Correction Détaillée du QCM Hadoop

HDFS, MapReduce et YARN

Fiche de révision générée par l'Assistant IA

Consignes de lecture

— Les réponses correctes sont indiquées par ✓.
— Les réponses incorrectes sont indiquées par ✗.
— Une explication suit chaque question pour comprendre le raisonnement.

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 distribué sur un cluster.

Analyse détaillée

Pourquoi c'est juste : HDFS (Hadoop Distributed File System) est conçu pour le stockage sur disque (et non en RAM) de fichiers très volumineux, répartis sur plusieurs machines (nœuds) du cluster.
Pourquoi les autres sont faux :

— **A :** HDFS utilise le disque dur, pas la RAM (c'est Spark qui privilégie la RAM).
— **C :** L'exécution des tâches est le rôle de MapReduce ou YARN, pas de HDFS qui ne s'occupe que du stockage.

Question 2

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

✗ A. Hadoop MapReduce
✗ B. Hadoop YARN
✓ **C. Hadoop NameNode**

Analyse détaillée

Pourquoi c'est juste : Le NameNode est le "maître" du système de fichiers. Il gère les métadonnées (l'arborescence des fichiers, les permissions, et la localisation des blocs sur les DataNodes).
Pourquoi les autres sont faux :

— **A :** MapReduce est un moteur de calcul.
— **B :** YARN gère les ressources du cluster (CPU/RAM).

Question 3

Quel est l'objectif du mécanisme de réplication des blocs de données dans le HDFS ?

✗ A. Améliorer les performances de lecture (pas l'objectif primaire).

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

✗ C. Réduire l'utilisation de l'espace disque.

Analyse détaillée

Pourquoi c'est juste : En copiant chaque bloc de données sur plusieurs machines (3 fois par défaut), Hadoop s'assure que si une machine tombe en panne, les données restent accessibles ailleurs.
Pourquoi les autres sont faux :

— **C :** La réplication *augmente* l'utilisation de l'espace disque (x3), elle ne la réduit pas.

Question 4

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

✗ A. 128 Ko

✗ B. 256 Mo

✗ C. 64 Mo

✓ **D. 128 Mo**

Analyse détaillée

Pourquoi c'est juste : Dans Hadoop 2.x et 3.x, la taille par défaut est de 128 Mo. Cela permet de minimiser le temps de recherche (seek time) par rapport au temps de transfert.
Note : Dans les très vieilles versions (Hadoop 1), c'était 64 Mo (Réponse C). Mais la question précise "dernière version".

Question 5

Quel avantage présente le HDFS par rapport à un système de fichiers traditionnel... ?

✗ A. Il est plus rapide pour les opérations de lecture/écriture (sur de petits fichiers).

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

✗ C. Il stocke les données en utilisant une structure relationnelle.

Analyse détaillée

Pourquoi c'est juste : HDFS est optimisé pour le débit (throughput) sur du matériel standard ("commodity hardware") qui peut tomber en panne fréquemment.
Pourquoi les autres sont faux :

— **A :** HDFS a une latence élevée ; il est moins rapide qu'un système classique pour les petits accès aléatoires.

— **C :** HDFS stocke des fichiers bruts, pas des tables relationnelles (c'est le rôle de Hive ou HBase par-dessus HDFS).

Question 6

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

✗ A. Les données sont restaurées à partir des DataNodes (Impossible, ils n'ont pas les métadonnées).

✗ B. Une copie de sauvegarde... prend automatiquement le relais (Vrai uniquement en mode Haute Disponibilité/HA, mais mal formulé ici comme "copie de sauvegarde").

✗ C. La perte est inévitable.

✓ **D. Les administrateurs devraient gérer le problème manuellement.**

Analyse détaillée

Analyse complexe :

— Dans un cluster Hadoop standard (sans HA), le NameNode est un point de défaillance unique (SPOF). S'il crashe, l'administrateur doit intervenir manuellement pour restaurer les métadonnées depuis une sauvegarde (FSImage) ou promouvoir le Secondary NameNode. C'est la réponse "scolaire" classique.

