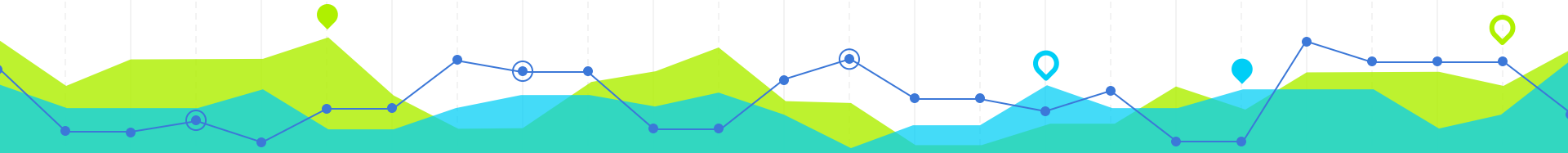
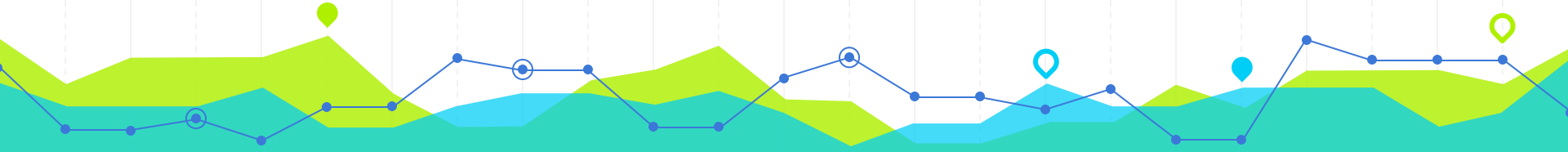


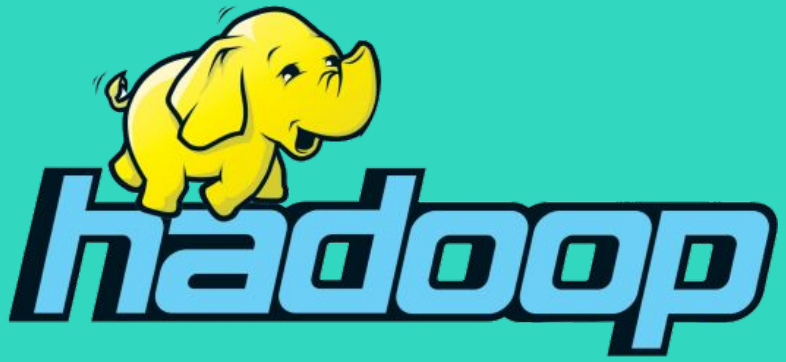
Réalisé par:
Chahoud Marwane & Marzouki Fouad

Hadoop

Année Universitaire: 2016-2017 - ENSAK

Sommaire





Introduction 1

Motivation

- Production massive de données => Volume,Variété et Vélocité
- Sources de données :
  - Les applications => des logs, des réseaux de capteurs, des rapports de transactions, des traces de GPS… etc
  - Les individus => des photographies, des vidéos, des musiques, des données sur l'état de santé (rythme cardiaque, pression ou poids

Problématiques

- Stockage et à l'analyse des données
- La capacité de stockage des disques durs augmente, le temps de lecture croît

Nécessité

- Paralléliser les traitements en stockant sur plusieurs unités
- Duplication des données comme dans un système RAID

Apache Hadoop ?

- Système de stockage distribué via HDFS (duplication)
- Système d'analyse des données MAPReduce

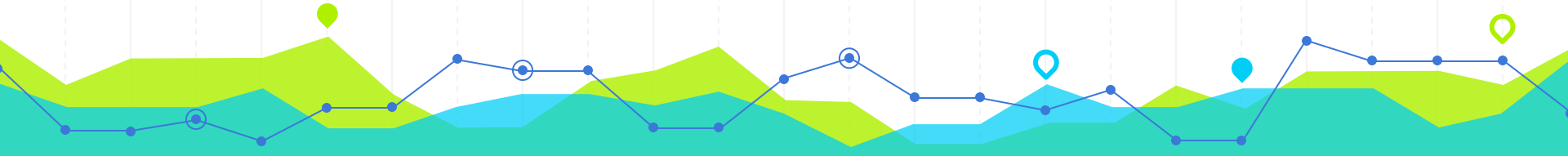
Domaines d’application

Santé

- L’ensemble des données socio-démographiques et de santé disponibles auprès de différentes sources.

Intérêts de l’exploitation:

- Identification de facteurs de risque de maladie
- Aide au diagnostic
- Aide aux choix et au suivi de l’efficacité des traitements



Télécommunication

- Anticiper les risques de résiliation d’abonnement
- Retrouver les informations de géolocalisation
- Traiter les données mobiles, données de transaction, de consultation d'url, de consommation média ou d'objets connectés

Énergie

- Créer un réseau de distribution plus intelligent capable d’optimiser l’ensemble de la chaîne énergétique (Smart Grid)

- Proposer des applications de gestion des énergies

Transport

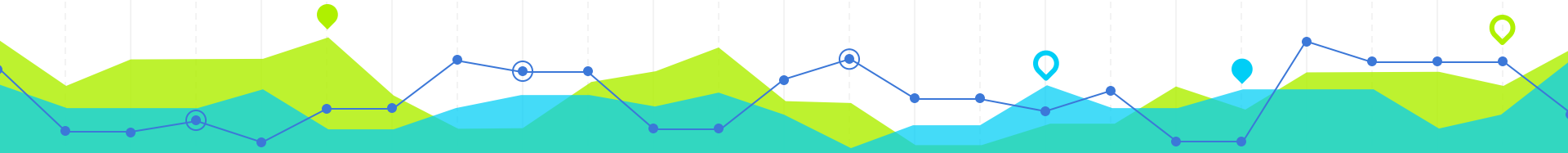
- Se renseigner sur son itinéraire
- Utiliser des données publiques, des données temps réel mais aussi des données de ses utilisateurs pour calculer le trajet le plus court
- Réduire les délais et les coûts des moyens de transport

