Introduction au Big Data avec Spark et Hadoop

Glossaire : Introduction au Big Data avec Spark et Hadoop

Bienvenue ! Ce glossaire alphabétique contient de nombreux termes utilisés dans ce cours. Ce glossaire complet comprend également des termes supplémentaires reconnus par l'industrie qui ne sont pas utilisés dans les vidéos de cours. Ces termes sont essentiels pour que vous puissiez les reconnaître lorsque vous travaillez dans l'industrie, participez à des groupes d'utilisateurs et à d'autres programmes de certification professionnelle.

Temps de lecture estimé : 20 minutes

Terme	Définition
Agrégation de données	L'agrégation est un processus Spark SQL fréquemment utilisé pour présenter des statistiques agrégées. Les fonctions d'agrégation couramment utilisées telles que count(), avg(), max() et d'autres sont intégrées aux DataFrames. Les utilisateurs peuvent également effectuer une agrégation par programmation à l'aide de requêtes SQL et de vues de table.
AIOps	Implique l'utilisation de l'intelligence artificielle pour automatiser ou améliorer les opérations informatiques. Elle permet de collecter, d'agréger et de traiter de grands volumes de données opérationnelles. Elle permet également d'identifier les événements et les modèles dans les systèmes d'infrastructure de grande taille ou complexes. Elle permet de diagnostiquer rapidement la cause profonde des problèmes afin que les utilisateurs puissent les signaler ou les résoudre automatiquement.
Service de stockage simple d'Amazon (Amazon S3)	Protocole d'interface de magasin d'objets inventé par Amazon. Il s'agit d'un composant Hadoop qui comprend le protocole S3. S3 fournit une interface pour les services Hadoop, tels qu'IBM Db2 Big SQL, afin de consommer des données hébergées par S3.
Analyser les données à l'aide de printSchema	Dans cette phase, les utilisateurs examinent le schéma ou les types de données des colonnes DataFrame à l'aide de la méthode du schéma d'impression. Il est impératif de noter les types de données dans chaque colonne. Les utilisateurs peuvent appliquer la fonction select() pour examiner en détail les données d'une colonne spécifique.
Détection d'anomalies	Processus d'apprentissage automatique qui identifie les points de données, les événements et les observations qui s'écartent du comportement normal d'un ensemble de données. La détection d'anomalies à partir de données de séries chronologiques est un problème qu'il est essentiel de résoudre pour les applications industrielles.
Apache	Ce serveur HTTP open source implémente les normes HTTP actuelles pour être hautement sécurisé, facilement configurable et hautement extensible. La licence Apache Software de l'Apache Software Foundation le construit et le distribue.
Apache Cassandra	Il s'agit d'une base de données NoSQL évolutive spécialement conçue pour ne pas avoir de point de défaillance unique.
Apache HBase	Un magasin de données NoSQL robuste qui gère efficacement les ressources de stockage et de calcul indépendamment de l'écosystème Hadoop.
Apache Mesos	Gestionnaire de cluster à usage général avec des avantages supplémentaires. L'utilisation de Mesos présente certains avantages. Il peut fournir un partitionnement dynamique entre Spark et d'autres frameworks Big Data et un partitionnement évolutif entre plusieurs instances Spark. Cependant, l'exécution de Spark sur Apache Mesos peut nécessiter une configuration supplémentaire, en fonction de vos exigences de configuration.
Apache Nutch	Un logiciel d'exploration Web extensible et évolutif utilisé pour regrouper des données à partir du Web.
Apache Spark	Un framework de calcul distribué open source conçu pour traiter des données à grande échelle et effectuer des analyses de données. Il fournit des bibliothèques pour diverses tâches de traitement de données, notamment le traitement par lots, le traitement de flux en temps réel, l'apprentissage automatique et le traitement de graphes. Spark est connu pour sa rapidité et sa simplicité d'utilisation, ce qui en fait un choix populaire pour les applications Big Data.
Architecture d'Apache Spark	Se compose des processus pilote et exécuteur. Le cluster comprend le gestionnaire de cluster et les nœuds de travail. Le contexte Spark planifie les tâches du cluster et le gestionnaire de cluster gère les ressources du cluster.
Modes de cluster Apache Spark	Apache Spark propose différents modes de cluster pour le calcul distribué, notamment Standalone, YARN (Yet Another Resource Negotiator) et Apache Mesos. Chaque mode possède des caractéristiques et des complexités de configuration spécifiques.
Interface utilisateur d'Apache Spark	Fournit des informations précieuses, organisées sur plusieurs onglets, sur l'application en cours d'exécution. L'onglet Tâches affiche les tâches de l'application, y compris l'état de la tâche. L'onglet Étapes indique l'état des tâches au sein d'une étape. L'onglet Stockage affiche la taille de tous les RDD ou DataFrames qui ont persisté dans la mémoire ou le disque. L'onglet Environnement inclut toutes les variables d'environnement et les propriétés système pour Spark ou la JVM. L'onglet Exécuteur affiche un résumé qui montre l'utilisation de la mémoire et du disque pour tous les exécuteurs utilisés pour l'application. Des onglets supplémentaires s'affichent en fonction du type d'application utilisé.
Gardien de zoo Apache	Un service centralisé de gestion des informations de configuration pour maintenir des liens sains entre les nœuds. Il assure la synchronisation entre les applications distribuées. Il est également utilisé pour suivre les pannes de serveur et les partitions réseau en déclenchant un message d'erreur, puis en réparant les nœuds défaillants.
Problèmes de dépendance des applications	Les applications Spark peuvent avoir de nombreuses dépendances, notamment des fichiers d'application tels que des fichiers de script Python, des fichiers JAR Java et même des fichiers de données requis. Les applications dépendent des bibliothèques utilisées et de leurs dépendances. Les dépendances doivent être rendues disponibles sur tous les nœuds du cluster, soit par préinstallation, y compris les dépendances dans le script spark-submit fourni avec l'application, soit sous forme d'arguments supplémentaires.
Interface de programmation d'application (API)	Ensemble de règles bien définies qui aident les applications à communiquer entre elles. Il fonctionne comme une couche intermédiaire pour le traitement du transfert de données entre les systèmes, permettant aux entreprises d'ouvrir leurs données et fonctionnalités d'application aux partenaires commerciaux, aux développeurs tiers et à d'autres services internes.
Problèmes de ressources d'application	Les cœurs de processeur et la mémoire peuvent devenir un problème si une tâche se trouve dans la file d'attente de planification et que les travailleurs disponibles n'ont pas suffisamment de ressources pour exécuter les tâches. Lorsqu'un travailleur termine une tâche, le processeur et la mémoire sont libérés, ce qui permet la planification d'une autre tâche. Cependant, si l'application demande plus de ressources qui peuvent devenir disponibles, les tâches peuvent ne jamais être exécutées et finir par expirer. De même, supposons que les exécuteurs exécutent de longues tâches qui ne se terminent jamais. Dans ce cas, leurs ressources ne deviennent jamais disponibles, ce qui entraîne également l'impossibilité d'exécuter les tâches futures, ce qui entraîne une erreur de dépassement de délai. Les utilisateurs peuvent facilement accéder à ces erreurs lorsqu'ils consultent l'interface utilisateur ou les journaux d'événements.

