Déployer un modèle dans le cloud

Cyril REGAN

2 octobre 2020

Problématique

Projet :

- Application de détection par photo d'un fruit pour consulter des informations

Besoin :

- Lancement d'un moteur de classification d'images en environnement big data

=> 1ère étape : construire une architecture big data + extraire les feautures des images sur le cloud.

Données & difficultées

Données :

Données initiales kaggle : **Fruits 360**

- 67692 images
- 131 fruits et légumes

Difficultés :

- Spark + architecture cloud AWS EC2, S3, IAM

=> Supporter l'augmentation massive de données

Table of Contents

Type de Cloud

- **IaaS** (Infrastructure as a Service) : accès à tout ou partie de son infrastructure technique.
- **PaaS** (Platform as a Service) : accès à l'infrastructure + gestion nbr machines
- **SaaS** (Software as a Service) : accès à un logiciel sous forme de service.

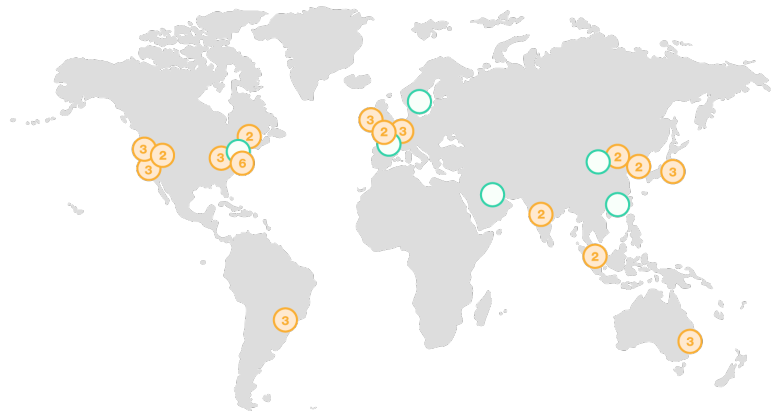


Figure – Serveurs aws

Principaux fournisseurs de cloud : aws, Microsoft,Google, IBM

Table of Contents

Services aws

- EC2 (Elastic Compute Cloud : IaaS) : serveurs sous forme de machines virtuelles dans le cloud.
- S3 (Amazon Simple Storage Service : PaaS) : service de stockage et de distribution de fichiers
- IAM (Identity and Access Management) : contrôle l'accès aux services et ressources d'aws.

... et pleins d'autres mais pas utilisés dans cette étude

Table of Contents

MapReduce

Parallèle vs distribué :

- Calculs parallèles : threads exécutés en même temps et partagent une mémoire commune.
- Calculs distribués : nœuds de calculs distants, autonomes sans partage de ressources => passage à l'échelle horizontale : suffit d'ajouter des nœuds pour $\nearrow$ la capacité de calcul.

MapReduce

- SPLIT : création de sous-lots,
- MAP : opération appliquée à chaque lot,
- SHUFFLE and SORT : regroupement et trie des résultats intermédiaires,
- REDUCE : agrégation des résultats intermédiaires.

