# Introduction aux nouvelles technologies Big Data

**Durée estimée :** 30 minutes

## Introduction

Dans le monde en constante évolution du big data, de nouvelles technologies émergent pour répondre au besoin croissant d'un traitement des données plus rapide, plus évolutif et plus efficace. Les systèmes traditionnels comme Hadoop et Spark ont été l'épine dorsale du traitement du big data, mais les technologies modernes telles qu'Apache Pulsar, Apache Druid et PrestoDB gagnent en popularité pour leurs capacités uniques. Cette lecture explorera ces technologies et comparera leurs fonctionnalités à Hadoop et Spark, vous offrant ainsi une compréhension plus large du paysage actuel du big data.

## Objectifs

À la fin de cette lecture, les apprenants seront capables de :

1. Expliquer les fonctionnalités principales et les cas d'utilisation des technologies Big Data modernes
2. Différencier les solutions traditionnelles et émergentes
3. Identifier les scénarios spécifiques dans lesquels chaque technologie (Pulsar pour la messagerie et le streaming, Druid pour l'analyse en temps réel et PrestoDB pour les requêtes distribuées) est la mieux adaptée
4. Évaluer les compromis en termes d'évolutivité, de performances et de latence de ces technologies émergentes par rapport aux systèmes Big Data traditionnels
5. Déterminer la pile technologique adaptée aux applications du monde réel

---

## 1. Apache Pulsar

### Aperçu

Apache Pulsar est une plateforme de messagerie et de streaming distribuée open source, cloud-native, développée par Yahoo en 2016. Elle a été conçue pour gérer la messagerie à haut débit et à faible latence pour les systèmes distribués à grande échelle et à l'échelle mondiale. Pulsar combine les fonctionnalités des systèmes de mise en file d'attente de messages (comme Kafka) et des plateformes de streaming en temps réel, ce qui le rend extrêmement polyvalent pour les architectures de données modernes.

### Caractéristiques principales

- **Modèle de messagerie unifié :** Pulsar prend en charge à la fois la mise en file d'attente des messages et la diffusion en temps réel dans un seul système. Cela réduit le besoin d'intégrer plusieurs solutions pour différents cas d'utilisation.
- **Multi-location et isolation :** l'architecture Pulsar est conçue pour la multi-location, ce qui permet à plusieurs clients ou équipes de partager le même cluster Pulsar tout en maintenant une forte isolation entre leurs charges de travail. Pulsar est donc idéal pour les grandes organisations ayant divers cas d'utilisation.
- **Géo-réplication :** Pulsar prend en charge nativement la géo-réplication, permettant une réplication transparente des données sur plusieurs emplacements géographiques, ce qui est essentiel pour la création de systèmes distribués à l'échelle mondiale.
- **Compactage et conservation des sujets :** Pulsar prend en charge le compactage des sujets, qui conserve uniquement les valeurs les plus récentes pour chaque clé, et les politiques de conservation des messages qui peuvent stocker des messages pendant des périodes définies.
- **Évolutivité et performances :** Pulsar sépare le calcul et le stockage (via Apache BookKeeper), ce qui permet une mise à l'échelle horizontale avec un impact minimal sur les performances. Son architecture garantit une diffusion des messages à faible latence et peut s'adapter à des millions de sujets.

### Cas d'utilisation

- **Analyse en temps réel :** Pulsar est parfaitement adapté à la capture d'événements en temps réel et à leur diffusion vers des moteurs d'analyse en aval comme Apache Druid ou Apache Spark.
- **Architectures pilotées par événements :** la capacité de Pulsar à gérer à la fois la mise en file d'attente et le streaming des messages le rend idéal pour les applications pilotées par événements telles que les microservices.
- **Orchestration du pipeline de données :** il peut servir d'épine dorsale à des pipelines de données complexes, garantissant une livraison de messages en temps réel et tolérante aux pannes.

### Limites

- Courbe d'apprentissage pour le déploiement et l'exploitation
- Communauté et écosystème limités par rapport à Kafka

---

## 2. Druide Apache

### Aperçu

Apache Druid est une base de données d'analyse en temps réel conçue pour l'ingestion et l'interrogation rapides de données basées sur des événements. Développée à l'origine par Metamarkets (plus tard acquise par Snap Inc.), Druid excelle dans la gestion de volumes importants de données de séries chronologiques, telles que les journaux, les métriques et les flux de clics.

### Caractéristiques principales

- **Stockage des données en colonnes :** Druid stocke les données dans un format en colonnes, ce qui permet une agrégation et un filtrage rapides de grands ensembles de données. Cette structure est optimisée pour les requêtes analytiques qui analysent de grandes quantités de données.
- **Ingestion de données en temps réel :** Druid peut ingérer des données en streaming à partir de sources telles qu'Apache Kafka ou Pulsar et fournir des résultats de requête quasi instantanés. Sa capacité à combiner l'ingestion de données en streaming et par lots le rend extrêmement polyvalent.
- **Analyse de séries chronologiques :** l'architecture Druid est optimisée pour les données temporelles, ce qui la rend idéale pour les cas d'utilisation tels que l'analyse de données de séries chronologiques, la surveillance des tendances et la détection d'anomalies.
- **Évolutivité horizontale :** Druid peut évoluer horizontalement en ajoutant davantage de nœuds, ce qui lui permet de gérer efficacement des pétaoctets de données.
- **Optimisé pour OLAP :** Druid est spécialement conçu pour les charges de travail de traitement analytique en ligne (OLAP), qui impliquent des requêtes complexes qui regroupent de grands ensembles de données.

