PROJET 8 – « DÉPLOYEZ UN MODÈLE DANS LE CLOUD »

Soutenance de projet – parcours Data Scientist

25 Avril 2020

Sommaire

I - PROBLÉMATIQUE

Rappel de la problématique

Présentation du jeu de données

Problématique

- **Fruits! :** Startup AgriTech
- **Produits :**
  - Application smartphone grand public de reconnaissance de fruit et affichage d'informations
  - Développement de robots cueilleurs intelligents



- **Objectif :** Mettre en place l'architecture Big Data
  - Preprocessing et réduction de dimension
  - Anticipation du passage à l'échelle futur dans un contexte d'adoption massive
- **Moyens :** Scripts pyspark + solution évolutive

Jeu de données

- **Origine:**
  - Images de fruits et labels associés (Fruits 360, *Mihai Oltean*)
  - 120 variétés de fruits différents (un dossier par variété)
  - Plusieurs variétés du même fruit (exemple : pomme « red » et « golden »)

- **Caractéristiques :**
  - Images 100x100 JPEG RGB
  - Photos studio sur fond blanc de fruits centrée sur le fruit
  - Photos sous tous les angles (timelapse + rotation 3 axes)

- **Jeu d'entraînement :** 53 000 images
- **Jeu de Test :** 18 000 images
- **Jeu multi fruits non labellisé :** 103 images

Banana

Apple Red 1

« Ratés »

II – LE BIG DATA

Qu’est-ce que le big data

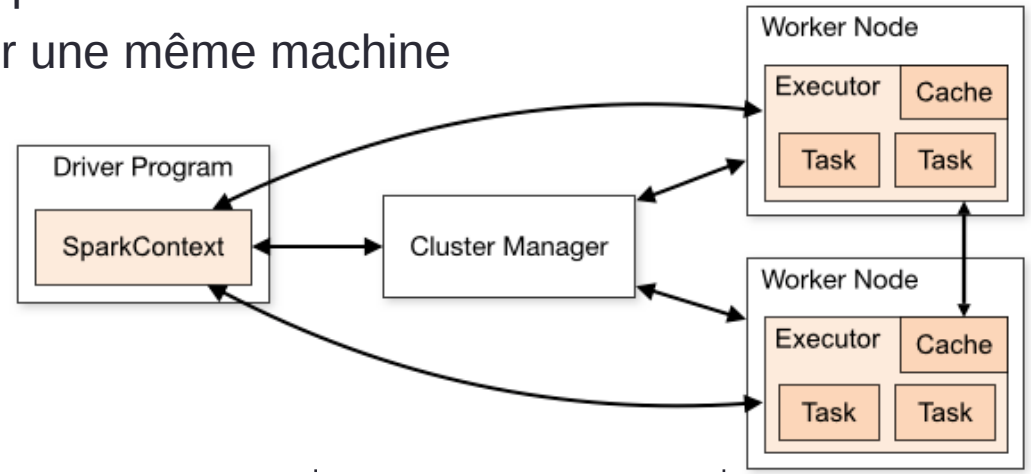
Comment répondre à ses enjeux?

Qu’est-ce que le Big Data ?

- En Français : les données massives
- Les **« V » :**
  - **Volume :** trop important pour être stocké et/ou traité sur une seule machine avec des performances acceptables.
    - Dépassement de la capacité de RAM
    - Dépassement des capacités de stockage
    - Etc.
  - **Vitesse** à laquelle les données sont produites
  - Large **Variété** de types de données
  - Etc.

Comment répondre à ces enjeux?

- **Stockage : système de fichier distribué** (ex : HDFS)
- **Capacités de calcul : Traitement par calculé distribué (MapReduce)**
  - Diviser les opérations en micro opérations distribuables entre différentes machines, réalisables en parallèle
  - Aggréger les résultats sur une même machine



