Hurence

Laurence.Hubert@hurence.com

Les outils d'analytique Big Data

Analytique temps réel

Quatre pôles d'activités

Conseil

Une expertise Big Data unique

Cadrages d'initiatives Big Data
Sizing et installation d'infrastructures Big Data
Développements Big Data du POC à la mise en production

Formations

Plus de 25 modules de formations Big Data

Hadoop, bases NoSQL, moteurs de recherche, développements sur stacks diverses temps réel ou non, machine learning, deep learning etc.

Support

Support d'infrastructures Hadoop

Monitoring, configuration, patchs, upgrades

Produits

TROPHÉE INNOVATION IT 2017

Suite LogIsland

Traitement temps réel d'événements sur des grosses volumétries (open source basé sur Spark / Kafka)

Briques Big Data “Analytique”

- Infrastructure minimale de serveurs avec de la RAM et du disque
- Un Data Lake
- Des chaînes d’injection de données massives batch et temps réel
- Des outils de traitement batch (analytique batch)
- Des outils de traitement temps-réel (analytique temps réel)

- La plupart des Data Lakes sont de type Hadoop (avec plusieurs distributions possibles Apache, Hortonworks, Cloudera, MapR)
- On pourrait concevoir un Data Lake sur une base de données Big Data (Exadata, Teradata, Cassandra ou autre…)
- Chaque approche a ses avantages et inconvénients

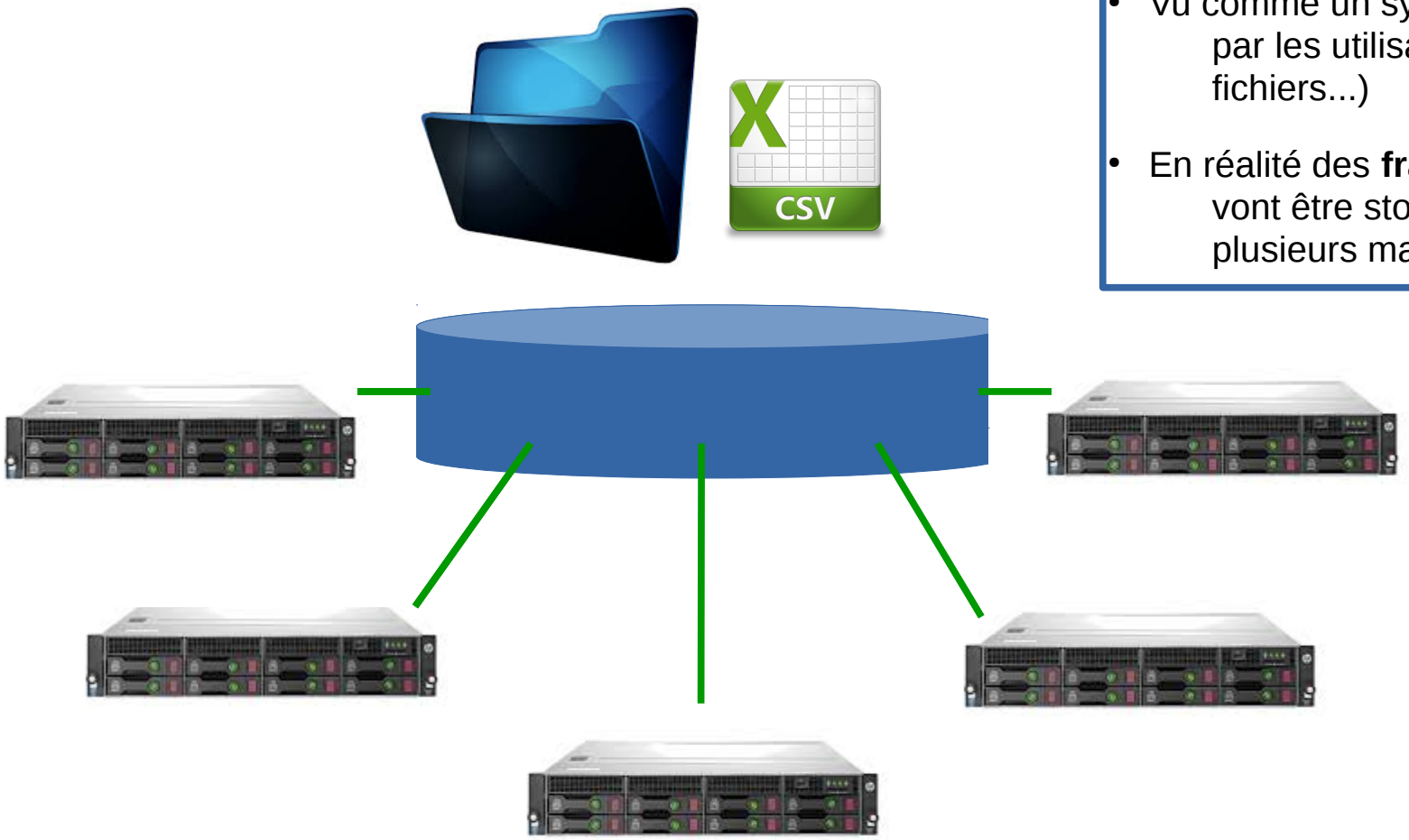
- L’idéal est quand une technologie s’applique bien en batch et en temps-réel (Spark avec Spark streaming)

