



et ses composants principaux

Cours 2017-2018



A propos

Olivier Large

Data scientist / data engineer @ Pollendata

olivier.large@pollendata.io

But du cours :

- Maîtriser le vocabulaire Hadoop
- Développer avec les principaux composants

Evaluation :

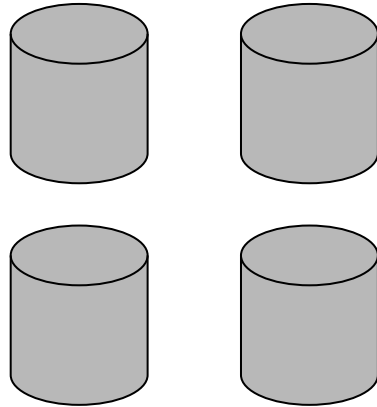
- 2 questionnaires à faire chez soi



1. **Hadoop - présentation générale**
2. Hadoop et ses briques principales
 - a. HDFS
 - b. YARN
3. Les frameworks de traitements
 - a. MapReduce
 - b. Hive et Pig
 - c. Oozie
4. Uses cases
 - a. Analyse de la qualité de l'air

Définition - cluster

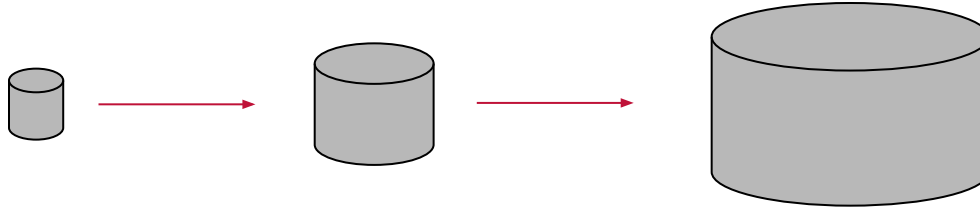
Groupe de ressources (type serveur) qui agit comme un seul et même système



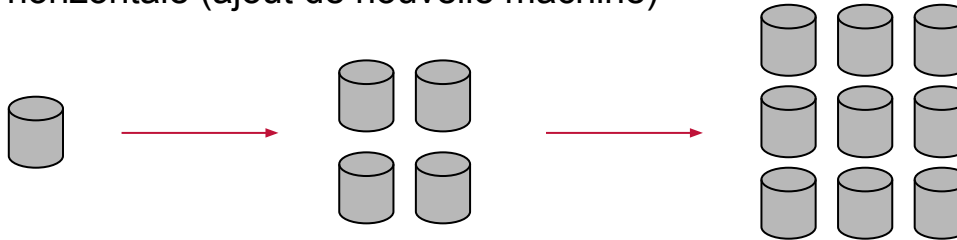
Définition - scalabilité

Désigne la capacité d'un produit à s'adapter à un changement d'ordre de grandeur de la demande.

scalabilité verticale (améliorer une seule machine)



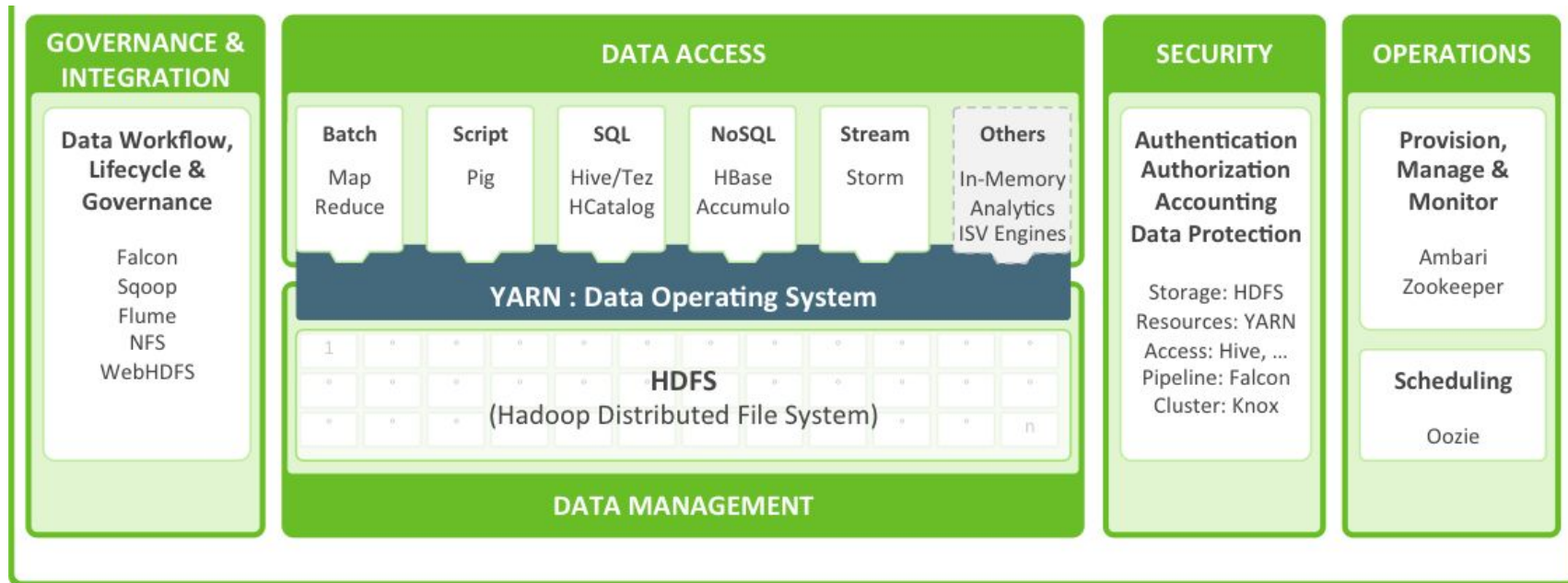
scalabilité horizontale (ajout de nouvelle machine)



Hadoop - vue d'ensemble

- Ensemble de technologies distribuées permettant de stocker et traiter de gros volumes de données (> To).
- Open Source
- Développé initialement par Doug Cutting (Yahoo)
- Compte aujourd'hui ~100 committers actifs (principalement chez Hortonworks)
- Scalabilité horizontale forte
- Sécurisation native des données (triple réplication sur des disques différents)

Hadoop - écosystème





Hadoop - usages types

- Archivage actif (les données stockées/archivées peuvent être directement exploitées par des applications)
- Traitement de données non structurées (connaissance client, détection de fraude, analyse d'images)
- Délégation de traitements lourds en profitant de la parallélisation (pour raccourcir le temps sur des requêtes complexes, etc.)