Terme	Définition
ID d'application	Un identifiant unique que Spark attribue à chaque application. Ces fichiers journaux apparaissent pour chaque processus d'exécution et de pilote exécuté par l'application.
Big Data	Ensembles de données dont le type ou la taille dépasse la capacité des bases de données relationnelles traditionnelles à gérer, capturer et traiter les données avec une faible latence. Les caractéristiques du Big Data incluent un volume, une vitesse et une variété élevés.
Analyse de Big Data	Utilise des techniques d'analyse avancées sur des ensembles de données volumineux et diversifiés, comprenant des données structurées, semi-structurées et non structurées provenant de différentes sources et tailles, allant des téraoctets aux zettaoctets. Il aide les entreprises à tirer des enseignements des données collectées par les appareils IoT.
Outils de programmation Big Data	Les outils de programmation sont le dernier composant des outils commerciaux de Big Data. Ces outils de programmation effectuent des tâches analytiques à grande échelle et opérationnalisent le Big Data. Ils fournissent également toutes les fonctions nécessaires à la collecte, au nettoyage, à l'exploration, à la modélisation et à la visualisation des données. Parmi les outils populaires que vous pouvez utiliser pour la programmation, citons R, Python, SQL, Scala et Julia.
Bloc	Quantité minimale de données écrites ou lues, et offre également une tolérance aux pannes. La taille de bloc par défaut peut être de 64 Mo ou 128 Mo, selon la configuration du système de l'utilisateur. Chaque fichier stocké n'a pas besoin d'occuper l'espace de stockage de la taille de bloc préconfigurée.
Ensemble de données Bootstrap (BSDS)	Un ensemble de données séquencées par clé (KSDS) VSAM (Virtual Storage Access Method) utilisé pour stocker les informations essentielles requises par IBM MQ (mise en file d'attente des messages). Le BSDS comprend généralement un inventaire de tous les ensembles de données de journaux actifs et archivés connus d'IBM MQ. IBM MQ utilise cet inventaire pour suivre les ensembles de données de journaux actifs et archivés. Le BSDS joue un rôle essentiel dans le bon fonctionnement et la bonne gestion d'IBM MQ, en garantissant l'intégrité et la disponibilité des ensembles de données de journaux au sein du système de messagerie.
Intelligence d'affaires (BI)	Comprend divers outils et méthodologies conçus pour convertir efficacement les données en informations exploitables.
Catalyseur	Dans le cadre opérationnel de Spark, il utilise une paire de moteurs, à savoir Catalyst et Tungsten, de manière séquentielle pour l'amélioration et l'exécution des requêtes. La fonction principale de Catalyst consiste à dériver un plan de requête physique optimisé à partir du plan de requête logique initial. Ce processus d'optimisation implique la mise en œuvre d'une gamme de transformations telles que le refoulement des prédicats, l'élagage des colonnes et le repliement constant sur le plan logique.
Phases catalytiques	Catalyst analyse la requête, le DataFrame, le plan logique non résolu et le catalogue pour créer un plan logique dans la phase d'analyse. Le plan logique évolue vers un plan logique optimisé dans la phase d'optimisation logique. Il s'agit de l'étape d'optimisation basée sur des règles de Spark SQL. Des règles telles que le pliage, le refoulement et l'élagage sont applicables ici. Catalyst génère plusieurs plans physiques basés sur le plan logique dans la phase de planification physique. Un plan physique décrit le calcul sur des ensembles de données avec des définitions spécifiques expliquant comment effectuer le calcul. Un modèle de coût sélectionne ensuite le plan physique ayant le coût le plus faible. Cela explique l'étape d'optimisation basée sur les coûts. La génération de code est la phase finale. Dans cette phase, Catalyst applique le plan physique sélectionné et génère le bytecode Java à exécuter sur les nœuds.
Optimisation des requêtes Catalyst	Catalyst Optimizer utilise une structure de données arborescente et fournit les ensembles de règles d'arborescence de données en arrière-plan. Catalyst effectue les quatre tâches de haut niveau suivantes pour optimiser une requête : analyse, optimisation logique, planification physique et génération de code.
Algorithmes de classification	Un type d'algorithme d'apprentissage automatique qui aide les ordinateurs à apprendre à classer les éléments en différents groupes en fonction des modèles qu'ils trouvent dans les données.
Cloud computing	Permet aux clients d'accéder à l'infrastructure et aux applications via Internet sans nécessiter d'installation et de maintenance sur site. En exploitant le cloud computing, les entreprises peuvent utiliser la capacité du serveur à la demande et évoluer rapidement pour gérer les exigences informatiques étendues du traitement de grands ensembles de données et de l'exécution de modèles mathématiques complexes.
Fournisseurs de services cloud	Proposer une infrastructure et un support essentiels, en fournissant des ressources informatiques partagées qui englobent la puissance de calcul, le stockage, la mise en réseau et les logiciels d'analyse. Ces fournisseurs proposent également un modèle de logiciel en tant que service comprenant des solutions spécifiques, permettant aux entreprises de collecter, de traiter et de visualiser efficacement les données. AWS, IBM, GCP et Oracle sont des exemples marquants de fournisseurs de services cloud.
Cadre de gestion des clusters	Il gère les aspects informatiques distribués de Spark. Il peut exister sous forme de serveur autonome, Apache Mesos ou Yet Another Resource Network (YARN). Un cadre de gestion de cluster est essentiel pour faire évoluer le Big Data.
Groupes	Faites référence à des groupes de serveurs gérés collectivement et participant à la gestion de la charge de travail. Vous pouvez avoir des nœuds au sein d'un cluster, généralement des systèmes informatiques physiques individuels avec des adresses IP d'hôte distinctes. Chaque nœud d'un cluster peut exécuter un ou plusieurs serveurs d'applications. Les clusters sont un concept fondamental dans le calcul distribué et la gestion des serveurs, permettant l'allocation efficace des ressources et l'évolutivité des applications et des services sur plusieurs instances de serveur.
Interface de ligne de commande (CLI)	Utilisé pour entrer des commandes qui permettent aux utilisateurs de gérer le système.
Commitant	La plupart des projets open source disposent de processus formels de contribution au code et incluent différents niveaux d'influence et d'obligation envers le projet : contributeur, contributeur, utilisateur et groupe d'utilisateurs. En règle générale, les contributeurs peuvent modifier le code directement.
Matériel de base	Il s'agit de postes de travail ou d'ordinateurs de bureau à faible coût, compatibles IBM et exécutant plusieurs systèmes d'exploitation tels que Microsoft Windows, Linux et DOS sans adaptations ni logiciels supplémentaires.
Interface de calcul	Il s'agit d'une frontière partagée en informatique à travers laquelle deux ou plusieurs composants différents d'un système informatique échangent des informations.
Conteneurisation	Cela implique que les applications Spark sont plus portables. Cela facilite la gestion des dépendances et la configuration de l'environnement requis dans l'ensemble du cluster. Cela permet également un meilleur partage des ressources.
Optimisation basée sur les coûts	Le coût est mesuré et calculé en fonction du temps et de la mémoire consommés par une requête. L'optimiseur Catalyst sélectionne un chemin de requête qui entraîne une consommation minimale de temps et de mémoire. Comme les requêtes peuvent utiliser plusieurs chemins, ces calculs peuvent devenir assez complexes lorsque de grands ensembles de données font partie du calcul.

