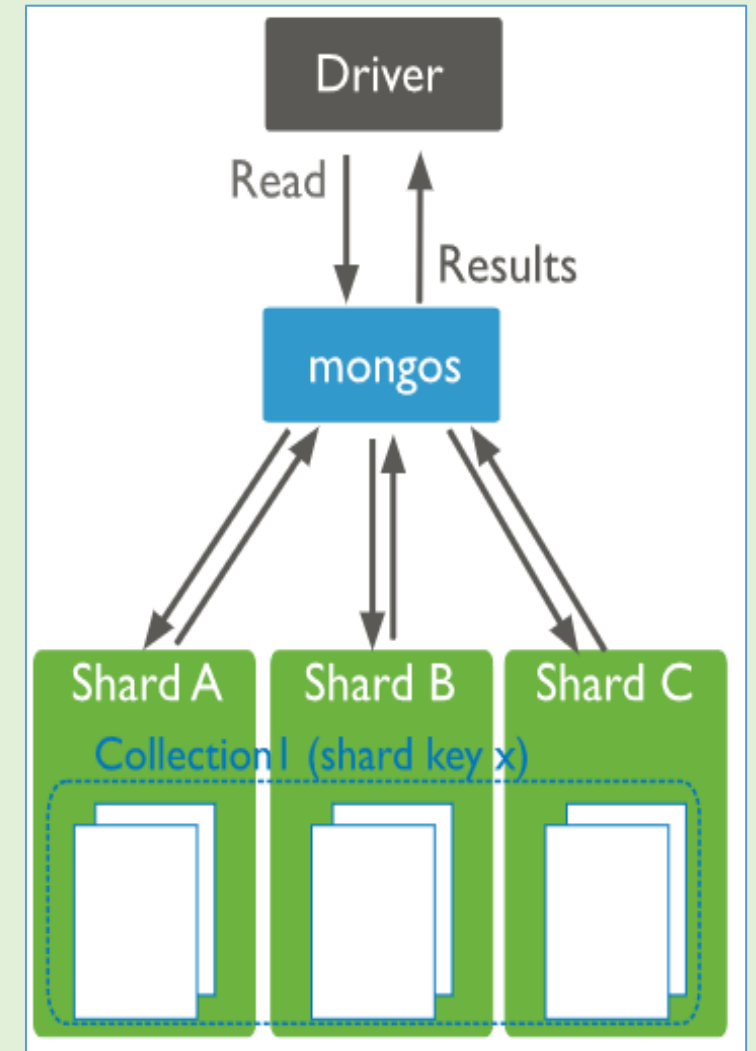
IV. SHARDING (2)

MongoDB utilise Mongos pour son sharding:

“MongoDB Sharded Cluster Query Router”

ou MongoDB Shard, est un service de routage pour les configurations de partitionnement MongoDB qui traite les requêtes de la couche application, et détermine l'emplacement de ces données dans le cluster fragmenté, afin de compléter ces opérations.

Du point de vue de l'application, une instance de mongos se comporte de la même manière que n'importe quelle autre instance de MongoDB.