Hadoop - positionnement

- Quelques limites :
 - Des compétences rares
 - Technologies jeunes (de nouvelles versions tous les mois)
 - Fonctions de gestion administratives peu évoluées



1. Hadoop - présentation générale
2. **Hadoop et ses briques principales**
 - a. HDFS
 - b. YARN
3. Les frameworks de traitements
 - a. MapReduce
 - b. Hive et Pig
 - c. Oozie
4. Uses cases
 - a. Analyse de la qualité de l'air

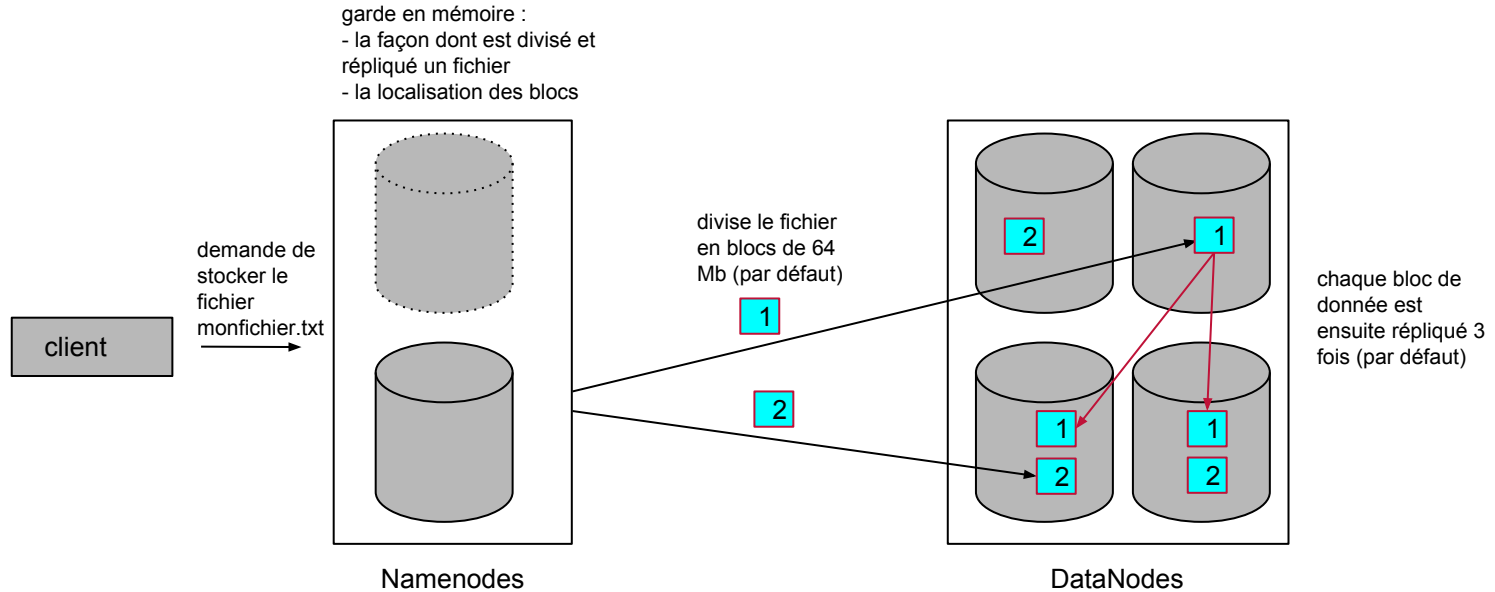
HDFS - définition

- **Hadoop Distributed File System : système de gestion de fichiers distribués dans Hadoop**
- **principe de base : haute tolérances aux pannes (à la base fait pour fonctionner sur des machines peu coûteuses) - les données sont répliquées**
- **optimisé pour stocker des fichiers volumineux**
- **fonctionne sur Unix, développé en Java**

2 composants :

- **Namenode : gardent un index des données**
- **Datanode : écrivent/lisent les fichiers stockés**

HDFS - schéma explicatif



HDFS - notes

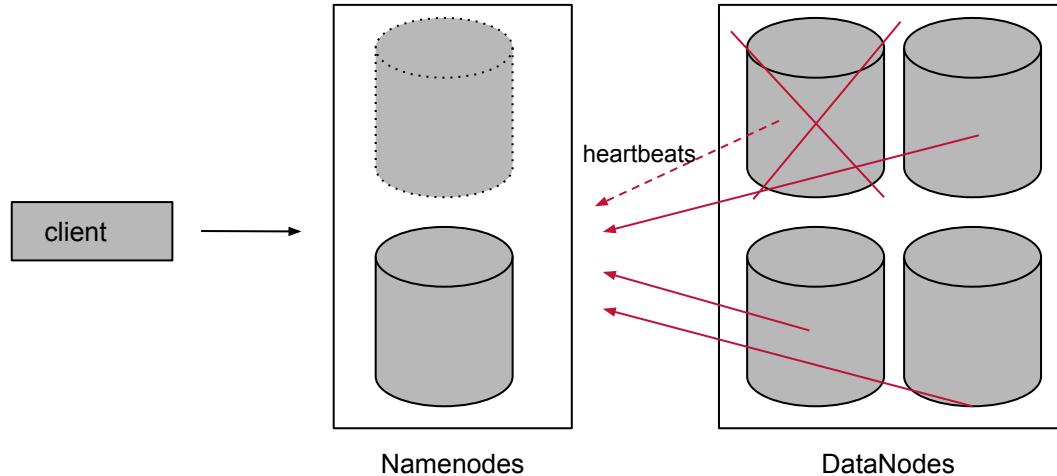
- **le facteur de réplication est de 3 par défaut**
- **la taille des blocks HDFS est de 64 Mo**
 - stocker un fichier de 1 ko ou de 60 Mo demandera un bloc de 64 Mo dans HDFS. Passer les 64 Mo en demandera un nouveau
 - motivations : les requêtes dans HDFS sont chères (rappatrient beaucoup de métadonnées). Séparer les données en blocs de 64 Mo est une optimisation sur les requêtes
 - cela amène à la question : comment stocker des petits fichiers dans Hadoop (vu dans la suite)
- **répartition intelligente (essaye de répartir la charge entre les machines)**
- **Stockage “rackaware” (stocke sur des racks différents)**