Terme	Définition
Créer une vue dans Spark SQL	Il s'agit de la première étape de l'exécution de requêtes SQL dans Spark SQL. Il s'agit d'une table temporaire utilisée pour exécuter des requêtes SQL. Les vues temporaires et globales sont prises en charge par Spark SQL. Une vue temporaire a une portée locale. La portée locale implique que la vue existe dans la session Spark actuelle sur le nœud actuel. Une vue temporaire globale existe dans l'application Spark générale. Cette vue peut être partagée entre différentes sessions Spark.
Planificateur DAG	Alors que Spark agit et transforme les données dans les processus d'exécution des tâches, le DAGScheduler facilite l'efficacité en orchestrant les nœuds de travail dans l'ensemble du cluster. Ce suivi des tâches rend possible la tolérance aux pannes, car il réapplique les opérations enregistrées aux données d'un état précédent.
Ingénierie des données	Il s'agit d'une pratique courante qui implique la conception et la création de systèmes de collecte, de stockage et d'analyse de données à grande échelle. Il s'agit d'une discipline qui trouve des applications dans différents secteurs. Les ingénieurs de données utilisent les outils Spark, notamment le moteur Spark principal, les clusters, les exécuteurs et leur gestion, Spark SQL et DataFrames.
Ingestion de données	Première étape du traitement des Big Data. Il s'agit d'un processus d'importation et de chargement de données dans IBM® WatsonX.data. Vous pouvez utiliser l'onglet Tâches d'ingestion de la page Gestionnaire de données pour charger les données de manière simple et sécurisée dans la console WatsonX.data.
Science des données	Discipline qui combine les mathématiques et les statistiques, la programmation spécialisée, l'analyse avancée, l'intelligence artificielle (IA) et l'apprentissage automatique avec une expertise spécifique dans le domaine pour révéler des informations exploitables cachées dans les données de l'organisation. Ces informations peuvent être utilisées dans la prise de décision et la planification stratégique.
Ensembles de données	Créés en extrayant des données à partir de packages ou de modules de données. Ils peuvent être utilisés pour rassembler une collection personnalisée d'éléments que vous utilisez fréquemment. Lorsque les utilisateurs mettent à jour leur ensemble de données, les tableaux de bord et les histoires qui utilisent l'ensemble de données sont également mis à jour.
Validation des données	La pratique consistant à vérifier l'intégrité, la qualité et l'exactitude des données utilisées dans les applications Spark ou les flux de travail de traitement des données. Ce processus de validation comprend la vérification des données pour détecter des problèmes tels que des valeurs manquantes, des valeurs aberrantes ou des erreurs de format de données. La validation des données est essentielle pour garantir que les données traitées dans les applications Spark sont fiables et adaptées à l'analyse ou au traitement ultérieur. Diverses techniques et bibliothèques, telles que l'API DataFrame d'Apache Spark ou des outils externes, peuvent être utilisées pour effectuer des tâches de validation des données dans les environnements Spark.
Entrepôt de données	Stocke les données historiques provenant de nombreuses sources différentes afin que les utilisateurs puissent les analyser et en extraire des informations.
Opérations DataFrame	Faites référence à un ensemble d'actions et de transformations qui peuvent être appliquées à un DataFrame, qui est une structure de données bidimensionnelle dans Spark. Les données d'un DataFrame sont organisées dans un format tabulaire avec des lignes et des colonnes, similaire à une table dans une base de données relationnelle. Ces opérations englobent un large éventail de tâches, notamment la lecture de données dans un DataFrame, l'analyse de données, l'exécution de transformations de données (telles que le filtrage, le regroupement et l'agrégation), le chargement de données à partir de sources externes et l'écriture de données dans divers formats de sortie. Les opérations DataFrame sont fondamentales pour travailler efficacement avec des données structurées dans Spark.
Trames de données	Collecte de données organisée de manière catégorique en colonnes nommées. Les DataFrames sont conceptuellement équivalents à une table dans une base de données relationnelle et similaires à un dataframe dans R ou Python, mais avec de plus grandes optimisations. Ils sont construits sur l'API SparkSQL RDD. Ils utilisent des RDD pour effectuer des requêtes relationnelles. De plus, ils sont hautement évolutifs et prennent en charge de nombreux formats de données et systèmes de stockage. Ils sont conviviaux pour les développeurs, offrant une intégration avec la plupart des outils de big data via Spark et des API pour Python, Java, Scala et R.
Programmation déclarative	Paradigme de programmation utilisé par un programmeur pour définir l'accomplissement du programme sans définir comment il doit être mis en œuvre. L'approche se concentre principalement sur ce qui doit être réalisé, plutôt que de préconiser la manière d'y parvenir.
Graphe acyclique dirigé (DAG)	Représentation conceptuelle d'une série d'activités. Un graphique illustre l'ordre des activités. Il se présente visuellement comme un ensemble de cercles, chacun représentant une activité, certains reliés par des lignes, représentant le flux d'une activité à une autre.
distinct ([numTasks])	Il permet de trouver le nombre d'éléments variés dans un ensemble de données. Il renvoie un nouvel ensemble de données contenant des éléments distincts de l'ensemble de données source.
Informatique distribuée	Un système ou une machine comportant plusieurs composants situés sur différentes machines. Chaque composant a sa propre tâche, mais les composants communiquent entre eux pour fonctionner comme un seul système pour l'utilisateur final.
Conducteur	Reçoit les instructions de requête soumises via la ligne de commande et envoie la requête au compilateur après avoir lancé une session.
Mémoire du conducteur	Désigne l'allocation de mémoire désignée pour le programme pilote d'une application Spark. Le programme pilote sert de coordinateur central des tâches, gérant la distribution et l'exécution des tâches Spark sur les nœuds du cluster. Il contient le flux de contrôle de l'application, les métadonnées et les résultats des transformations et actions Spark. La capacité de la mémoire du pilote est un facteur critique qui a un impact sur la faisabilité et les performances des applications Spark. Elle doit être configurée avec soin pour garantir une exécution efficace des tâches sans problèmes liés à la mémoire.
Programme de conduite	Il peut être exécuté en mode client ou en mode cluster. En mode client, l'émetteur de l'application (tel qu'un terminal de machine utilisateur) lance le pilote en dehors du cluster. En mode cluster, le programme pilote est envoyé et exécuté sur un nœud Worker disponible à l'intérieur du cluster. Le pilote doit pouvoir communiquer avec le cluster pendant son exécution, qu'il soit en mode client ou en mode cluster.
Configuration dynamique	Désigne une pratique employée dans le développement de logiciels pour éviter de coder en dur des valeurs spécifiques directement dans le code source de l'application. Au lieu de cela, les paramètres de configuration critiques, tels que l'emplacement d'un serveur maître, sont stockés en externe et peuvent être ajustés sans modifier le code de l'application.
Onglet Environnement	Contient plusieurs listes pour décrire l'environnement de l'application en cours d'exécution. Ces listes incluent les propriétés de configuration Spark, les profils de ressources, les propriétés pour Hadoop et les propriétés système actuelles.
Variables d'environnement	Méthode de configuration de l'application Spark dans laquelle les variables d'environnement sont chargées sur chaque machine, afin qu'elles puissent être ajustées machine par machine si le matériel ou le logiciel installé diffère entre les nœuds du cluster.
Exécuteur	Utilise une partie définie des ressources locales comme mémoire et cœurs de calcul, en exécutant une tâche par cœur disponible. Chaque exécuteur gère sa mise en cache de données comme indiqué par le pilote. En général, l'augmentation du nombre d'exécuteurs et de cœurs

