OPENCLASSROOMS

CentraleSupélec

Déployez un modèle dans le cloud

Formation Data Scientist - Janvier / Novembre 2022

Auteur: Eric TREGOAT

Mentor: Benjamin TARDY

Evaluateur: Patrick KAMNANG WANKO

Ordre du jour de la soutenance

❑ Présentation

- Rappel de la problématique et présentation du jeu de données
- Présentation de la réalisation de la chaîne de traitement des images dans un environnement Big Data dans le cloud
  - les différentes briques d'architecture choisies sur le cloud
  - leur rôle dans l'architecture Big Data
  - les étapes de la chaîne de traitement
- Conclusion et recommandations

❑ Questions-réponses

❑ Débriefing

Problématique et présentation du jeu de données

Problématique

- Reconnaissance de fruit à partir d'image
- Construction de l'architecture big data nécessaire

Mission

- Chaine de traitement de données dans un environnement big data
  - Preprocessing pour extraire les features des images
  - Réduction de dimension pour produire un dataset réduit
- Capacité de mise à l'échelle des données

Jeu de données

- Base de photos de fruits
- Photos couleur de dimension 100 x 100 pixels
- Vues sous de multiples angles

Echantillon d'images pour le projet

- 3 fruits (corn, mandarine, raspberry), chacun faisant l'objet d'un répertoire à son nom
- Pour chaque fruit / répertoire, 30 images jpeg de dimension 100 x 100 à différents angles de vue (échantillonnage périodique 30/nbr_images)

Architecture Big Data du projet

❑ Concentrée sur la couche « batch layer »

❑ Mise en œuvre dans le cloud sur AWS :

- **Région eu-west-3** (Paris) pour minimiser la latence
- **S3**: stockage des données d'entrée (images) et de sortie (dataset)
- **Cluster EMR** basé sur Spark / Hadoop: calculs de constitution du dataset
- **Instances EC2** uniformes à usage général m5.xlarge (4 vcpu, 16GB de RAM et 64 GB stockage local EBS)
  → 1 nœud maitre (driver)
  → 2 nœud esclaves (workers)

❑ Sécurité

- **IAM** : utilisateur avec accès administrateur pour le groupe OCP8 → paire de clés dans USER/.aws/credentials (local)
- **Paire de clés EC2 sur la région**

❑ Langage Python, notebook jupyter

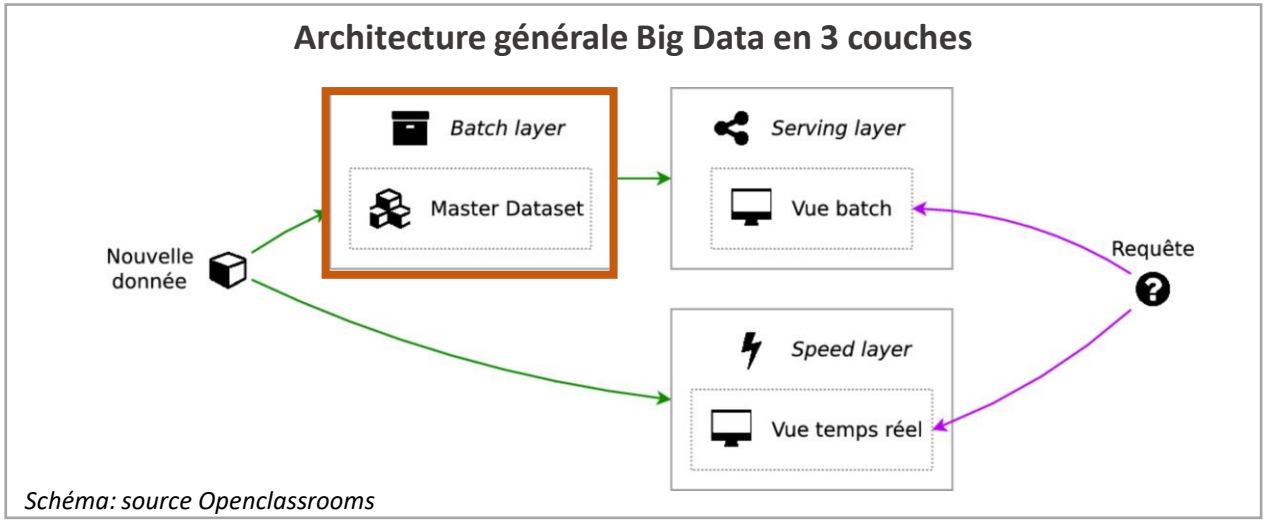
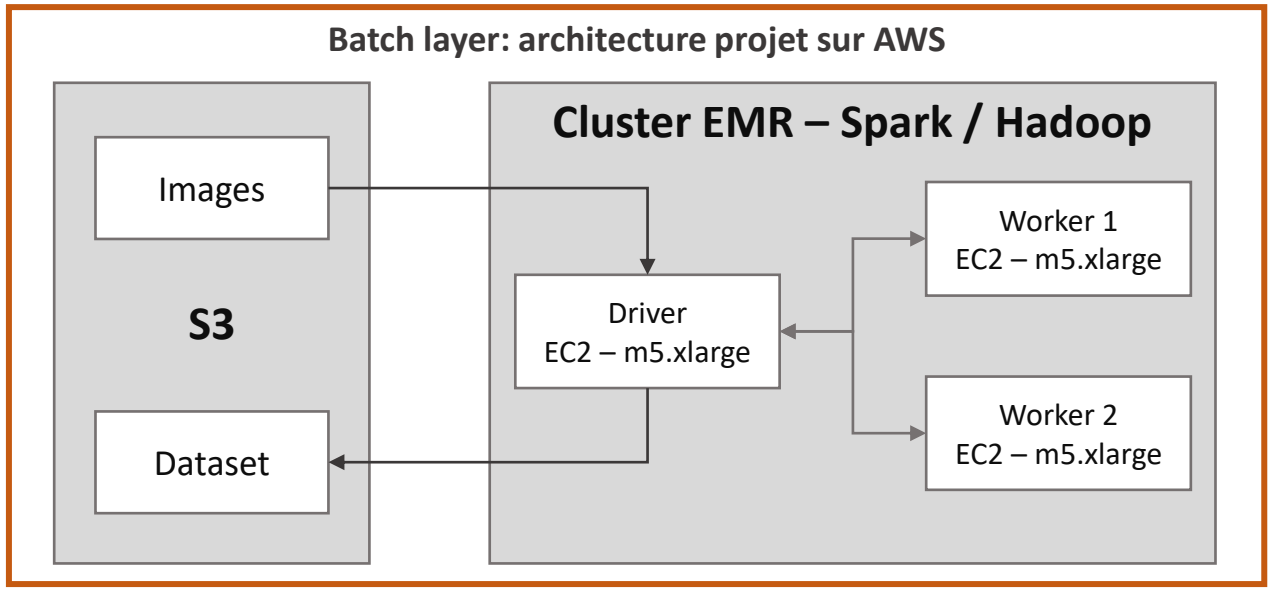


