Étude Comparative des Formats de Fichiers : Parquet, ORC, Avro et Apache Arrow

Par Jean Marie Fande NDIAYE

Enseignant : Dr. Djibril Mboup

Cours : BigData – ING3 GEIT IPSL 2024

Introduction

Dans le domaine de l'ingénierie des données et de l'analyse de big data, le choix du format de fichier pour le stockage et le traitement des données est crucial. Ce choix peut considérablement influencer les performances, l'efficacité et le succès global des opérations de données.

Cette étude se concentre sur quatre des formats de fichiers les plus populaires dans l'écosystème du big data : **Apache Parquet, Optimized Row Columnar (ORC), Avro** et **Apache Arrow**. Chacun de ces formats a ses propres caractéristiques, avantages et inconvénients, qui les rendent plus ou moins adaptés à différents cas d'utilisation.

Le but de cette étude est de fournir une comparaison claire et concise de ces formats, en mettant en évidence leurs forces, leurs faiblesses et les cas où chacun d'eux excelle. Cela permettra aux professionnels de la donnée de prendre des décisions éclairées sur le format le plus adapté à leurs besoins spécifiques.

Apache Parquet

Définition : Apache Parquet est un format de fichier de stockage columnar conçu pour les projets de l'écosystème Hadoop. Il est optimisé pour l'efficacité et les performances, particulièrement adapté aux requêtes complexes sur de grands ensembles de données.

Avantages :

- **Stockage columnar** : Parquet stocke les données par colonne, ce qui permet des Entrées/Sorties disques plus efficaces et une meilleure compression.
- **Évolution du schéma** : Parquet prend en charge les structures de données imbriquées complexes et permet l'évolution du schéma.
- **Compression** : Parquet offre de bons schémas de compression et d'encodage, réduisant l'espace de stockage et améliorant les performances pour la récupération de données en colonnes.

Inconvénients :

- **Charges de travail lourdes en écriture** : La compression et l'encodage par colonne augmentent le coût de l'écriture des données, ce qui peut être un inconvénient pour les charges de travail intensives en écriture.
- **Petits ensembles de données** : Les avantages du modèle de stockage en colonnes de Parquet ne sont pas aussi prononcés pour les petits ensembles de données.

Cas d'utilisation :

- Grandes structures de données complexes.
- Charges de travail lourdes en lecture.
- Analyse avec des outils comme Apache Spark ou Apache Arrow.

Optimized Row Columnar (ORC)

Définition : ORC est un format de fichier en colonnes autodescriptif et conscient des types, conçu pour les charges de travail Hadoop. Il optimise la gestion des données et la compression pour des performances de lecture améliorées.

Avantages :

- **Compression** : il fournit des taux de compression impressionnants, minimisant l'espace de stockage.
- **Types complexes** : il prend en charge les types complexes tels que les structs, les listes, les maps et les types union.
- **Transactions ACID** : il fonctionne bien avec les transactions ACID dans Hive, permettant des mises à jour, des suppressions et des fusions.

Inconvénients :

- **Support communautaire** : Comparé à Parquet, ORC a moins de support communautaire, ce qui signifie moins de ressources, de bibliothèques et d'outils disponibles.
- **Coûts d'écriture** : Comme Parquet, ORC peut avoir des coûts d'écriture élevés en raison de sa nature columnar.

Cas d'utilisation :

- Écriture à haute vitesse avec des frameworks basés sur Hive.
- Besoins de modifications de données (mises à jour et suppressions).
- Données complexes et imbriquées.

Avro

Définition : Avro est un format de sérialisation de données binaires compact et rapide, conçu pour des échanges de données efficaces dans des environnements distribués. Il prend en charge l'évolution du schéma, ce qui le rend idéal pour des données évolutives.

Avantages :

- **Stockage efficace des données** : c’est un format de sérialisation binaire compact et rapide.
- **Évolution du schéma** : il prend en charge l'évolution du schéma et les structures de données complexes.
- **Transmission des données** : il est bénéfique pour le stockage et la transmission efficace des données.