Vente

- Déterminer le comportement utilisateur ou les recommandations de recherche

Images et videos

- Effectuer le traitement d’images (satellite par exemple) et de vidéos



Architecture d'une requête SQL

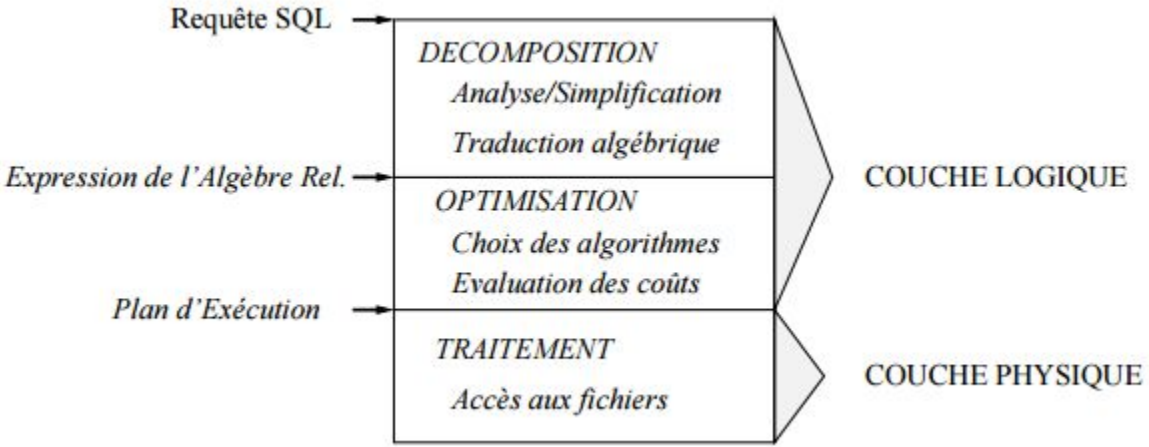
Préliminaire

1

Architecture d'une requête SQL

- **Décomposition:** requête SQL → expr. algèbre relationnelle
- **Optimisation:** expr. algèbre relationnelle → plan d'exécution
- **Évaluation (traitement)**: plan d'exécution→ résultats

⇒Ceci est exécuté dans un seul support physique pour une requête donnée.



MySQL vs Hadoop

- **Indexes**
- **Partitionnement**
- **Sharding:** partitionnement horizontale

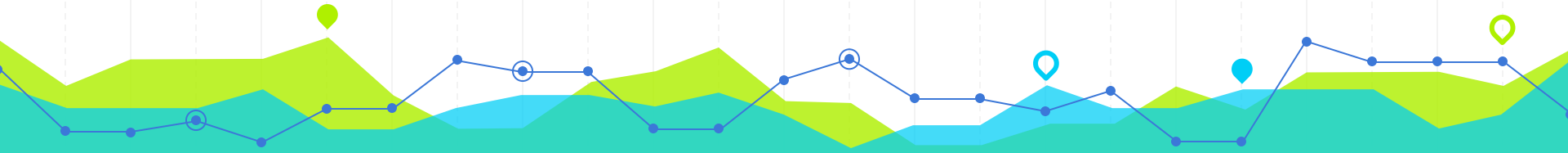
- **Full table scan**
- **Partitionnement**
- **MapReduce**

Dans hadoop, NO INDEX !!!

ETL vs ELT

- Extract data from external source
- Transform before loading
- Load data into MySQL

- Extract data from external source
- Load data into Hadoop
- Transform data / Analyze data / Visualize data / ...



Hadoop

Historique

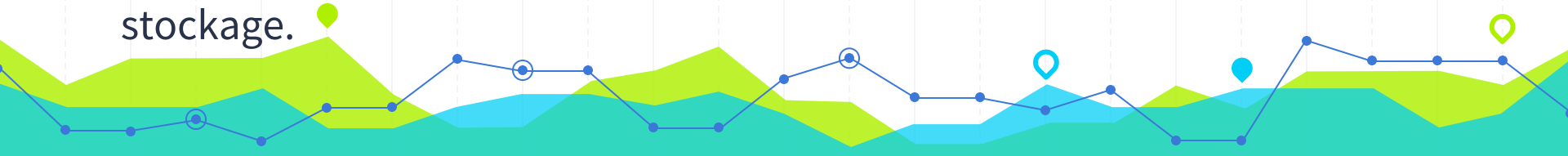
2

Hadoop ?

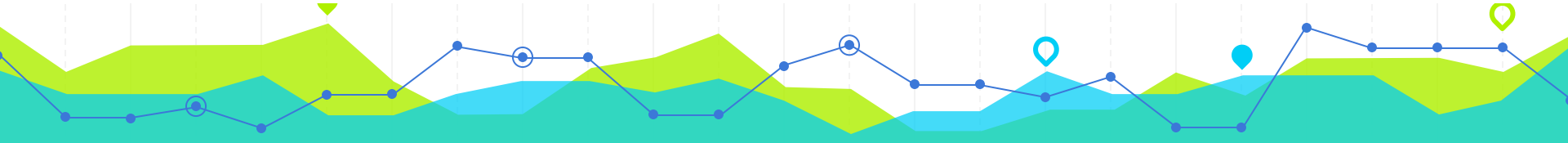
- Projet OpenSource par Apache: http://hadoop.apache.org/ ,
- Framework Développé en Java,
- Assure l'exécution de tâches MapReduce

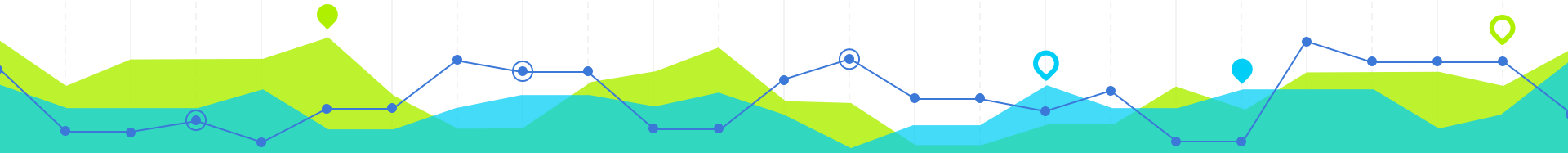
Historique:

- 2004: Développement de la 1er version d'Hadoop par **Doug Cutting**,
- 2006: **Doug Cutting (désormais chez Yahoo)** développe une 1er version exploitable de Apache Hadoop
- 2008: Hadoop utilisé chez Yahoo
- 2011: utilisé par de nombreuse entreprises et universités, **le cluster** Yahoo comporte **42000 machines** et des centaines de peta-octets d'espace de stockage.