HDFS - case of failure

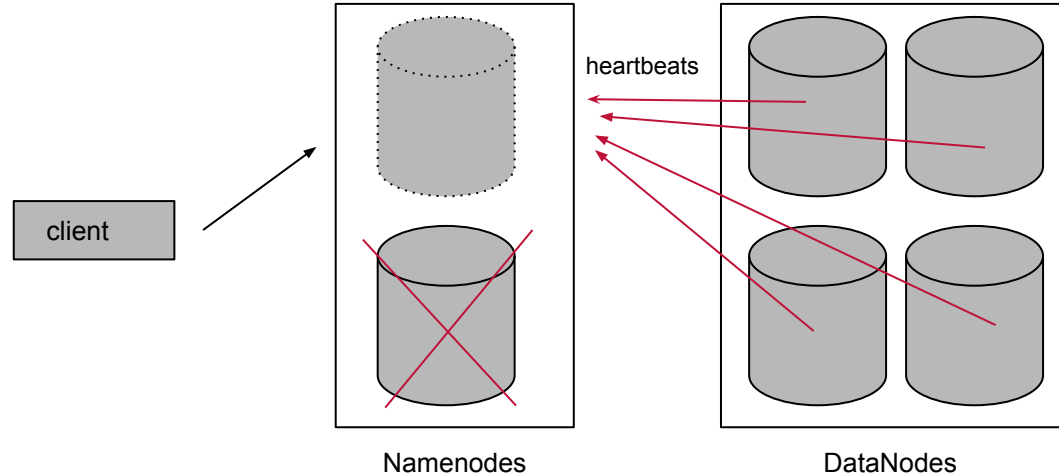
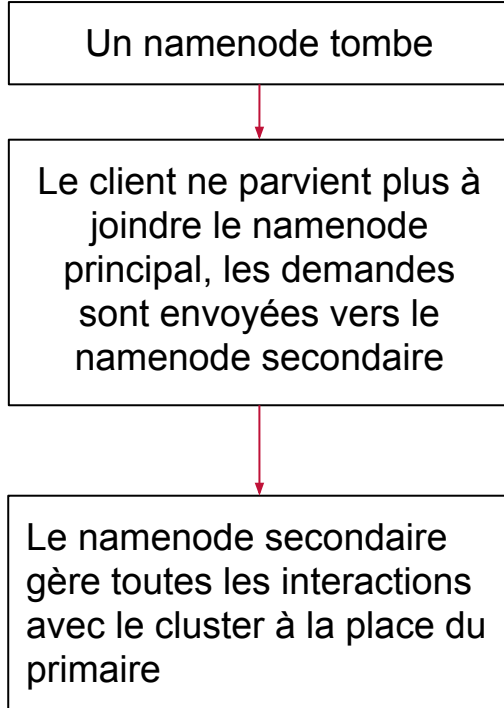
Un datanode tombe

Il n'envoie plus de heartbeat aux namenodes

Si le namenode ne reçoit rien pendant une certaine durée, les blocs de données sont répliqués vers d'autres machines



HDFS - case of failure



HDFS - usage

HUE :

Rechercher un nom de fichier

Accueil / user / filRouge

Actions

- Renommer
- Déplacer
- Copier
- Télécharger
- Modifier les autorisations

	Nom	Size	Utilisateur	Groupe	Autorisations	Date
	.		filRouge	filRouge	drwxr-xr-x	June 06, 2016 11:02 PM
	.		filRouge	filRouge	drwxr-xr-x	June 06, 2016 11:03 PM
	2015-06.csv	81,1 Mio	filRouge	filRouge	-rw-r--r--	June 06, 2016 10:28 PM
✓	2015-07.csv	179,1 Mio	filRouge	filRouge	-rwxr-xr-x	June 06, 2016 10:43 PM
	2015-08.csv	218,3 Mio	filRouge	filRouge	-rwxr-xr-x	June 06, 2016 10:48 PM
	2015-09.csv	159,0 Mio	filRouge	filRouge	-rwxr-xr-x	June 06, 2016 10:47 PM
	2015-10.csv	156,9 Mio	filRouge	filRouge	-rwxr-xr-x	June 06, 2016 10:51 PM
	2015-11.csv	191,1 Mio	filRouge	filRouge	-rwxr-xr-x	June 06, 2016 10:55 PM
	2015-12.csv	163,1 Mio	filRouge	filRouge	-rwxr-xr-x	June 06, 2016 10:55 PM