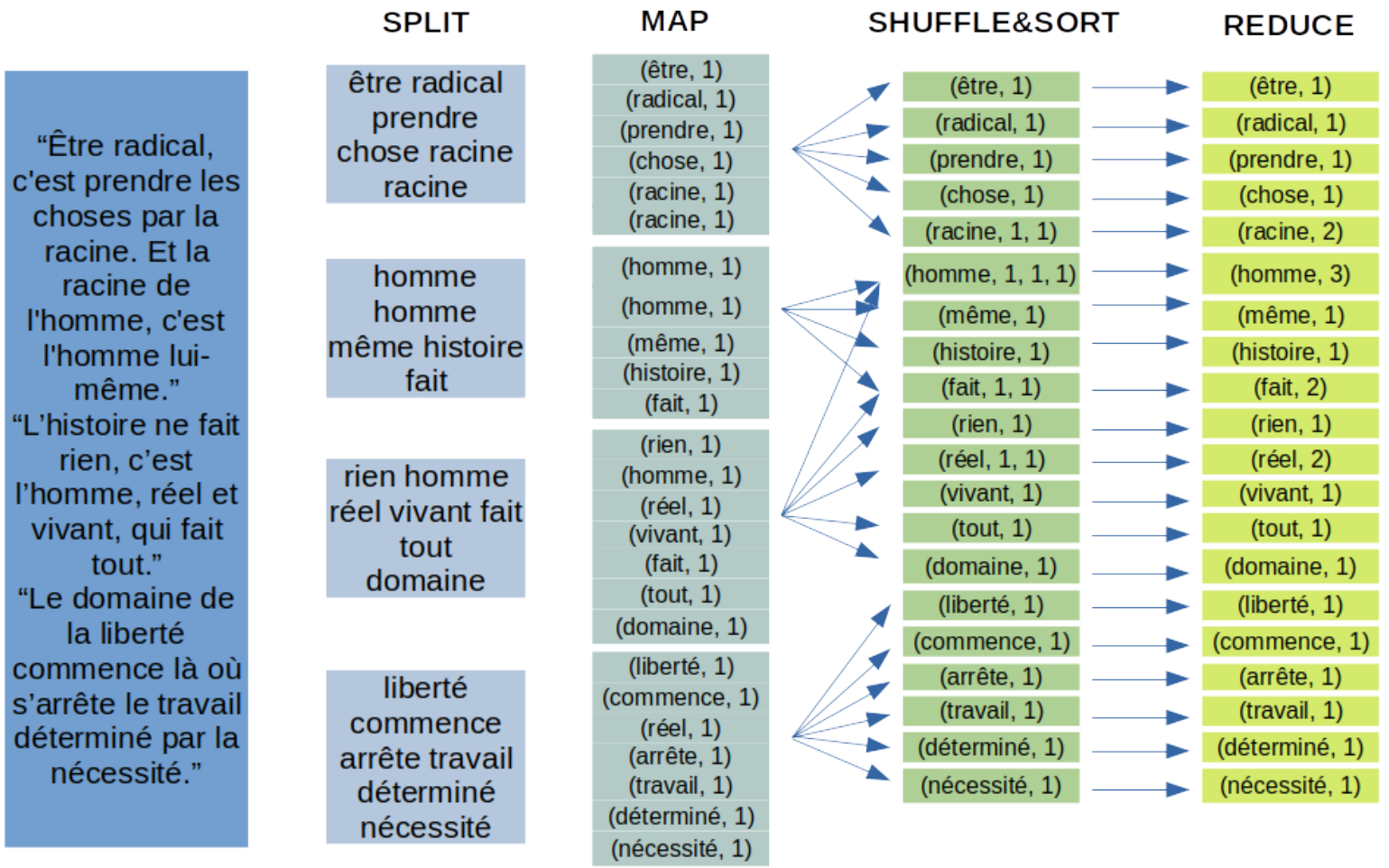


Figure – MAP REDUCE wordcount

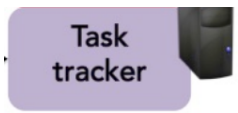
Table of Contents

Hadoop

Hadoop a l'architecture pour orchestrer du MapReduce + 1 système de fichiers distribués HDFS (Hadoop Distributed File System).

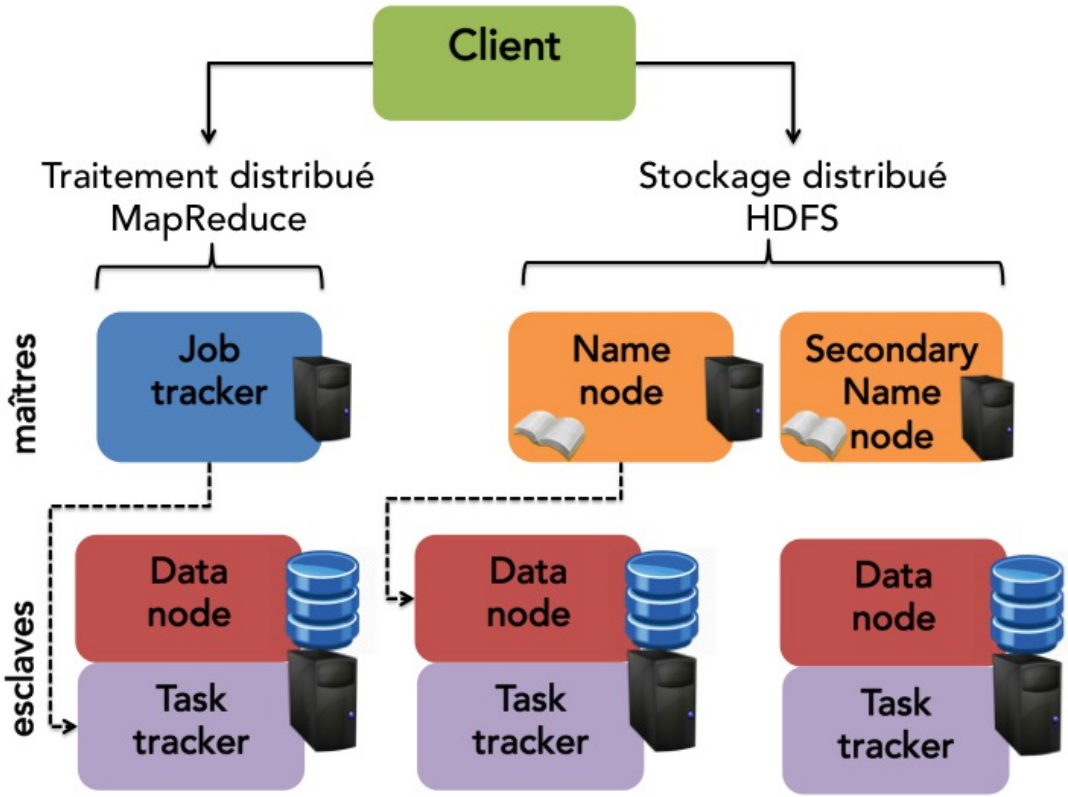
:

Ordonnancement des traitements (de tâches MapReduce) et gestion des ressources (qui fait quoi).



:

Exécution (des tâches MapReduce) et report au job tracker.



Inconvénients :

- Difficulté à transformer un algorithme en MapReduce,
- coût important si successions de MAP et REDUCE,
- et si le job tracker est défaillant ?

=> Hadoop + YARN (Yet Another Ressource Negociator) répond à ces pbs en exécutant des applications distribuées $\neq$ MapReduce, en séparant :

  - Gestion des ressources (Ressource manager) /
  - Gestion des jobs (AppMaster)

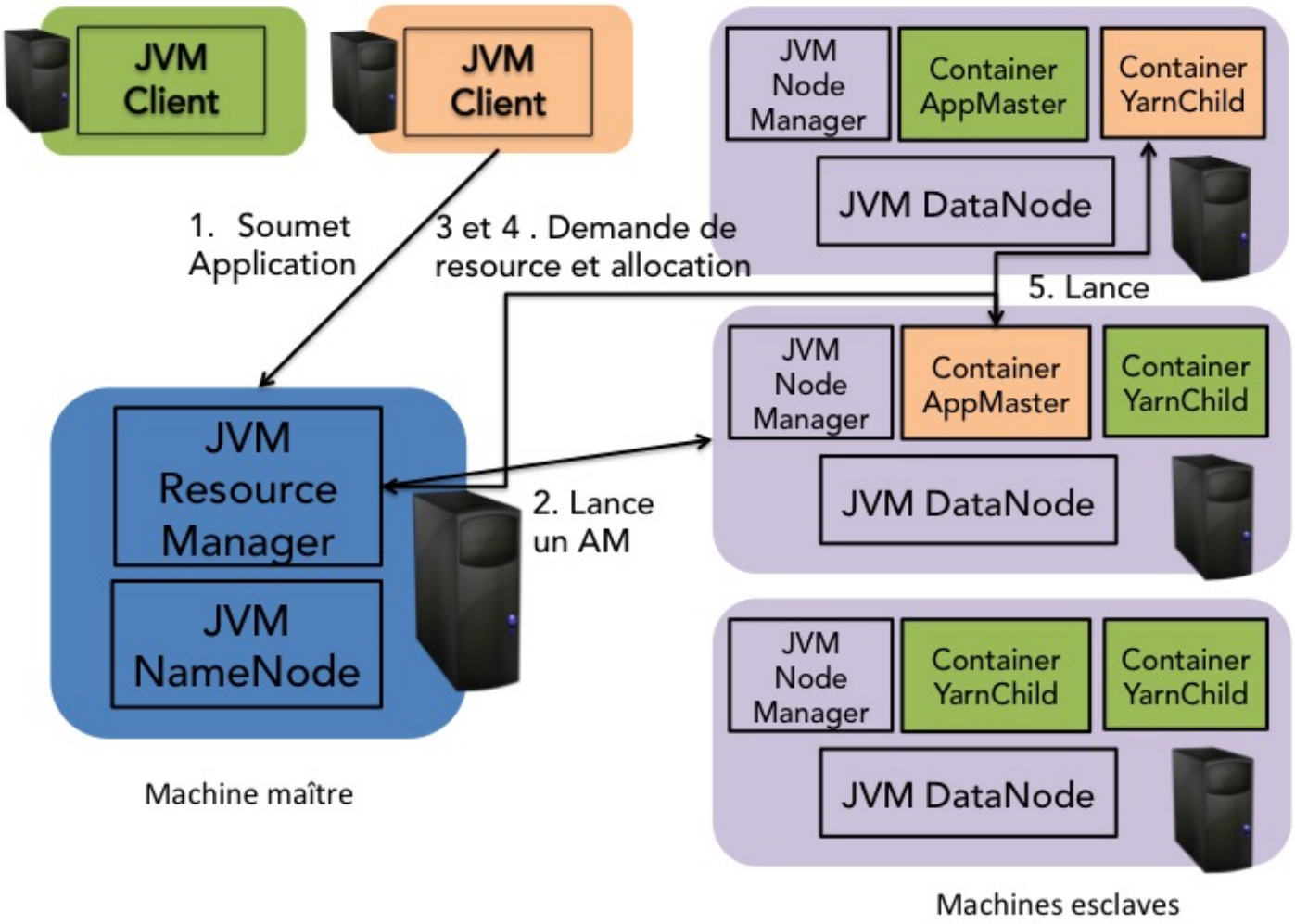


Figure – Fonctionnement Hadoop2

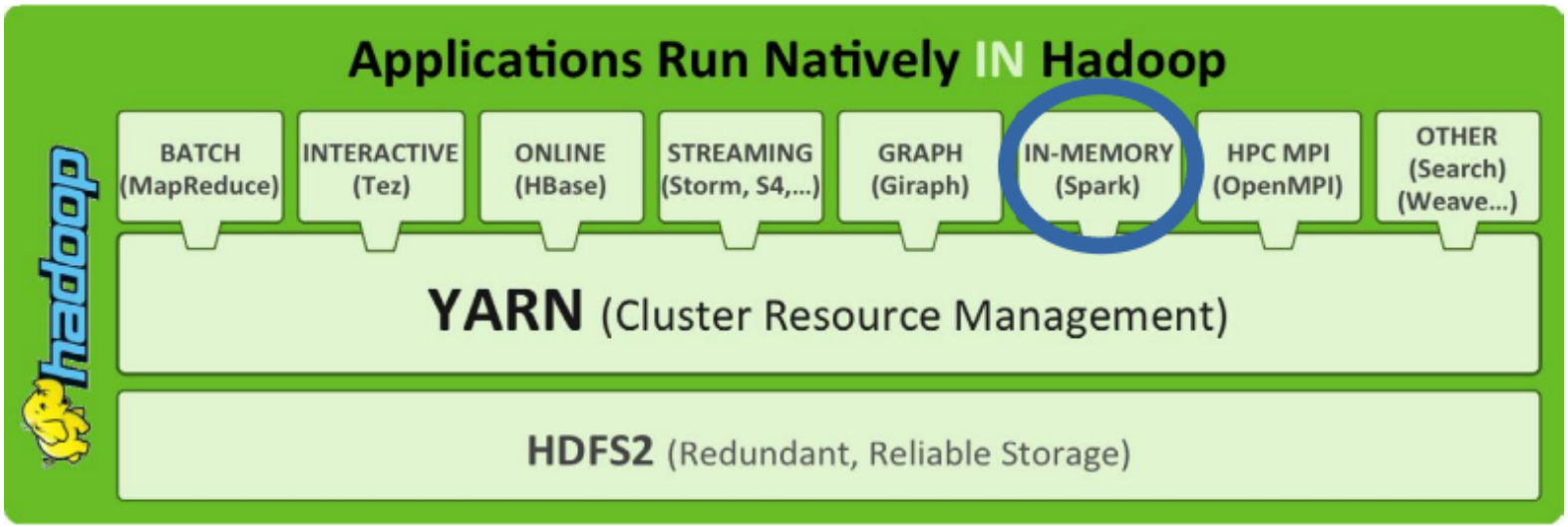


Figure – Hadoop2 + spark

Hadoop Streaming : Run programmes $\neq$ java (par ex python)

Table of Contents

Spark vs Hadoop MapReduce

Inconvénients d'hadoop :

- Écriture sur **disque** de MAP ou REDUCE coûteuses en temps.
- Limitation aux opérations MAP et REDUCE rend les **opérations complexes difficiles**.

