Data Science - 2024

NoSQL for Big-Data

Intro à MongoDB

Master Data Science

Last update: Déc. 2024

Pourquoi le NoSQL ?

- Les bases de données NoSQL sont conçues pour répondre aux limites des bases de données relationnelles (SQL).
- **Caractéristiques de NoSQL :**
  - Modèle flexible : Pas de schéma rigide.
  - Haute performance pour les lectures/écritures massives.
  - Scalabilité horizontale : Ajout de serveurs pour gérer plus de données.
- **Exemples d'utilisations :**
  - Applications en temps réel (réseaux sociaux, IoT).
  - Gestion de données semi-structurées ou non structurées.
  - Gestion de larges volumes de données, comme dans les applications Big Data.

Et pourquoi pas SQL ?

- Quand utiliser SQL : Systèmes bancaires, ERP.
- Quand utiliser NoSQL : Applications sociales, Big Data, analyses en temps réel.

SQL	NoSQL
Modèle relationnel (tables)	Modèle non relationnel (documents, graphes, etc.)
Schéma strict (colonnes/contraintes)	Schéma flexible, JSON-like
Transactions complexes (ACID)	Transactions simples (BASE souvent suffisant)
Scalabilité verticale	Scalabilité horizontale
Idéal pour des relations complexes	Idéal pour des données volumineuses et non structurées

Importance de MongoDB dans le Big-Data

- MongoDB est conçu pour traiter d'énormes volumes de données non structurées
- **Avantages dans le Big Data :**
  - Manipulation de données JSON/BSON, facilement intégrables dans les pipelines Big Data.
  - Compatible avec des outils comme Hadoop et Spark.
  - Sharding natif pour distribuer les données sur plusieurs nœuds.
  - Performances élevées pour les requêtes massives ou complexes.

- **Cas d'utilisation :**
  - Analyse des données clients.
  - Gestion des logs.
  - Systèmes recommandateurs.

Architecture de MongoDB

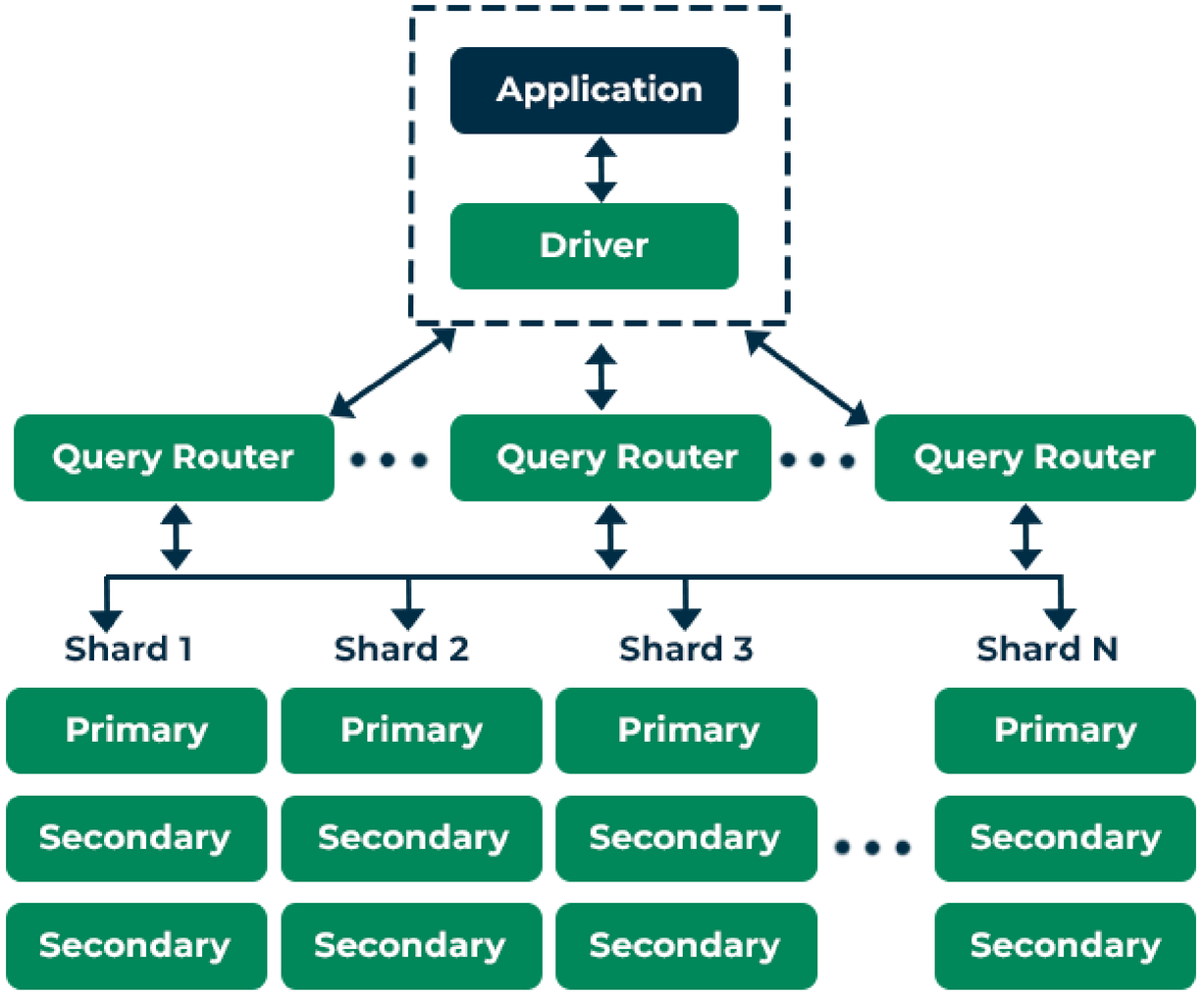
Architecture à trois niveaux :