Terme	Définition
	disponibles augmente le parallélisme du cluster. Les tâches s'exécutent dans des threads distincts jusqu'à ce que tous les cœurs soient utilisés. Lorsqu'une tâche se termine, l'exécuteur place les résultats dans une nouvelle partition RDD ou les transfère au pilote. Idéalement, limitez les cœurs utilisés au nombre total de cœurs disponibles par nœud.
Mémoire de l'exécuteur	Utilisé pour le traitement. Si la mise en cache est activée, une mémoire supplémentaire est utilisée. Une mise en cache excessive entraîne des erreurs de manque de mémoire.
Onglet Exécuteurs	Composant de certains outils et infrastructures informatiques distribués utilisés pour gérer et surveiller l'exécution des tâches au sein d'un cluster. Il présente généralement un tableau récapitulatif en haut qui affiche les mesures pertinentes pour les exécuteurs actifs ou terminés. Ces mesures peuvent inclure des statistiques liées aux tâches, l'entrée et la sortie de données, l'utilisation du disque et l'utilisation de la mémoire. Sous le tableau récapitulatif, l'onglet répertorie tous les exécuteurs individuels qui ont participé à l'application ou au travail, ce qui peut inclure le pilote principal. Cette liste fournit souvent des liens pour accéder aux messages de journal de sortie standard (stdout) et d'erreur standard (stderr) associés à chaque processus d'exécuteur. L'onglet Exécuteurs constitue une ressource précieuse pour les administrateurs et les opérateurs afin d'obtenir des informations sur les performances et le comportement des exécuteurs de cluster pendant l'exécution des tâches.
Écosystème Hadoop étendu	Il s'agit de bibliothèques ou de packages logiciels couramment utilisés ou installés sur le noyau Hadoop.
Extraire, charger et transformer (ELT)	Ce concept est né du traitement des big data. Toutes les données résident dans un lac de données. Un lac de données est un pool de données brutes dont l'objectif n'est pas prédéfini. Dans un lac de données, chaque projet forme des tâches de transformation individuelles selon les besoins. Il n'anticipe pas tous les scénarios d'utilisation des exigences de transformation comme dans le cas d'ETL et d'un entrepôt de données. Les organisations choisissent d'utiliser un mélange d'ETL et d'ELT.
Processus d'extraction, de transformation et de chargement (ETL)	Une approche systématique qui consiste à extraire des données de diverses sources, à les transformer pour répondre à des exigences spécifiques et à les charger dans un entrepôt de données ou un autre référentiel de données centralisé.
Tolérance aux pannes	Un système est tolérant aux pannes s'il peut continuer à fonctionner malgré la défaillance de certains composants. La tolérance aux pannes contribue à rendre votre infrastructure de démarrage à distance plus robuste. Dans le cas des serveurs de déploiement de système d'exploitation, l'ensemble du système est tolérant aux pannes si les serveurs de déploiement de système d'exploitation se sauvegardent mutuellement.
Système de fichiers	Une structure de répertoire complète qui comprend un répertoire racine (/) et d'autres répertoires et fichiers sous celui-ci. Elle est confinée à un volume logique. Les informations complètes sur le système de fichiers sont centralisées dans le fichier /etc/filesystems.
filtre (*fonc*)	It helps in filtering the elements of a data set basis its function. The filter operation is used to selectively retain elements from a data set or DataFrame based on a provided function (*func*). It allows you to filter and extract specific elements that meet certain criteria, making it a valuable tool for data transformation and analysis.
flatmap (*func*)	Similar to map (*func*) can map each input item to zero or more output items. Its function should return a Seq rather than a single item.
Flume	A distributed service that collects, aggregates, and transfers big data to the storage system. Offers a simple yet flexible architecture that streams data flows and uses an extensible data model, allowing online analytic applications.
For-loop	Extends from a FOR statement to an END FOR statement and executes for a specified number of iterations, defined in the FOR statement.
Functional programming (FP)	A style of programming that follows the mathematical function format. Declarative implies that the emphasis of the code or program is on the "what" of the solution as opposed to the "how to" of the solution. Declarative syntax abstracts out the implementation details and only emphasizes the final output, restating "the what." We use expressions in functional programming, such as the expression f of x, as mentioned earlier.
Hadoop	An open-source software framework that provides dependable distributed processing for large data sets through the utilization of simplified programming models.
Hadoop Common	Fundamental part of the Apache Hadoop framework. It refers to a collection of primary utilities and libraries that support other Hadoop modules.
Hadoop Distributed File System (HDFS)	A file system distributed on multiple file servers, allowing programmers to access or store files from any network or computer. It is the storage layer of Hadoop. It works by splitting the files into blocks, creating replicas of the blocks, and storing them on different machines. It is built to access streaming data seamlessly. It uses a command-line interface to interact with Hadoop.
Hadoop Ecosystem	It splits big data analytics processing tasks into smaller tasks. The small tasks are performed in conjunction using an algorithm (e.g., MapReduce) and then distributed across a Hadoop cluster (i.e., nodes that perform parallel computations on big data sets).
Hadoop Ecosystem stages	The four main stages are: Ingest, store, process, analyze, and access.
HBase	A column-oriented, non-relational database system that runs on top of the Hadoop Distributed File System (HDFS). It provides real-time wrangling access to the Hadoop file system. It uses hash tables to store data in indexes and allow for random data access, making lookups faster.
High-throughput	Throughput quantifies the data processed in a timeframe. The target system needs robust throughput for heavy workloads with substantial data changes from the source database to prevent latency spikes. Performance objectives are frequently outlined with throughput targets. High throughput is achieved when most messages are delivered successfully, whereas low successful delivery rates indicate poor throughput and network performance.
Hive	A data warehouse infrastructure employed for data querying and analysis, featuring an SQL-like interface. It facilitates report generation and utilizes a declarative programming language, enabling users to specify the data they want to retrieve.
Hive client	Hive provides different drivers for communication depending on the type of application. For example, for Java-based applications, it uses JDBC drivers, and other types of applications will use ODBC drivers. These drivers communicate with the servers.
Hive server	Used to execute queries and enable multiple clients to submit requests. It is built to support JDBC and ODBC clients.