L’infrastructure:
un super-calculateur Low Cost

- Des technologies matérielles plutôt « low-cost » (commodité)
- Des technologies logicielles plutôt « low-cost » (open source)
- Une facilité à rajouter de la puissance par simple ajout de machines
- **scalabilité horizontale** versus scalabilité verticale
- Une facilité à dé-commissionner du matériel défaillant ou obsolète
- Une redondance des serveurs clés (les serveurs sont en **H**igh **A**vailabilité)
- **Aucun SPOF** si on configure bien son cluster (théorie)
- **Zéro “downtime” (interruption de service)**

Data Lake version primitive

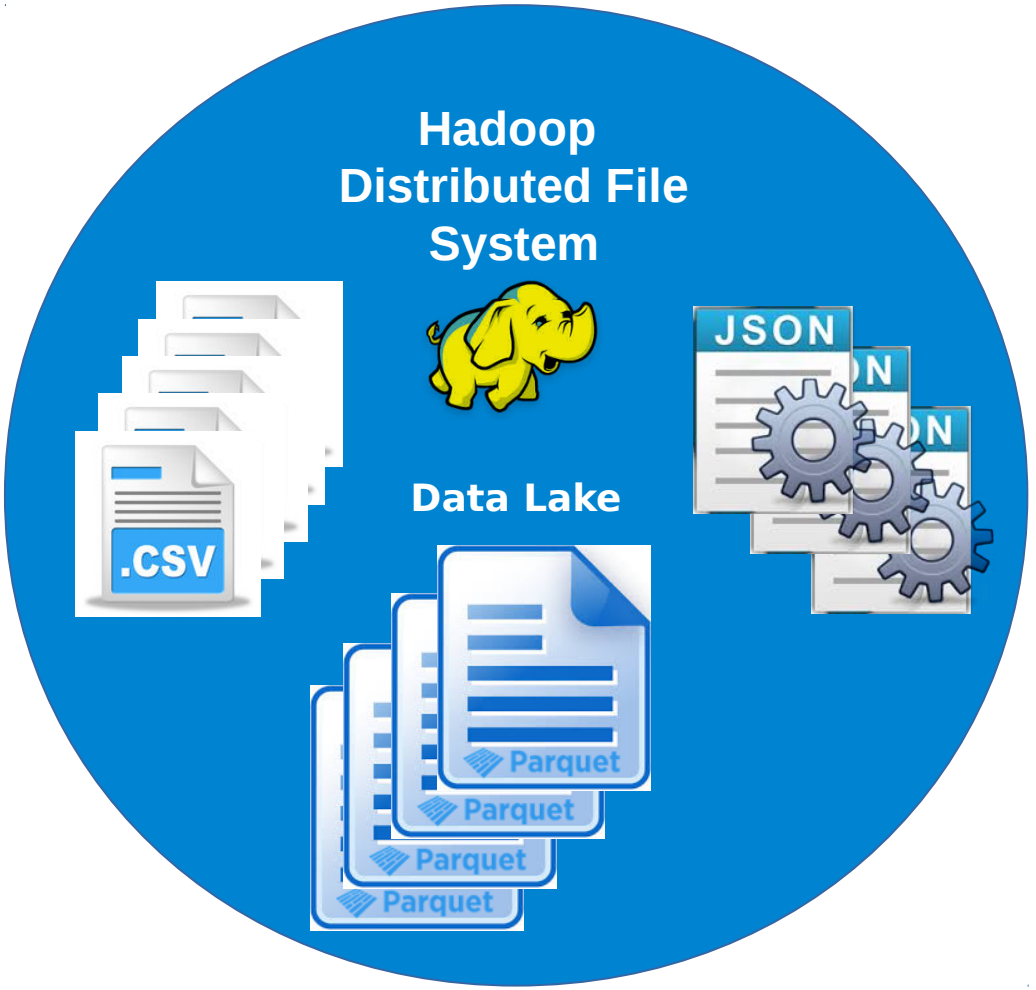
Un système de fichiers distribué



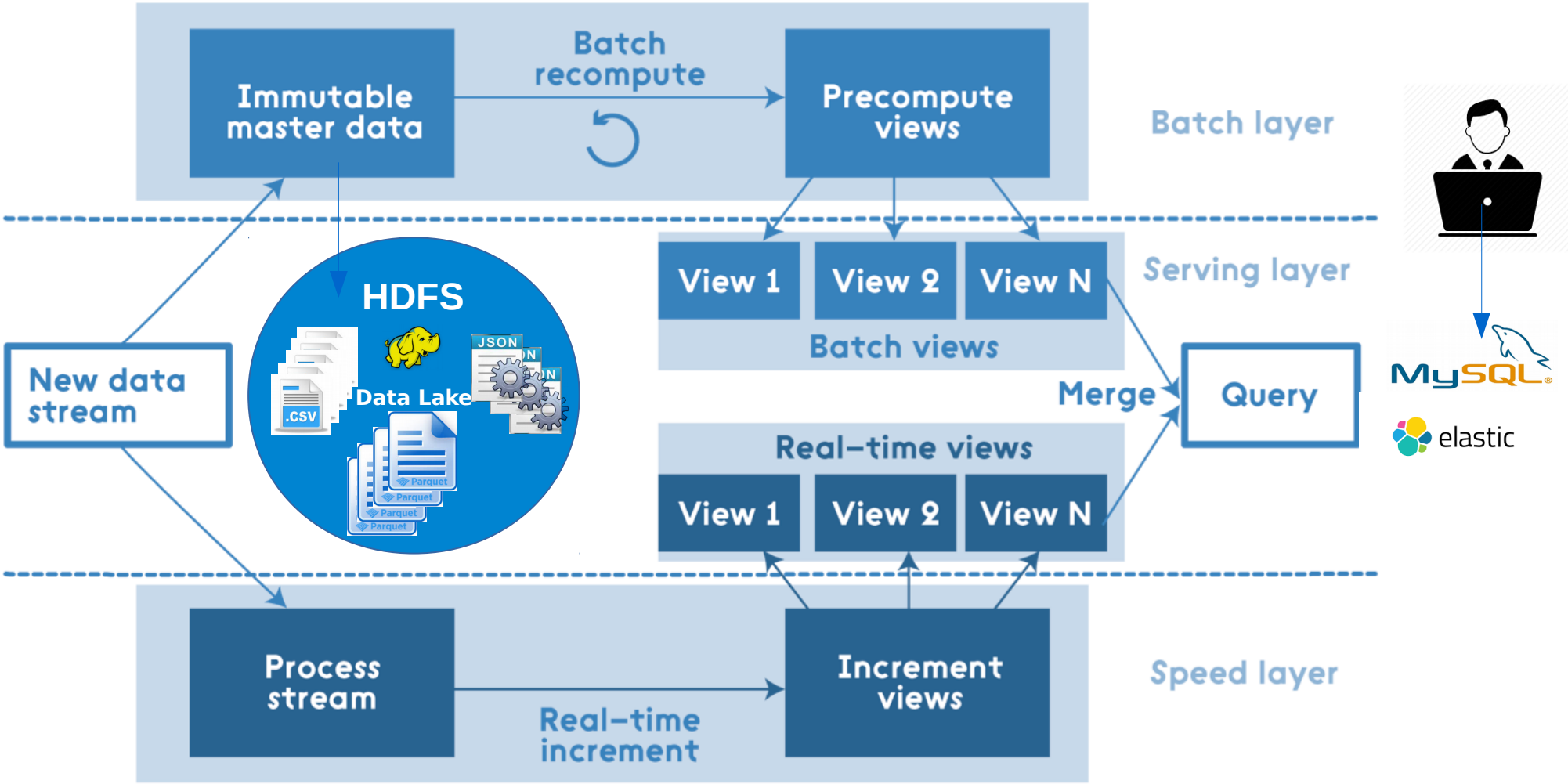
- Vu comme un système de fichier normal par les utilisateurs (folders, des fichiers...)
- En réalité des **fragments** de fichiers vont être stockés et **répliqués** sur plusieurs machines

Le Data Lake primaire

Les données sont dans un format **non propriétaire** sur un **système de fichier distribué**...