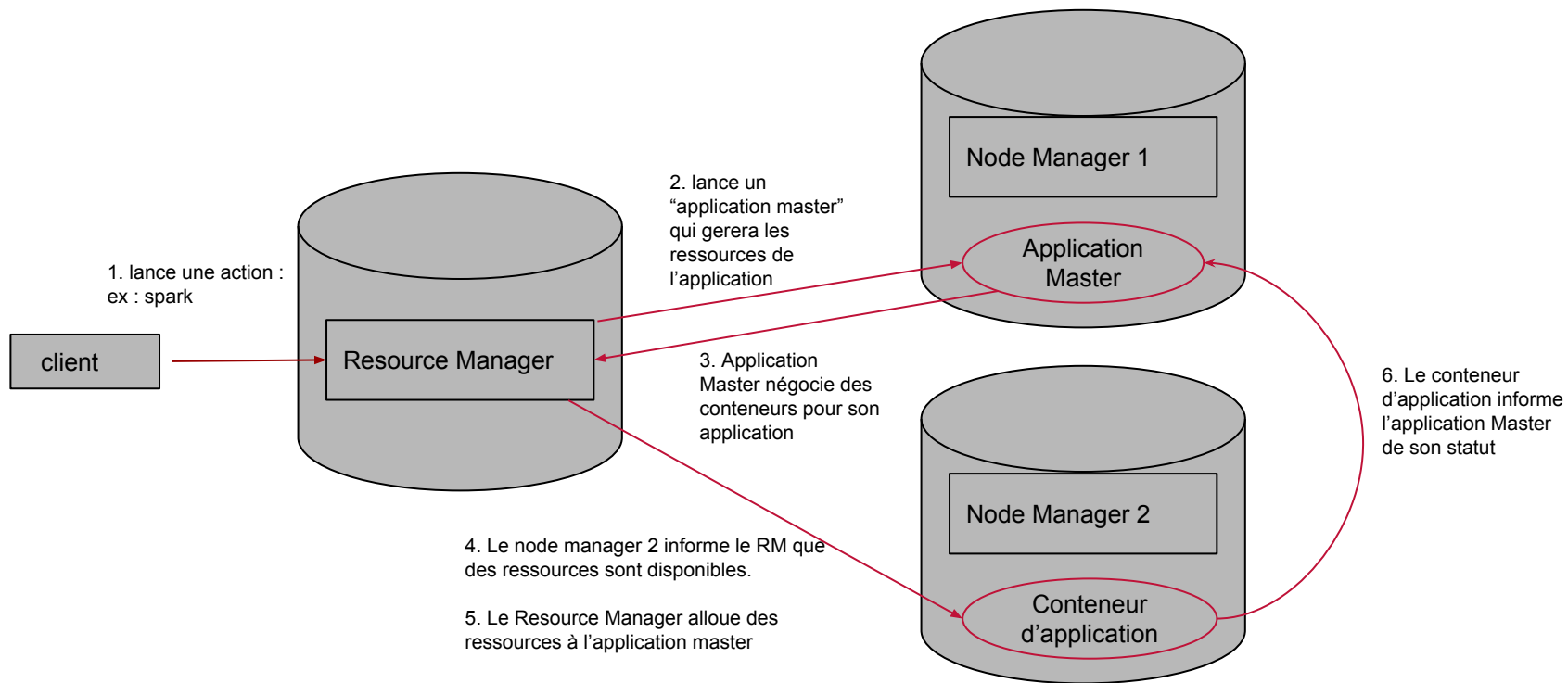
Command line

hadoop fs - ...

YARN - définition

- **Yet Another Resource Negotiator : système de gestion de ressources dans Hadoop**
- **Yarn gère l'allocation des ressources RAM, CPU, network pour chaque application lancée dans Hadoop**
- **3 composants principaux :**
 - Resource Manager (RM) : gère l'ensemble des ressources pour tout le cluster.
 - Node Manager : monitore les ressources par machine, reporte les informations au RM
 - Application Master (AM) : gère les ressources pour une application, négocie ses ressources au RM

YARN - schéma explicatif



YARN - notes

- **on peut définir des queues de priorité dans YARN**
 - les applications exécutées avec une queue de haut niveau de priorité pourront détruire des conteneurs d'applications avec un faible niveau
 - on peut associer des % de ressources maximales à chaque queues
- **YARN - long running application : les bases de données dans Hadoop sont aussi gérées par Yarn (exemple : HBase)**
- **YARN - label : on peut associer des labels aux node managers et associer des queues à une partie du cluster seulement**
- **Le futur de YARN : l'idée est que toute application, y compris un serveur web puisse tourner dans dans un docker embarqué dans un conteneur YARN**

YARN - usage

queue YARN

queue priority

HUE

Query Editors ▾ Data Browsers ▾ Workflows ▾

Navigateur de fichiers Job Browser 1 filRouge

Job Browser

Nom d'utilisateur filRouge Texte Rechercher un texte

Réussi En cours d'exécution Echoué Détruit

Journaux	ID	Nom	Application Type	Statut	Utilisateur	Maps	Reduces	File d'attente	Priorité	Durée	Envoyé	
	1462264675501_0223	PySparkShell	SPARK	RUNNING	filRouge	100%	100%	root.filRouge	N/A	33s	06/07/16 13:08:03	Détruire
	1462264675501_0222	PySparkShell	SPARK	SUCCEEDED	filRouge	100%	100%	root.filRouge	N/A	1h:15m:4s	06/06/16 23:09:50	
	1462264675501_0214	PySparkShell	SPARK	SUCCEEDED	filRouge	100%	100%	root.filRouge	N/A	2h:8m:48s	06/06/16 16:55:35	
	1462264675501_0209	PySparkShell	SPARK	SUCCEEDED	filRouge	100%	100%	root.filRouge	N/A	11m:51s	06/06/16 16:42:23	
	1462264675501_0207	PySparkShell	SPARK	SUCCEEDED	filRouge	100%	100%	root.filRouge	N/A	3m:9s	06/06/16 16:38:15	
	1462264675501_0198	PySparkShell	SPARK	SUCCEEDED	filRouge	100%	100%	root.filRouge	N/A	1m:35s	06/06/16 16:23:04	
	1462264675501_0196	PySparkShell	SPARK	SUCCEEDED	filRouge	100%	100%	root.filRouge	N/A	2h:44m:14s	06/06/16 16:22:29	