Qui Utilise Hadoop





Version d'Hadoop

1.0 et 2.0

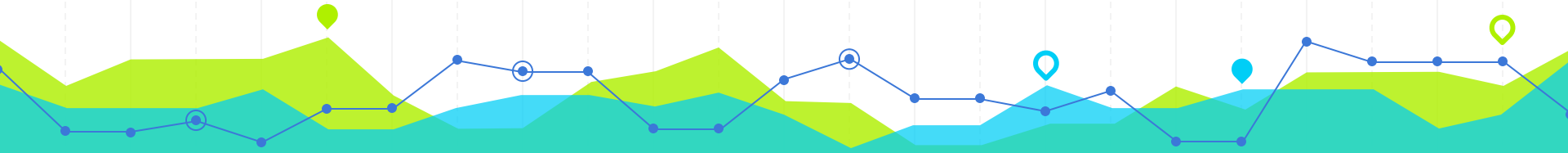
3

Version d'Hadoop 1.0 et 2.0

- Hadoop1.0 supporté juste des app MapReduce,
- Hadoop 2.0 vient avec le framework **YARN (Yet another Resource Navigator)** qui supporte des app Non- Mapreduce,

- Hadoop 1.0 a été développé pour supporter les distribution Unix, Hadoop 2.0 est disponible sur Windows,
- Hadoop 2.0 Rapidité d'accès aux données et aux cache de données,

- Hadoop 2.0 HDFS Snapshots: sauvegarder l'état du système, pour Syst recovery, Update, Protection, ...

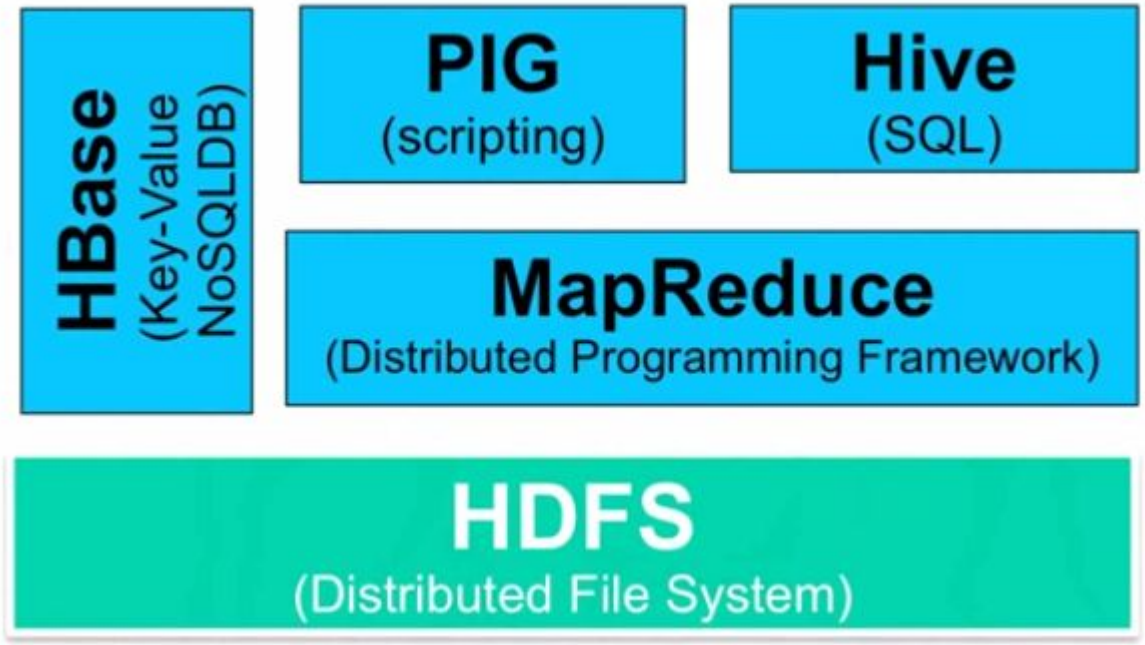


Architecture d’Hadoop

HDFS, MapReduce, Hive, PIG

4

Architecture d'Hadoop



HDFS

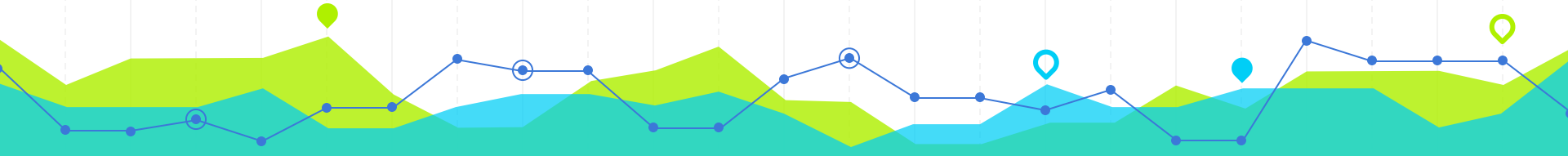
- HDFS pour: Hadoop Distributed File System
- Système de fichiers distribué associé à Hadoop. Stockage des données d'E/S,
- Les données sont séparées sur plusieurs machines
- Repose sur deux serveurs: **NameNode et DataNode**

NameNode: unique sur le cluster. Stocke les métadonnées (informations relative aux noms des fichiers) de manière centralisé.