Lambda architectures



Le Lac Majeur

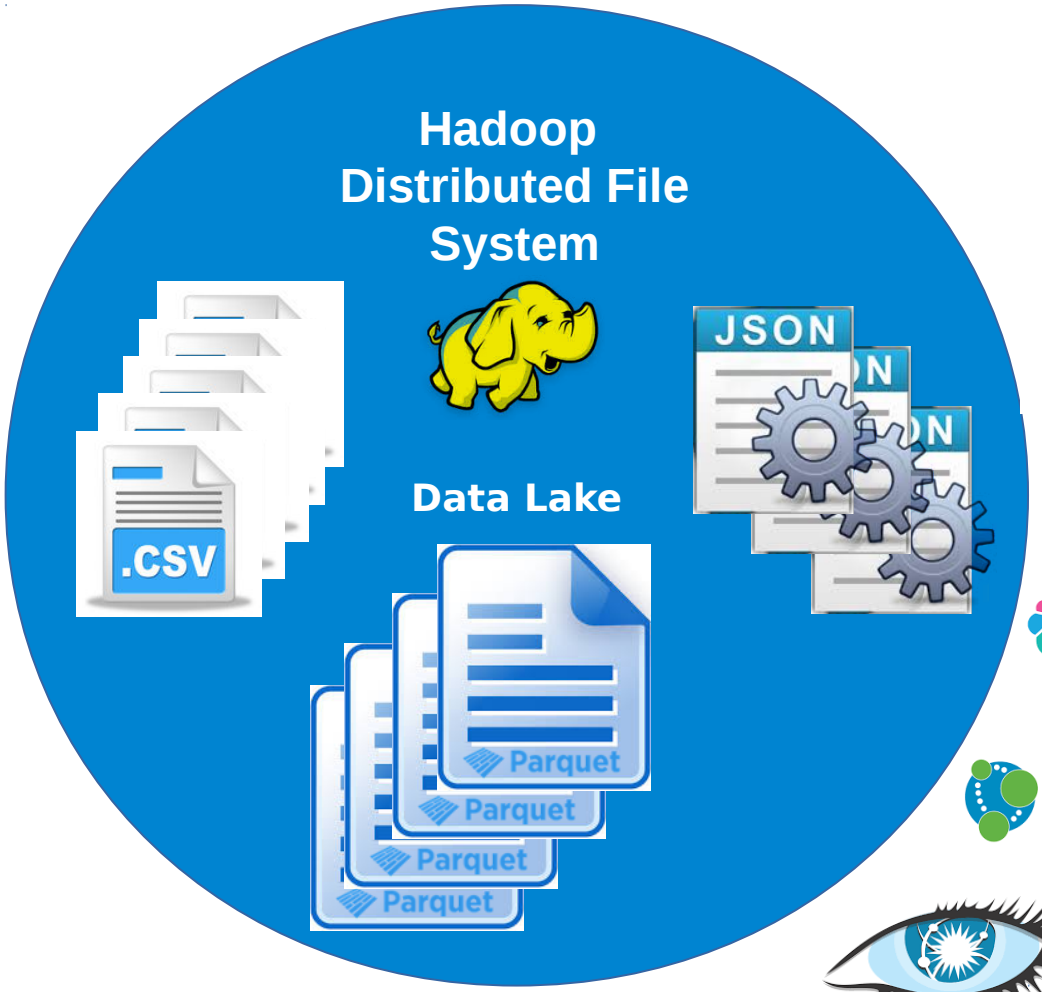
et des lacs secondaires

Sur du Big Data on ne pourra pas faire de l'analytique sur de grosses volumétries avec un utilisateur face à l'ordinateur…

Le Data Lake n'est pas juste un grand système de fichiers.

