Déployez un modèle dans le cloud

Projet 8

OpenClassrooms

Emanuele Partenza

Fruits!

Sommaire

1. Problématique et jeu de données
2. Architecture Big Data
3. PySpark
4. AWS
5. Etapes du projet
6. Démonstration d’exécution
7. Conclusions

Problématique et Objectif

- La très jeune start-up de l'AgriTech, nommée "**Fruits**!", cherche à proposer des solutions innovantes pour la récolte des fruits;

- Elle souhaite mettre à disposition du grand public une application mobile qui permettrait aux utilisateurs de prendre en photo un fruit et d'obtenir des informations sur ce fruit.

- Développer une première chaine de traitement des données qui comprendra le *preprocessing* et une étape de réduction de dimension;
- Tenir compte du fait que le volume de données va augmentes très rapidement après la livraison de ce projet.

Jeu de données

Données issues d'un kernel Kaggle :

90483 images et 131 classes

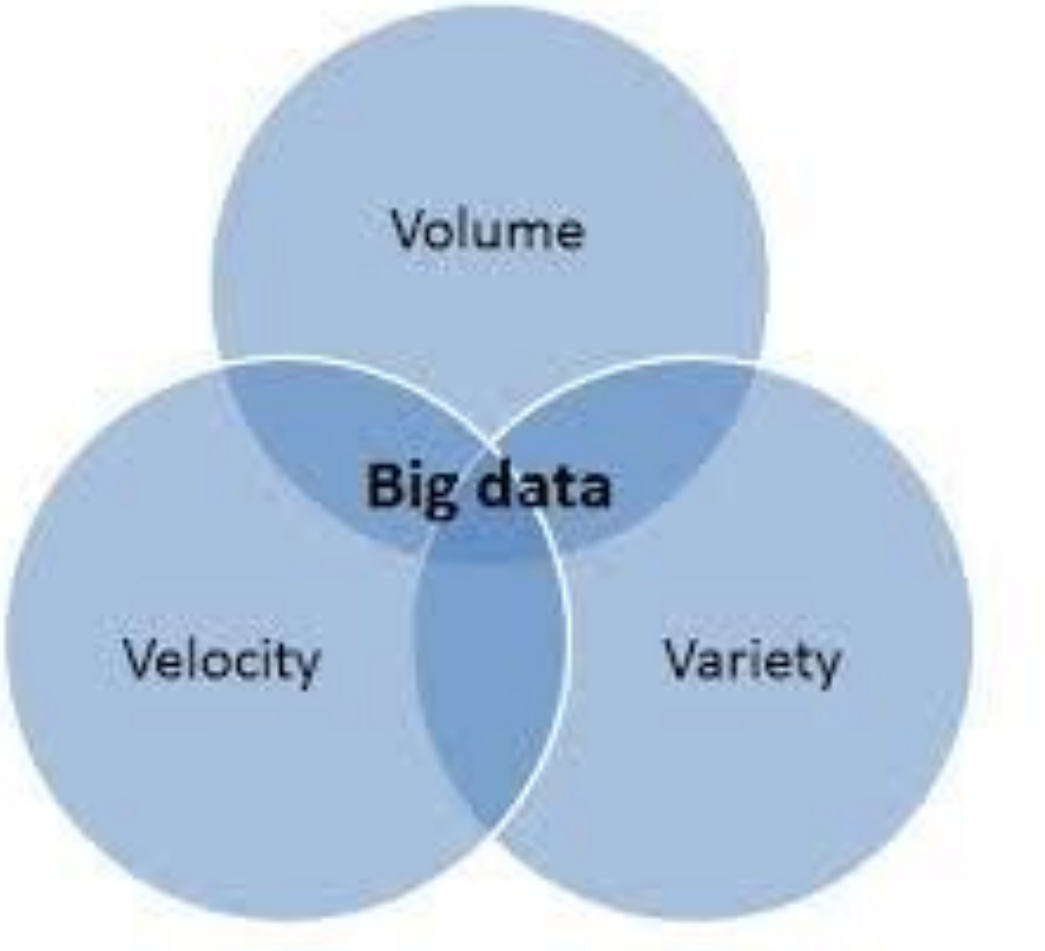
2 jeux de données training (67692) et 1 test set d(22688)