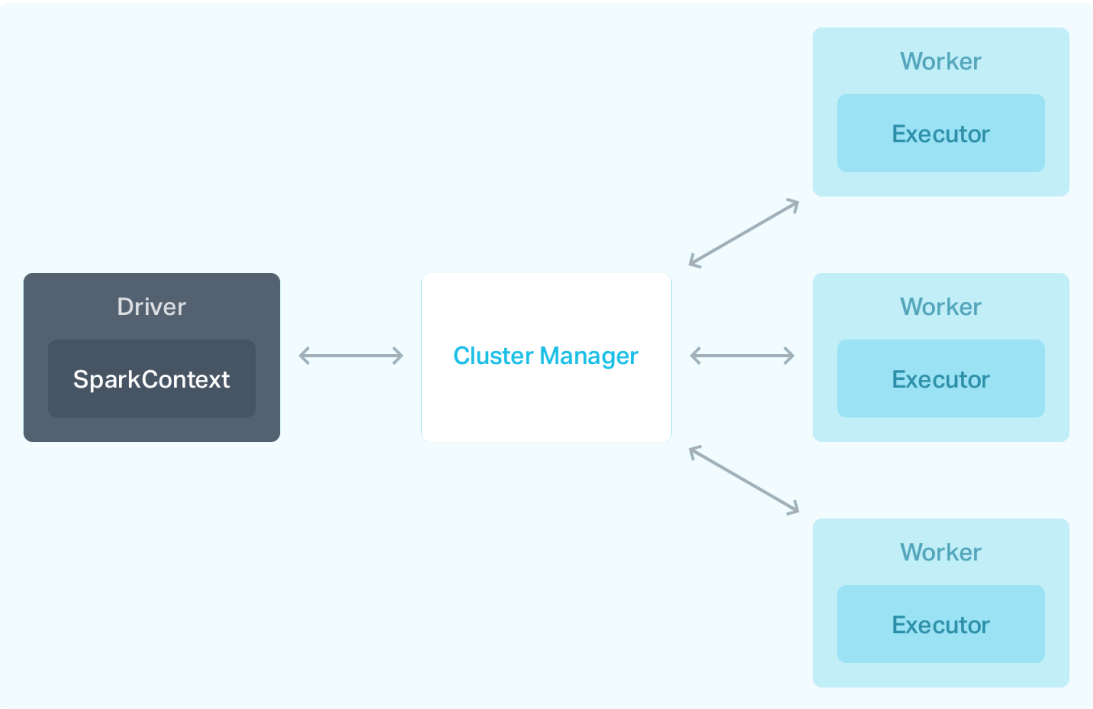
Apache Spark résout ces deux problèmes :

- Écriture en RAM (accélération de 10 à 100 fois)
- Propose d'autres opérations (que MAP et REDUCE) distribuées (transformation automatique en MAP et REDUCE).

Table of Contents

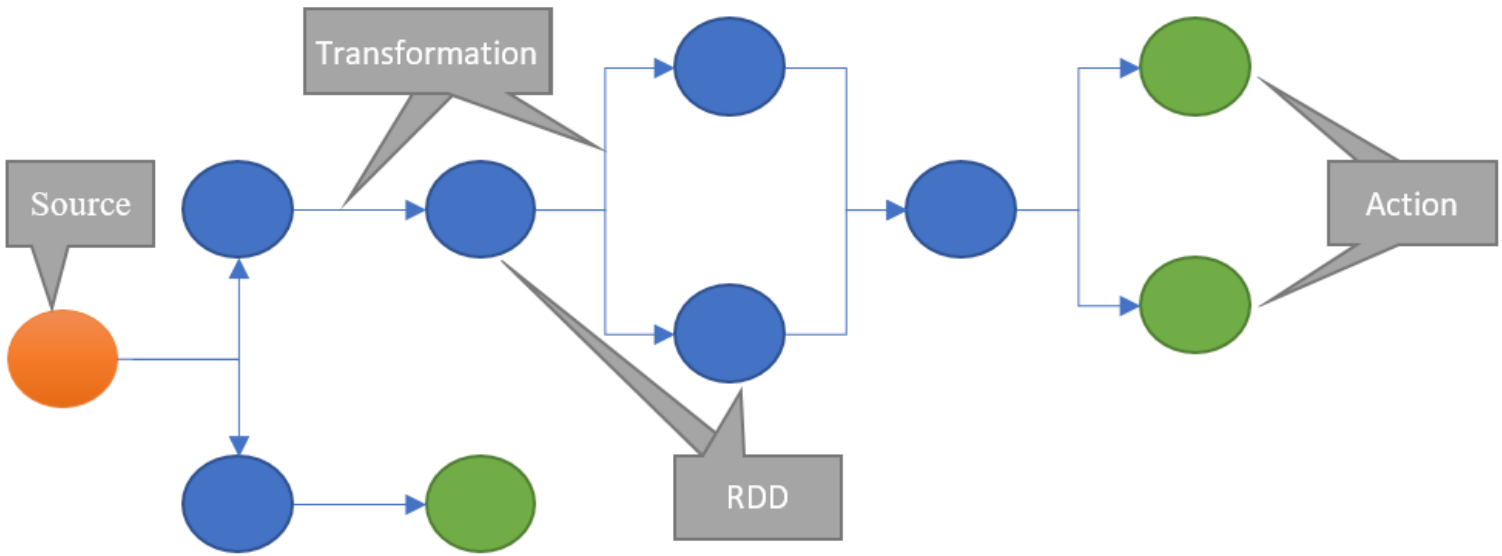
Architecture

- workers (machines physiques) : instancient un executor qui exécute les tâches.
- driver : répartie les tâches sur différents executors.
- cluster manager : instancie les différents workers.