C'est aussi un ensemble de **technologies rapides et/ou particulières (moteurs de recherche, bases graphe)** dans lesquelles on stocke des vues… par exemple les données du jour, des données aggrégées (des stocks), des données en graphe (des arbres généalogiques): nos lacs secondaires.

Les technologies que l'on met "autour" du Data Lake primaire dépendent de la culture et des cas d'usage de l'entreprise.



ORACLE
Qlik® Sense
MySQL
PostgreSQL
elastic
Solr
neo4j
ArangoDB
cassandra
Couchbase
mongoDB.

SQL parallélisé

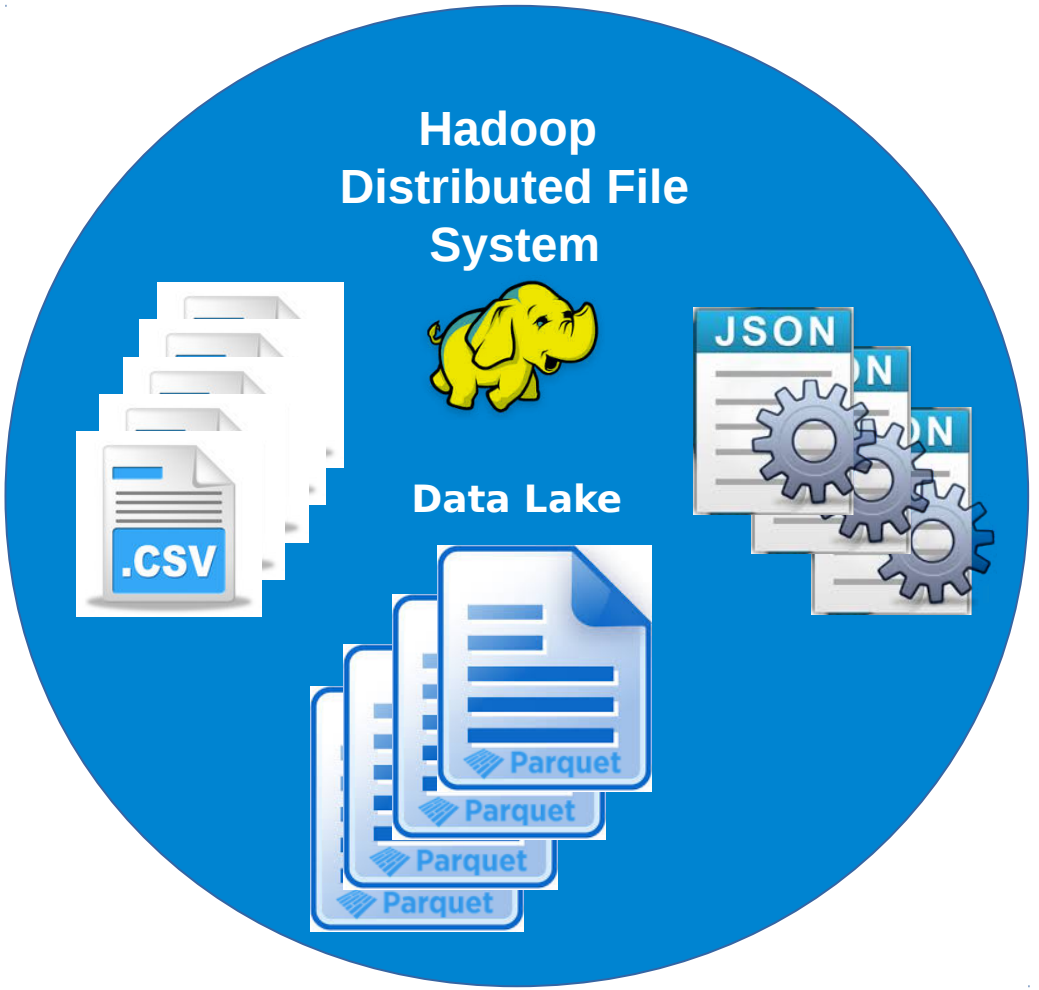
La première analytique ce sont des requêtes SQL parallélisées sur nos fichiers vus commes des bases de données...