Questionnaire

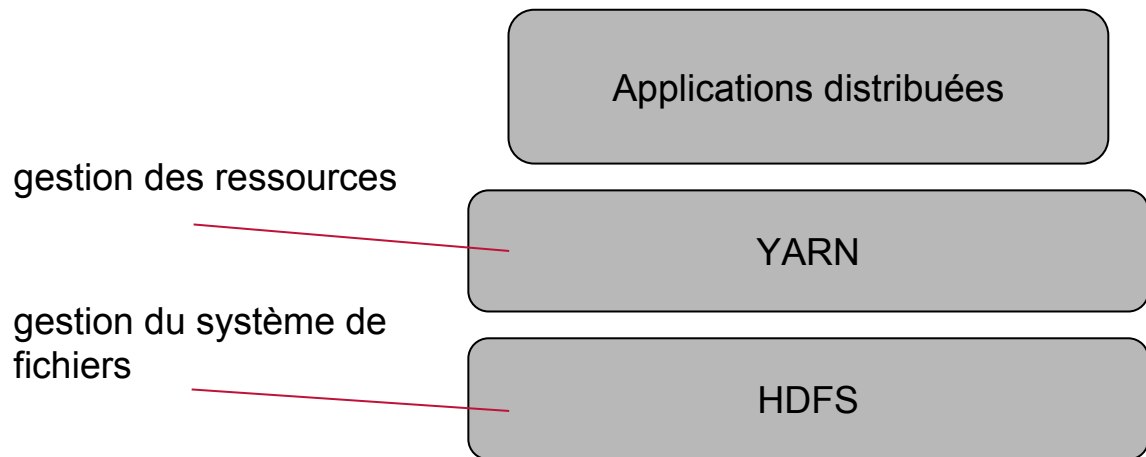
Quelles sont les bonnes réponses ?

- **A. HDFS est un système de gestion de fichiers optimisé pour stocker de gros fichiers.**
- **B. HDFS ne réplique pas les données dans le cluster. Il faut insérer les données plusieurs fois pour faire des sauvegardes.**
- **C. YARN manage les ressources de mon application**
- **D. Avec YARN, je peux faire passer mon programme en priorité élevée et ainsi récupérer les ressources initialement allouées aux programmes de mes camarades**



1. Hadoop et ses briques principales
 - a. Hadoop - vue d'ensemble
 - b. HDFS
 - c. YARN
2. **Les frameworks de traitements**
 - a. MapReduce
 - b. Hive et Pig
 - c. Oozie
3. Uses cases
 - a. Analyse de la qualité de l'air

Rappel : schéma Hadoop basique

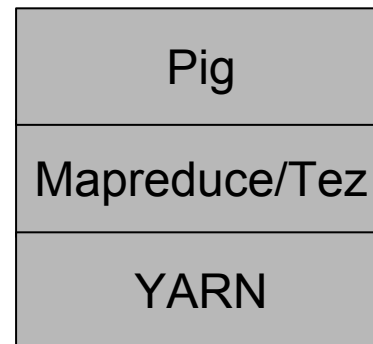


MapReduce - définition

- **framework permettant de paralléliser une application**
- **2 étapes : map (traite des tâches parallélisées), reduce (travaille sur un groupement données)**
- **“data locality optimization” : les traitements s’effectuent préférentiellement là où sont stockées les données**
 - => “moving calcul is cheaper than moving data”
- **Aujourd’hui, grande perte d’intérêt pour ce framework au profit de Spark ou Tez/Impala. Seulement intéressant pour de très grandes quantités de données.**



- **ETL de Hadoop : permet de créer des chaînes de traitement dans Hadoop**
- **Langage Pig Latin**
- **Surcouche de MapReduce (ou Tez, Impala...)**



Pig usage

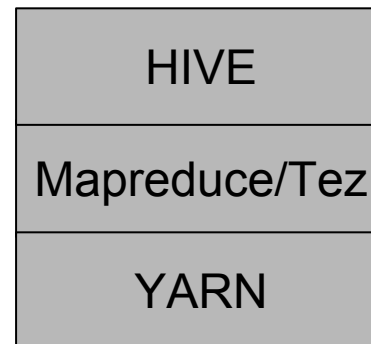
Pig Script for example query

```
1. visits = load '/data/visits' as (user, url, time);
2. gVisits = group visits by url;
3. visitCounts = foreach gVisits generate url, count(visits);
4. urlInfo = load '/data/urlInfo' as (url, category, pRank);
5. visitCounts = join visitCounts by url, urlInfo by url;
6. gCategories = group visitCounts by category;
7. topUrls = foreach gCategories generate top(visitCounts,10);
8. store topUrls into '/data/topUrls';
```

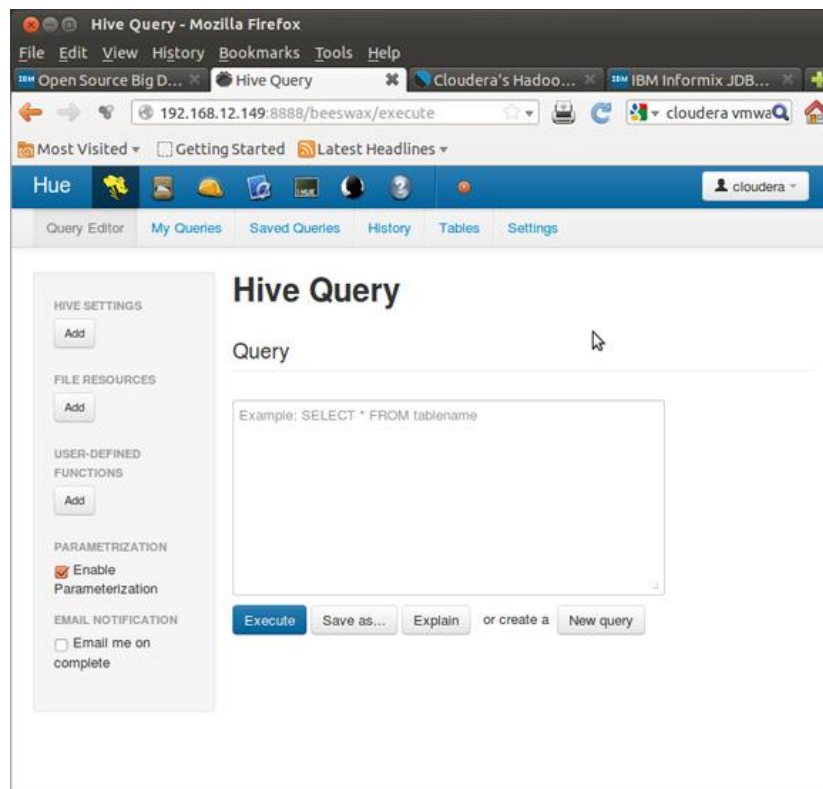
Pig Slides adapted from Olston et al. (SIGMOD 2008)