131 dossiers :

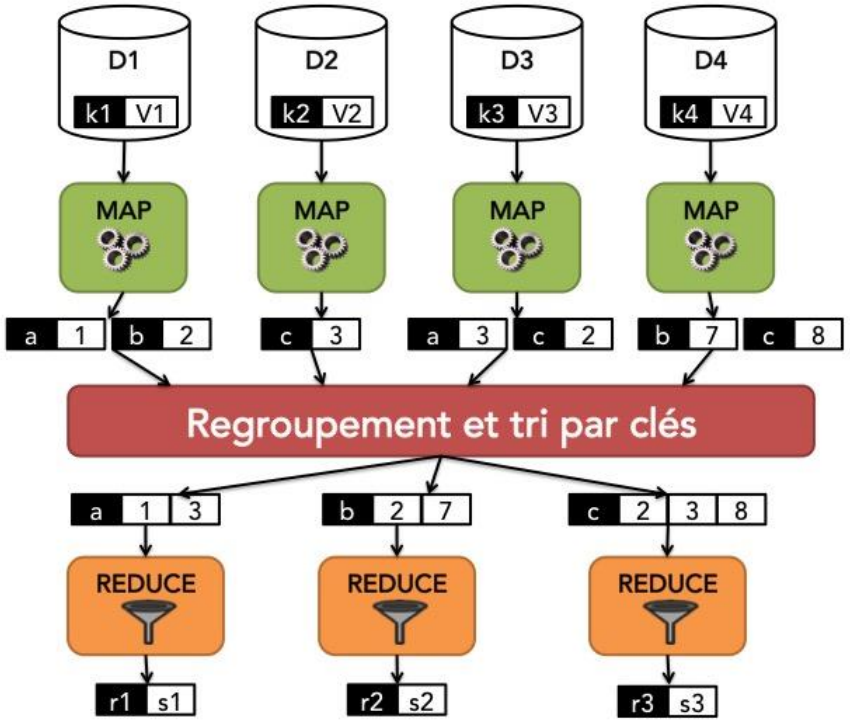
- Un fruit ou un légume;
- Une image avec un fond blanc et sous 3 axes;
- 100x100 pixels en JPG RGB;
- Plusieurs variétés pour certains fruits.

