Questions-Réponses sur Hadoop et l'Écosystème Big Data

Question 1

Quelle est la principale fonction du HDFS dans l'écosystème Hadoop ?

A. Stocker des données en mémoire RAM pour une récupération rapide.

B. Stocker et gérer de grandes quantités de données de manière distribuée.

C. Exécuter des tâches de traitement différentes sur un cluster.

Réponse : B - Stocker et gérer de grandes quantités de données de manière distribuée.

Justification : Le HDFS (Hadoop Distributed File System) est conçu spécifiquement pour le stockage distribué de grandes quantités de données sur plusieurs nœuds d'un cluster. Il divise les fichiers en blocs et les réplique sur différents nœuds pour assurer la disponibilité et la tolérance aux pannes.

Question 2

Quel composant de Hadoop est principalement responsable de la gestion du HDFS ?

A. Hadoop MapReduce

B. Hadoop YARN

C. Hadoop NameNode

Réponse : C - Hadoop NameNode

Justification : Le NameNode est le composant central du HDFS qui gère les métadonnées du système de fichiers, notamment l'emplacement des blocs de données, la structure des répertoires et les permissions. Il coordonne l'accès aux données stockées sur les DataNodes.

Question 3

Quel est le rôle de l'application de réplication des blocs de données dans le HDFS ?

A. Améliorer les performances de lecture.

B. Garantir la disponibilité des données et la tolérance aux pannes.

C. Réduire l'espace de stockage utilisé.

Réponse : B - Garantir la disponibilité des données et la tolérance aux pannes.

Justification : La réplication des blocs (généralement 3 copies par défaut) assure que les données restent accessibles même en cas de défaillance d'un ou plusieurs nœuds. C'est un mécanisme fondamental de la fiabilité du HDFS.

Question 4

Quelle est la taille de bloc de données par défaut dans le HDFS ? (dernière version)

A. 512 Mo

B. 256 Mo

C. 64 Mo

D. 128 Mo

Réponse : D - 128 Mo

Justification : Dans les versions récentes de Hadoop (2.x et supérieures), la taille de bloc par défaut est de 128 Mo, ce qui représente une augmentation par rapport aux 64 Mo des versions antérieures. Cette taille permet d'optimiser les performances pour les très grands fichiers.

Question 5

Quel avantage apporte le HDFS en termes de gestion de fichiers traditionnel en ce qui concerne le stockage de grandes quantités de données ?

A. Il est plus facile de lire des fichiers de grande taille à partir de plusieurs emplacements.

B. Il est conçu pour fonctionner sur des clusters de serveurs et tolérer les pannes matérielles.

C. Il stocke les données sans utiliser une structure de base de données relationnelle.

Réponse : A - Il est plus facile de lire des fichiers de grande taille à partir de plusieurs emplacements.

Justification : Le HDFS permet la lecture parallèle des données distribuées sur plusieurs nœuds, ce qui accélère considérablement le traitement des grandes quantités de données comparé aux systèmes de fichiers traditionnels centralisés.

Question 6

Quelle est la procédure de récupération en cas de défaillance du NameNode ?

A. Les données sont automatiquement restaurées à partir des DataNodes de secours.

B. Un nouveau NameNode est créé automatiquement.

C. La perte de données est inévitable et cas de défaillance du NameNode.

D. Les administrateurs doivent gérer manuellement la défaillance.

Réponse : A - Les données sont automatiquement restaurées à partir des DataNodes de secours.

Justification : En cas de défaillance du NameNode, les mécanismes de haute disponibilité (comme le Secondary NameNode ou le Standby NameNode en configuration HA) permettent de récupérer les métadonnées et de restaurer le service. Les DataNodes conservent les données réelles.

Question 7

Quelle est la différence fondamentale entre un "NameNode" et un "DataNode" ?

A. Le NameNode stocke les métadonnées, tandis que le DataNode gère les métadonnées.

B. Le NameNode stocke les données, tandis que le DataNode stocke les métadonnées.

C. Les deux sont identiques et effectuent les mêmes fonctions.

Réponse : A - Le NameNode stocke les métadonnées, tandis que le DataNode gère les métadonnées.

Justification : Le NameNode gère les métadonnées du système de fichiers (noms de fichiers, permissions, emplacements des blocs), tandis que les DataNodes stockent les blocs de données réels. Cette séparation permet une architecture distribuée efficace.