DataNode: plusieurs par cluster. Stocke les fichiers en des blocs de données (64KB par défaut)

Différence entre HDFS et un système de fichiers classique

- HDFS n'est pas solidaire du noyau du système
- HDFS est un système distribué
- HDFS utilise des tailles de blocs largement supérieures à ceux des systèmes classiques
- HDFS fournit un système de réplication des blocs dont le nombre de réplications est configurable



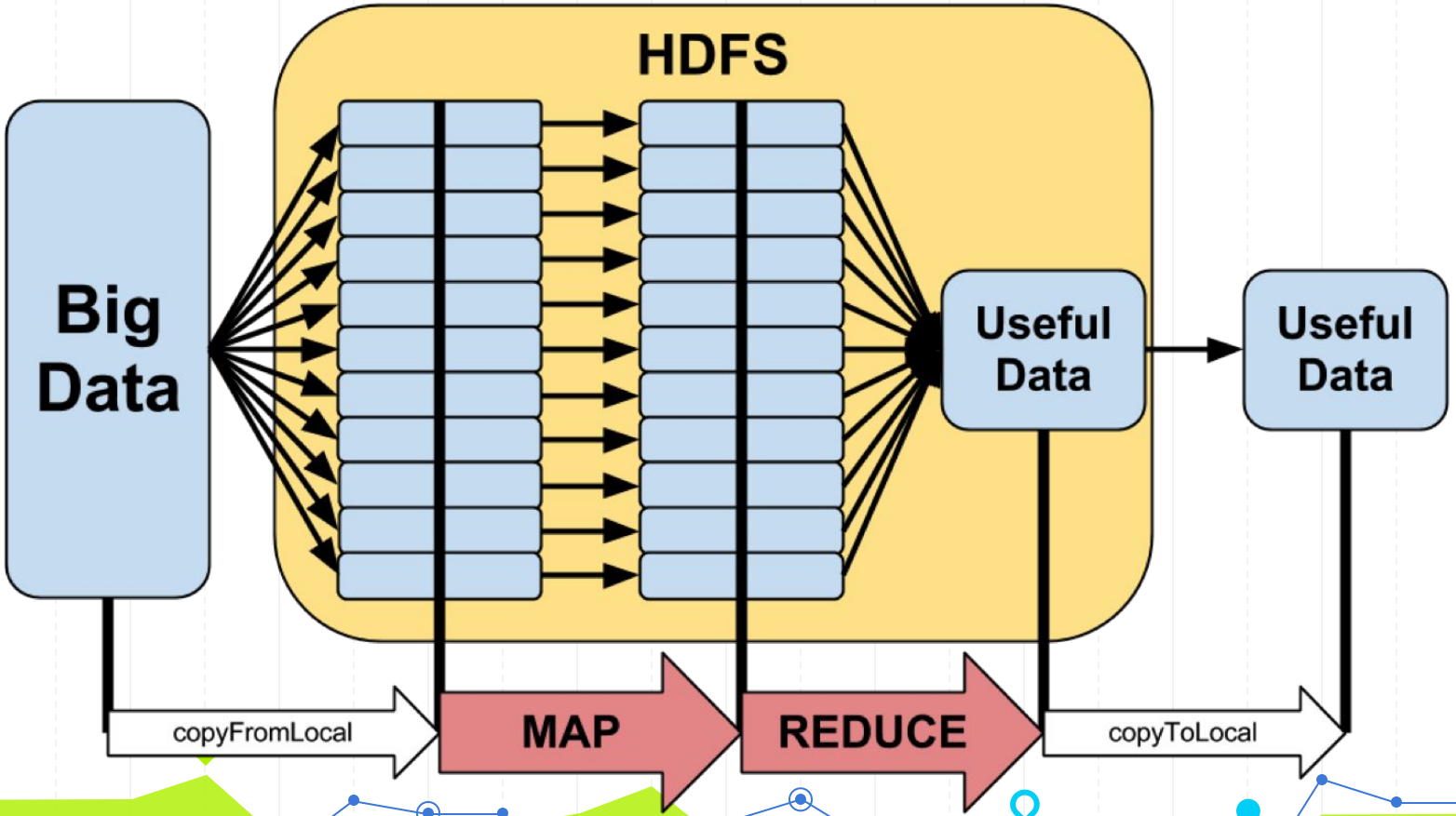
MapReduce

- Lire et écrire les fichiers.
- L’exécution des requêtes dans les **DataNode dans HDFS**

MapReduce

- Basé sur la stratégie algorithmique: ***“Diviser pour régner”***: découper un problème complexe en plusieurs problème simple.
- Pour un problème donné **P**, en le découpe en sous problème **Pi**, de telle manière à exécuter ces sous problème facilement et en parallèle(chaque machine exécute chaque tâche sur un ensemble de fragment d’entrée).

MapReduce


HDFS
Big Data
Useful Data
Useful Data
copyFromLocal
MAP
REDUCE
copyToLocal

MapReduce

On distingue 4 étapes dans l'exécution d'un programme MapReduce:

- **Le découpage** des données d'entrée **(split op)**,
- **Le mapping:** une fonction s'exécute sur chaque fragment qui génère une série de **(key/Value),**
- **Le regroupement** de tous les couples **(shuffle op)**,
- **L'opération reduce**, une fonctions qui associe des couple **(key/Value)** aux groupes associés.

Principe de MapReduce

On cherche à énumérer tous les mots distincts d'une source textuelle, avec pour chacun d'entre eux le nombre de fois qu'ils sont présents au sein de la source.

Imaginons cette exemple s’exécute sur une bibliothéque nationale

Exemple:

```
celui qui croyait au ciel
celui qui ny croyait pas
fou qui fait le delicat
fou qui songe a ses querelles
```

Principe de MapReduce

```
celui qui croyait au ciel
celui qui ny croyait pas
fou qui fait le delicat
fou qui songe a ses querelles
```

Opérations à effectuer:

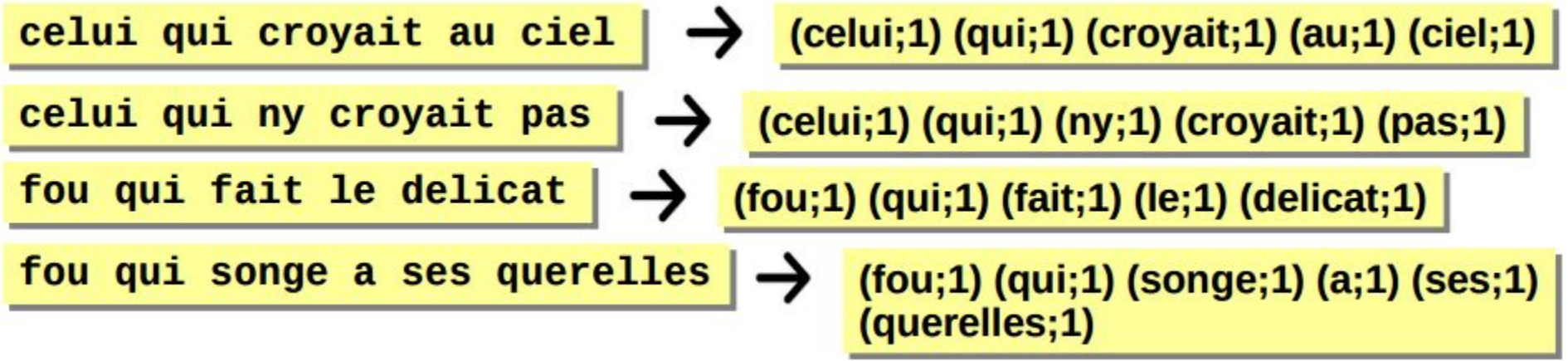
- **Map:** va produire une série de couples (Key;Value),
- **Shuffle:** les couples seront regroupés par clef distincte,
- **Reduce:** réduire les groupes triés par clef distincte

Principe de MapReduce

1er étape:

- Dans un premier temp, on génère des couples (clef;valeur), quelque soit le mot rencontré, la clef est 1

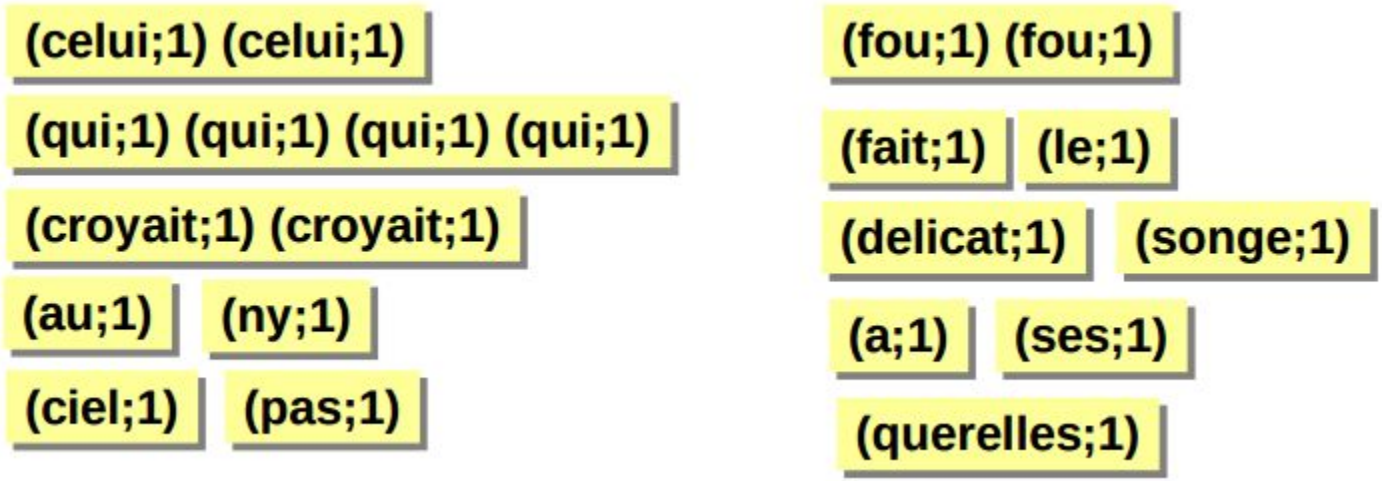
map(clé1, valeur1) → List(clé2, valeur2)



Principe de MapReduce

2éme étape:

- Exécution de shuffle, regroupement des couples (clé;valeur)

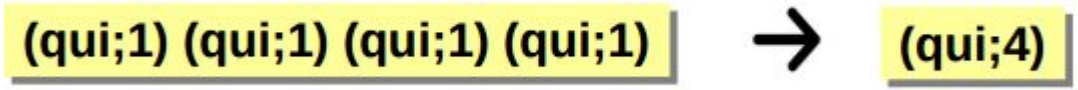


Principe de MapReduce

3éme étape:

- Chacun de ses groupes distincts sera passé en entrée de la fonction reduce
- Le rôle de reduce est simplement de réduire les couples (cle; valeur) reçue en entrée et les additionner pour avoir un unique couple (cle;valeur)

reduce(clé2, List(valeur2)) → List(valeur2)



Principe de MapReduce

Résultat:

- L’avantage du map/reduce, permet de

Compter le nombre d’occurence d’un fichier Volumineux en quelque seconde.

```
qui: 4
celui: 2
croyait: 2
fou: 2
au: 1
ciel: 1
ny: 1
pas: 1
fait: 1
[…]
```

Hive

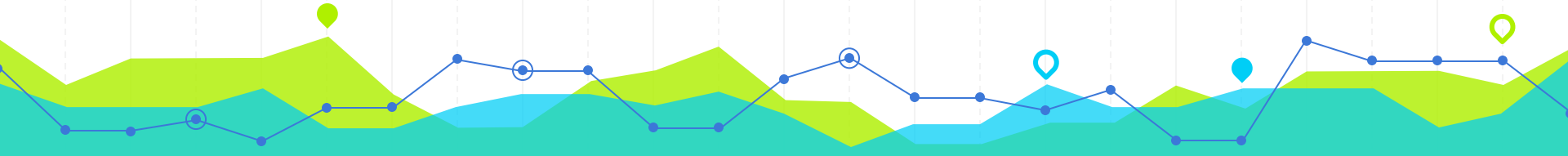
- Crée une base de données relationnelle dans le système de fichiers HDFS

Pig

- Language script, traduit en MapReduce,

Hbase

- Enregistrement de données Key/Value - NoSQL.