- ~ DataWarehouse de Hadoop. Ajoute une structure (couche de métadonnées) sur les fichiers dans HDFS et une interface de requête en ~SQL.
- Une requête SQL dans Hive exécutera des tâches parallélisées. Une requête SQL HIVE est traduite dans l'un des frameworks de distribution (mapreduce, Impala, Tez, etc.)
- Très utilisé dans les grandes entreprises
- Connecteurs JDBC
- Contraintes ACID (avec configuration)



HIVE usage



HIVE et PIG : complémentarité

- Hive et Pig possèdent beaucoup de fonctionnalités redondantes (extraction, transformation)
- Dans certaines entreprises (exemple : Yahoo), elles sont utilisées de façon complémentaires, comme dans leur “data factory”, notamment pour l’analyse de logs :



source : <https://developer.yahoo.com/blogs/hadoop/pig-hive-yahoo-464.html>

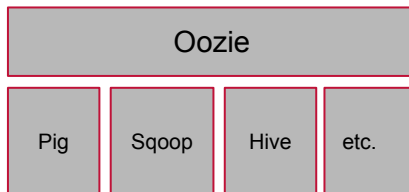
Questionnaire

Quelles sont les bonnes réponses ?

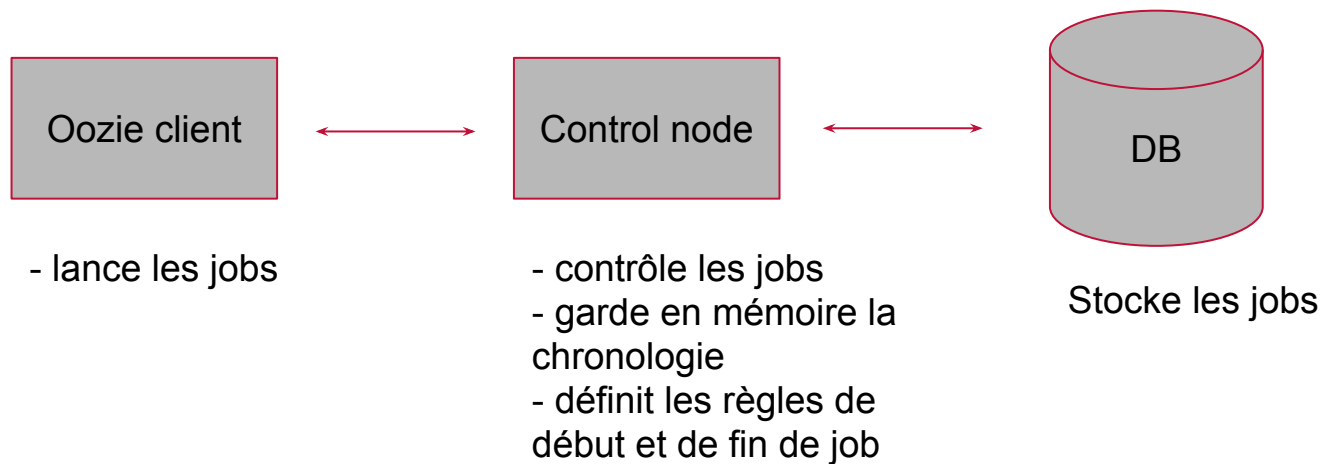
- A. Pig permet de requêter en SQL dans son datawarehouse
- B. MapReduce permet de stocker des données
- C. Dans un cluster Hortonworks, il vaut mieux utiliser Tez en background de Hive plutôt que MapReduce
- D. Connecter un outil de visualisation (type Qlikview, Tableau) à Hive est impossible.

Oozie

- **Open Source - java web app**
- **Workflow scheduler dans Hadoop**
- **Un workflow est un ensemble d'actions (MR, interactions HDFS, Pig, script Java ou shell, etc.)**
- **Oozie permet de créer les jobs, de les piloter et les automatiser**
- **2 types de jobs dans Oozie :**
 - Oozie workflow : les jobs sont des Directed Acyclical Graphs (DAGs), une séquence d'actions à exécuter.
 - Oozie coordinator : Oozie workflow de façon récurrente (exemple : lancés tous les matins au moment de l'arrivée de nouvelle donnée)