RDD

- Un **RDD** (Resilient Distributed Datasets) est créé à l'issue d'une opération de transformation (exemple : MAP).
- Une **transformation** ne calcule pas immédiatement les résultats, mais seulement lorsqu'on en a besoin (lazy evaluation).
- Une **action** (comme .collect) nécessite de calculer les résultats.



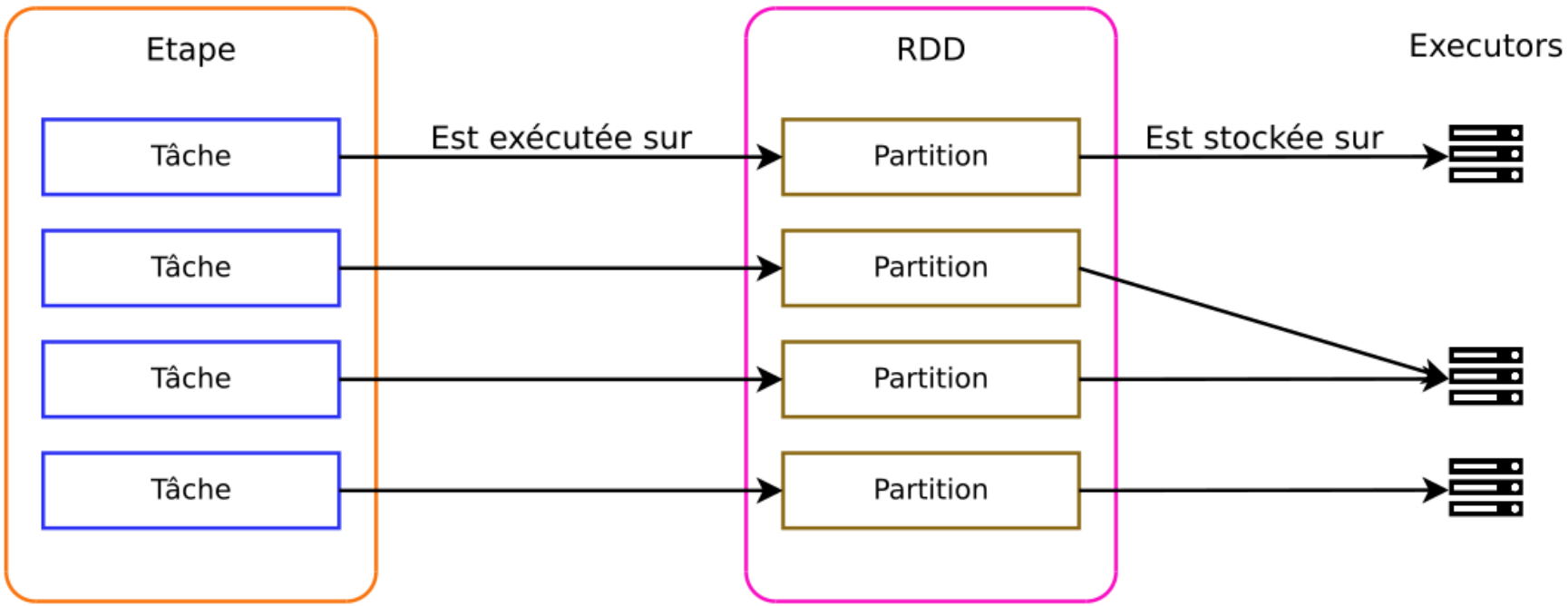


Figure – Fonctionnement RDD

Table of Contents

Spark SQL et sparkML

- **Spark SQL** : permet d'introduire les **DataFrames** (type RDD de données structurées)
- **Spark ML** : modèles d'apprentissage machine en architecture distribuée.
- **Sparkdl** : librairie spécialisée dans le Deep Learning.