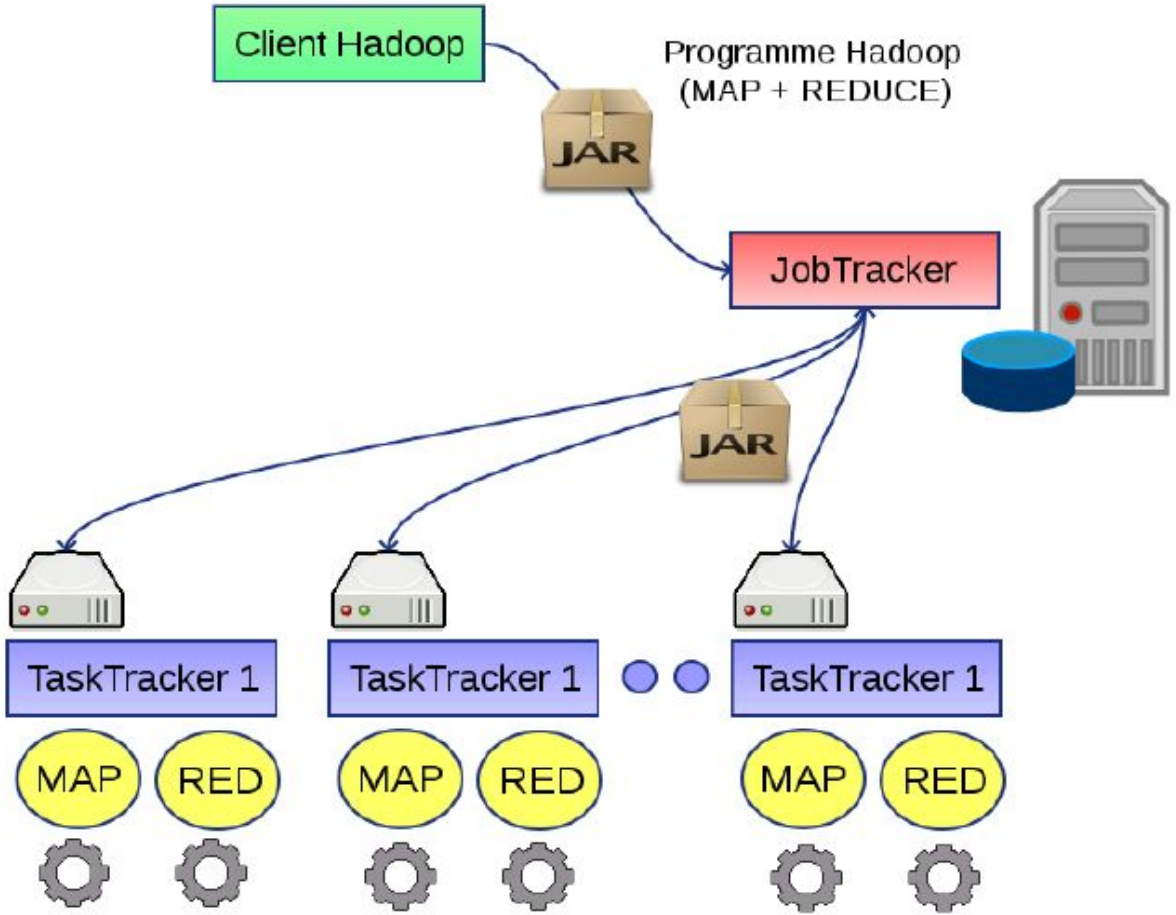
Un cluster hadoop repose sur deux serveurs:

JobTracker

- Unique sur le cluster. Reçoit des tâches Map/Reduce à exécuter (sous forme d'un fichier .jar), organise leurs exécution sur le cluster