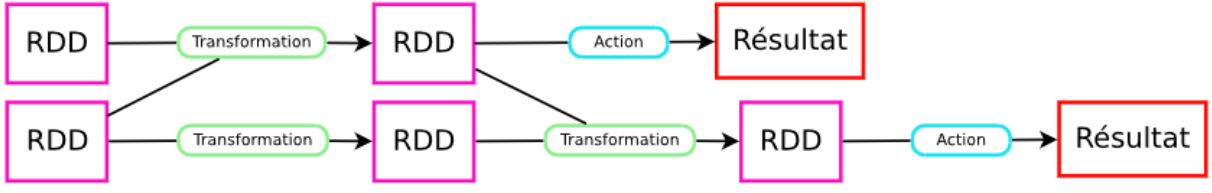
Application maître :
Configuration / Initialisation
Aggrégation des calculs

Cluster Manager :
Gestion des ressources
Distribution des calculs entre les workers

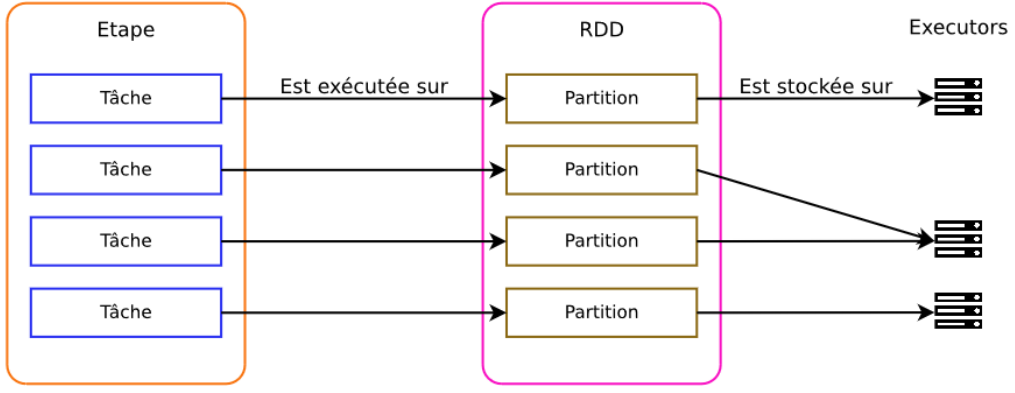
Workers :
Exécution des tâches en parallèle

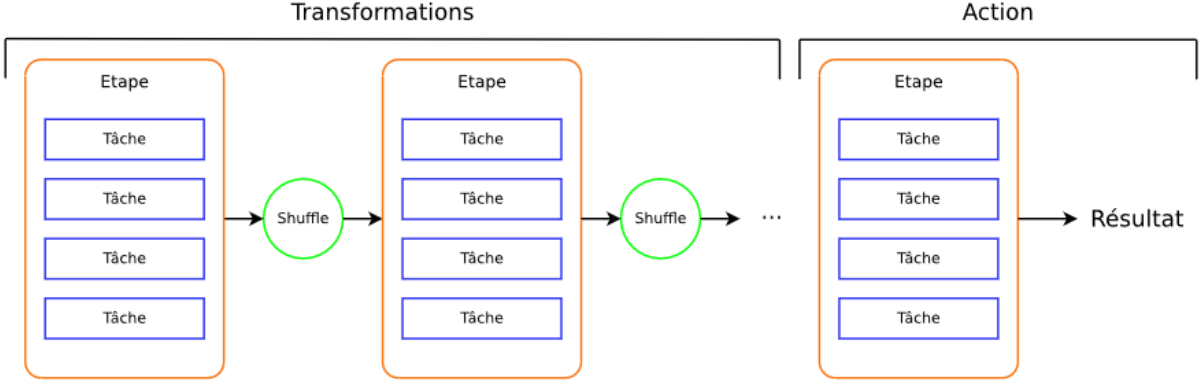
Comment répondre à ces enjeux?

- **Tolérance aux pannes**
  - Utilisation de Resilient Distributed Datasets (RDD)
    - Division des données en partition
    - Duplication des données (3 machines par défaut)
  - Graphe Acyclique Orienté (DAG) :
    - Panne : Régénération à partir des noeuds parents
    - Noeuds (RDD ou Résultats) : liés par des actions et transformations



Comment répondre à ces enjeux?





Shuffle = redistribution des données entre les noeuds

III – ARCHITECTURE RETENUE ET CHAÎNE DE TRAITEMENT

Quel prétraitement?