SQL parallélisé sur le lac primaire

Driver OBDC

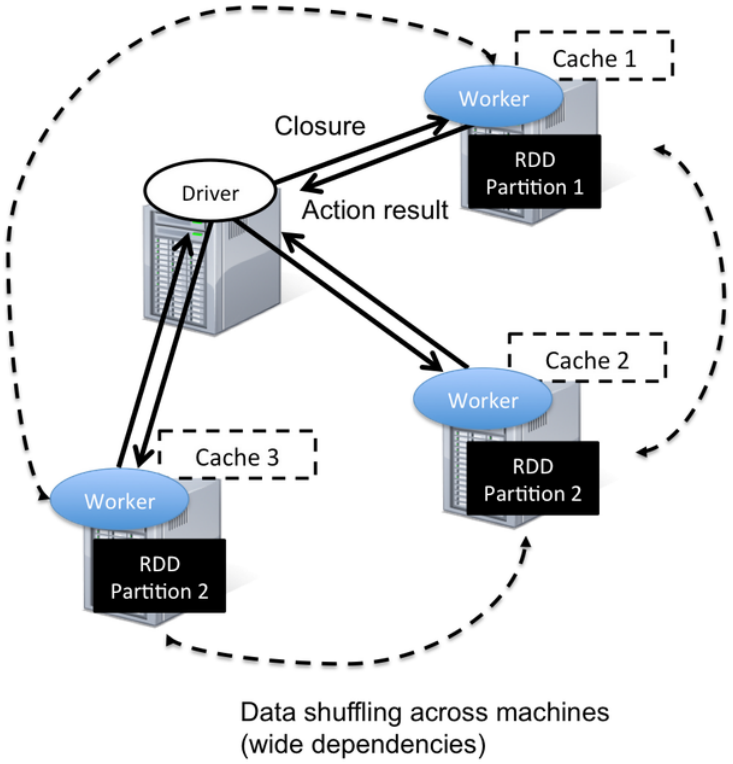
Hive, Spark SQL, Drill, Impala Hawk, BigSql etc...

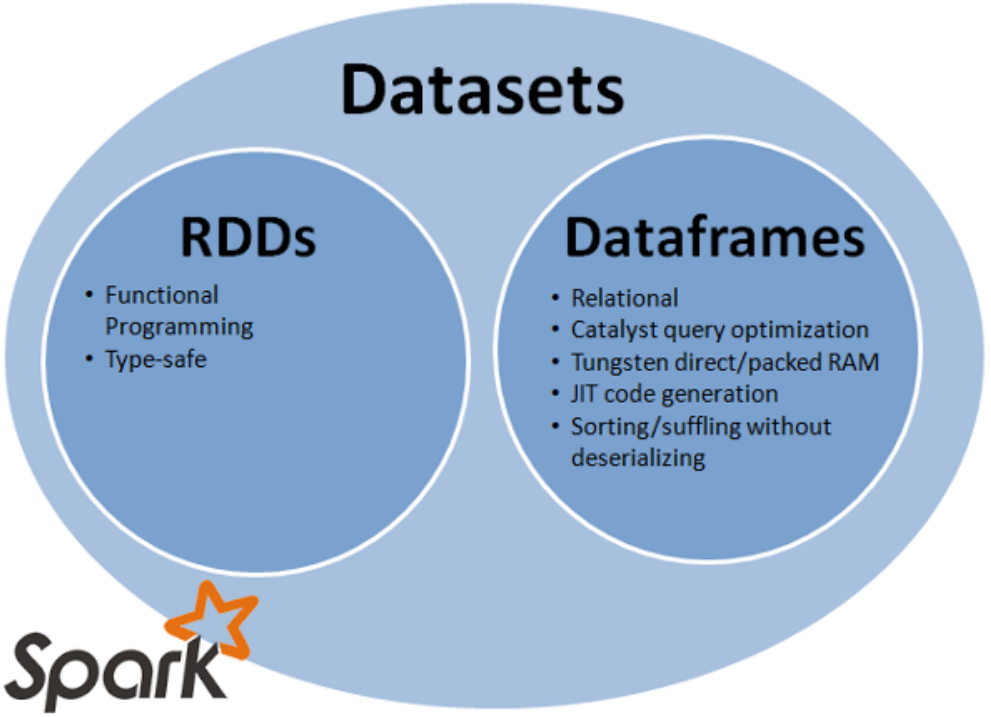
Traite les fichiers comme de vraies tables d'un DBMS

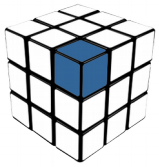


Analytique Avancée

- En général l'analytique avancée consistera à construire des **modèles** sur la **couche batch** (sur l'ensemble des données du Data Lake) et à appliquer ces **modèles** dans la couche temps réel. Pour cela un outil magique Spark!