Table of Contents

Modèles

- Lecture sur s3 des images avec la librairie *pyspark.ml.image*
- Extraction de caractéristiques avec le réseau de neurone **ResNet50**
  Réseau pré-entraîné sur "imagenet". Il est architecturé en hadoop-spark avec la librairie ***sparkdl***.*DeepImageFeaturizer*
- Réduction de dimension par ACP (Analyse en Composante Principale) avec la librairie *pyspark.ml.feature.PCA* et *pyspark.ml.linalg.Vectors*
- Instance Ec2 m5ad.4xlarge : **16** vCPU, **64** Go RAM,**16** Go ssd

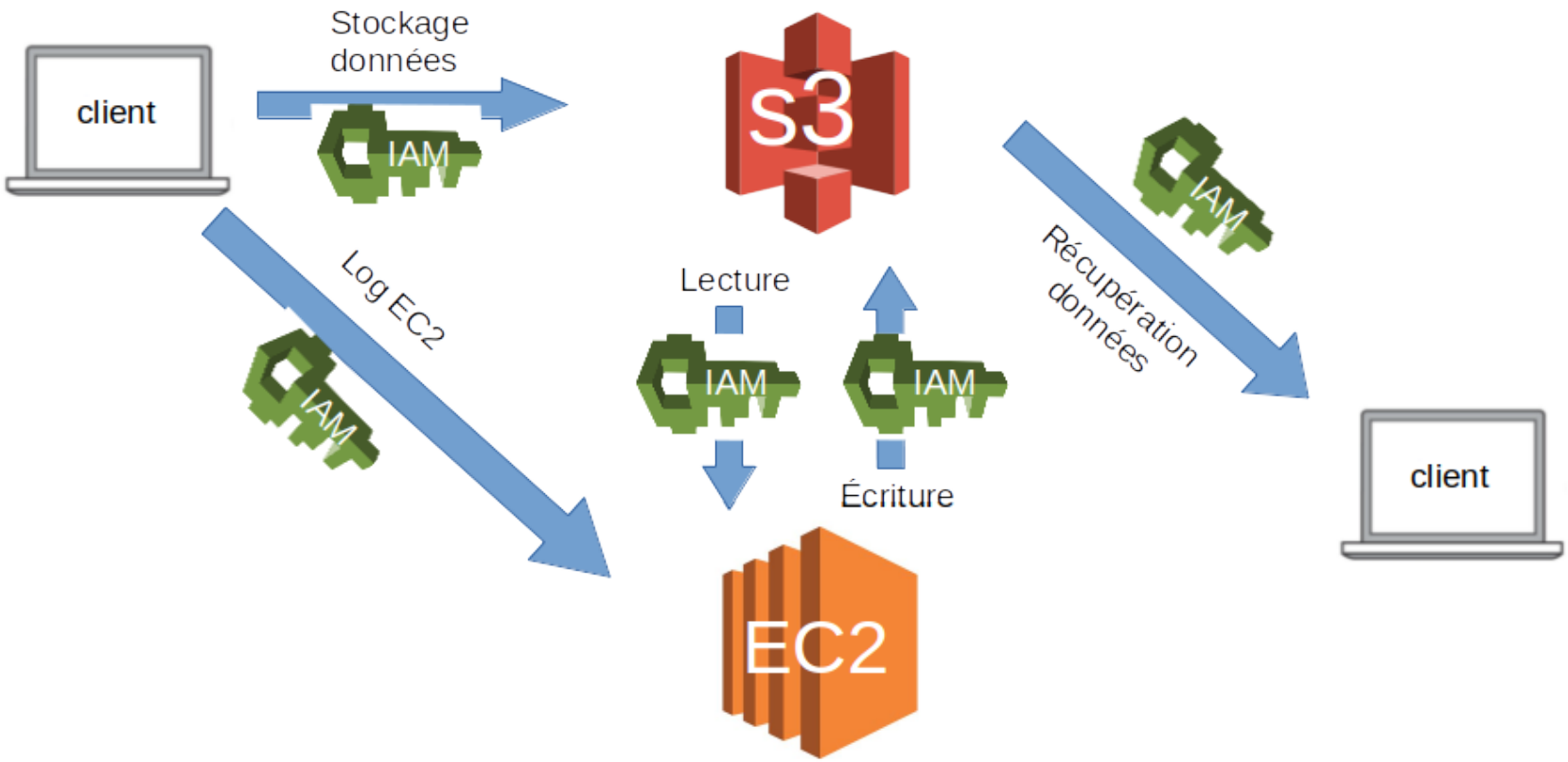
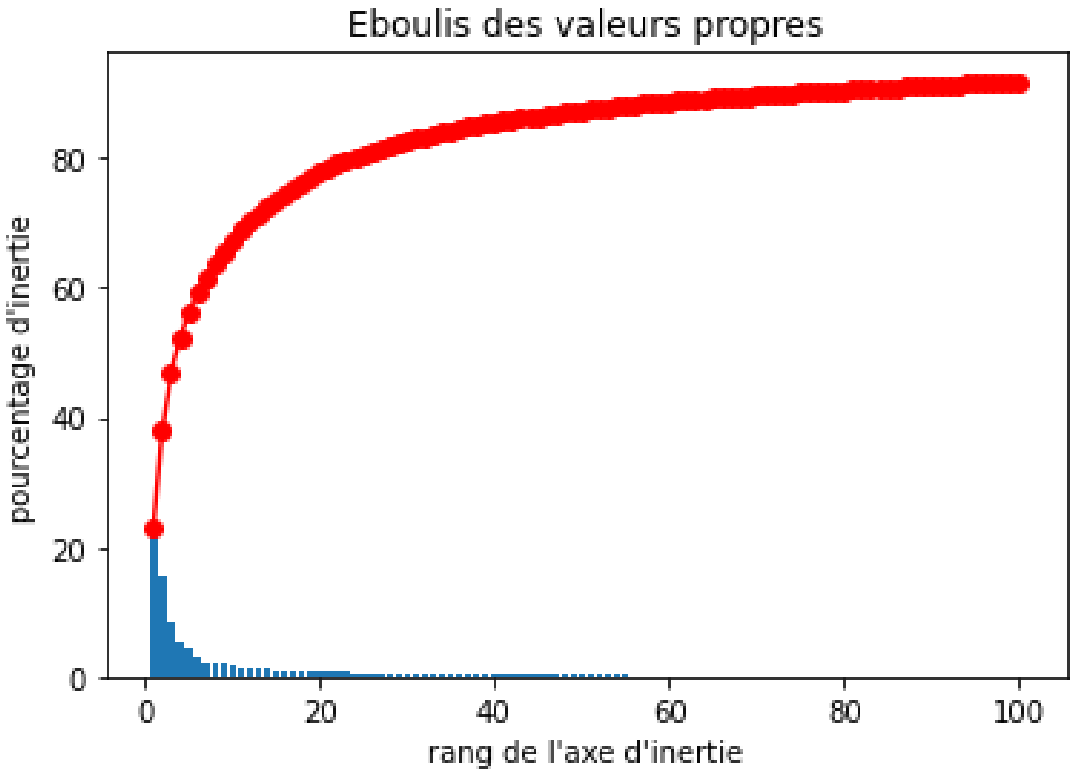