Objectif : préparer les algorithmes pour le learning

Réduction de dimensions

Extraction d'information des images

Solutions envisageables

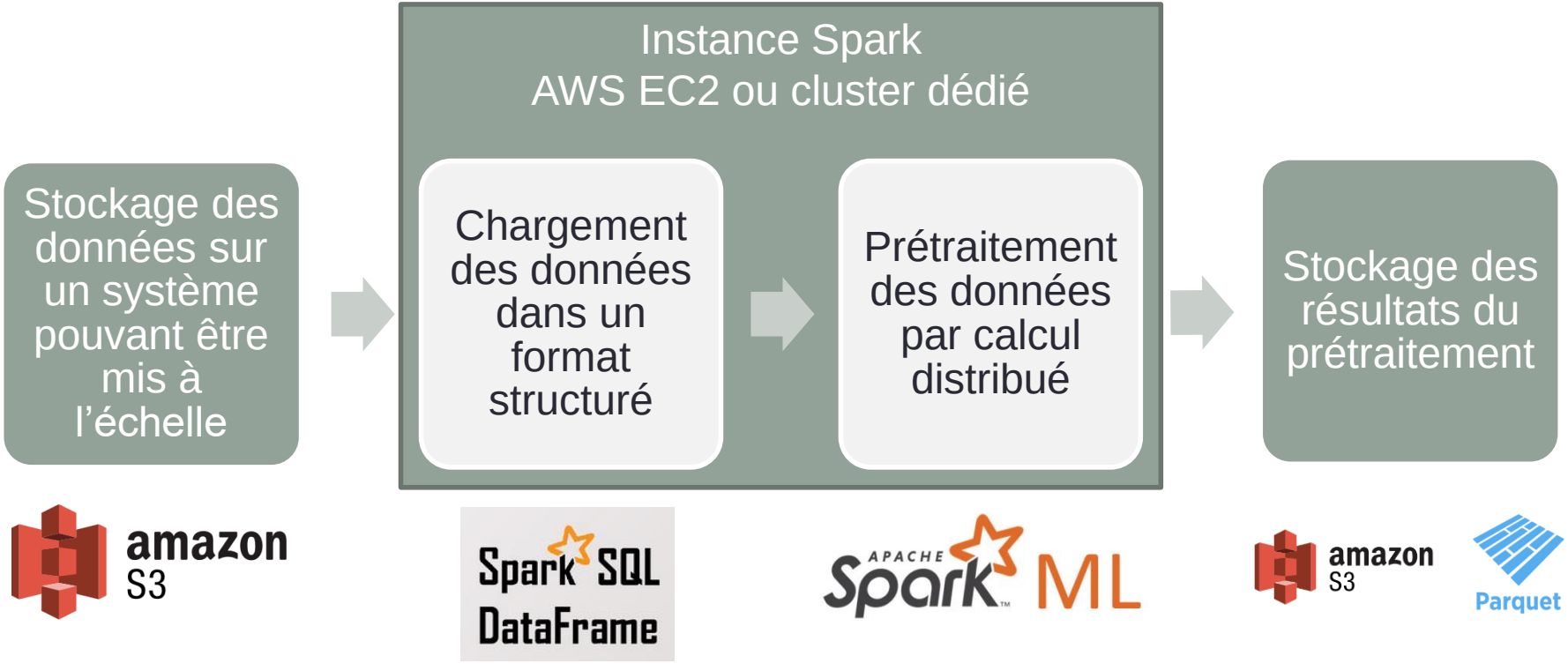
Egalisation historgramme et redimensionnement

Traitement d'image + extraction de features

ORB, SURF, SIFT, etc.

Réseaux préentraînés (Transfer Learning)