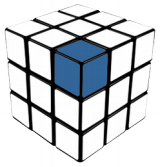






Le miracle des Data Frames

- Un Data Frame est un format interne en colonne et donc peut se requêter en SQL !
- On peut monter aujourd'hui en Data Frame des données en format JSON, Parquet, CSV, Avro, ORC etc.
- On peut aussi monter les données de beaucoup de bases de données et moteurs de recherche : Cassandra, MongoDB, Couchbase, Elasticsearch, SolR…
- On peut donc absolument tout croiser sans contrainte (les jointures sont possibles, etc.)
- Et on peut faire de la Data Science facilement sur toutes ces Data Frames...



Les modèles

- La Data Science en Spark… un régal (quelques libraries DeepLearning, MLLib etc.)…
- Un exemple de reconnaissance d'images.. on va apprendre à reconnaitre des images de personnes… là Steve Jobs et Mark Zuckerberg…

```
from sparkdl import readImages
from pyspark.sql.functions import lit

img_dir = "/PATH/TO/personalities/"

#Read images and Create training & test DataFrames for transfer learning
jobs_df = readImages(img_dir + "/jobs").withColumn("label", lit(1))
zuckerberg_df = readImages(img_dir + "/zuckerberg").withColumn("label", lit(0))
jobs_train, jobs_test = jobs_df.randomSplit([0.6, 0.4])
zuckerberg_train, zuckerberg_test = zuckerberg_df.randomSplit([0.6, 0.4])

#dataframe for training a classification model
train_df = jobs_train.unionAll(zuckerberg_train)

#dataframe for testing the classification model
test_df = jobs_test.unionAll(zuckerberg_test)
```

```
from pyspark.ml.classification import LogisticRegression
from pyspark.ml import Pipeline
from sparkdl import DeepImageFeaturizer

featurizer = DeepImageFeaturizer(inputCol="image", outputCol="features", modelName="Inceptic
lr = LogisticRegression(maxIter=20, regParam=0.05, elasticNetParam=0.3, labelCol="label")
p = Pipeline(stages=[featurizer, lr])
p_model = p.fit(train_df)
```

L’utilisation du modèle

- Une fois qu’on a un modèle il suffit de donner des images et le modèle va nous prédire si la photo est Steve Jobs ou Mark Zuckerberg
- Le code suivant va sortir des lignes avec le fichier et la prediction !

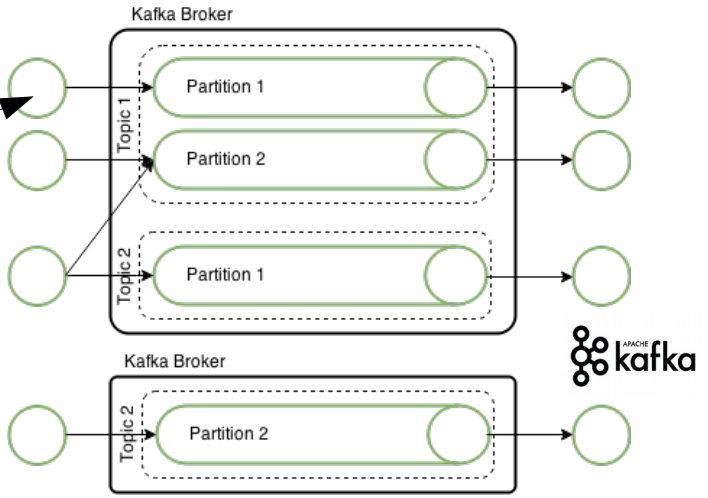
```
1  predictions = p_model.transform(test_df)
2
3  predictions.select("filePath", "prediction").show(truncate=False)
```

