Introduction au Big Data avec Spark et Hadoop

Module 5 Glossaire : Options d'environnement de développement et d'exécution

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 : 15 minutes

Terme	Définition
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.
Apache Hadoop YARN (Encore un autre négociateur de ressources)	Gestionnaire de cluster du projet Hadoop. Il est populaire dans l'écosystème du Big Data et prend en charge de nombreux autres frameworks en plus de Spark. Les clusters YARN ont leurs propres dépendances, configurations et exigences de configuration, leur déploiement est donc plus complexe que celui de Spark autonome.
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.
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 les modes autonome, YARN (Yet Another Resource Negotiator) et Apache Mesos. Chaque mode possède des caractéristiques spécifiques et des complexités de configuration.
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.
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.
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.
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.
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 de travail 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.
Variables d'environnement	Méthode de configuration d'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. Une utilisation courante consiste à s'assurer que chaque machine du cluster utilise le même exécutable Python en définissant la variable d'environnement « PYSPARK_PYTHON ».
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 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.
Cloud hybride	Unifie et combine les infrastructures de cloud public et privé et sur site pour créer une infrastructure informatique unique, optimale en termes de coûts et flexible.
Moteur d'analyse IBM	Fonctionne avec Spark pour fournir une solution d'analyse flexible et évolutive. Il utilise une infrastructure de cluster Apache Hadoop pour séparer le stockage et le calcul en stockant les données dans un stockage d'objets tel qu'IBM Cloud Object Storage. Cela implique que les utilisateurs peuvent exécuter des nœuds de calcul uniquement lorsque cela est nécessaire.
Conducteur de spectre IBM	Une plateforme multilocataire pour le déploiement et la gestion de Spark et d'autres frameworks sur un cluster avec des ressources partagées. Cela permet d'exécuter plusieurs applications et versions Spark ensemble sur un seul grand cluster. Les ressources du cluster peuvent être divisées de manière dynamique, évitant ainsi les temps d'arrêt. IBM Spectrum Conductor fournit également à Spark une sécurité de niveau entreprise.

