Chapitre 2 : Hadoop

Aiway

Your way to AI

Kesako ?

Aiway
Your way to AI

- Hadoop est un framework libre et open source écrit en Java destiné à faciliter la création d'applications distribuées

- Permet aux applications de travailler avec des milliers de nœuds et des pétaoctets de données

Les grandes parties de Hadoop

Aiway
Your way to AI

- Le projet Hadoop consiste en deux grandes parties
  - Un système de stockage : ***HDFS*** (**H**adoop **D**istributed **F**ile **S**ystem)
  - Un système de traitement de données : ***MapReduce***
- Principe :
  - Diviser les données et les sauvegarder sur une collection de machines, appelée cluster
  - Traiter les données directement là où elles sont stockées, plutôt que de les copier à partir d'un serveur distribué
- Il est possible d'ajouter des machines au cluster au fur et à mesure → *scalabilité verticale*

Hadoop - Vocabulaire

Aiway
Your way to AI

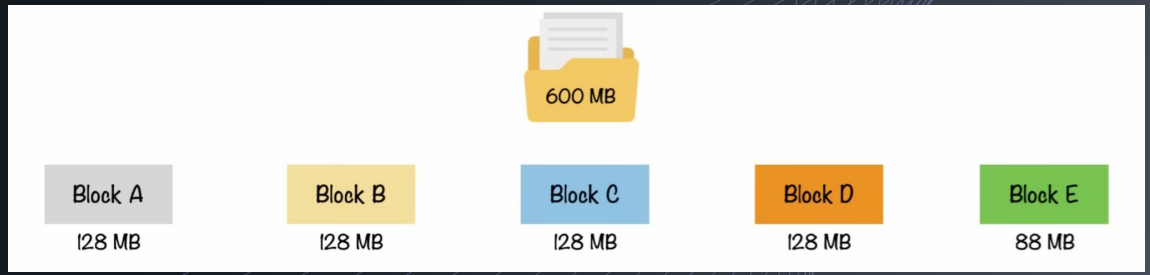
- Un cluster est un ensemble de machines physiques ou virtuelles connectées entre elles et interagissant en vue d'accomplir des tâches définies
- Chaque machine est un noeud du cluster
- Un cluster Hadoop est composé de deux types de noeuds
  - Un NameNode, le noeud maître, un noeud unique qui dispose de l'arborescence du FS et des metadata, dont le rôle est de dispatcher les données, indiquer leur localisation et superviser les traitements parallèles
  - Des DataNode, les noeuds esclaves, qui stockent les données fragmentées et effectuent les traitements adéquats dessus

HDFS

Aiway
Your way to AI

- HDFS est un système de stockage distribué permettant de sauvegarder des données sur un ensemble de machines
  - Les données sont fragmentées et stockées sous forme de blocs
  - Chaque bloc fait une taille maximale de 128 MB
  - Les blocs sont répartis sur l'ensemble des machines du cluster

- Exemple :



HDFS - Replication

Aiway
Your way to AI

- Que se passe t-il lorsqu'une machine devient indisponible ? Les données sont-elles perdues ?
  - Hadoop offre une haute tolérance aux pannes
  - Mise en place de répliques
    - Chaque bloc est ajouté non pas sur un noeud, mais sur N noeuds afin d'optimiser l'accessibilité aux données
    - Le facteur de réplication défini le nombre N de copies créées pour un bloc (3 répliques par défaut)
    - Les répliques permettent de garantir une haute accessibilité aux données malgré les pannes éventuelles des machines
    - Les noeuds destination sont choisi aléatoirement et lorsqu'un des noeud tombe, le NameNode le détecte et rajoute de nouvelles répliques → N répliques existent à tout moment