— *Note :* En production moderne avec Haute Disponibilité (HA), il y a un "Standby NameNode" qui prend le relais (Option B). Cependant, le terme "copie de sauvegarde" dans la réponse B est techniquement imprécis pour désigner un nœud Standby actif. La réponse D est souvent attendue dans les QCMs fondamentaux.

Question 7

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

✗ A. Le NameNode stocke les données... (Inverse).

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

✗ C. Les deux sont responsables du stockage des données.

Analyse détaillée

Moyen mnémotechnique :

— **Name**Node = Gère les **Noms** de fichiers (Métadonnées : qui est où ?).

— **Data**Node = Stocke la **Data** réelle (le contenu des fichiers).

Question 8

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

✗ A. Partitionner les données (c'est le Shuffle).

✗ B. Appliquer un filtrage (trop réducteur).

✗ C. Regrouper les données (c'est le Reduce).

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

Analyse détaillée

Définition : Le Mapper prend une donnée brute en entrée et émet une série de paires (K, V) (Clé, Valeur) qui seront ensuite triées.

Question 9

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

- ✗ A. Générer un fichier de sortie (C'est le résultat, mais pas la fonction logique).
- ✓ **B. Agréger les données intermédiaires produites par les mappers.**
- ✗ C. Lire les données d'entrée (rôle du Map).
- ✗ D. Combine... (C'est le rôle du Combiner ou du Shuffle).

> **Analyse détaillée**
>
> **Explication :** Le Reducer reçoit une clé et une liste de toutes les valeurs associées à cette clé $[K, (V1, V2, V3...)]$ pour les agréger (somme, moyenne, concaténation, etc.).

Question 10

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

- ✗ A. Gérer la distribution des données (rôle de HDFS).
- ✗ B. Allouer des ressources... et gérer les tâches (Incomplet ou mélange).
- ✗ C. Stocker des données (rôle de HDFS).
- ✓ **D. Il est responsable de la gestion des ressources et du suivi de l'exécution des Jobs.**

> **Analyse détaillée**
>
> **Pourquoi c'est juste :** YARN signifie *Yet Another Resource Negotiator*. Il sépare la gestion des ressources (CPU, RAM) de la logique de traitement (MapReduce, Spark, etc.).

Question 11

Quelle est la principale caractéristique de la tolérance aux pannes dans Hadoop ?

- ✓ **A. La duplication des données sur chaque nœud du cluster.**
- ✗ B. L'isolation des nœuds défectueux (C'est une conséquence, pas la méthode).
- ✗ C. La prévention... par l'alimentation électrique.

> **Analyse détaillée**
>
> **Correction sémantique :** Le terme exact est "Réplication" (sur différents nœuds), mais l'option A ("Duplication") est la seule qui décrit le concept de copier les données pour survivre à une panne.

Question 12

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

- ✗ A. Python (Possible via Hadoop Streaming, mais pas natif).
- ✓ **B. Java**
- ✗ C. C#
- ✗ D. SQL (Utilisé par Hive, pas pour écrire du MapReduce pur).

Analyse détaillée

Pourquoi c'est juste : Hadoop est écrit en Java. Bien qu'on puisse utiliser Python ou C++, Java reste le langage natif et le plus performant pour MapReduce.

Question 13

Quelle est la fonction principale du framework Hadoop MapReduce ?

- ✗ A. Stocker de grandes quantités (rôle de HDFS).
- ✗ B. Exécuter des opérations SQL (rôle de Hive).
- ✓ **C. Traiter et analyser des données massives en parallèle.**

Analyse détaillée

Pourquoi c'est juste : C'est le cœur du sujet. MapReduce divise un gros traitement en petits morceaux exécutés en parallèle sur le cluster.