Figure – Schéma cloud aws

Table of Contents

Résultats

La dimension des caractéristiques en sortie de **ResNet50** est de 2048. L'ACP est tronquée à 100 caractéristiques, ce qui permet de conserver une inertie de 92% :



Résultats : dataframe de colonnes *[label,chemin, caractéristiques réduites]* (.parquet) sur s3 pour préparer une classification :

```
+--------------------+-----+--------------------+
|                path|label|            features|
+--------------------+-----+--------------------+
|s3a://projet8usao...|  Fig|[3.50662003143329...|
|s3a://projet8usao...|  Fig|[1.82374190714284...|
|s3a://projet8usao...|  Fig|[4.38541701222242...|
|s3a://projet8usao...|  Fig|[3.41234314132443...|
|s3a://projet8usao...|  Fig|[3.88504892864175...|
|s3a://projet8usao...|  Fig|[3.31374232345248...|
|s3a://projet8usao...|  Fig|[3.46202173463615...|
|s3a://projet8usao...|  Fig|[3.51689750167102...|
|s3a://projet8usao...|  Fig|[3.62530665362521...|
|s3a://projet8usao...|  Fig|[3.89476108850136...|
|s3a://projet8usao...|  Fig|[3.92573152988034...|
|s3a://projet8usao...|  Fig|[3.73297460117602...|
|s3a://projet8usao...|  Fig|[1.85503512134103...|
|s3a://projet8usao...|  Fig|[4.24449059747582...|
|s3a://projet8usao...|  Fig|[4.23644848645660...|
|s3a://projet8usao...|  Fig|[4.02262519723578...|
|s3a://projet8usao...|  Fig|[4.09222322807256...|
|s3a://projet8usao...|  Fig|[3.55146248116995...|
|s3a://projet8usao...|  Fig|[3.47415381298948...|
|s3a://projet8usao...|  Fig|[3.94689457347835...|
+--------------------+-----+--------------------+
only showing top 20 rows
```

Conclusion

Ce projet a permis de

- Construire une architecture big data avec spark pour une classification d'images
- D'héberger données + machines sur le cloud (aws EC2, s3, IAM)

Perspectives

du projet :

- Effectuer la classification d'image sur le set d'origine
- Construire l'application
- Passage à l'échelle horizontal pour absorber la charge

personnelles :

- La configuration de modèle d'architecture big data est très stricte et nécessite la plus grande vigilance : c'est un travail fastidieux qu'il ne faut pas le négliger avant de se lancer dedans.
- Mais le jeu en vaut la chandelle ! car traitement impossible en local

Merci de votre attention !