Projet - Concevoir un pipeline Hadoop résilient et dimensionné

- **Module :** Big Data et traitement massif
- **Durée :** 6 heures
- **Organisation :** équipes de 2 personnes
- **Fichiers fournis :** github.com/supavenir/hadoop-td3 (git clone https://github.com/supavenir/hadoop-td3)
- **Environnement :** Hadoop sur Docker Compose (cluster fourni dans le dossier `environnement/` : celui du TD1, avec Hive et Python en plus)
- **Technologies autorisées :** HDFS, YARN, MapReduce Streaming et/ou Hive
- **Technologie exclue pour ce projet :** Spark, traité dans l'autre partie du module
- **AWS :** interdit pour ce projet (aucun risque de facturation)

1. Contexte : le lendemain de la grande promotion

Vous rejoignez l'équipe data de **MarchéPlus**, une enseigne de commerce en ligne. Après une grande campagne promotionnelle, la direction veut recevoir chaque matin un rapport fiable avant l'ouverture des équipes commerciales.

Aujourd'hui, les données sont traitées par un script exécuté sur une machine unique. Le rapport est parfois en retard, les relances produisent des résultats incohérents et personne ne sait combien de temps les données doivent rester disponibles.

Votre équipe doit proposer et démontrer une solution Hadoop pour ce traitement. Il ne s'agit pas seulement de faire fonctionner un job : vous devez **faire des choix d'architecture, les justifier et expliciter leurs limites**.

Besoin métier

Chaque nuit, **trois flux distincts** sont déposés par trois systèmes sources différents :

- **les commandes**, issues de la plateforme de vente ;
- **les consultations de produits**, issues du site web (journaux de navigation) ;
- **les annulations et retours**, issus du service client.

Le traitement doit produire, pour chaque journée :

- le chiffre d'affaires net par catégorie (commandes moins annulations et retours) ;
- le nombre de commandes par catégorie ;
- les cinq produits les plus consultés par catégorie ;
- une liste des anomalies de données à examiner.

Les équipes métier veulent consulter les résultats **avant 6 h**. Une journée de données peut être corrigée ou complétée le lendemain : le pipeline doit donc permettre une reprise ou un recalcul sans publier silencieusement un résultat incohérent.

Contraintes communiquées par l'entreprise

- En moyenne, **120 Go de données compressées** arrivent chaque nuit, tous flux confondus.
- Pendant les promotions, le volume peut atteindre **1,5 fois le volume moyen**.
- Les fichiers arrivent entre minuit et 3 h ; le traitement doit se terminer avant 5 h 30, afin de laisser une marge avant la publication.
- Les trois flux n'arrivent pas à la même heure et **l'un d'eux peut manquer ou arriver incomplet**.
- Les données brutes doivent être conservées **90 jours**.
- Les données intermédiaires de calcul ne sont nécessaires que pendant le traitement et les reprises.
- Les résultats agrégés doivent rester disponibles pendant **deux ans**.
- Les identifiants clients ne doivent pas apparaître dans les rapports métier.
- L'équipe dispose d'un budget limité et ne veut pas laisser un cluster actif en permanence.
- Une interruption d'un nœud ne doit pas entraîner automatiquement la perte définitive des données ou du résultat.

À vous d'identifier les ambiguïtés.

Le cahier des charges ne précise pas, par exemple, si les fichiers tardifs modifient les rapports déjà publiés, ni quelle règle métier appliquer à une commande annulée après la clôture de la journée, ni ce qu'il faut faire si le flux d'annulations n'est pas arrivé à 5 h. Formulez des hypothèses explicites plutôt que d'ignorer ces questions.

2. Données mises à disposition

Un jeu de données réduit est fourni, **volontairement bien plus petit que les volumes réels**. Il se compose de trois fichiers CSV (avec en-tête), plus un lot correctif, dans le dossier `donnees/`.