- Application: définit les couches applicatives en dessous (on est là-bas)
- Réseau (permet l'interaction entre les couches d'en bas et d'en haut)
- Persistance (stockage des infos sur des serveurs physiques)

MongoDB Architecture

A Computer Science portal for geeks. It contains well written, well thought and well explained computer science and programming articles,...

GeeksforGeeks / Oct 10



Concepts fondamentaux - Collection

- Une collection dans MongoDB est un regroupement de documents
- Equivalent à une table en SQL
- **Caractéristiques :**
  - Aucune contrainte de schéma rigide.
  - Chaque collection peut contenir des documents avec des structures différentes.
- **Exemple :**
  - `db.createCollection("utilisateurs")`

- **Ajout de documents dans une collection :**
  - `db.utilisateurs.insertOne({ nom: "Alice", age: 25 })`

Concepts fondamentaux - Document

- Les documents sont les éléments de base de MongoDB, représentés en JSON (JavaScript Object Notation).

- **Exemple de document :**
  - {"nom": "Alice","age": 25,"tags": ["développeur", "MongoDB"]}

- **Avantages des documents :**
  - Représentation naturelle des données.
  - Flexibilité pour stocker des structures complexes (listes, objets imbriqués).
  - Facilite les modifications et les ajouts.

Concepts fondamentaux - Indexation

- Les index permettent d'améliorer les performances des requêtes en réduisant le temps de recherche.

- **Création d'un index simple :**
  - db.utilisateurs.createIndex({ nom: 1 }) // Index sur le champ "nom"

- **Index composés :**
  - Utilisés pour accélérer les recherches basées sur plusieurs champs.
  - db.utilisateurs.createIndex({ nom: 1, age: -1 }
-
- **Impact des index :**
  - Améliore les performances des lectures.
  - Augmente le coût en stockage.

Requêtes et filtres

- **Requêtes basiques :**
  - db.utilisateurs.find({ nom: "Alice" })
- **Opérateurs avancés :**
  - $gt, $lt : Supérieur, inférieur.
  - $in : Correspondance avec une liste.
  - **Exemple :**
    - db.utilisateurs.find({ age: { $gt: 20 } })
- **Projection :**
  - Limiter les champs renvoyés.
    - db.utilisateurs.find({}, { nom: 1, _id: 0 })

Scalabilité – Sharding

- MongoDB permet de distribuer les données sur plusieurs nœuds pour une meilleure scalabilité.

- Concepts clés :
  - Shard : Une portion de données.
  - Cluster : Ensemble de shards gérés par MongoDB.
  - Mongos : Routeur des requêtes pour le cluster.

- Exemple de sharding :
  - sh.enableSharding("maBase")
  - sh.shardCollection("maBase.utilisateurs", { "nom": 1 })

Sécurité dans MongoDB

- Authentification : Ajout d'utilisateurs avec rôles spécifiques.

```
db.createUser({
        user: "admin",
        pwd: "password",
        roles: ["readWrite", "dbAdmin"]
})
```

- SSL/TLS : Chiffrement des connexions.
    - Configuration de Bind IP :
    - Limiter les connexions aux adresses IP autorisées :
        - net:bindIp: 127.0.0.1