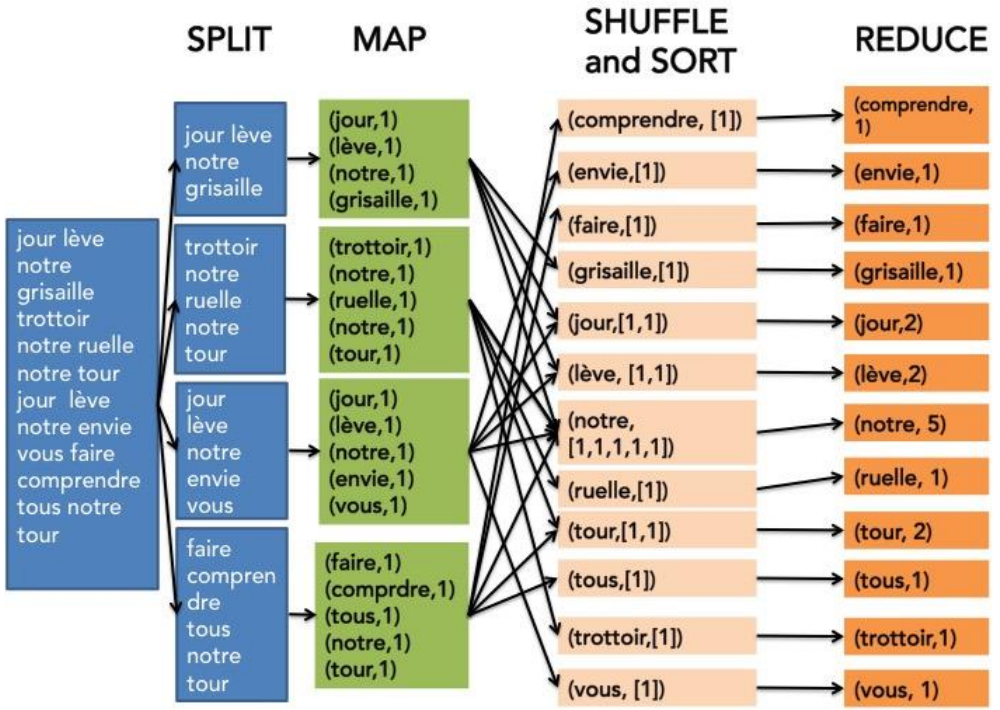
Pourquoi le BigData

Dans l'application mobile que ***Fruits!*** a l'intention de développer on à un fort ***volume*** de données, susceptible d'***augmenter rapidement*** et dans le temps ne seront probablement ***pas dans le même format***.