Schémas des trois flux

Flux 1 : `commandes_AAAAMMJJ.csv` (une ligne par ligne de commande)

```
commande_id,ligne_id,client_id,produit_id,categorie,quantite,prix_unitaire,devise,s
tatut,horodatage_commande
```

Flux 2 : `consultations_AAAAMMJJ.csv` (une ligne par page produit vue)

```
session_id,client_id,produit_id,categorie,url,horodatage_vue
```

Flux 3 : `annulations_AAAAMMJJ.csv` (une ligne par annulation ou retour)

```
annulation_id,commande_id,ligne_id,type,motif,quantite,montant_rembourse,horodatage
_annulation
```

Lot correctif : `corrections_AAAAMMJJ.csv` (une ligne par correction, livré le lendemain de la journée AAAAMMJJ qu'il corrige)

```
commande_id,ligne_id,client_id,produit_id,categorie,quantite,prix_unitaire,devise,s
tatut,horodatage_commande,type_correction,correction_id,horodatage_correction
```

Les dix premières colonnes reprennent le schéma du flux 1 : le lot correctif ne porte que sur les commandes. `type_correction` vaut `AJOUT` (ligne de commande absente du lot initial), `MODIFICATION` (nouvelle version d'une ligne de commande existante) ou `SUPPRESSION` (ligne de commande à retirer). `horodatage_correction` indique quand la correction a été émise.

Remarques sur les schémas :

- `type` vaut `ANNULATION` (avant expédition) ou `RETOUR` (après livraison) ;
- le flux 3 **ne contient pas la catégorie** : il faut la retrouver en rattachant l'annulation à la ligne de commande correspondante (**jointure**) ;
- le flux 2 n'a **aucun lien avec les commandes** : il alimente uniquement le top 5 ;
- `client_id` est présent dans les flux 1 et 2 ; il est **interdit** dans les sorties métier.

Qualité des données

Comme ceux que livrent les systèmes sources réels, ce jeu de données n'est pas propre. Explorez-le avant de coder : c'est à vous d'identifier les problèmes qu'il contient et de décider comment les traiter.

Pour vos tests (fichier corrompu, flux manquant, lignes ajoutées), travaillez sur des copies : les résultats que vous présentez doivent être calculés sur le jeu fourni, sans modification.

Règles métier à définir

Avant de coder, décidez et documentez :

- comment calculer le chiffre d'affaires net (prix, quantité, devise, annulation partielle ou totale, retour) ;
- quelle clé identifie de manière unique une ligne de commande, et comment traiter les doublons ;
- que faire des lignes invalides ou incomplètes ;
- que faire d'une annulation orpheline ;
- à quelle journée rattacher une annulation (celle de la commande ou celle de l'annulation) ;
- comment classer les événements tardifs ;
- que faire si un flux est absent ou incomplet au moment du traitement ;
- comment départager les produits à égalité dans le top 5 ;
- quelles données peuvent apparaître dans les sorties.

Il n'y a pas nécessairement une seule réponse possible. En revanche, une règle non documentée ne pourra pas être considérée comme un choix maîtrisé.

3. Votre mission

Concevez une solution Hadoop qui répond au besoin de MarchéPlus, puis démontrez-en une version réduite sur l'environnement disponible.

Paliers de réalisation

Les paliers sont cumulatifs. **Le palier 1 est obligatoire** ; les suivants se valorisent dans la grille d'évaluation.

Palier	Contenu	Statut
1	Nettoyage des flux, **jointure commandes/annulations**, CA net par catégorie, sortie des rejets, un contrôle de cohérence	Obligatoire
2	Nombre de commandes par catégorie, mécanisme de publication fiable, test d'échec et de relance	Attendu
3	Top 5 des produits consultés par catégorie, traitement du lot correctif, tests de montée en charge	Extension

A. Architecture et parcours des données

Définissez le chemin des données depuis leur arrivée jusqu'à la publication des résultats.

Précisez notamment :

- où se trouvent les données brutes ;
- quelles données sont stockées dans HDFS et pendant combien de temps ;
- où se trouvent les résultats qui doivent survivre à la suppression du cluster ;
- quels composants interviennent dans le traitement ;
- à quel moment un résultat peut être considéré comme publiable.

B. Stockage et capacité

Estimez le volume à stocker **dans l'espace de travail HDFS** (le stockage à 90 jours et à deux ans est traité séparément, voir section D), en tenant compte :

- du volume moyen et du volume de pointe ;
- du **taux de décompression** (les 120 Go sont compressés : posez une hypothèse explicite) ;
- de la durée de conservation retenue pour les données de travail ;
- du facteur de réplication HDFS ;
- de la marge de capacité que vous souhaitez garder.

Expliquez pourquoi **une réplication HDFS ne suffit pas à assurer la persistance des données après la destruction du cluster**.

C. Traitement et qualité des données

Implémentez au minimum (palier 1) :

- le nettoyage et la validation de chaque flux ;
- **la jointure entre les lignes de commande et les annulations/retours**, afin de rattacher la catégorie et de calculer le net ;
- une agrégation du chiffre d'affaires net par catégorie ;
- un traitement des lignes invalides, avec une sortie permettant de les examiner ;
- une vérification de la cohérence d'au moins un résultat (par exemple : brut − annulations = net, ou comptage des lignes en entrée = lignes valides + rejetées).