Inconvénients :

- **Performance de requête** : il peut ne pas être le plus efficace pour les performances de requête comparé aux formats columnaires comme Parquet et ORC.

Cas d'utilisation :

- Applications nécessitant une évolution du schéma.
- Traitement haute performance.
- Stockage et transmission dans des environnements cloud.

Apache Arrow

Définition : Apache Arrow est une structure de données en mémoire conçue pour une efficacité de calcul en mémoire. Elle complète Parquet en fournissant une structure de données en colonnes optimisée pour les opérations en mémoire.

Avantages :

- **Format en mémoire** : Arrow est conçu pour le calcul en mémoire, offrant une efficacité de calcul en mémoire.
- **Complément à Parquet** : il agit comme une structure de données en mémoire pour Parquet, combinant les avantages de la structure de données en colonnes avec le calcul en mémoire.

Inconvénients :

- **Pas un format de stockage** : c’est principalement utilisé pour le calcul en mémoire et non pour le stockage sur disque.

Cas d'utilisation :

- Calcul en mémoire performant.
- Utilisé avec Parquet pour le stockage sur disque et le calcul en mémoire.

Tableau Comparatif des Formats de Fichiers : Parquet, ORC, Avro et Apache Arrow

Critère	Parquet	ORC	Avro	Apache Arrow
Type de stockage	Columnar	Columnar	Row-based	In-memory
Compression	Excellente, plusieurs schémas de compression	Excellente, taux de compression élevé	Modérée, compact	N/A (se concentre sur le stockage en mémoire)
Efficacité des requêtes	Très élevée pour les requêtes complexes	Très élevée pour les requêtes complexes	Bonne, mais moins efficace que Parquet/ORC	Très élevée pour les opérations en mémoire
Schema Evolution	Supporte l'évolution du schéma	Supporte l'évolution du schéma	Excellent support de l'évolution du schéma	N/A (utilisé principalement pour le calcul)
Support des types complexes	Oui	Oui	Oui	Oui
Cas d'utilisation	Requêtes analytiques, grands ensembles de données	Charges de travail Hadoop, ACID transactions	Échange de données, compatibilité des schémas	Calcul en mémoire, en complément de Parquet
Coûts d'écriture	Élevés pour les charges lourdes d'écriture	Élevés pour les charges lourdes d'écriture	Modérés	N/A
Support communautaire	Large, bien supporté	Modéré, moins large que Parquet	Large, bien supporté	En croissance rapide
Compatibilité	Hadoop, Spark, Impala	Hadoop, Hive	Hadoop, Spark	Outils de calcul en mémoire, intégré avec Parquet
Effet sur la performance	Améliore la performance des requêtes analytiques	Améliore la performance des requêtes analytiques	Améliore la compatibilité et l'évolutivité	Améliore l'efficacité du calcul en mémoire

Conclusion

Le choix du format de fichier pour le stockage et le traitement des données est crucial dans le domaine du big data. Cette étude comparative a examiné quatre formats populaires : Apache Parquet, ORC, Avro et Apache Arrow, chacun ayant des avantages et des inconvénients spécifiques.

- **Apache Parquet** se distingue par son efficacité dans le traitement de grandes quantités de données et les requêtes complexes.
- **Optimized Row Columnar (ORC)** est optimisé pour les charges de travail Hadoop, offrant une excellente compression et la gestion de types de données complexes.
- **Avro** est idéal pour les échanges de données binaires compacts et rapides, avec une gestion efficace de l'évolution du schéma.
- **Apache Arrow** excelle en tant que structure de données en mémoire pour un calcul rapide et efficace.

Chaque format a ses cas d'utilisation spécifiques, et le choix dépendra des besoins particuliers du projet. En comprenant les forces et faiblesses de chaque format, les professionnels peuvent prendre des décisions éclairées pour optimiser la gestion et l'analyse de leurs données.