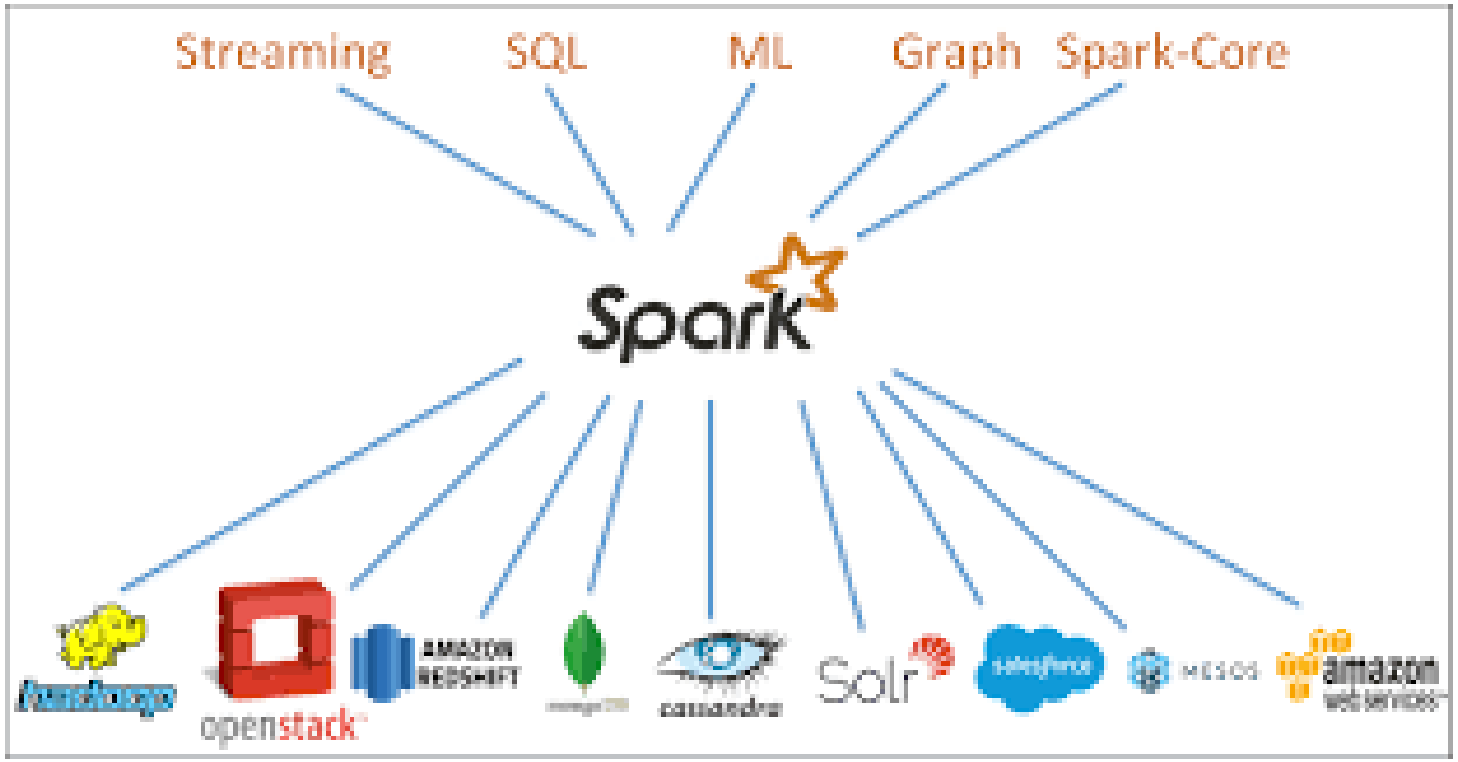


Calcul distribué / MapReduce





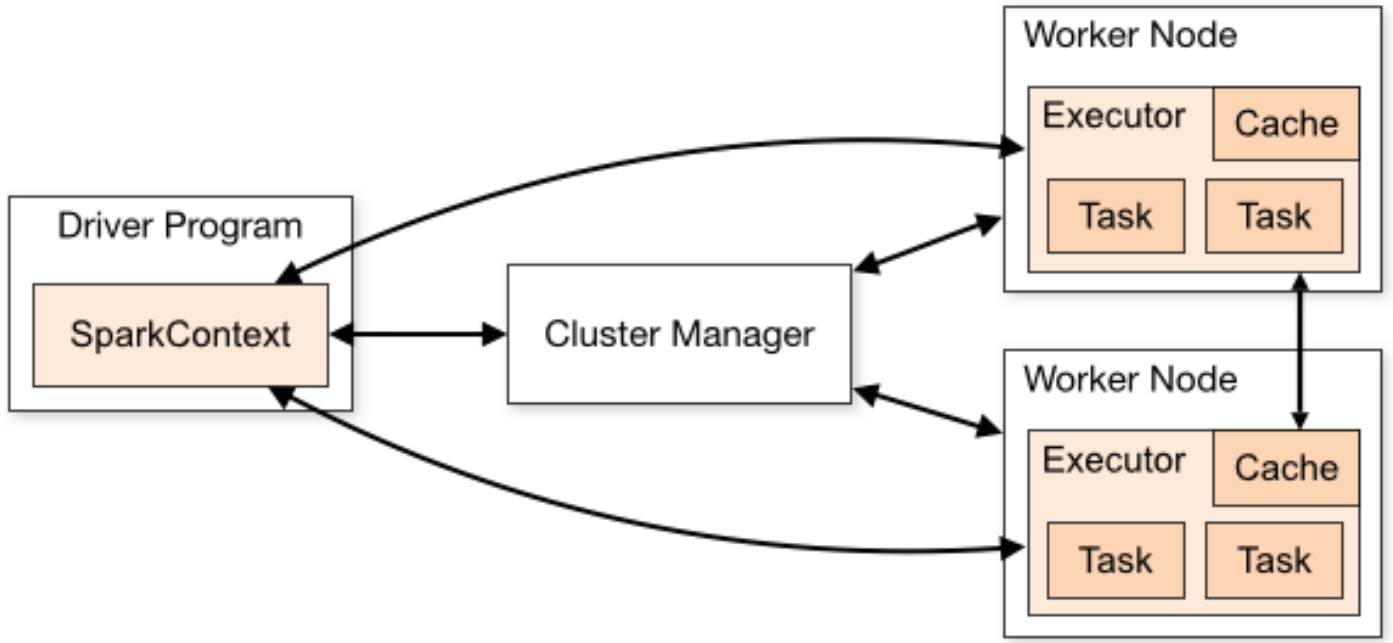
Spark



- Ecrit les données sur RAM et non sur disque ce qui a des conséquences importantes sur la rapidité du traitement
- Élargit le cadre map/reduce en proposant des opération supplémentaires pouvant être réalisé de manière distribuée