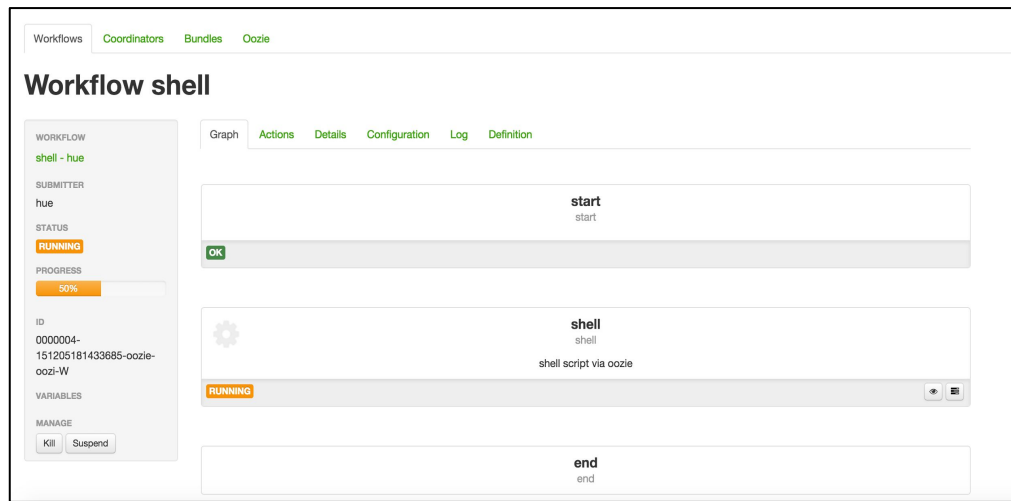
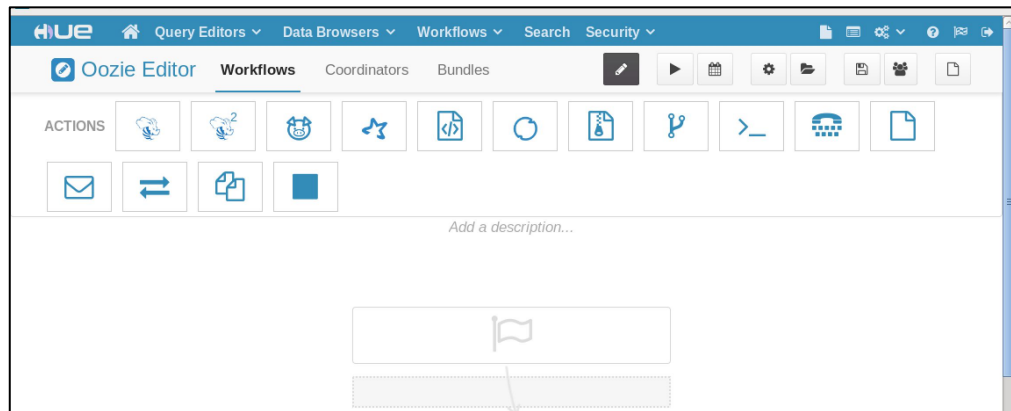
Oozie - Architecture



Oozie - usage

Depuis Hue :

- drag&drop pour créer son job





1. Hadoop et ses briques principales
 - a. Hadoop - vue d'ensemble
 - b. HDFS
 - c. YARN
2. Les frameworks de traitements
 - a. MapReduce
 - b. Hive et Pig
 - c. Oozie
3. Uses cases
 - a. Analyse de la qualité de l'air

Use case : analyse de la qualité de l'air

Une entreprise chargée du suivi de la qualité de l'air vous demande de développer une chaîne de traitement afin de valider et stocker des données.

L'entreprise vous envoie des données chaque jour sur un de vos serveurs.

Les données correspondent à 24h de mesures avec les différentes quantités de polluants. Les données sont au pas minute.

Le traitement est assuré en deux parties :

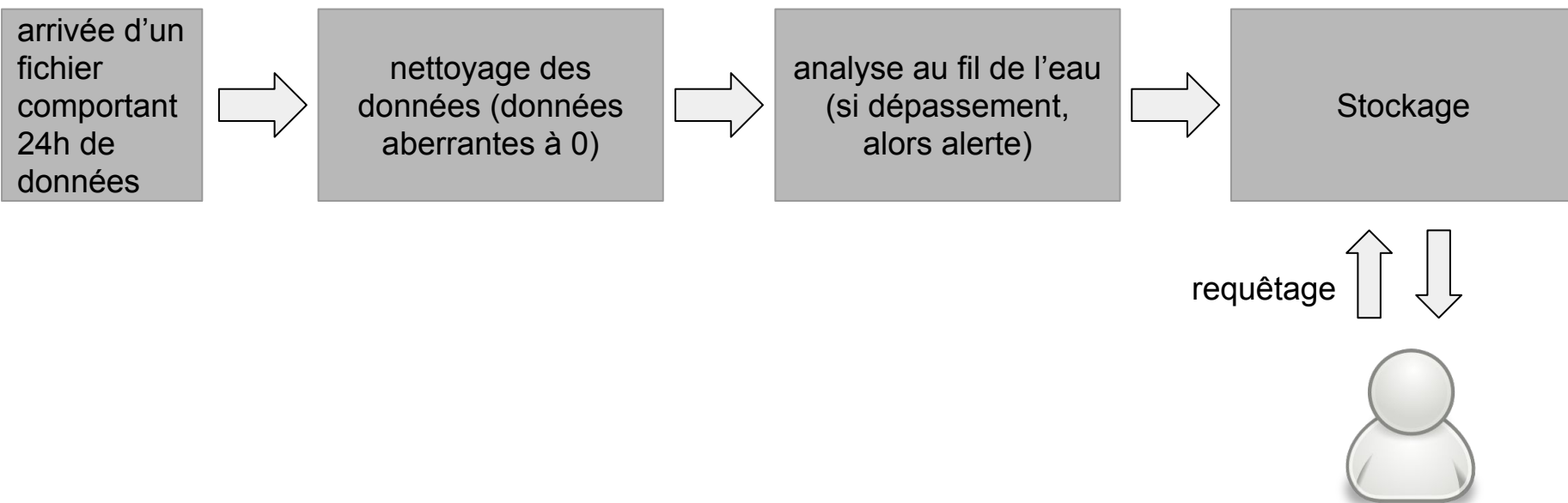
- qualité des données : les données utilisées doivent être nettoyées, les valeurs aberrantes (celles à 10000) doivent être mises à la même valeur que la donnée précédente.
- Les données doivent être moyennées à l'heure.
- analyse au fil de l'eau : vérifier si les polluants dépassent certaines quantités (les seuils sont présents dans le fichier seuil attaché). Ecrire un rapport d'alerte des seuils dépassés sur la journée dans HDFS

Les données doivent être stockées et pouvoir être requêtées en SQL par des personnes du métier.