Schéma: source Openclassrooms



OPENCLASSROOMS
CentraleSupélec

Architecture logicielle du cluster

- **EMR version 6.8.0 la plus récente disponible**
  - Hadoop 3.2.1
  - Spark 3.3.0
- **Inclusion des librairies pour Jupyter notebook:**
  - JupyterEnterpriseGateway 2.1.0
  - Livy 0.7.1
- **Installation de librairies complémentaires à l'amorçage:**
  - Tensorflow==2.10.0
  - Pillow==9.2.0
  - Pyarrow==9.0.0
  - Boto3==1.25.1

Configuration de la session spark

- **Objet**
  - Configuration différenciée driver vs worker
  - Recherche du meilleur parti de la capacité des instances
  - Configuration avec Livy pour notebook (pour script: fichier spark-defaults.conf ou option de spark-submit)
  - Anticipation de la mise à l'échelle
- **Driver**
  - spark.driver.cores 1
  - spark.driver.memory 8g (max = 8.64g)
  - spark.driver.maxResultSize 0 (spark-defaults.conf)
- **Workers**
  - spark.executor.instances 2
  - spark.executor.cores 4 (/ worker instance)
  - spark.executor.memory 1g (/ process, max ~ 2g)

Capacité instance m5.xlarge:
- 4 vcpu
- 16 GB de RAM

Memory heap / instance

Spark memory

spark.memory.fraction
0.6 * (heap - memory overhead)

Storage (50%) + execution (50%)

User memory

Memory overhead (≥ 384 MB)

memoryOverheadFactor :
0.1 * instance memory > 384 MB

AWS spark-defaults.conf :
spark.yarn.executor.memoryOverheadFactor = 0.1875

Chaine de traitement

N°	Traitement	Sortie
0	Configuration et lancement de la session spark	Session spark configurée
1	Lecture des données d'entrée	Liste des objets (images) en entrée
2	Fonctions de création du modèle CNN et d'extraction des features	Modèle CNN EfficientNetB0 Fonction d'extraction des features
3	Extraction itérative des features des images	Dictionnaire d'encodage des labels Dataframe des labels encodé y Dataframe des features X
4	Enregistrement du dataset dans S3	Dictionnaire → json Dataframes → parquet
5	Standardisation des features de X et enregistrement du modèle dans S3	scaler → enregistrement dans S3 avec 'write' Dataframe X avec features standardisées
6	Réduction de dimension avec un PCA a) Modèle PCA et fit sur X standardisé b) Enregistrement dans S3 du dataset-features réduit Xpca c) Variance expliquée: vecteur + somme, et enregistrement dans S3 d) Enregistrement du modèle PCA	 Xpa → parquet Enregistrement variance → pickle Pca → enregistrement dans S3 avec 'write'

Conclusion et recommandations

❑ Faisabilité de réaliser la problématique

établie avec le démonstrateur en objet de ce projet

❑ Recommandations et points d'attention pour le passage à l'échelle

- Clarifier l'organisation et mettre en place des droits IAM correspondants
- Effectuer une acquisition itérative des images (<1000 à la fois avec boto3) pour constituer progressivement le dataset d'entrée
- Limiter le nombre de features extraites des images (ajout de la couche de GlobalAveragePooling2D à revoir en fonction du machine learning) car leur nombre impacte le besoin mémoire du driver pour les opérations de sérialisation
- La taille mémoire du driver est un point d'attention qui dépendra des opérations de sérialisation qui seront effectuées et la possibilité ou pas d'effectuer les traitements par batch
- Pas de difficulté anticipée pour la mise à l'échelle horizontale (nombre d'exécuteurs), à mettre en regard du temps d'exécution et des coûts
- Cluster EMR (instances) à faire évoluer (ou rendre scalable) avec l'évolution des besoins
- Pas de difficulté anticipée sur le stockage des images en entrée et des données de sortie
- PCA: nombre de features (réduites) à revoir en fonction de la variance expliquée (fonction du nombre d'images)

Echanges avec l’évaluateur

- **Questions-réponses**
- **Débriefing**

OPENCLASSROOMS CentraleSupélec

Contact: Eric TREGOAT

eric.tregoat@gmail.com

06 49 99 79 59

https://www.linkedin.com/in/erictregoat/