Terme	Définition
IBM Watson	Crée des environnements prêts pour la production pour l'IA et l'apprentissage automatique en fournissant des services, une assistance et des flux de travail holistiques. La réduction de la configuration et de la maintenance permet de gagner du temps afin que les utilisateurs puissent se concentrer sur la formation de Spark pour améliorer ses capacités d'apprentissage automatique. IBM Cloud Pak for Watson AIOps propose des solutions avec Spark qui peuvent corréler les données de votre chaîne d'outils d'exploitation pour apporter des informations ou identifier les problèmes en temps réel.
Archives Java (JAR)	Format de fichier standard utilisé pour regrouper des classes Java et des ressources associées dans un seul fichier compressé. Les fichiers JAR sont couramment utilisés pour regrouper des bibliothèques, des classes et d'autres ressources Java dans une seule unité pour la distribution et le déploiement.
Java	Technologie dotée d'un langage de programmation et d'une plateforme logicielle. Pour créer et développer une application à l'aide de Java, les utilisateurs doivent télécharger le kit de développement Java (JDK), disponible pour Windows, macOS et Linux.
Kubernetes (K8s)	Un framework populaire pour exécuter des applications conteneurisées sur un cluster. Il s'agit d'un système open source hautement évolutif qui offre des déploiements flexibles sur le cluster. Spark utilise un planificateur Kubernetes natif intégré. Il est portable, il peut donc être exécuté de la même manière sur le cloud ou sur site.
Mode local	Exécute une application Spark en tant que processus unique localement sur la machine. Les exécuteurs sont exécutés en tant que threads distincts dans le processus principal qui appelle « spark-submit ». Le mode local ne se connecte à aucun cluster et ne nécessite aucune configuration en dehors d'une installation Spark de base. Le mode local peut être exécuté sur un ordinateur portable. Cela est utile pour tester ou déboguer une application Spark, par exemple, pour tester un petit sous-ensemble de données afin de vérifier l'exactitude avant d'exécuter l'application sur un cluster. Cependant, le fait d'être limité par un seul processus signifie que le mode local n'est pas conçu pour des performances optimales.
Configuration de la journalisation	Méthode de configuration de l'application Spark dans laquelle la journalisation Spark est contrôlée par le fichier de valeurs par défaut log4j, qui dicte le niveau de messages, tels que les informations ou les erreurs, enregistrés dans le fichier ou envoyés au pilote pendant l'exécution de l'application.
Propriétés	Méthode de configuration d'application Spark dans laquelle les propriétés Spark sont utilisées pour ajuster et contrôler la plupart des comportements d'application, y compris la définition des propriétés avec le pilote et leur partage avec le cluster.
Python	Langage de programmation dynamique de haut niveau, interprété et polyvalent, facile à apprendre, axé sur la lisibilité du code. Il fournit un cadre robuste pour la création d'applications rapides et évolutives pour z/OS, avec un riche écosystème de modules pour développer de nouvelles applications de la même manière que vous le feriez sur n'importe quelle autre plateforme.
Échelle	General-purpose programming language that supports functional and object-oriented 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.
Spark application	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 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 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 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 Shuffle	Performed when a task requires other data partitions. It marks the boundary between stages.
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.
Cluster autonome Spark	Il comporte deux composants principaux : les travailleurs et le maître. Les travailleurs s'exécutent sur des nœuds de cluster. Ils démarrent un processus d'exécution avec un ou plusieurs cœurs réservés. Il doit y avoir un maître disponible qui peut s'exécuter sur n'importe quel nœud de cluster. Il connecte les travailleurs au cluster et en assure le suivi grâce à l'interrogation par pulsation. Cependant, si le maître est associé à un travailleur, ne réservez pas tous les cœurs et la mémoire du nœud pour le travailleur.
Spark autonome	Inclus avec l'installation de Spark. Il est idéal pour configurer un cluster simple. Aucune dépendance supplémentaire n'est requise pour la configuration et le déploiement. Spark Standalone est spécialement conçu pour exécuter Spark et constitue souvent le moyen le plus rapide de mettre en place un cluster et d'exécuter des applications.
Tâches Spark	Les tâches d'un travail donné fonctionnent sur différents sous-ensembles de données appelés partitions et peuvent être exécutées en parallèle.
Soumettre Spark	Spark est fourni avec une interface unifiée pour la soumission d'applications, appelée script « spark-submit » qui se trouve dans le répertoire « bin/ ». « Spark-submit » peut être utilisé pour tous les types de cluster pris en charge et accepte de nombreuses options de configuration pour l'application ou le cluster. L'interface unifiée signifie que vous pouvez passer de l'exécution de Spark en mode local

Terme	Définition
	au cluster en modifiant un seul argument. « Spark-submit » fonctionne de la même manière, quelle que soit la langue de l'application. Par exemple, un cluster peut exécuter des applications Python et Java simultanément en transmettant les fichiers requis.
Configuration statique	Paramètres écrits par programmation dans l'application. Ces paramètres ne sont généralement pas modifiés, car ils nécessitent de modifier l'application elle-même. Utilisez une configuration statique pour un élément qui est peu susceptible d'être modifié ou modifié entre les exécutions de l'application, comme le nom de l'application ou d'autres propriétés liées à l'application uniquement.
Uber-JAR	Un Uber-JAR est un fichier d'archive Java (JAR) unique qui contient non seulement le code de l'application, mais également toutes ses dépendances, y compris celles transitives. Le but d'un Uber-JAR est de créer un package autonome qui peut être facilement transporté et exécuté au sein d'un cluster ou d'un environnement informatique.
Travailleur	Nœud de cluster pouvant lancer des processus d'exécution pour exécuter des tâches.

Auteur(s)

- Niha Ayaz Sultan