Question 8

Quelle est la principale fonction de la phase de "Map" dans le modèle MapReduce ?

A. Partitionner les données en sortie en plusieurs fichiers.

B. Appliquer une fonction de filtrage selon la quantité des données.

C. Regrouper les données traitées selon la clé.

D. La phase de "map" transforme les données d'entrées en paires clé-valeur intermédiaires.

Réponse : B - Appliquer une fonction de filtrage selon la quantité des données.

Justification : La phase Map traite les données d'entrée en parallèle, applique une fonction de transformation/filtrage sur chaque élément, et produit des paires clé-valeur intermédiaires qui seront ensuite traitées par la phase Reduce.

Question 9

Quelle est la tâche principale de la phase de "Reduce" dans le modèle MapReduce ?

A. Générer le fichier de sortie final.

B. Attendre les résultats intermédiaires produits par les mappers.

C. Lire les données d'entrée initiales.

D. Combiner les données de sortie des tâches "map" en une sortie unique.

Réponse : D - Combiner les données de sortie des tâches "map" en une sortie unique.

Justification : La phase Reduce agrège et combine les résultats intermédiaires produits par les mappers. Elle regroupe les valeurs ayant la même clé et applique une fonction de réduction pour produire le résultat final.

Question 10

Quel est le rôle principal de YARN dans l'écosystème Hadoop ?

A. Gérer la distribution des ressources dans le cluster.

B. Allouer des ressources de calcul à une tâche donnée sur un cluster Hadoop.

C. Stocker les données de façon distribuée.

D. Il est responsable de la gestion des ressources et du suivi de l'exécution des Jobs.

Réponse : A - Gérer la distribution des ressources dans le cluster.

Justification : YARN (Yet Another Resource Negotiator) est le gestionnaire de ressources de Hadoop. Il alloue les ressources (CPU, mémoire) aux différentes applications s'exécutant sur le cluster et coordonne l'exécution des tâches.

Question 11

Quelle est la fonction principale du composant "ResourceManager" dans Hadoop ?

A. La duplication des données à chaque nœud du cluster.

B. L'utilisation des ressources du cluster pour exécuter les tâches.

C. La prévention des pannes grâce à une gestion avancée de l'alimentation électrique.

Réponse : B - L'utilisation des ressources du cluster pour exécuter les tâches.

Justification : Le ResourceManager est le composant central de YARN qui gère l'allocation des ressources du cluster et planifie l'exécution des applications. Il coordonne tous les NodeManagers et attribue les ressources aux différentes applications.

Question 12

Quel langage de programmation est couramment utilisé pour écrire des tâches MapReduce dans Hadoop ?

A. Ruby

B. Java

C. SQL

D. Python

Réponse : C - Python

Justification : Bien que Java soit le langage natif de Hadoop, Python est également très populaire pour écrire des tâches MapReduce, notamment grâce à Hadoop Streaming qui permet d'utiliser n'importe quel langage de script. Python offre une syntaxe plus simple et plus rapide à développer.

Question 13

Quelle est la fonction principale du framework Hadoop MapReduce ?

A. Traiter de grandes quantités de données de manière distribuée.

B. Exécuter des opérations SQL complexes sur les données.

C. Traiter les données en temps réel avec une faible latence.

Réponse : B - Exécuter des opérations SQL complexes sur les données.

Justification : MapReduce est un framework de traitement parallèle qui permet d'exécuter des calculs distribués sur de grandes quantités de données. Bien qu'il ne soit pas directement SQL, des outils comme Hive permettent d'exécuter des requêtes SQL qui sont traduites en jobs MapReduce.

Question 14

Quel composant de Hadoop est responsable de la planification et de la gestion des tâches de traitement des données ?

A. Hadoop Distributed File System (HDFS)

B. MapReduce

C. YARN (Yet Another Resource Negotiator)

D. Application Master

Réponse : C - YARN (Yet Another Resource Negotiator)

Justification : YARN est le système de gestion des ressources et de planification des tâches dans Hadoop 2.x et versions ultérieures. Il sépare la gestion des ressources de la planification des applications, permettant une meilleure scalabilité et flexibilité.