Terme	Définition
Hive services	Client interactions are done through the Hive services. Any query operations are done here. The command-line interface acts as an interface for the Hive service. The driver takes in query statements, monitors each session's progress and life cycle, and stores metadata generated from the query statements.
Hive tables	Spark supports reading and writing data stored in Apache Hive.
Hive Web Interface	A web-based user interface that interacts with Hive through a web browser. It offers a graphical user interface (GUI) used to browse tables, execute Hive queries, and manage Hive resources.
HMaster	The master server that monitors the region server instances. It assigns regions to region servers and distributes services to different region servers. It also manages any changes that are made to the schema and metadata operations.
Hue	An acronym for Hadoop user experience. It allows you to upload, browse, and query data. Users can run Pig jobs and workflow in Hue. It also provides an SQL editor for several query languages, like Hive and MySQL.
Hybrid cloud	Unifies and combines public and private cloud and on-premises infrastructure to create a single, cost-optimal, and flexible IT infrastructure.
IBM Analytics Engine	Works with Spark to provide a flexible, scalable analytics solution. It uses an Apache Hadoop cluster framework to separate storage and compute by storing data in object storage such as IBM Cloud Object Storage. This implies users can run compute nodes only when required.
IBM Spectrum Conductor	A multitenant platform for deploying and managing Spark and other frameworks on a cluster with shared resources. This enables multiple Spark applications and versions to be run together on a single large cluster. Cluster resources can be divided up dynamically, avoiding downtime. IBM Spectrum Conductor also provides Spark with enterprise-grade security.
IBM Watson	Creates production-ready environments for AI and machine learning by providing services, support, and holistic workflows. Reducing setup and maintenance saves time so that users can concentrate on training Spark to enhance its machine-learning capabilities. IBM Cloud Pak for Watson AIOps offers solutions with Spark that can correlate data across your operations toolchain to bring insights or identify issues in real time.
Immutable	This type of object storage allows users to set indefinite retention on the object if they are unsure of the final duration of the retention period or want to use event-based retention. Once set to indefinite, user applications can change the object retention to a finite value.
Impala	A scalable system that allows nontechnical users to search for and access the data in Hadoop.
Imperative programming paradigm	In this software development paradigm, functions are implicitly coded in every step used in solving a problem. Every operation is coded, specifying how the problem will be solved. This implies that pre-coded models are not called on.
In-memory processing	The practice of storing and manipulating data directly in a computer's main memory (RAM), allowing for faster and more efficient data operations compared to traditional disk-based storage.
InputSplits	Created by the logical division of data. They serve as an input to a single Mapper job.
Internet of Things (IoT)	A system of physical objects connected through the internet. A "thing or device" can include a smart device in our homes or a personal communication device such as a smartphone or computer. These collect and transfer massive amounts of data over the internet without manual intervention by using embedded technologies.
Iterative process	An approach to continuously improving a concept, design, or product. Creators produce a prototype, test it, tweak it, and repeat the cycle to get closer to the solution.
JAR (Java Archive)	A standard file format used to package Java classes and related resources into a single compressed file. JAR files are commonly used to bundle Java libraries, classes, and other assets into a single unit for distribution and deployment.
Java	Technology equipped with a programming language and a software platform. To create and develop an application using Java, users are required to download the Java Development Kit (JDK), available for Windows, macOS, and Linux.
Java virtual machines (JVMs)	The platform-specific component that runs a Java program. At run time, the VM interprets the Java bytecode compiled by the Java Compiler. The VM is a translator between the language and the underlying operating system and hardware.
JavaScript Object Notation (JSON)	A simplified data-interchange format based on a subset of the JavaScript programming language. IBM Integration Bus provides support for a JSON domain. The JSON parser and serializer process messages in the JSON domain.
JDBC client	Component in the Hive client, which allows Java-based applications to connect to Hive.
Job details	Provides information about the different stages of a specific job. The timeline displays each stage, where the user can quickly see the job's timing and duration. Below the timeline, completed stages are displayed. In the parentheses beside the heading, users will see a quick view that displays the number of completed stages. Then, view the list of stages within the job and job metrics, including when the job was submitted, input or output sizes, the number of attempted tasks, the number of succeeded tasks, and how much data was read or written because of a shuffle.
Jobs tab	Commonly found in Spark user interfaces and monitoring tools, it offers an event timeline that provides key insights into the execution flow of Spark applications. This timeline includes crucial timestamps such as the initiation times of driver and executor processes, along with the creation timestamps of individual jobs within the application. The Jobs tab serves as a valuable resource for monitoring the chronological sequence of events during Spark job execution.
JSON data sets	Spark infers the schema and loads the data set as a DataFrame.
Kubernetes (K8s)	A popular framework for running containerized applications on a cluster. It's an open-source system that is highly scalable and provides flexible deployments to the cluster. Spark uses a built-in native Kubernetes scheduler. It is portable, so it can be run in the same way on cloud or on-premises.
Lambda calculus	A mathematical concept that implies every computation can be expressed as an anonymous function that is applied to a data set.