Un Cluster Spark est composé de :

- Un ou plusieurs **workers** : chaque worker instancie un ***executor*** chargé d'exécuter les différentes ;
- Un ***driver*** : chargé de répartir les taches sur les différents ***executors***.
- Un ***cluster manager*** : chargé d'instancier les différents ***workers***.



AWS

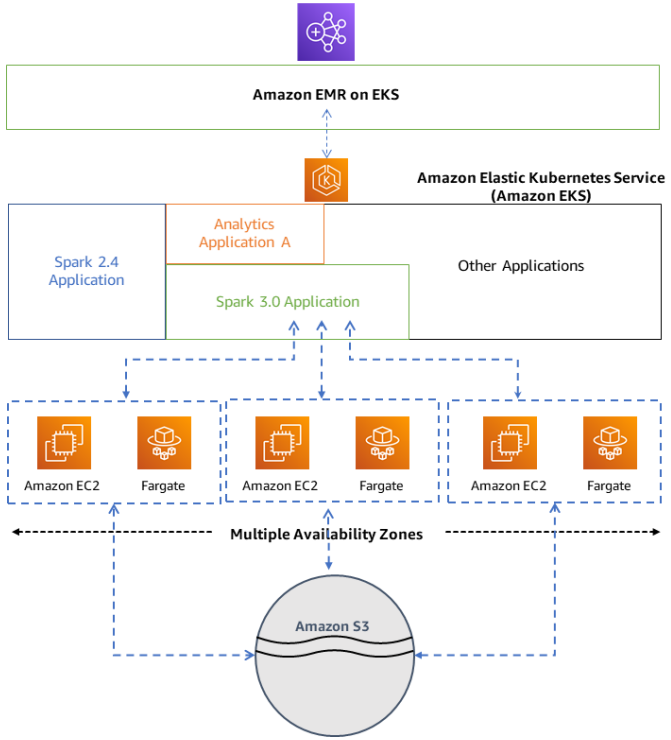
S3 : stockage des données

EMR : construction des clusters pour la distribution des calcules sur plusieurs instances

IAM : configurer les rôles sur aws

EMR Architecture

- Amazon EMR sur EKS associe de manière souple les applications à l'infrastructure sur laquelle elles s'exécutent
- Chaque couche d'infrastructure assure l'orchestration de la couche suivante
- Lorsque vous soumettez une tâche à EMR, votre définition de tâche contient tous ses paramètres spécifiques à l'application
- EMR utilise ces paramètres pour indiquer à EKS quels pods et conteneurs déployer
- EKS met ensuite en ligne les ressources informatiques d'EC2 et AWS Fargate nécessaire pour exécuter la tâche.



TensorFlow

APACHE Spark™

jupyterhub

hadoop

Configurations
.JSON et
Bootstrap .sh

$ bootstrap.sh

```
1    sudo python3 -m pip install -U setuptools
2    sudo python3 -m pip install -U pip
3    sudo python3 -m pip install wheel
4    sudo python3 -m pip install pillow
5    sudo python3 -m pip install pandas==1.2.5
6    sudo python3 -m pip install pyarrow
7    sudo python3 -m pip install boto3
8    sudo python3 -m pip install s3fs
9    sudo python3 -m pip install fsspec
10   sudo python3 -m pip install keras
```