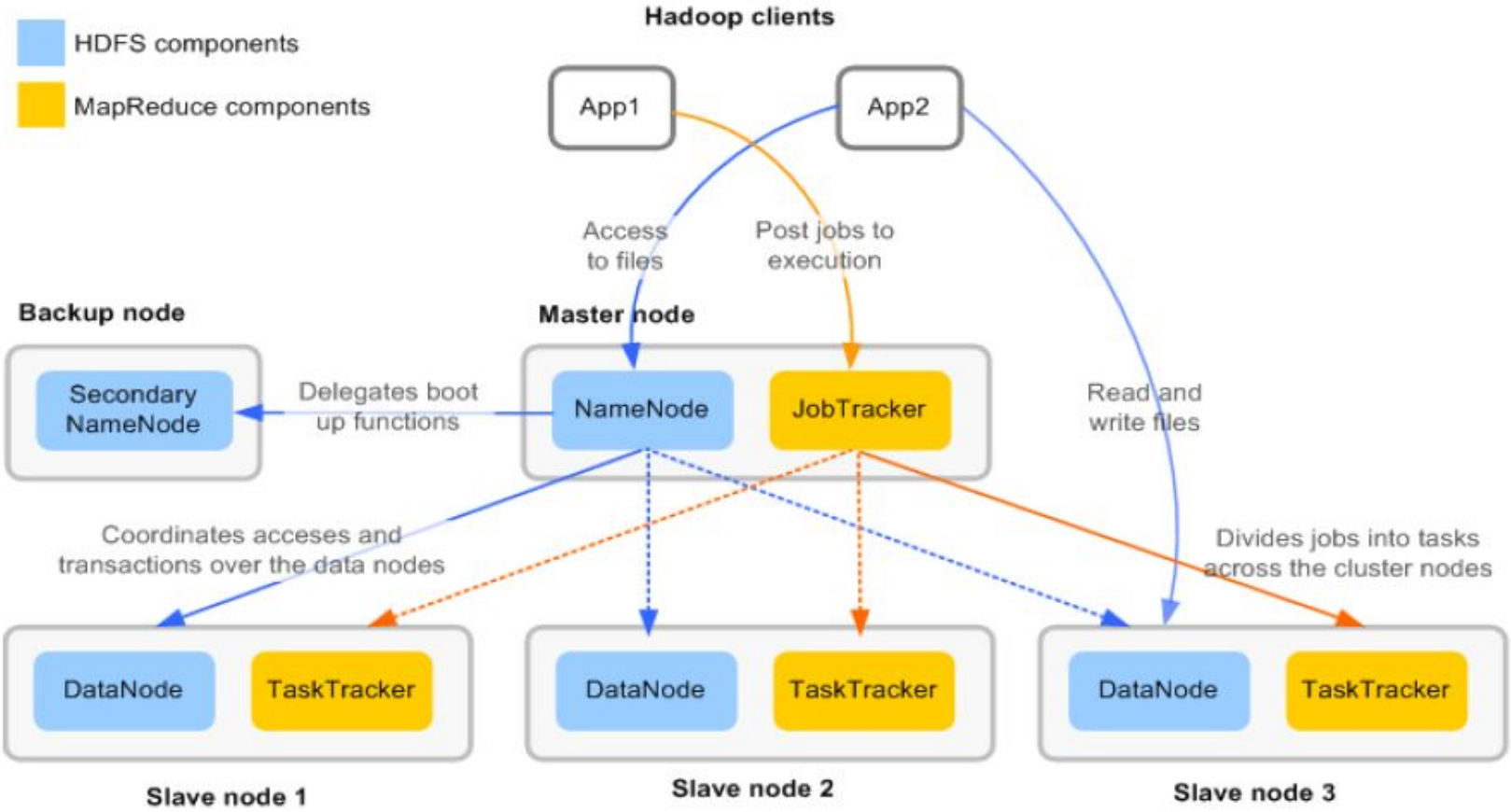
TaskTracker

- Plusieurs par cluster. Exécute le job Map/Reduce (sous la forme de tâche map et reduce)
- Chaque TaskTracker constitue une unité de calcul du cluster

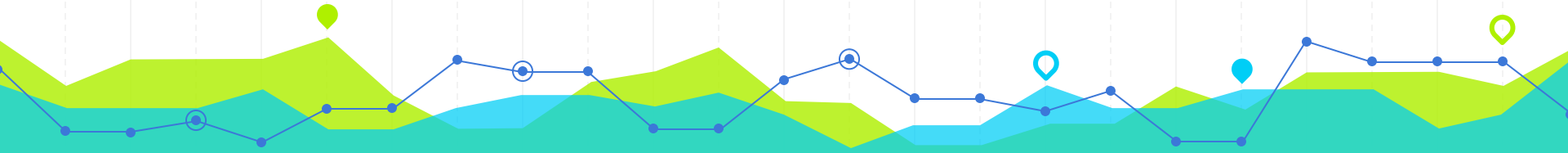

Client Hadoop
Programme Hadoop
(MAP + REDUCE)
JAR
JobTracker
JAR
TaskTracker 1
TaskTracker 1
TaskTracker 1
MAP
RED
MAP
RED
MAP
RED

Exécution d’une tâche

- Tous les TaskTracker signalent leurs statuts continuellement via le paquet “heartbeat”
- En cas de défaillance d’un TaskTracker (heartbeat manquant ou tâche échouée), le JobTracker avise en conséquence: redistribution de la tâche sur un autre noeud, etc …
- Hadoop permet d’afficher des stats via la console hadoop