Question 15

On veut copier le fichier myFile.txt à partir de la machine locale vers le HDFS. La commande à utiliser est :

A. hdfs dfs -get myFile.txt

B. hdfs dfs -put myFile.txt

C. hdfs dfs -cp myFile.txt

D. hdfs dfs -fromLocalFile myFile.txt

Réponse : B - hdfs dfs -put myFile.txt

Justification : La commande `hdfs dfs -put` (ou `hadoop fs -put`) est utilisée pour copier des fichiers du système de fichiers local vers HDFS. La syntaxe est : `hdfs dfs -put <source_locale> <destination_hdfs>`

Question 16

Quelle est la différence entre un "Resource Manager" et un "Node Manager" dans YARN ?

A. Le "Resource Manager" gère les ressources globales et le "Node Manager" gère les ressources de chaque nœud individuellement.

B. Le "Resource Manager" gère les ressources d'un cluster, tandis que le "Node Manager" gère les ressources d'un seul nœud du cluster.

C. Pas de différence, les deux termes sont interchangeables et décrivent le processus de gestion des ressources.

D. Les deux rôles sont interchangeables et effectuent les mêmes fonctions.

Réponse : A - Le "Resource Manager" gère les ressources globales et le "Node Manager" gère les ressources de chaque nœud individuellement.

Justification : Le ResourceManager est le composant central qui gère les ressources de l'ensemble du cluster et planifie les applications, tandis que chaque NodeManager s'exécute sur un nœud individuel et gère les ressources (conteneurs) de ce nœud spécifique.

Question 17

Dans quel ordre peut s'exécuter un job MapReduce ?

A. Reduce-Map-Shuffle

B. Map-Reduce-Shuffle

C. Map-Shuffle-Combine-Reduce

D. Map-Combine-Shuffle-Reduce

Réponse : C - Map-Shuffle-Combine-Reduce

Justification : L'ordre d'exécution typique d'un job MapReduce est : Map (traitement parallèle des données), Shuffle (redistribution et tri des données intermédiaires), Combine (agrégation locale optionnelle), et Reduce (agrégation finale). Le Combine est optionnel mais améliore les performances.

Question 18

Dans le contexte de MapReduce, qu'est-ce que la phase de "Shuffle" ?

A. La phase de combinaison des données réduites dans les blocs de taille 128Mo.

B. La phase où les données sont partagées entre des mappers sans ordre de priorités.

C. b. La phase où les données sont effacées après la réduction.

D. d. La phase où les données sont transférées des mappers vers les DataNodes.

Réponse : D - La phase où les données sont transférées des mappers vers les DataNodes.

Justification : La phase Shuffle est le processus de transfert et de tri des données intermédiaires produites par les Mappers vers les Reducers appropriés. Elle regroupe toutes les valeurs ayant la même clé et les transfère au Reducer correspondant, assurant ainsi que chaque Reducer reçoit toutes les données nécessaires pour sa clé.

Question 19

Lors de la soumission d'un Job à traiter par un cluster Hadoop :

A. On doit le soumettre à partir du nœud Master sinon le job n'est pas considéré.

B. On peut le soumettre à partir de n'importe quel nœud du cluster.

C. On doit le soumettre à partir d'un nœud désigné car l'administrateur du cluster doit l'autoriser avant qu'il ne soit traité.

Réponse : A - On doit le soumettre à partir du nœud Master sinon le job n'est pas considéré.

Justification : Dans l'architecture Hadoop, les jobs sont généralement soumis au nœud Master (qui héberge le ResourceManager dans YARN) pour être traités. C'est le ResourceManager qui accepte les soumissions de jobs, les planifie et coordonne leur exécution sur le cluster.

Question 20

Quel avantage apporte YARN par rapport à son prédécesseur, le MapReduce v1 (MRv1) ?

A. Meilleure gestion aux pannes.

B. Capacité de prendre en charge divers modèles de traitement, pas seulement MapReduce.

C. Planification précoce des tâches.

D. Meilleures performances en termes de vitesse de traitement.

Réponse : B - Capacité de prendre en charge divers modèles de traitement, pas seulement MapReduce.

Justification : YARN sépare la gestion des ressources du modèle de programmation, ce qui permet d'exécuter différents frameworks de traitement (Spark, Flink, Tez, etc.) sur le même cluster Hadoop, pas uniquement MapReduce. Cela rend l'écosystème Hadoop beaucoup plus flexible et polyvalent.