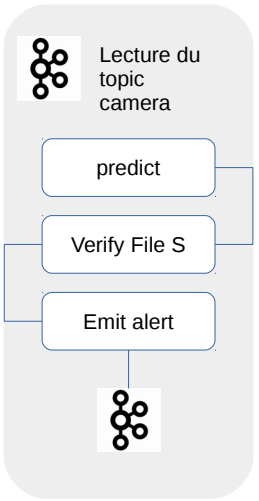
```
>>> predictions.select("filePath", "prediction").show(truncate=False)
Using TensorFlow backend.
+-----------------------------------------------------------------+----------+
|filePath                                                         |prediction|
+-----------------------------------------------------------------+----------+
|file:/home/zsellami/Téléchargements/image/jobs/steve5.jpg        |1.0       |
|file:/home/zsellami/Téléchargements/image/jobs/steve.jpg         |1.0       |
|file:/home/zsellami/Téléchargements/image/jobs/steve11.jpg       |1.0       |
|file:/home/zsellami/Téléchargements/image/jobs/steve13.jpg       |1.0       |
|file:/home/zsellami/Téléchargements/image/jobs/steve15.jpg       |1.0       |
|file:/home/zsellami/Téléchargements/image/jobs/steve6.jpg        |1.0       |
|file:/home/zsellami/Téléchargements/image/zuckerberg/mark3.jpg   |1.0       |
|file:/home/zsellami/Téléchargements/image/zuckerberg/mark7.jpg   |0.0       |
|file:/home/zsellami/Téléchargements/image/zuckerberg/mark8.jpg   |0.0       |
|file:/home/zsellami/Téléchargements/image/zuckerberg/mark9.jpg   |1.0       |
|file:/home/zsellami/Téléchargements/image/zuckerberg/mark13.jpg  |1.0       |
|file:/home/zsellami/Téléchargements/image/zuckerberg/mark14.jpg  |0.0       |
|file:/home/zsellami/Téléchargements/image/zuckerberg/mark15.jpg  |1.0       |
|file:/home/zsellami/Téléchargements/image/zuckerberg/mark2.jpg   |1.0       |
|file:/home/zsellami/Téléchargements/image/zuckerberg/mark5.jpg   |0.0       |
|file:/home/zsellami/Téléchargements/image/zuckerberg/mark6.jpg   |1.0       |
+-----------------------------------------------------------------+----------+
```

L’utilisation du modèle en temps-réel

agent

generation
d’un .png
tous les x frames
Et envoi sous forme
d’évènement dans Kafka





Spark Streaming

LogIsland :

l'entreprise en événements

Productors | Collectors | LogIsland | Visualizers

nifi

Machines Logs

logstash

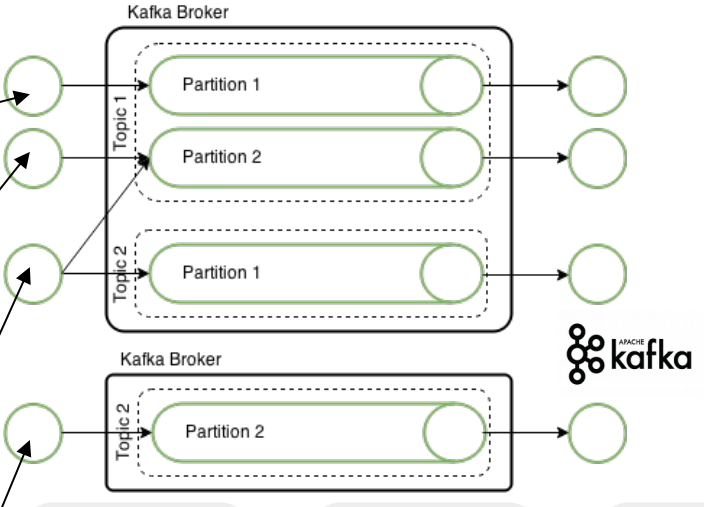
Sensors data

sqoop

CRM data

talend

Business data



Sensors queue

Normalize Field

Verify threshold

Emit alert

Sensors.yml

production queue

SQL on window count (products)

Compute TRS

Verify threshold

Emit alert

Production.yml

All queues All Events

Put ElasticSearch

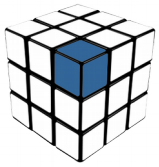
Put HDFS (Data Lake)

Storage.yml

elastic

hadoop HDFS

Qlik Sense



Ressources

- source : https://github.com/Hurence/logisland/releases
- docker : https://hub.docker.com/r/hurence/logisland/tags/
- maven : https://search.maven.org/#search%7Cga%7C1%7Clogisland
- documentation : http://logisland.readthedocs.io/en/latest/concepts.html
- support : https://gitter.im/logisland/logisland
- contact : Laurence.Hubert@hurence.com