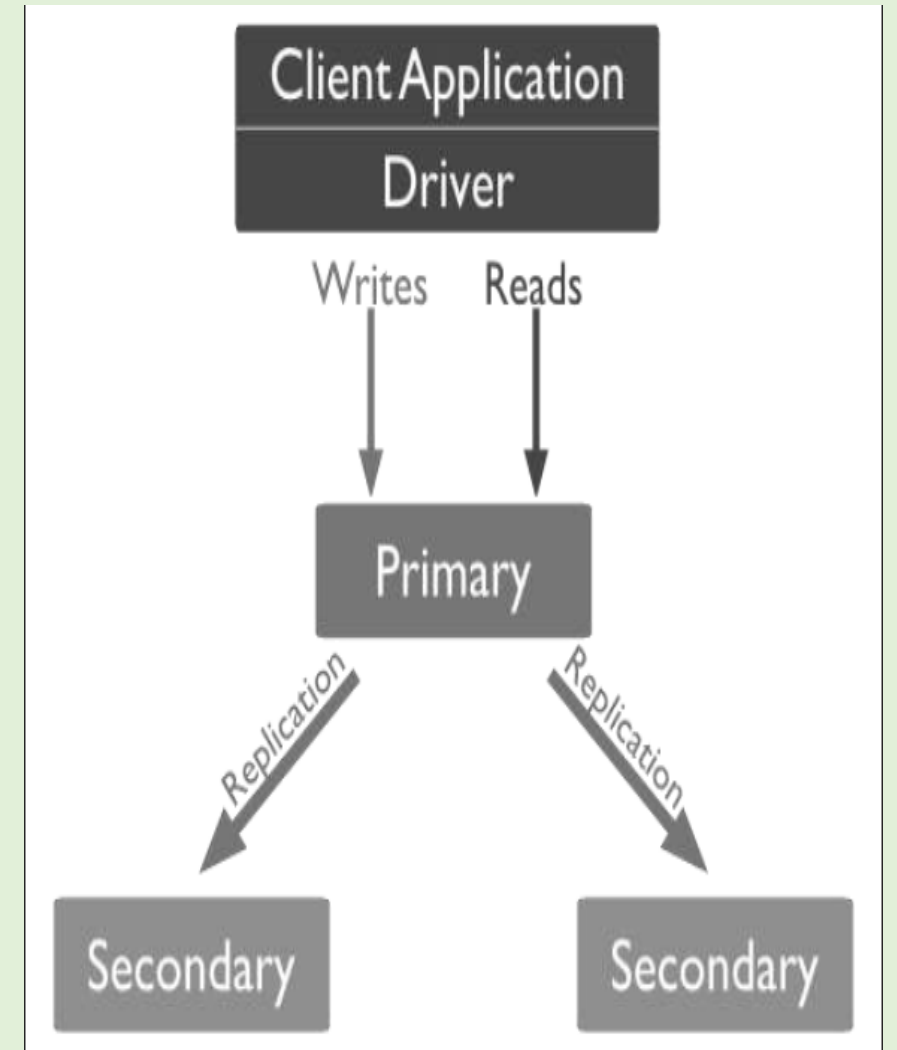


V. REPLICATION

La réplication consiste à écrire les données sur plusieurs serveurs.

Elle est connue sous « replica set » dans MongoDB, et sert à garder les données en toute sécurité en préservant une haute disponibilité des données (High Availability, 24 * 7) avec un plan de reprise d'activité (Disaster recovery).

Replica Set possède les membres suivants :
Primary, Secondary, Priority 0, Hidden

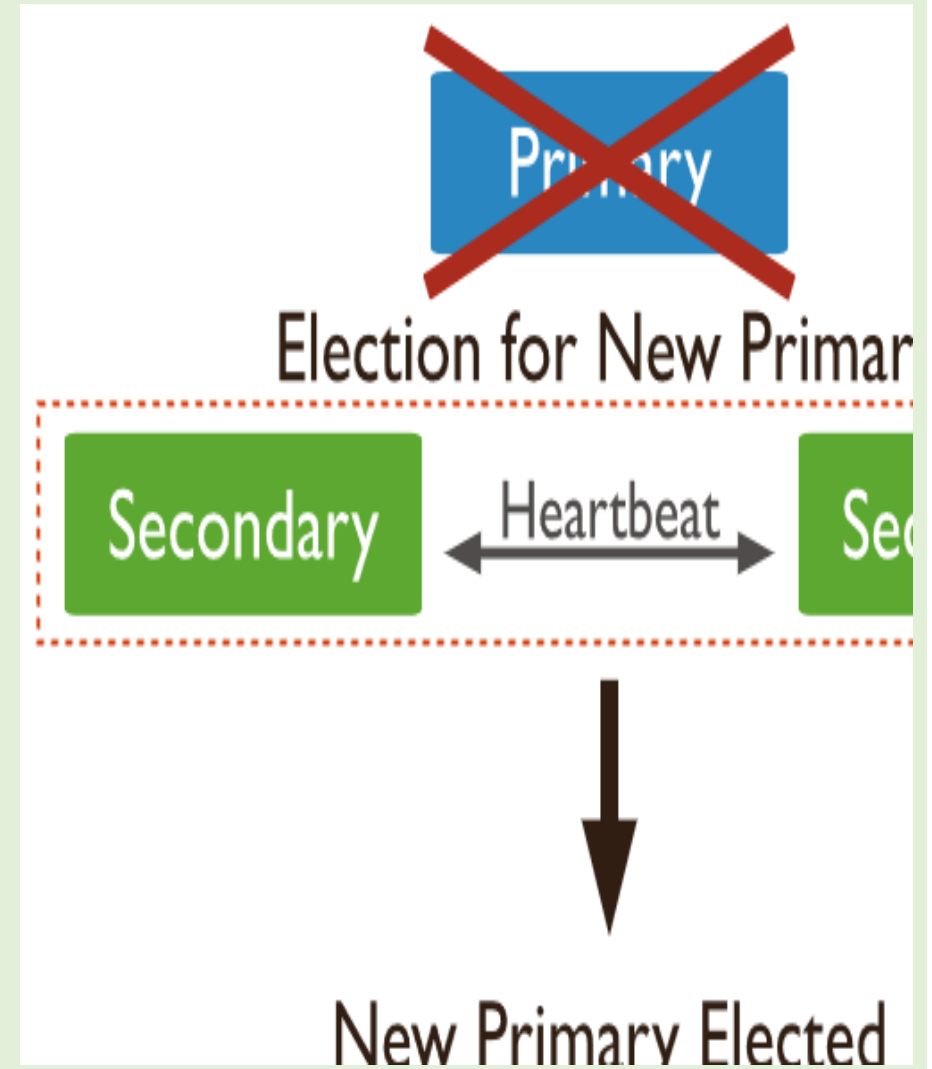


V. REPLICATION (2)

En général, un minimum de 3 nœuds sont requis :
1 primary et 2 secondary.