Décomposition de la problématique



Instance Spark

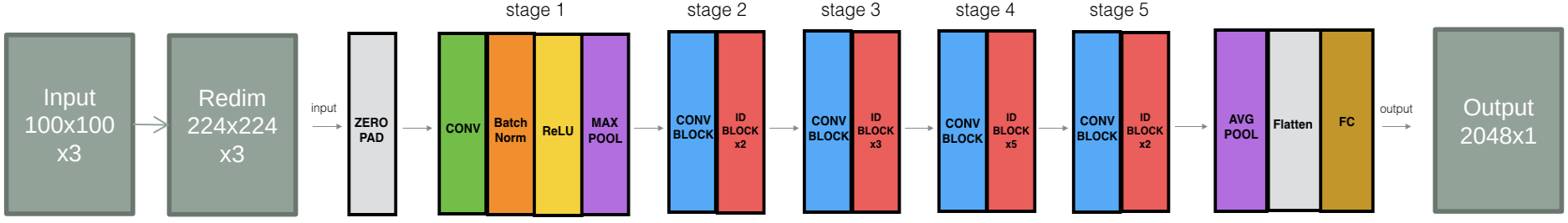
Création d'un **SparkContext**	Chargement des images dans l'arborescence: stockage dans un spark DataFrame	Extraction du chemin vers chaque fichier	Obtention dela catégorie du fruit (split String du nom de dossier)	Prétraitement et Réduction Dimensionnelle	Enregistrement au format « Parquet »
SparkContext. getOrCreate()	spark.read()	Input_file_name()	UDF (user defined function) Str.split()	ResNet50	Spark.save

Les calculs ne sont réellement exécutés que lorsqu'une action est réalisée : affichage des données, enregistrement, requête, etc.

```
+--------------------+--------------+--------------------+
|                path|     categorie|  image_preprocessed|
+--------------------+--------------+--------------------+
|s3a://p8openclass...|Apple_Golden_1|[0.00841228850185...|
|s3a://p8openclass...|Apple_Golden_1|[0.11171255260705...|
|s3a://p8openclass...|Apple_Golden_1|[0.31495276093482...|
|s3a://p8openclass...|Apple_Golden_1|[0.41679331660270...|
|s3a://p8openclass...|Apple_Golden_1|[0.26685762405395...|
```

Le prétraitement en détail

- **Réseau RESNET50 :**
  - Approche Transfer Learning
  - 23 M paramètres pré entraînés
  - 50 couches de neurones



- Combine prétraitement et la réduction de dimension

Zoom sur l'infrastructure AWS

- Stockage fichiers sur S3 :
  - upload via AWS CLI ou Interface Web
  - Lecture des fichiers depuis Spark
  - Enregistrement de fichier depuis spark vers S3
- Instance EC2 (T2.xlarge) / OS Ubuntu Server 18.04
- Configuration : Python 3 / Java 8 / Spark / Hadoop / Hadoop-AWS/ Spark MLLib / Pillow
- Configuration sur machine distante : accès via SSH
  - Chargement clés IAM / AWS
  - Installation des logiciels et packages
  - Mise en place Jupyter Notebook accessible à distance pour exécution du code / analyse des résultats

Comment passer à l'échelle?

- Aucune modification de code Spark/Python à apporter : évolution sans coupure de charge
- Stockage des fichiers :
  - S3 : OK
  - Alternative : HDFS sur n serveurs
- Évolution de l'infrastructure de calcul:
  - Instance EC2 de plus grande capacité RAM/Processeur
  - Remplacement par un cluster Elastic Map Reduce avec plusieurs instances EC2 (1 Maître + n esclaves) :
    - Configuration automatique
  - Alternative hors AWS : Créer un cluster avec plusieurs noeuds
- Dans un second temps : augmentation du nombre d'instances esclaves / noeuds

CONCLUSION

Conclusion et perspectives

- **Enseignements**
  - Prise en main Pyspark
  - Découverte de l'écosystème AWS
  - Administration d'un serveur Linux par SSH
  - Découverture du format distribué parquet

- **Difficultés rencontrées**
  - Nombreuses possibilités techniques : choix complexes
  - Débug complexe dû à des erreurs peu explicites (superposition Spark/hadoop/Java/S3)

Perspectives

- Aller plus loin :
  - Prétraitement pour cas réels (recadrage, plusieurs fruits, arrière plan, etc.)
  - Entraîner le modèle (approche transfer learning)
  - Déployer le modèle en production sur un cluster
  - Monitoring...

- Tester les solutions existantes sur le marché : API Pl@ntnet

- Pousser le cas d’usage :
  - Identifier la maturité des fruits pour les cueillir au bon moment
  - Identifier les pathologies ou les fruits abîmés

MERCI DE VOTRE ATTENTION

AVEZ VOUS DES QUESTIONS?