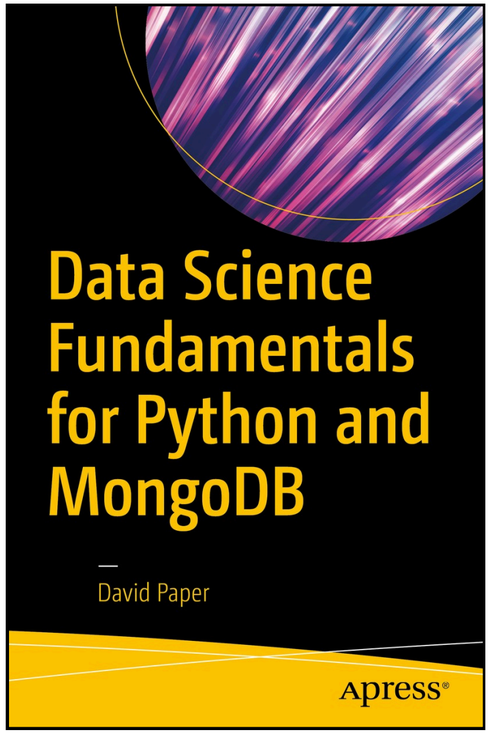
Utilisation de MongoDB

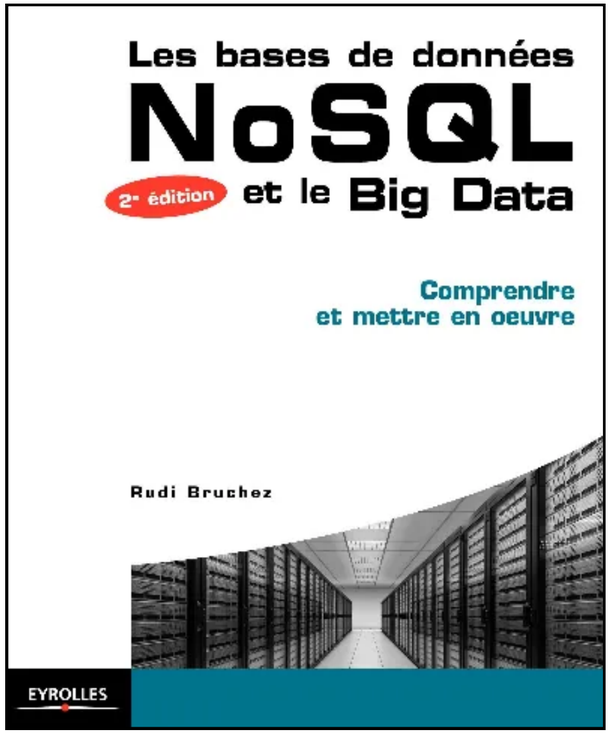
- MongoDB est utilisé dans des domaines d'applications variés :
  - E-commerce : Gestion des catalogues de produits.
  - Réseaux sociaux : Stockage des messages, relations utilisateurs.
  - IoT : Traitement des données des capteurs.

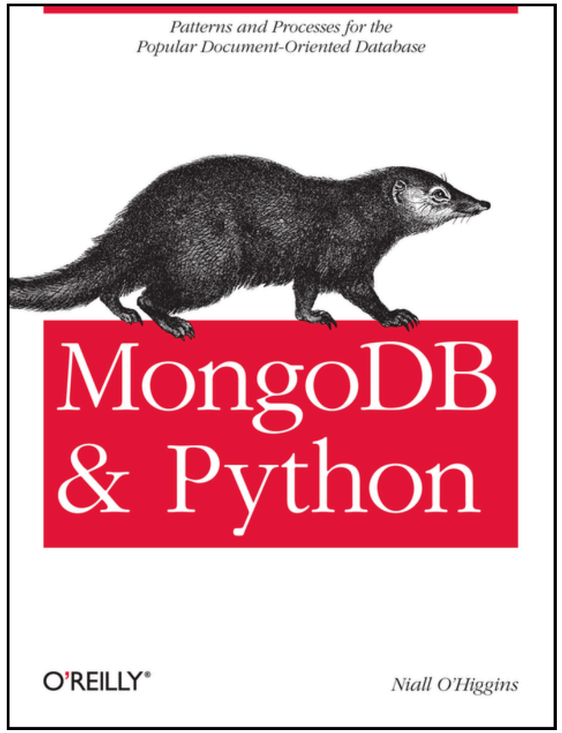
- **Utilisé le plus souvent comme Backend :**
    - Framework : Express.js ou Flask.
    - Base de données : MongoDB.
  - Frontend :
    - Framework : React ou Vue.js.
  - Workflow :
    - Stockage des données dans MongoDB.
    - Requêtes via des APIs pour affichage en temps réel.

Documentation officielle et lectures

Documentation officielle : https://www.mongodb.com/docs/







Quelques liens utiles :

- https://www.mongodb.com
- https://learn.mongodb.com/
- https://www.mongodb.com/docs