### Cas d'utilisation

- **Tableaux de bord interactifs :** Druid est couramment utilisé pour alimenter des tableaux de bord d'analyse interactifs où les utilisateurs doivent interroger de grands ensembles de données en temps réel.
- **Détection de fraude :** sa capacité à analyser de grands volumes de données en temps réel le rend adapté à la détection de schémas de fraude ou d'activités suspectes.
- **Analyse opérationnelle :** les entreprises utilisent Druid pour surveiller et analyser les mesures de leurs systèmes opérationnels, telles que la surveillance des journaux système ou de l'activité des utilisateurs.

### Limites

- Complexité dans l'installation et la configuration
- Flexibilité limitée pour les données non chronologiques

---

## 3. PrestoDB

### Aperçu

PrestoDB est un moteur de requêtes SQL distribué open source, développé par Facebook, qui permet d'effectuer des requêtes sur de grands ensembles de données stockés dans divers systèmes tels que HDFS, S3, MySQL et les magasins NoSQL. Presto est conçu pour exécuter des requêtes complexes à des vitesses interactives, ce qui en fait une solution idéale pour les entreprises à la recherche d'une interrogation rapide sans déplacer de données.

### Caractéristiques principales

- **Requêtes fédérées :** Presto peut interroger des données provenant de plusieurs sources, en combinant des ensembles de données de différents systèmes dans une seule requête. Cela élimine le besoin de processus ETL (Extraction, Transformation, Chargement), ce qui facilite le travail avec des sources de données disparates.
- **Vitesses de requête interactives :** Presto est optimisé pour l'exécution de requêtes SQL avec une faible latence, ce qui en fait un choix populaire pour les outils de business intelligence (BI) où l'interrogation rapide de grands ensembles de données est essentielle.
- **Architecture enfichable :** Presto offre une architecture extensible qui permet l'intégration avec une large gamme de sources de données via des connecteurs personnalisés.
- **Évolutivité :** Presto peut évoluer horizontalement sur plusieurs clusters, ce qui lui permet de gérer de grands ensembles de données répartis sur différents systèmes de stockage.

### Cas d'utilisation

- **Interrogation de lac de données :** Presto est fréquemment utilisé pour interroger des données dans des lacs de données distribués tels que ceux construits sur HDFS ou AWS S3.
- **Analyse interactive :** les outils de veille économique et de reporting utilisent Presto pour fournir aux utilisateurs un accès rapide et interactif à de grands ensembles de données.
- **Exploration de données ad hoc :** les scientifiques et les ingénieurs de données utilisent Presto pour l'analyse exploratoire des données sur plusieurs sources de données.

### Limites

- Non conçu pour le traitement des transactions ou les charges de travail en streaming
- Nécessite un réglage minutieux pour les requêtes à grande échelle

---

## Comparaison avec Hadoop et Spark

| | Fonctionnalité | Apache Hadoop | Apache Spark | Apache Pulsar | Druide Apache | PrestoDB |
|---|---|---|---|---|---|---|
| 1 | **Fonctionnalités de base** | Traitement par lots | Traitement par lots et par flux | File d'attente de messages + Streaming | Base de données d'analyse en temps réel | Moteur de requête SQL |
| 2 | **Cas d'utilisation principal** | ETL à grande échelle | Analyses en temps réel et par lots | Microservices pilotés par événements | Requêtes en temps réel + historiques | Interrogation interactive |
| 3 | **Informatique** | Basé sur disque, MapReduce | Traitement en mémoire | Traitement des flux et des messages | Ingestion en temps réel | Requêtes SQL sur plusieurs sources |
| 4 | **Latence** | Haut | Faible (en mémoire) | Faible | Faible (optimisé pour OLAP) | Faible |
| 5 | **Évolutivité** | Élevé mais nécessite un réglage | Haut | Très élevé (géo-réplication) | Évolutif horizontalement | Évolutif horizontalement |
| 6 | **Ingestion de données** | Lot | Lot + Stream | Flux | Lot + Stream | Lot |
| 7 | **Points forts** | Haute tolérance aux pannes, écosystème mature | Calcul rapide en mémoire, bibliothèques ML | Messagerie et streaming unifiés | Analyse de séries chronologiques, agrégations rapides | Requêtes interactives rapides sur plusieurs sources de données |
| 8 | **Faiblesses** | Latence élevée, non adapté au temps réel | Complexe pour les tâches à petite échelle | Configuration complexe, écosystème plus petit | Flexibilité limitée en matière de séries non temporelles | Nécessite un réglage pour des performances optimales |

---

## Conclusion

Si les plateformes Big Data traditionnelles telles que Hadoop et Spark restent fondamentales, la complexité croissante et les exigences en temps réel des systèmes de données modernes ont conduit à l'essor d'outils spécialisés comme Apache Pulsar, Apache Druid et PrestoDB. Ces technologies fournissent des solutions d'analyse en temps réel, des architectures pilotées par événements et des requêtes rapides sur des données distribuées, ce qui les rend essentielles pour les organisations confrontées à des défis de données modernes. La compréhension de ces outils émergents est bénéfique pour créer des systèmes de données efficaces, évolutifs et à l'épreuve du temps.

---

## Auteur

**Rajashree Patil**