Terme	Définition
Lambda functions	Calculus functions, or operators. These are anonymous functions that enable functional programming. They are used to write functional programming code.
List processing language (Lisp)	The functional programming language that was initially used in the 1950s. Today, there are many functional programming language options, including Scala, Python, R, and Java.
Loading or exporting the data	In the ETL pipeline's last step, data is exported to disk or loaded into another database. Also, users can write the data to the disk as a JSON file or save the data into another database, such as a Postgres (PostgresSQL) database. Users can also use an API to export data to a database, such as a Postgres database.
Local mode	Runs a Spark application as a single process locally on the machine. Executors are run as separate threads in the main process that calls 'spark-submit'. Local mode does not connect to any cluster or require configuration outside a basic Spark installation. Local mode can be run on a laptop. That's useful for testing or debugging a Spark application, for example, testing a small data subset to verify correctness before running the application on a cluster. However, being constrained by a single process means local mode is not designed for optimal performance.
Logging configuration	Spark Application configuration method in which Spark logging is controlled by the log4j defaults file, which dictates what level of messages, such as info or errors, are logged to the file or output to the driver during application execution.
Low latency data access	A type of data access allowing minimal delays, not noticeable to humans, between an input processed and corresponding output offering real-time characteristics. It is crucial for internet connections using trading, online gaming, and voice over IP.
Machine data	Refers to information generated by various sources, including Internet of Things (IoT) sensors embedded in industrial equipment as well as weblogs that capture user behavior and interactions.
Machine learning	A full-service cloud offering that allows developers and data scientists to collaborate and integrate predictive capabilities with their applications.
Map	MapReduce converts a set of data into another set of data and the elements are fragmented into tuples (key or value pairs).
map (*func*)	It is an essential operation capable of expressing all transformations needed in data science. It passes each element of the source through a function func, thereby returning a newly formed distributed data set.
MapReduce	A program model and processing technique used in distributed computing based on Java. It splits the data into smaller units and processes big data. It is the first method used to query data stored in HDFS. It allows massive scalability across hundreds or thousands of servers in a Hadoop cluster.
Meta store	Stores the metadata, the data, and information about each table, such as the location and schema. The meta store, file system, and job client, in turn, communicate with Hive storage and computing to perform the following: Metadata information from tables is stored in some databases and query results, and data loaded from the tables is stored in a Hadoop cluster on HDFS.
Modular development	Techniques used in job designs to maximize the reuse of parallel jobs and components and save user time.
Multiple related jobs	Spark application can consist of many parallel and often related jobs, including multiple jobs resulting from multiple data sources, multiple DataFrames, and the actions applied to the DataFrames.
Node	A single independent system responsible for storing and processing big data. HDFS follows the primary and secondary concept.
NoSQL databases	NoSQL databases are built from the ground up to store and process vast amounts of data at scale and support a growing number of modern businesses. NoSQL databases store data in documents rather than relational tables. Types of NoSQL databases include pure document databases, key-value stores, wide-column databases, and graph databases such as MongoDB, CouchDB, Cassandra, and Redis.
Open Database Connectivity (ODBC) client	Component in the Hive client, which allows applications based on the ODBC protocol to connect to Hive.
Open-source software	Not only is the runnable version of the code free, but the source code is also completely open, meaning that every line of code is available for people to view, use, and reuse as needed.
Optimizer	Performs transformations on the execution and splits the tasks to help speed up and improve efficiency.
Parallel computing	A computing architecture in which multiple processors execute different small calculations fragmented from a large, complex problem simultaneously.
Parallel programming	It resembles distributed programming. It is the simultaneous use of multiple compute resources to solve a computational task. Parallel programming parses tasks into discrete parts solved concurrently using multiple processors. The processors access a shared pool of memory, which has control and coordination mechanisms in place.
Parallelization	Parallel regions of program code executed by multiple threads, possibly running on multiple processors. Environment variables determine the number of threads created and calls to library functions.
Parquet	Columnar format that is supported by multiple data processing systems. Spark SQL allows reading and writing data from Parquet files, and Spark SQL preserves the data schema.
Parser	A program that interprets the physical bit stream of an incoming message and creates an internal logical representation of the message in a tree structure. The parser also regenerates a bit stream for an outgoing message from the internal message tree representation.
Partitioning	This implies dividing the table into parts depending on the values of a specific column, such as date or city.
Persistent cache	Information is stored in "permanent" memory. Therefore, data is not lost after a system crash or restart, as if it were stored in cache memory.
Pig Hadoop component	Famous for its multi-query approach, it analyzes large amounts of data. It is a procedural data flow language and a procedural programming language that follows an order and set of commands.

Terme	Définition
Price analytics	Helps understand market segmentation, identify the best price points for a product line, and perform margin analysis for maximum profitability.
Primary node	Also known as the name node, it regulates file access to the clients and maintains, manages, and assigns tasks to the secondary node. The architecture is such that per cluster, there is one name node and multiple data nodes, the secondary nodes.
Properties	Spark Application configuration method in which Spark properties are used to adjust and control most application behaviors, including setting properties with the driver and sharing them with the cluster.
Python	Easy-to-learn, high-level, interpreted, and general-purpose dynamic programming language focusing on code readability. It provides a robust framework for building fast and scalable applications for z/OS, with a rich ecosystem of modules to develop new applications the same way you would on any other platform.
R	An open-source optimized programming language for statistical analysis and data visualization. Developed in 1992, it has a robust ecosystem with complex data models and sophisticated tools for data reporting.
Rack	The collection of about forty to fifty data nodes using the same network switch.
Rack awareness	When performing operations such as read and write, the name node maximizes performance by choosing the data nodes closest to themselves. This could be done by selecting data nodes on the same rack or nearby racks. It is used to reduce network traffic and improve cluster performance. To achieve rack awareness, the name node keeps the rack ID information.
RDD actions	It is used to evaluate a transformation in Spark. It returns a value to the driver program after running a computation. An example is the reduce action that aggregates the elements of an RDD and returns the result to the driver program.
RDD transformations	It helps in creating a new RDD from an existing RDD. Transformations in Spark are deemed lazy as results are not computed immediately. The results are computed after evaluation by actions. For example, map transformation passes each element of a data set through a function. This results in a new RDD.
Read	In this operation, the client will send a request to the primary node to acquire the location of the data nodes containing blocks. The client will read files closest to the data nodes.
Read the data	When reading the data, users can load data directly into DataFrames or create a new Spark DataFrame from an existing DataFrame.
Reduce	Job in MapReduce that uses output from a map as an input and combines data tuples into small sets of tuples.
Redundancy	Duplication of data across multiple partitions or nodes in a cluster. This duplication is implemented to enhance fault tolerance and reliability. If one partition or node fails, the duplicated data on other partitions or nodes can still be used to ensure that the computation continues without interruption. Redundancy is critical in maintaining data availability and preventing data loss in distributed computing environments like Spark clusters.
Region	The basic building element and most negligible unit of the HBase cluster, consisting of column families. It contains multiple stores, one for each column family and has two components: HFile and MemStore .
Region servers	These servers receive read and write requests from the client. They assign the request to a region where the column family resides. They serve and manage regions present in a distributed cluster. The region servers can communicate directly with the client to facilitate requests.
Relational database	Data is organized into rows and columns collectively, forming a table. The data is structured across tables, joined by a primary or a foreign key.
Relational Database Management System (RDBMS)	Traditional RDBMS is used to maintain a database and uses the structured query language known as SQL. It is suited for real-time data analysis, like data from sensors. It allows for as many read-and-write operations as a user may require. It can handle up to terabytes of data. It enforces that the schema must verify loading data before it can proceed. It may not always have built-in support for data partitioning.
Replication	The process of creating a copy of the data block. It is performed by rack awareness as well. It is done by ensuring data node replicas are in different racks. So, if a rack is down, users can obtain the data from another rack.
Replication factor	Defined as the number of times you make a copy of the data block. Users can set the number of copies they want, depending on their configuration.
Resilient Distributed Datasets (RDDs)	A fundamental abstraction in Apache Spark that represents distributed collections of data. RDDs allow you to perform parallel and fault-tolerant data processing across a cluster of computers. RDDs can be created from existing data in storage systems (like HDFS), and they can undergo various transformations and actions to perform operations like filtering, mapping, and aggregating. The "resilient" aspect refers to resilient distributed datasets (RDDs) ability to recover from node failures, and the "distributed" aspect highlights their distribution across multiple machines in a cluster, enabling parallel processing.
Scala	A general-purpose programming language that supports both object-oriented and functional programming. The most recent representative in the family of programming languages. Apache Spark is written mainly in Scala, which treats functions as first-class citizens. Functions in Scala can be passed as arguments to other functions, returned by other functions, and used as variables.
Scalability	The ability of a system to take advantage of additional resources, such as database servers, processors, memory, or disk space. It aims at minimizing the impact on maintenance. It is the ability to maintain all servers efficiently and quickly with minimal impact on user applications.
Schema	It is a collection of named objects. It provides a way to group those objects logically. A schema is also a name qualifier; it provides a way to use the same natural name for several objects and to prevent ambiguous references to those objects.
Secondary node	This node is also known as a data node. There can be hundreds of data nodes in the HDFS that manage the storage system. They perform read and write requests at the instructions of the name node. They also create, replicate, and delete file blocks based on instructions from the name node.
Semi-structured data	Semi-structured data (e.g., JSON, CSV, XML) is the "bridge" between structured and unstructured data. It does not have a predefined data model and is more complex than structured data, yet easier to store than unstructured data.