Le traitement peut utiliser **MapReduce Streaming**, **Hive**, ou une combinaison des deux. Expliquez pourquoi vous avez retenu cette approche, et **comment vous avez réalisé la jointure** (par exemple jointure côté reducer en Streaming, ou `JOIN` Hive) ainsi que son coût (données déplacées entre les nœuds, risque de déséquilibre si quelques clés concentrent beaucoup de lignes).

Le nettoyage des données est attendu en **MapReduce Streaming**, sauf justification contraire dans le registre des décisions.

Calcul de contrôle, avant votre premier job. Copiez les deux extraits ci-dessous dans `extrait_commandes.csv` et `extrait_annulations.csv`. Ouvrez-les dans un tableur (Google Sheets, LibreOffice Calc…) et, en appliquant **vos** règles métier, calculez pour chaque catégorie le CA brut, les annulations déduites, le CA net et le nombre de commandes, ainsi que le nombre d'annulations orphelines. Lancez ensuite vos jobs ou vos requêtes Hive sur ces extraits et comparez : tout écart doit être compris et corrigé avant de passer au jeu complet.

```
commande_id,ligne_id,client_id,produit_id,categorie,quantite,prix_unitaire,devise,s
tatut,horodatage_commande
CMD1073051,1,CL008566,P00097,electromenager,1,117.90,EUR,PAYEE,2026-11-15T00:38:22
CMD1073061,1,CL009872,P00108,electromenager,3,70.49,EUR,PAYEE,2026-11-15T01:03:16
CMD1073061,2,CL009872,P00039,informatique,1,29.90,EUR,EXPEDIEE,2026-11-15T01:03:16
CMD1073089,1,CL007415,P00105,electromenager,1,135.90,EUR,PAYEE,2026-11-15T01:49:01
CMD1073089,2,CL007415,P00127,maison,1,10.99,EUR,PAYEE,2026-11-15T01:49:01
CMD1073093,1,CL008035,P00105,electromenager,1,135.90,EUR,PAYEE,2026-11-15T01:53:44
CMD1073104,1,CL014189,P00108,electromenager,3,70.49,EUR,LIVREE,2026-11-15T02:13:00
CMD1073104,2,CL014189,P00260,mode,1,20.99,EUR,ANNULEE,2026-11-15T02:13:00
CMD1073337,1,CL000029,P00110,electromenager,1,185.49,EUR,LIVREE,2026-11-15T07:12:37
CMD1073337,2,CL000029,P00261,mode,2,8.90,EUR,PAYEE,2026-11-15T07:12:37
CMD1074687,1,CL000014,P00125,maison,2,5.95,EUR,PAYEE,2026-11-15T12:34:48
CMD1074687,2,CL000014,P00261,mode,2,8.90,EUR,EXPEDIEE,2026-11-15T12:34:48
CMD1074687,3,CL000014,P00251,mode,2,10.49,EUR,PAYEE,2026-11-15T12:34:48
CMD1074687,4,CL000014,P00009,informatique,1,23.95,EUR,PAYEE,2026-11-15T12:34:48
CMD1075106,1,CL007765,P00040,informatique,4,129.90,EUR,PAYEE,2026-11-15T13:31:47
```

```
annulation_id,commande_id,ligne_id,type,motif,quantite,montant_rembourse,horodatage
_annulation
ANN0196112,CMD1073337,1,RETOUR,taille_inadaptee,1,185.49,2026-11-15T09:20:38
ANN0196652,CMD1073104,2,ANNULATION,changement_avis,1,20.99,2026-11-15T13:56:54
ANN0198204,CMD1075106,1,ANNULATION,paiement_modifie,1,129.90,2026-11-15T14:29:51
ANN0198163,CMD1074687,5,ANNULATION,paiement_modifie,1,23.95,2026-11-15T15:42:55
ANN0198196,CMD1075106,1,ANNULATION,changement_avis,1,129.90,2026-11-15T17:47:52
```

N'utilisez pas Spark pour cette séance.

D. Relance et publication des résultats

Persistance hors cluster. Le dossier `persistant/` de la machine hôte simule le stockage objet durable (type S3). Vous devez démontrer qu'un résultat validé survit à la destruction du cluster (`docker compose down -v`, recréation, restauration).