Question 14

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

- ✗ A. HDFS
- ✗ B. MapReduce (C'est le modèle de programmation, pas le gestionnaire).
- ✓ **C. YARN (Yet Another Resource Negotiator)**
- ✗ D. Application Master (C'est un sous-composant géré par YARN pour une appli spécifique).

Analyse détaillée

Pourquoi c'est juste : Depuis Hadoop 2, YARN est le chef d'orchestre qui planifie qui fait quoi et quand.

Question 15

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

- ✗ A. hdfs dfs -get (Copie du HDFS VERS local).
- ✓ **B. hdfs dfs -put myfile.txt**
- ✗ C. hdfs dfs -cp (Copie de HDFS VERS HDFS).
- ✗ D. hdfs dfs -fromLocal (La commande exacte est `-copyFromLocal`, la syntaxe est fausse ici).

Analyse détaillée

Astuce : `put` = mettre (envoyer vers le cluster). `get` = prendre (récupérer du cluster).

Question 16

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

- ✗ A. Incomparables...
- ✓ **B. Le "Resource Manager" gère les ressources du cluster, tandis que le "Node Manager" gère les ressources individuelles sur un nœud.**
- ✗ C. Le "Node Manager" planifie... (Inverse).

> **Analyse détaillée**
>
> **Hiérarchie :**
>
> — **Resource Manager (RM) :** 1 seul par cluster (Le Chef).
>
> — **Node Manager (NM) :** 1 par machine (L'ouvrier qui surveille la RAM/CPU de sa propre machine).

Question 17

Dans quel ordre peut s'exécuter un job MapReduce ?

- ✗ A. Reduce-Shuffle-Map (Impossible).
- ✗ B. Map-Reduce-Shuffle (Le tri/shuffle doit se faire avant la réduction).
- ✗ C. Map-Shuffle-Combine-Reduce (Le Combine se fait localement avant le Shuffle pour optimiser).
- ✓ **D. Map-Combine-Shuffle-Reduce**

> **Analyse détaillée**
>
> **L'ordre logique :** 1. **Map** : Traitement initial. 2. **Combine** (Optionnel mais fréquent) : Pré-agrégation locale pour réduire le trafic réseau. 3. **Shuffle** : Transfert et tri des données à travers le réseau vers les Reducers. 4. **Reduce** : Agrégation finale.

Question 18

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

- ✗ A. La phase de combinaison...
- ✓ **B. La phase où les données mappées sont triées et réparties vers les réducteurs appropriés.**
- ✗ C. La phase où les données sont effacées...
- ✗ D. La phase de copie...

> **Analyse détaillée**
>
> **Définition :** Le "Shuffle and Sort" est la magie de Hadoop. C'est le moment où toutes les valeurs ayant la même clé (venant de différents mappers) sont envoyées physiquement vers le même Reducer.

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 (Faux, mauvaise pratique).

✓ **B. On peut le soumettre de n'importe quel nœud du cluster (configuré comme client/gateway).**

✗ C. On doit le soumettre à partir d'un nœud distant...

Analyse détaillée

Explication : Un job est soumis via un "Hadoop Client". Ce client peut être installé sur n'importe quel nœud (souvent un "Edge Node" ou une passerelle) tant qu'il a accès aux fichiers de configuration du cluster pour contacter le Resource Manager.

Question 20

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

✗ A. Meilleure tolérance aux pannes (Déjà présent avant).

✗ B. Planification précoce...

✓ **C. Capacité de prendre en charge divers modèles de traitement, pas seulement MapReduce.**

✗ D. Meilleures performances... (Vague).

Analyse détaillée

Le grand changement : Avec MRv1, on ne pouvait faire QUE du MapReduce. Avec YARN, on peut faire tourner Spark, Storm, HBase, Flink sur le même cluster en même temps. YARN est devenu un "Système d'exploitation de données".

Résumé Visuel : Architecture Hadoop v2