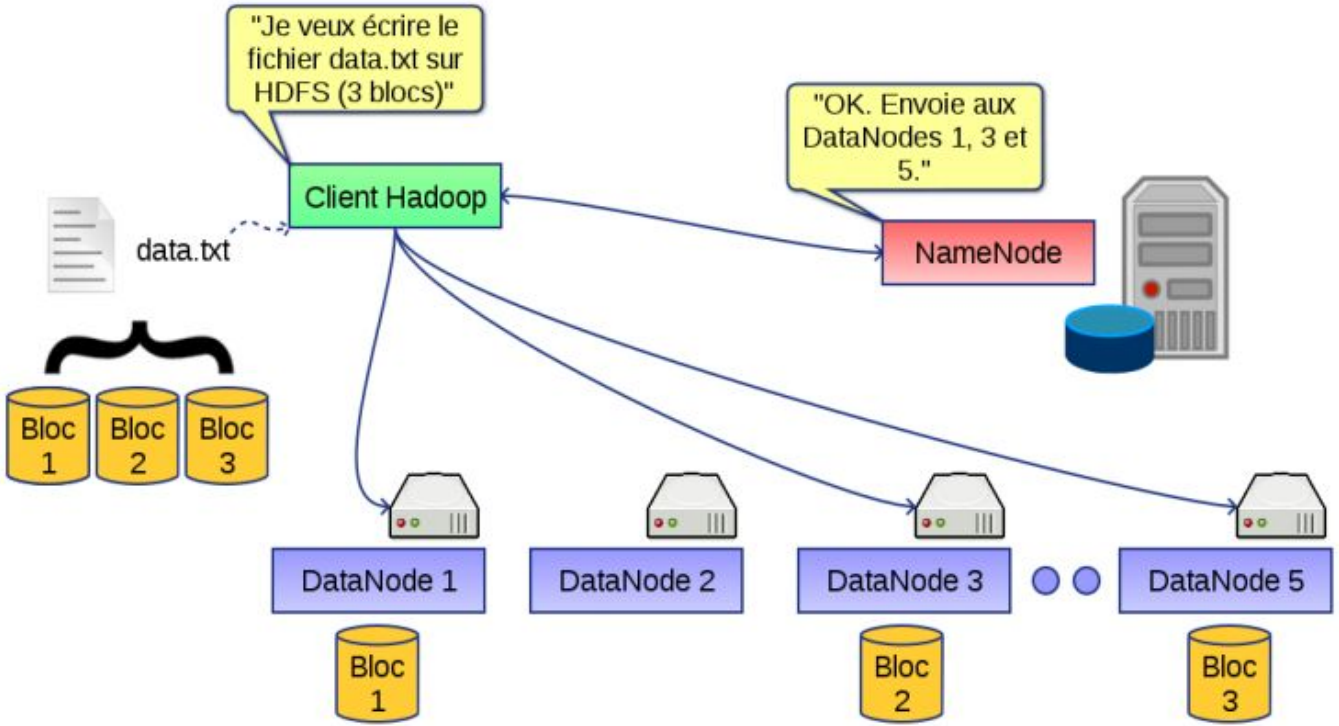
Source: documentation Hadoop



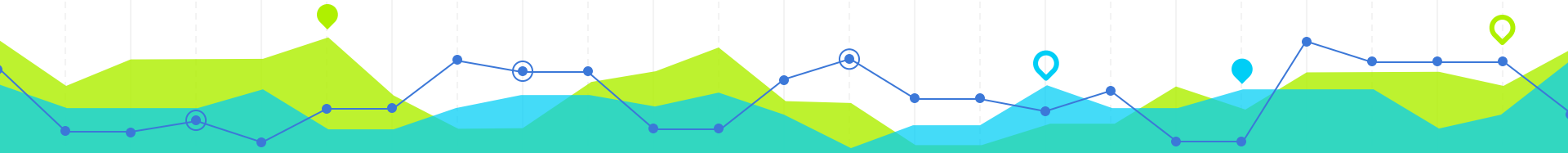
Ecriture d’un fichier

5

Ecriture HDFS



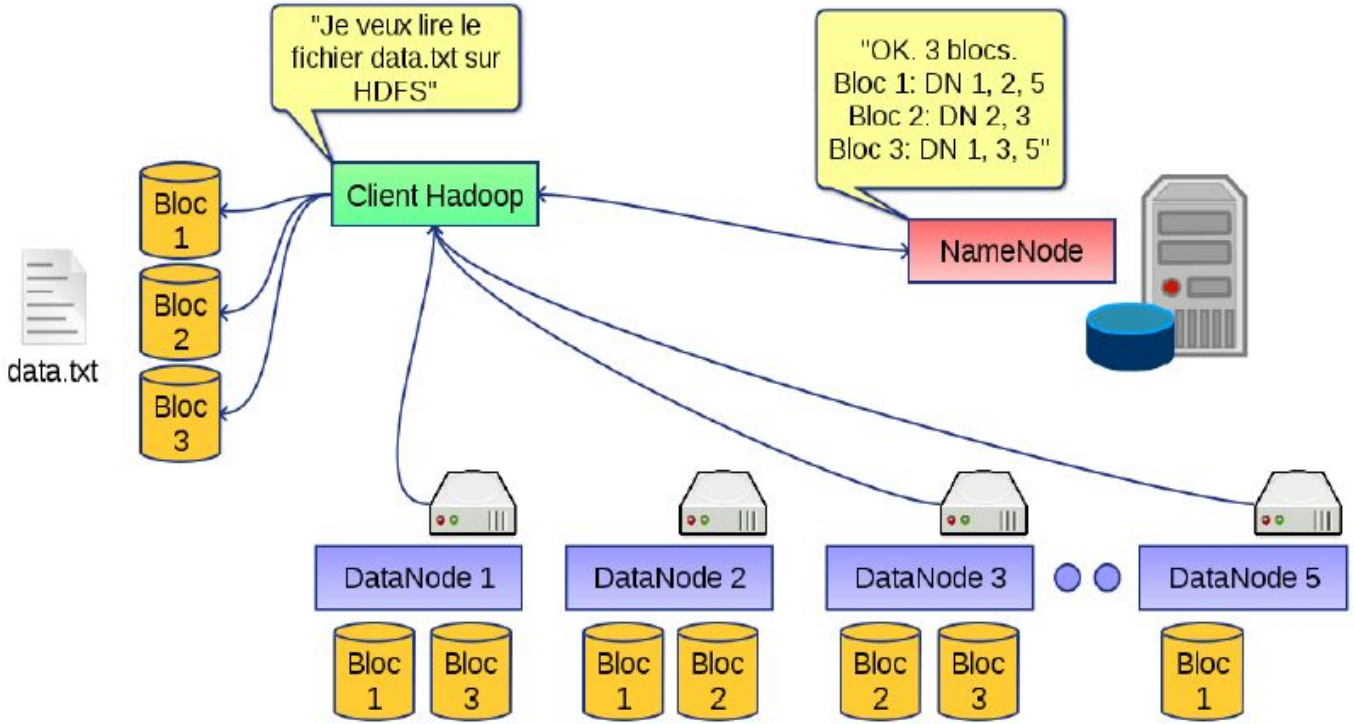
- Le client indique au NameNode qu'il souhaite écrire un bloc.
- Celui-ci lui indique le DataNode à contacter.
- Le client envoie le bloc au Datanode.
- Les DataNodes répliquent le bloc entre eux.
- Le cycle se répète pour le bloc suivant.



Lecture d’un fichier

6

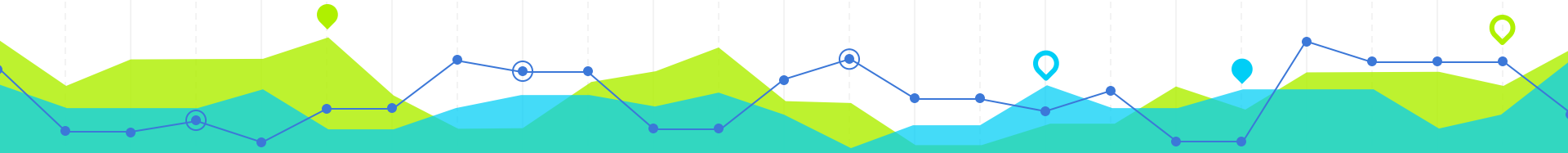
Lecture HDFS



- Le client indique au NameNode qu'il souhaite lire un fichier.
- Celui-ci lui indique sa taille et les différents DataNode contenant les N blocs.
- Le client récupère chacun des blocs à un des DataNodes.
- Si un DataNode est indisponible, le client le demande à un autre.

Partie Pratique

“Nombres d’occurrences des mots”



Conclusion

7

- La programmation MapReduce est fortement utilisé auprès des grandes entreprise comme google, la preuve c'est que c'est en java, facile à utiliser même pour un simple programmeur
- Il y'a d'autre projet similaire à Hadoop, comme: *Apache Spark, Apache HBase, Apache Sqoop, …* pour Spark, utilise son propre "engine" (à part MapReduce) et il est **100x** plus rapide que Hadoop !

- Hadoop est un framework libre-facile, se familiariser avec nécessite de la pratique et une occasion pour s'ouvrir sur d'autre technologie plus performant.
- Nom d'Hadoop s'agit d'un éléphant en peluche du fils de l'auteur **Doug Cuttin**

Merci pour votre attention
Des questions ?!