Dans le cas d'échec du primary, "élection" établit un nouveau primary, et un secondary est élu primaire.

Après la récupération de l'ancien primary, il rejoint le « replica set » comme un nœud secondary.



VI. COMPARAISON SQL ET NOSQL

✓ Structure de Données SQL vs NOSQL

- SQL organise le stockage de données sur le principe de tables reliées entre elles, dans un schéma prédéfini ; la structure et les types des données sont fixés à l'avance.
- NOSQL stocke et manipule des documents qui correspondent à des collections d'objets sans schéma prédéfini. NOSQL est plus flexible et pardonnable, mais la possibilité de stocker des données n'importe où peut entraîner des problèmes de cohérence.

✓ Intégrité de Données SQL vs NOSQL

- SQL permet d'appliquer des règles d'intégrité de données à l'aide de contraintes de clés étrangères.
- Par contre, MongoDB acceptera tout ce qui est fourni.

VI. COMPARAISON SQL ET NOSQL (2)

✓ La Logique de Jointure SQL vs NOSQL

- Les requêtes SQL offrent une puissante clause JOIN.
- NOSQL en général, et Mongo DB en particulier, n'ont pas toujours d'équivalent de JOIN.

✓ La Normalisation SQL vs La Dé-Normalisation NOSQL

- La normalisation de SQL revient à la façon dont les données sont pas reliées par des clés étrangères, et pas dupliquées.
- Par contre, dans NOSQL, les requêtes sont plus rapides, mais la mise à jour des données est plus lente.

✓ Performance SQL vs NOSQL

Ce n'est pas surprenant car Le principe de « dé-normalisation » induit une représentation plus simple et permet donc de récupérer toutes les informations sur un élément spécifique dans une seule requête.

Il n'y a donc pas besoin de liens JOIN ou de requêtes SQL complexes.

Mais la redondance des informations alourdit considérablement les opérations de mise à jour.

VI. COMPARAISON SQL ET NOSQL (3)

✓ Transaction SQL vs Transaction NOSQL

Dans les bases de données SQL, plusieurs mises à jour peuvent être exécutées au sein d'une même transaction afin de garantir le succès ou l'échec de l'exécution du code SQL.

Dans une bdd NOSQL, la modification d'un document unique est atomique.

✓ Propriétés d'ACID

SQL est conforme ACID, tandis que MongoDB est compatible ACID au niveau du document.

Ce que MongoDB ne possède pas, ce sont les transactions, c'est-à-dire les mises à jour de plusieurs documents qui peuvent être annulées et qui sont conformes à l'ACID.

✓ Propriétés de CAP

MongoDB est CP : quand une crise arrive et MongoDB doit décider quoi faire, il choisira la cohérence sur la disponibilité. Il cessera d'accepter les écritures système jusqu'à ce qu'il croit qu'il peut en toute sécurité compléter ces écritures.

VI. COMPARAISON SQL ET NOSQL (4)

✓ SQL vs NOSQL dans les Nouvelles Technologies

Le « load balancing » sur un serveur SQL est complexe car dû à la nature même dont les données sont stockées, organisées et reliées entre elles, en plus licences et coût élevé.

L'organisation des données en documents et la « dé-normalisation » des collections permettent le partitionnement, et le déploiement dans le Cloud est plus simple et facile à gérer, ce qui conduit à une croissance pratiquement illimitée.

A ce titre, les bdd NOSQL sont résolument orientée “Big Data”.

En plus, la représentation des données en collection et le résultat en flux JSON des requêtes permet de consommer les données rapidement et facilement par les applications front (web, mobile).

VII. CONCLUSION

MongoDB est une base NOSQL avec un syntaxe léger, facile à comprendre, simple à installer et à utiliser, et puissant.

Bdd Scalable, flexible, performante et basée sur Json.

Adoptée pour répondre à différents besoins, d'où une grande diversité de fonctionnalités, comme le « Big Data » et le « Cloud »: une vaste gamme de cas d'utilisation.

MongoDB pourrait être la bdd la plus avantageuse parmi toutes les bases de données à l'avenir, étant déjà le numéro un dans les bdd NOSQL, 5ème parmi toutes les bdd, et reconnu par Gartner et Forester : il est devenu la norme de facto pour les bases de données de nouvelle génération.

VIII. INSTALLATION DE MONGODB

INSTALLATION DE MONGODB

IX. INSTALLATION DE MONGODB COMPASS

INSTALLATION DE MONGODB COMPASS

X. COMMNADES DE MONGODB

COMMNADES DE MONGODB