Terme	Définition
Sentiment analysis	Utilizes social media conversations to gain insights into consumer opinions about a product. It is used to develop effective marketing strategies and establish customer connections based on their sentiments and preferences.
Serialization	Required to coordinate access to resources that are used by more than one program. An example of why resource serialization is needed occurs when one program is reading from a data set and another program needs to write to the data set.
Shuffle	Phase in which interim map output from mappers is transferred to reducers. Every reducer fetches interim results for all values associated with the same key from multiple nodes. This is a network-intensive operation within the Hadoop cluster nodes.
Social data	Comes from the likes, tweets and retweets, comments, video uploads, and general media that are uploaded and shared via the world's favorite social media platforms. Machine-generated data and business-generated data are data that organizations generate within their own operations.
Spark Application	A Spark application refers to a program or set of computations written using the Apache Spark framework. It consists of a driver program and a set of worker nodes that process data in parallel. Spark applications are designed for distributed data processing, making them suitable for big data analytics and machine learning tasks.
Spark Cluster Manager	Communicates with a cluster to acquire resources for an application to run. It runs as a service outside the application and abstracts the cluster type. While an application is running, the Spark Context creates tasks and communicates to the cluster manager what resources are needed. Then the cluster manager reserves executor cores and memory resources. Once the resources are reserved, tasks can be transferred to the executor processes to run.
Spark Configuration Location	Located under the "conf" directory in the installation. By default, there are no preexisting files after installation, however, Spark provides a template for each configuration type with the filenames shown here. Users can create the appropriate file by removing the '.template' extension. Inside the template files are sample configurations for standard settings. They can be enabled by uncommenting.
Spark Context	Communicates with the Cluster Manager. It is defined in the Driver, with one Spark Context per Spark Application.
Spark Core	Often popularly referred to as "Spark." The fault-tolerant Spark Core is the base engine for large-scale parallel and distributed data processing. It manages memory and task scheduling. It also contains the APIs used to define RDDs and other datatypes. It parallelizes a distributed collection of elements across the cluster.
Spark data persistence	Also known as caching data in Spark. Ability to store intermediate calculations for reuse. This is achieved by setting persistence in either memory or both memory and disk. Once intermediate data is computed to generate a fresh DataFrame and cached in memory, subsequent operations on the DataFrame can utilize the cached data instead of reloading it from the source and redoing previous computations. This feature is crucial for accelerating machine learning tasks that involve multiple iterations on the same data set during model training.
Spark driver program	A program that functions as software situated on the primary node of a machine. It defines operations on RDDs, specifying transformations and actions. To simplify, the Spark driver initiates a SparkContext linked to a designated Spark Master. Furthermore, it transfers RDD graphs to the Master, the location from which the stand-alone cluster manager operates.
Spark History server	Web UI where the status of running and completed Spark jobs on a provisioned instance of Analytics Engine powered by Apache Spark, is displayed. If users want to analyze how different stages of the Spark job are performed, they can view the details in the Spark history server UI.
Spark jobs	Computations that can be executed in parallel. The Spark Context divides Jobs into Tasks to be executed on the Cluster.
Spark logging	Controlled using log4j and the configuration is read through "conf/log4j-properties." Users can adjust a log level to determine which messages (such as debug, info, or errors) are shown in the Spark logs.
Spark memory management	Spark memory stores the intermediate state while executing tasks such as joining or storing broadcast variables. All the cached and persisted data will be stored in this segment, specifically in the storage memory.
Spark ML	Spark's machine learning library for creating and using machine learning models on large data sets across distributed clusters.
Spark RDD persistence	Optimization technique that saves the result of RDD evaluation in cache memory. Using this technique, the intermediate result can be saved for future use. It reduces the computation overhead.
Spark Shell	Available for Scala and Python, giving you access to Spark APIs for working with data as Spark jobs. Spark Shell can be used in local or cluster mode, with all options available.
Spark Shell Environment	When Spark Shell starts, the environment automatically initializes the SparkContext and SparkSession variables. This means you can start working with data immediately. Expressions are entered in the shell and evaluated in the driver. Entering an action on a shell DataFrame generates Spark jobs that are sent to the cluster to be scheduled as tasks.
Spark Shuffle	Performed when a task requires other data partitions. It marks the boundary between stages.
Spark SQL memory optimization	The primary aim is to improve the run-time performance of a SQL query by minimizing the query time and memory consumption, thereby helping organizations save time and money.
Spark SQL	A Spark module for structured data processing. Users can interact with Spark SQL using SQL queries and the DataFrame API. Spark SQL supports Java, Scala, Python, and R APIs. Spark SQL uses the same execution engine to compute the result independently of the API or language used for computation. Developers can use the API to help express a given transformation. Unlike the basic Spark RDD API, Spark SQL includes a cost-based optimizer, columnar storage, and code generation to perform optimizations that equip Spark with information about the structure of data and the computation in process.
Spark Stages	Represents a set of tasks an executor can complete on the current data partition. Subsequent tasks in later stages must wait for that stage to be completed before beginning execution, creating a dependency from one stage to the next.
Spark Standalone	Included with the Spark installation. It is best for setting up a simple cluster. There are no additional dependencies required to configure and deploy. Spark Standalone is specifically designed to run Spark and is often the fastest way to get a cluster up and running applications.