Décrivez ce qui se passe si :

- le job échoue à mi-parcours ;
- le job est relancé avec le même lot ;
- des fichiers de sortie existent déjà ;
- un des trois flux est absent, vide ou incomplet ;
- un fichier tardif ou corrigé arrive le lendemain.

Proposez un mécanisme qui évite qu'un résultat incomplet ou périmé soit présenté comme le résultat officiel

(par exemple : sorties temporaires et sorties validées, renommage atomique, fichier marqueur de validation, versionnage par lot). Expliquez votre logique.

Test d'échec obligatoire (palier 2). Provoquez au moins une défaillance réelle et documentez le comportement observé : arrêt d'un DataNode, interruption d'un job avec `yarn application -kill`, fichier d'entrée corrompu ou flux manquant. Vous devez montrer ce qui se passe, pas seulement le décrire.

E. Dimensionnement et montée en charge

Proposez une configuration cible pour le scénario réel, même si vous ne pouvez tester que sur un petit cluster.

Distinguez les rôles des nœuds et expliquez :

- dans quel cas vous ajouteriez des nœuds de calcul ;
- dans quel cas vous ajouteriez des nœuds qui stockent les blocs HDFS ;
- pourquoi ces deux décisions ne sont pas équivalentes ;
- quels indicateurs vous surveilleriez pour vérifier que le dimensionnement convient ;
- quelles limites pourraient apparaître lorsque le volume augmente (NameNode, petits fichiers, jointure déséquilibrée, entrées-sorties disque ou réseau).

Faites apparaître **au moins un calcul de capacité et un calcul de débit** (par exemple : quel débit de traitement faut-il pour absorber le volume de pointe dans la fenêtre disponible ?). Indiquez les hypothèses utilisées et précisez pourquoi une extrapolation à partir du jeu de données réduit reste approximative.

Pour tester la montée en charge (palier 3), fabriquez un jeu plus volumineux à partir des fichiers fournis, par exemple en les dupliquant. Une simple concaténation transforme chaque copie en doublon de la précédente : rendez les identifiants uniques dans chaque copie, ou expliquez l'effet de ce choix sur vos mesures. Ce jeu sert à mesurer des durées et des débits, pas à produire des résultats métier.

4. Contraintes de réalisation

- Travaillez avec l'environnement local fourni (Docker Compose).
- **N'utilisez pas AWS** pour ce projet.
- Le cluster de démonstration ne doit pas être présenté comme équivalent à une infrastructure de production.
- Ne vous contentez pas d'un schéma théorique : votre pipeline doit produire un résultat vérifiable sur le jeu de données réduit.
- Vous pouvez vous répartir le travail, mais chaque membre de l'équipe doit être capable d'expliquer l'architecture, la jointure et les limites de la solution.

5. Usage de l'IA générative



Première phase sans IA. L'exploration des données, les hypothèses, le schéma d'architecture et le calcul de contrôle sur l'extrait se font **sans assistant IA**, y compris celui intégré au tableur, avec les outils du shell (`head`, `wc`, `grep`, `awk`), un tableur, le sujet et votre binôme. Faire valider avant de commencer l'implémentation.

Pourquoi : c'est la partie du projet où l'on apprend le plus, et celle qu'une IA ferait le plus volontiers à votre place. Repérer vous-mêmes ce qui ne va pas dans les données, et prévoir le résultat avant de lancer un job, c'est ce qui vous permettra ensuite de juger si ce que produit votre pipeline, ou une IA, est juste.

Après cette phase, l'usage ponctuel d'un assistant IA est autorisé, comme il le serait dans un contexte professionnel. En revanche :

- **vous restez responsables** de tout ce que vous livrez : un script produit par une IA que vous ne savez pas expliquer sera considéré comme non maîtrisé ;
- **les choix d'architecture et les règles métier doivent être les vôtres** : l'IA peut servir à explorer des options ou à débloquer une syntaxe, pas à décider à votre place ;
- **vérifiez systématiquement** ce qu'elle propose (versions, options Hadoop, comportement réel sur votre cluster) ; une réponse plausible peut être fausse ;
- ne transmettez pas de données sensibles ;
- **tracez votre usage** : une demi-page indiquant pour quelles tâches vous avez utilisé l'IA, ce que vous avez corrigé ou rejeté, et une erreur ou imprécision que vous avez relevée dans ses réponses.