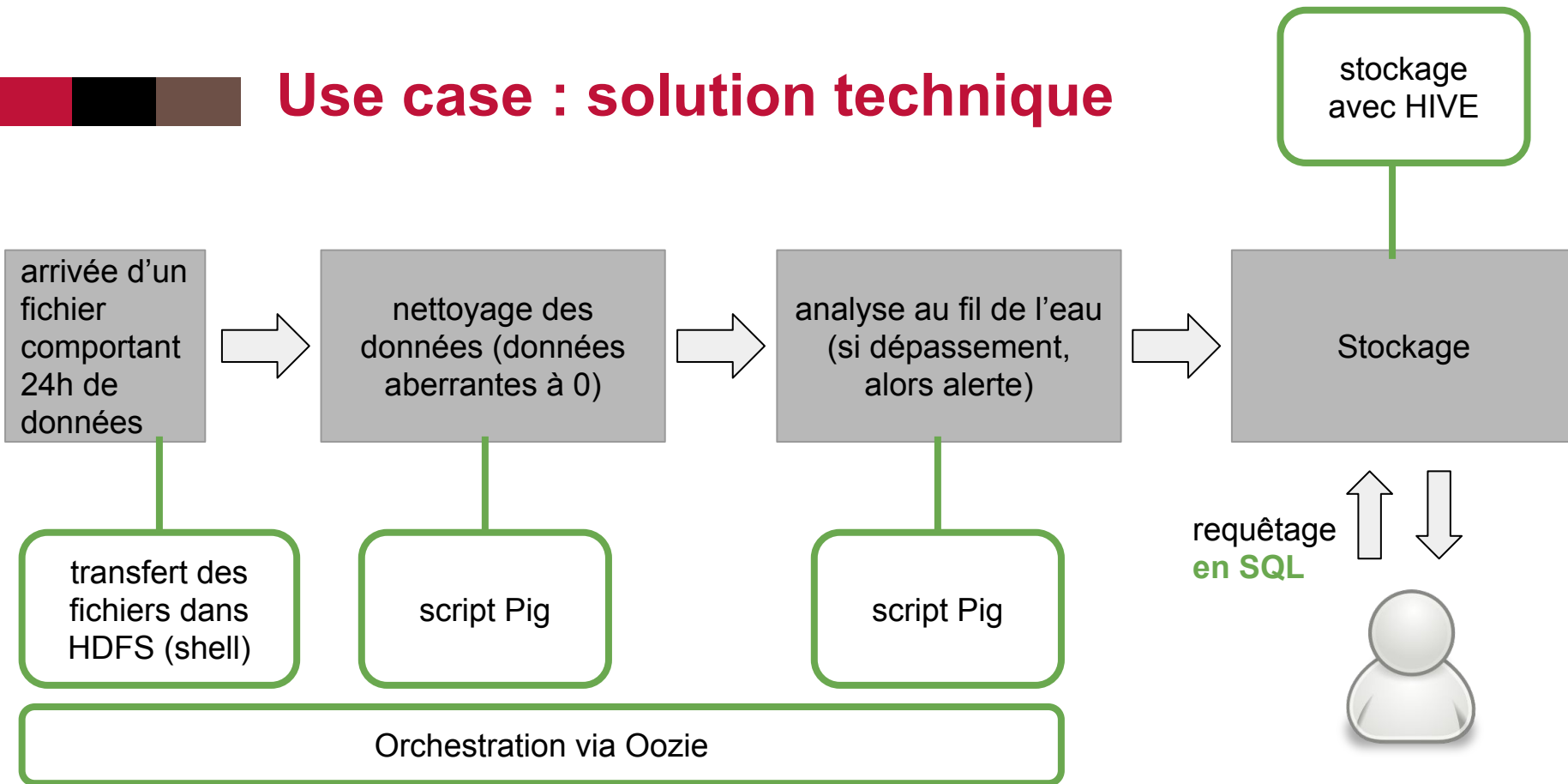
Celles-ci doivent pouvoir répondre à la question :

“sur cette nouvelle journée insérée, dans quel ville le taux de NO2 a-t-il le plus augmenté entre 0h et 12h ?”

Use case : chaîne de traitement



Use case : solution technique



Matériel :

- Machine virtuelle Cloudera
- Développements des scripts via Hue

Scripts à développer :

- bash d'envoi des fichiers sur HDFS
- script de transformation des données avec Pig
- insertion des données dans Hive
- orchestration via Oozie
- recherches via Hive

Annexe - HUE

Interface web qui interagit avec les différentes briques d'Hadoop

The screenshot shows the HUE web interface. The top navigation bar is blue and contains the HUE logo, a home icon, and several menu items: 'Query Editors', 'Data Browsers', 'Workflows', 'Navigateur de fichiers', 'Job Browser', and 'filRouge'. Below the navigation bar, the main content area is titled 'Mes documents'. On the left, there is a sidebar with sections: 'ACTIONS' (containing 'Nouveau document' and 'trash'), 'MES PROJETS' (containing a message about no projects), and 'PARTAGÉ AVEC MOI' (containing a message about no shared projects). The main content area has a search bar and a message: 'Ce projet ou cette balise ne contient actuellement aucun document.' Red arrows point from text labels to specific parts of the interface: 'accès autres services' points to the 'Data Browsers' menu; 'accès HDFS' points to the 'Navigateur de fichiers' menu; 'accès YARN RM' points to the 'Job Browser' menu; and 'gestion des utilisateurs' points to the 'filRouge' menu.

accès autres services

accès HDFS

accès YARN RM

gestion des utilisateurs