Terme	Définition
Spark Standalone cluster	Has two main components: Workers and the Master. The workers run on cluster nodes. They start an executor process with one or more reserved cores. There must be one master available which can run on any cluster node. It connects workers to the cluster and keeps track of them with heartbeat polling. However, if the master is together with a worker, do not reserve all the node's cores and memory for the worker.
Spark tasks	Tasks from a given job operate on different data subsets called partitions and can be executed in parallel.
SparkContext	When a Spark application is being run, as the driver program creates a SparkContext, Spark starts a web server that serves as the application user interface. Users can connect to the UI web server by entering the hostname of the driver followed by port 4040 in a browser once that application is running. The web server runs for the duration of the Spark application, so once the SparkContext stops, the server shuts down, and the application UI is no longer accessible.
Spark-submit	Spark comes with a unified interface for submitting applications called the 'spark-submit' script found in the 'bin/' directory. 'Spark-submit' can be used for all supported cluster types and accepts many configuration options for the application or cluster. 'Unified interface' means you can switch from running Spark in local mode to cluster by changing a single argument. 'Spark-submit' works the same way, irrespective of the application language. For example, a cluster can run Python and Java applications simultaneously by passing in the required files.
SQL Procedural code	A set of instructions written in a programming language within an SQL database environment. This code allows users to perform more complex tasks and create custom functions, procedures, and control structures, enabling them to manipulate and manage data in a more controlled and structured manner.
SQL queries in Spark SQL	Spark SQL allows users to run SQL queries on Spark DataFrames.
Sqoop	An open-source product designed to transfer bulk data between relational database systems and Hadoop. It looks in the relational database and summarizes the schema. It generates MapReduce code to import and export data. It helps develop any other MapReduce applications that use the records stored in HDFS.
Stages tab	Displays a list of all stages in the application, grouped by the current state of either completed, active, or pending. This example displays three completed stages. Click the Stage ID Description hyperlinks to view task details for that stage.
Static configuration	Settings that are written programmatically into the application. These settings are not usually changed because they require modifying the application itself. Use static configuration for something that is unlikely to be changed or tweaked between application runs, such as the application name or other properties related to the application only.
Storage tab	Displays details about RDDs that have been cached or persisted to memory and written to disk.
Streaming	Implies HDFS provides a constant bitrate when transferring data, rather than having the data transferred in waves.
Streaming analytics	Help leverage streams to ingest, analyze, monitor, and correlate data from real-time data sources. They also help to view information and events as they unfold.
String data type	It is the IBM® Informix® ESQL/C data type that holds character data that is null-terminated and does not contain trailing blanks.
Structured data	Structured data, typically categorized as quantitative data, is highly organized and easily decipherable by machine learning algorithms. Developed by IBM in 1974, structured query language (SQL) is the programming language used to manage structured data.
Syntax error	If this error is detected while processing a control statement, the remaining statement is skipped and not processed. Any operands in the portion of the statement preceding the error are processed.
toDS() function	Converts data into a typed data set for efficient and type-safe operations in PySpark.
Transactional Data	Generated from all the daily transactions that take place both online and offline, such as invoices, payment orders, storage records, and delivery receipts.
Transform the data	In this step of the ETL pipeline, users plan for required data set transformations, if any. The transformation aims at retaining only the relevant data. Transformation techniques include data filtering, merging with other data sources, or performing columnar operations. Columnar operations include actions such as multiplying each column by a specific number or converting data from one unit to another. Transformation techniques can also be used to group or aggregate data. Many transformations are domain-specific data augmentation processes. The effort needed varies with the domain and the data.
Tungsten	Catalyst and Tungsten are integral components of Spark's optimization and execution framework. Tungsten is geared toward enhancing both CPU and memory performance within Spark. Unlike Java, which was initially designed for transactional applications, it seeks to bolster these aspects by employing methods more tailored to data processing within the Java Virtual Machine (JVM). To achieve optimal CPU performance, it also adopts explicit memory management, employs cache-friendly data structures through STRIDE-based memory access, supports on-demand JVM bytecode, minimizes virtual function dispatches, and capitalizes on CPU register placement and loop unrolling.
Uber-JAR	An Uber-JAR is a single Java Archive (JAR) file that contains not only the application code but also all its dependencies, including transitive ones. The purpose of an Uber-JAR is to create a self-contained package that can be easily transported and executed within a computing cluster or environment.
Unified memory	Unified regions in Spark shared by executor memory and storage memory. If executor memory is not used, storage can acquire all the available memory, and vice versa. If the total storage memory usage falls under a certain threshold, executor memory can discard storage memory. Due to complexities in implementation, storage cannot evict executor memory.
Unstructured data	Information lacking a predefined data model or not fitting into relational tables.
User code	Made up of the driver program, which runs in the driver process, and the functions and variables serialized that the executor runs in parallel. The driver and executor processes run the application user code of an application passed to the Spark-submit script. The user code in the driver creates the SparkContext and creates jobs based on operations for the DataFrames. These DataFrame operations become serialized closures sent throughout the cluster and run on executor processes as tasks. The serialized closures contain the necessary functions, classes, and variables to run each task.

Terme	Définition
Variety	The diversity of data or the various data forms that need to be stored. Variety is one of the four main components used to describe the dimensions of big data.
Velocity	The speed at which data arrives. Velocity is one of the four main components used to describe the dimensions of big data.
Veracity	La certitude des données, comme celle des grandes quantités de données disponibles, rend difficile de déterminer si les données collectées sont exactes. La véracité est l'un des quatre principaux éléments utilisés pour décrire les dimensions du big data.
Volume	L'augmentation de la quantité de données stockées au fil du temps. Le volume est l'un des quatre principaux éléments utilisés pour décrire les dimensions du Big Data.
Travailleur	Nœud de cluster pouvant lancer des processus d'exécution pour exécuter des tâches.
Nœud de travail	Une unité dans un système distribué qui exécute des tâches et traite des données selon les instructions d'un coordinateur central.
Flux de travail	Inclure les tâches créées par SparkContext dans le programme du pilote. Les tâches en cours s'exécutent en tant que tâches dans les exécuteurs et les tâches terminées transfèrent les résultats vers le pilote ou écrivent sur le disque.
Écrire	Dans cette opération, le nœud Name s'assure que le fichier n'existe pas. Si le fichier existe, le client reçoit un message d'exception d'E/S. Si le fichier n'existe pas, le client reçoit l'autorisation de commencer à écrire des fichiers.
Encore un autre négociateur de ressources (YARN)	Il fait office de gestionnaire de ressources fourni avec Hadoop et est généralement le gestionnaire de ressources par défaut pour de nombreuses applications Big Data, telles que HIVE et Spark. Bien qu'il reste un gestionnaire de ressources robuste, il est important de noter que des gestionnaires de ressources basés sur des conteneurs plus contemporains, tels que Kubernetes, émergent progressivement comme les nouvelles pratiques standard dans le domaine.

Auteur(s)

- Niha Ayaz Sultan
- Rashi Kapoor