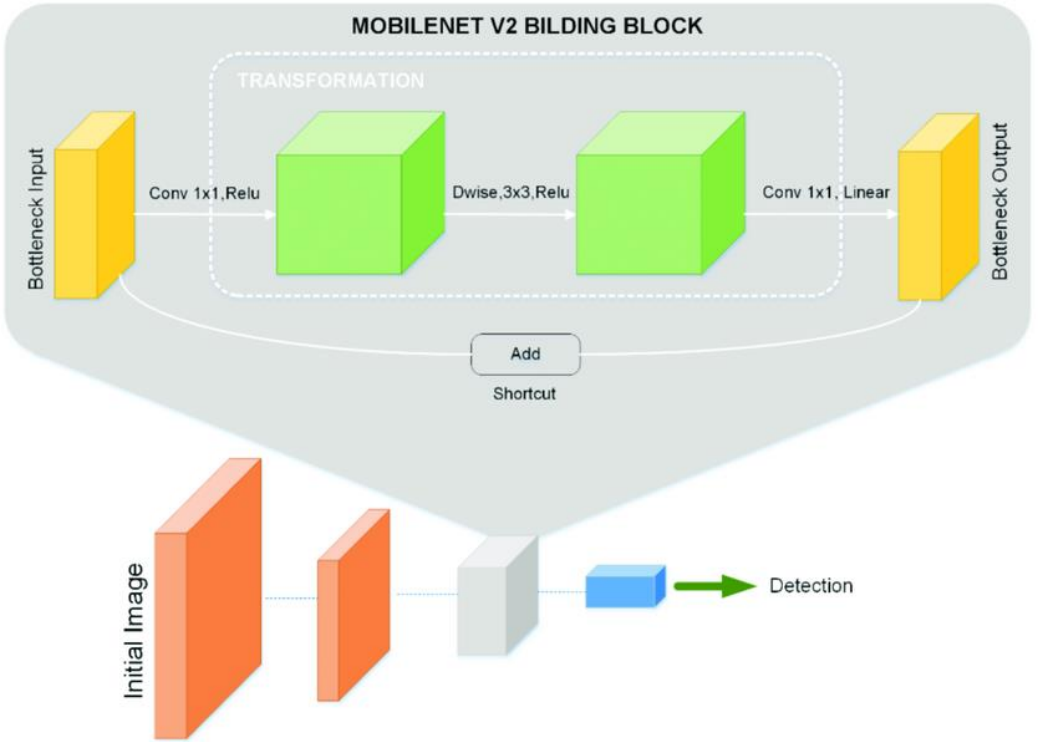
{} configuration.json > ...

```
1    [{"classification":"jupyter-s3-conf","properties":{"s3.persistence.bucket":"emanuelepartenza-projet8","s3.persistence.enabled":"true"}}]
```

Les différentes étapes

MobilNetV2

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	



Resultats

```
Temps d'execution 524.02 secondes
+--------------------+-------------+--------------------+--------------------+--------------------+
|                path|        label|            features|      scaledFeatures|         pcaFeatures|
+--------------------+-------------+--------------------+--------------------+--------------------+
|s3://emanuelepart...|Apple Golden 1|[0.0,0.0104257911...|[-0.8562760789128...|[7.34856954418275...|
|s3://emanuelepart...|Apple Golden 1|[0.0,0.0834322348...|[-0.8562760789128...|[6.5955599854301,...|
|s3://emanuelepart...|Apple Golden 1|[0.01090431213378...|[-0.8360288905009...|[4.43380727582671...|
|s3://emanuelepart...|  Cantaloupe 2|[0.34131395816802...|[-0.2225223919461...|[6.96026744806433...|
|s3://emanuelepart...|  Cantaloupe 2|[0.01884618028998...|[-0.8212823848133...|[3.51006709241068...|
+--------------------+-------------+--------------------+--------------------+--------------------+
only showing top 5 rows
```

Exécution du script PySpark sur le Cloud

Conclusions