6. Étapes conseillées

1. **Sans IA :** lire le cas, explorer les données, relever les ambiguïtés et formuler les hypothèses.
2. **Sans IA :** dessiner l'architecture, choisir l'approche pour la jointure et répartir les tâches.
3. **Sans IA :** calculer dans un tableur le résultat attendu sur l'extrait (section C).
4. Point avec l'enseignant : hypothèses, schéma et calcul de contrôle.
5. Mettre en œuvre le traitement, d'abord sur l'extrait, puis sur le jeu complet (palier 1, puis 2).
6. Tester les erreurs, la relance et la cohérence des résultats.
7. Estimer la capacité, discuter le scaling et les limites.
8. Finaliser le dossier et préparer la restitution.
9. Présentation et questions individuelles.

Vous pouvez organiser votre travail autrement, mais ne sacrifiez pas les tests et la restitution : un pipeline qui tourne sans test de reprise ni dossier ne couvre que le palier 1.

7. Livrable

Remettez un dossier court comprenant :

1. un schéma d'architecture montrant les données, les stockages et les traitements ;
2. un registre des hypothèses et décisions ;
3. les scripts, requêtes ou commandes nécessaires pour reproduire le traitement ;
4. le calcul de contrôle sur l'extrait (tableur) et sa comparaison avec la sortie de vos jobs ;
5. une preuve d'exécution : sortie du job, nombre de lignes traitées par flux, lignes rejetées, résultat métier obtenu ;
6. la description de la jointure et la vérification de sa justesse (par exemple : nombre d'annulations rattachées et nombre d'orphelines) ;
7. un test de reprise ou d'échec, accompagné d'une explication du comportement observé ;
8. une estimation de capacité et de dimensionnement, avec vos hypothèses ;
9. une synthèse précisant ce qui devrait changer avant un déploiement réel ;
10. la note d'usage de l'IA (une demi-page).

Registre des hypothèses et décisions

Sujet	Hypothèse ou choix	Justification	Conséquence ou limite
Fichier tardif			
Doublons et clé d'unicité			
Annulation orpheline			
Rattachement d'une annulation à une journée			

Sujet	Hypothèse ou choix	Justification	Conséquence ou limite
Flux manquant ou incomplet			
Choix de la jointure			
Données conservées dans HDFS			
Relance du job			
Publication du résultat			
Dimensionnement			

Restitution orale

Votre présentation doit répondre à la question suivante :

Quelle décision de votre équipe vous semble la plus importante pour éviter un résultat erroné ou une dépense inutile, et pourquoi ?

Chaque membre pourra être interrogé individuellement sur n'importe quelle partie de la solution.

8. Critères de performance

Critère	Insuffisant	Satisfaisant	Excellent
Cohérence de l'architecture	Schéma générique, composants mal reliés	Parcours complet et cohérent avec les contraintes	Choix justifiés par les contraintes, alternatives discutées
Compréhension de Hadoop	Confusion entre rôles HDFS, YARN, moteur de calcul	Rôles correctement distingués	Impact de chaque composant sur la reprise et le scaling expliqué
Qualité du traitement	Résultats non vérifiables, rejets ignorés	Résultats vérifiés, rejets tracés, jointure correcte	Contrôles de cohérence multiples, orphelins et écarts expliqués
Gestion de la reprise	Relance écrase ou duplique les sorties	Sorties temporaires/validées, relance idempotente	Échec réellement testé, flux manquant et lot correctif gérés
Dimensionnement	Chiffres sans hypothèses	Calculs de capacité et de débit expliqués	Limites identifiées (NameNode, déséquilibre de jointure, I/O) et marges justifiées
Prise de recul	Prototype présenté comme production	Limites du prototype listées	Plan chiffré et priorisé avant déploiement réel
Maîtrise et usage de l'IA	Code non expliqué, aucune trace	Usage tracé, sorties vérifiées	Erreurs de l'IA identifiées et corrigées, décisions clairement humaines
Défense des choix	Un seul membre répond	Chacun explique sa partie	Chacun explique l'ensemble et les compromis



Une solution qui traite le petit fichier le plus rapidement possible n'est pas nécessairement la meilleure. Vous serez aussi évalués sur votre capacité à distinguer **stockage temporaire et persistant**, **calcul et stockage**, **résultat calculé et résultat validé**, et à expliquer les compromis retenus.

From:
http://slamwiki2.kobject.net/ - **SlamWiki 2.1**

Permanent link:
http://slamwiki2.kobject.net/eadl/bloc5/fm3/td3

Last update: **2026/11/05 00:45**