HDFS - Défaillance du NameNode

Aiway
Your way to AI

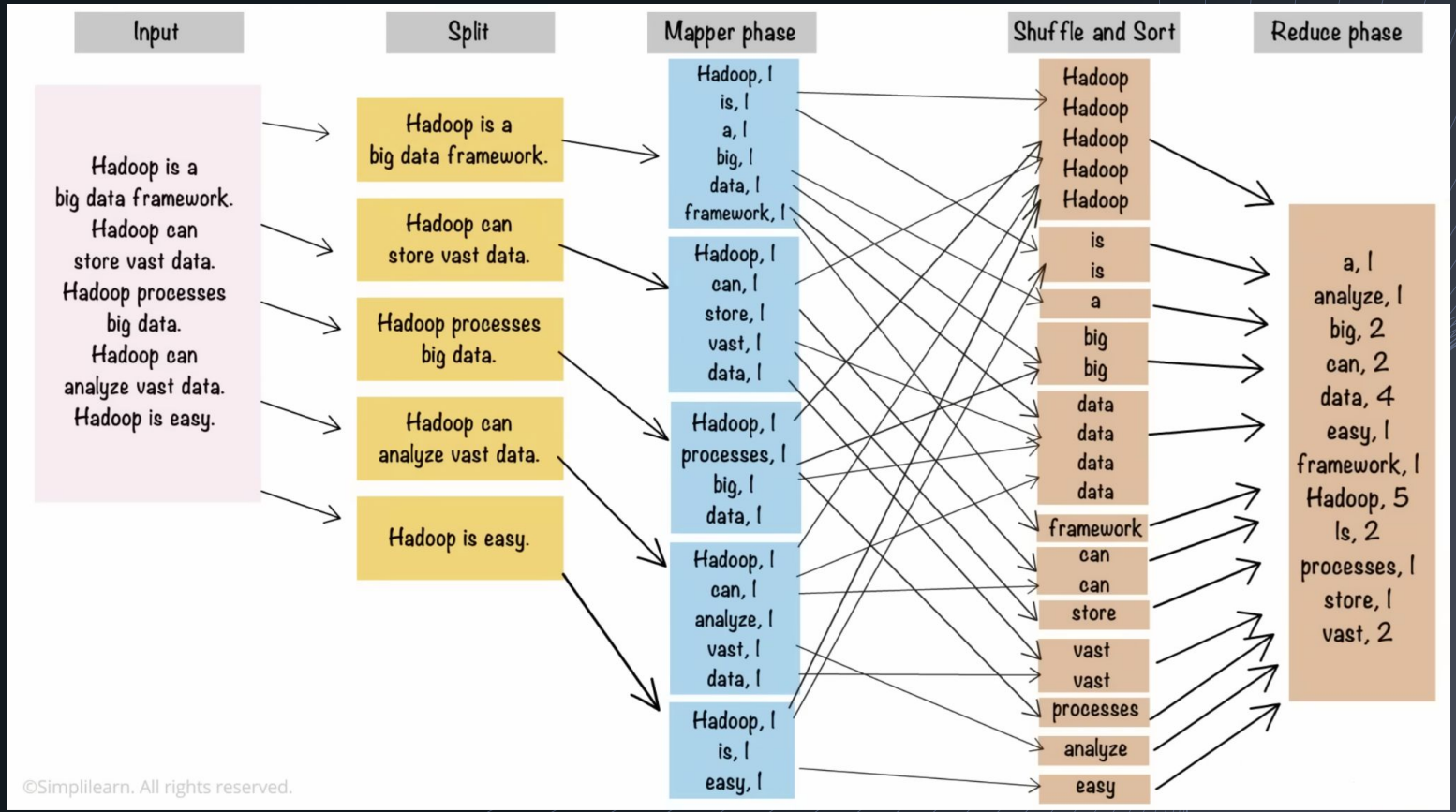
- Les DataNodes sont remplaçables grâce à la réplication
- Que se passe t-il en cas de panne du NameNode
  - Si panne réseau, les données sont temporairement indispo
  - Si panne disque du NN, perte définitive des données
- Pour éviter la perte de données, le NN est dupliqué sur le système de fichiers du réseau
  - On parle de standby NameNode, qui est inactif jusqu'à ce que le NN principal ne devienne indisponible, au quel cas il le replace

MapReduce

Aiway
Your way to AI

- Procédé de traitement parallèle des données
  - Les données à traiter sont d'abord découpées en un certain nombre de fragments et réparties parmi les DataNode.
    Si les données sont déjà réparties sur les noeuds, chacun d'eux traite les données dont il dispose
  - Le *map* est la phase de traitement où pour chaque élément une étiquette <clé, valeur> est crée
  - Le *shuffle and sort* est la phase lors de laquelle les étiquettes ayant la même clé sont rassemblés ensemble
  - Le *reduce* est l'étape au cours de laquelle toutes les étiquettes sont transmises aux noeuds en charge de centraliser les résultats et qui vont les réduire, appliquer une opération dessus

MapReduce - Exemple



MapReduce - Résultat final

Aiway
Your way to AI

- Si un seul reducer est utilisé, la sortie de ce dernier est le résultat final

- Si plusieurs reducers sont utilisés, une dernière opération de tri est nécessaire afin d'ordonner les résultats partiels en sortie

Démons de MapReduce

Aiway
Your way to AI

- JobTracker
  - Divise le travail sur les Mappers et Reducers, s'exécutant sur les différents noeuds
- TaskTracker
  - S’exécute sur chacun des noeuds pour exécuter les vraies tâches de MapReduce
  - Choisit en général de traiter un bloc sur la même machine que lui
  - S’il est déjà occupé, la tâche revient) un autre tracker, qui utilisera le réseau (phénomène rare)

YARN

Aiway
Your way to AI

- Pour permettre la réalisation de tâches en parallèle sur plusieurs machine, il est nécessaire d'avoir une gestion des ressources à assigner à une tâche donnée
- ***YARN*** - **Y**et **A**nother **R**esources **N**egociator
  - Gestionnaire de ressources composé de
    - *Resource manager* : attribue les ressources
    - *Node manager* : gère les noeuds et surveille l'utilisation des ressources au sein d'un noeud
    - *Containers* : contiennent les ressources physiques disponibles
    - *Application master* : gère le processus qui requiert les ressources
- Permet une gestion efficace des ressources sur les noeuds

Hadoop Design Patterns

Aiway
Your way to AI

- Les design patterns représentent les types de traitements les plus utilisés avec Hadoop
- Trois catégories principales :
  - Patrons de Filtrage (Filtering Patterns)
    - Echantillonnage de données
    - Listes des top-n
  - Patrons de Récapitulation (Summarization Patterns)
    - Comptage des enregistrements
    - Trouver les min et max
    - Statistiques
    - Indexes
  - Patrons Structurels
    - Combinaison de données relationnelles

Patrons de Filtrage

Aiway
Your way to AI

- Ne modifie pas les données
- Trient, parmi les données présentes, lesquelles garder et lesquelles enlever
- On peut obtenir :
  - Des filtres simples : définition d'une fonction indiquant le critère de filtrage
  - L'échantillonnage (sampling) : création d'un petit ensemble d'enregistrements à partir d'un grand ensemble, en retenant des échantillons
  - L'échantillonnage aléatoire : retenir un échantillon représentatif des données initiales
  - Les listes Top-N

Patrons de Filtrage - Exemple 1

Aiway
Your way to AI

- Exemple de Filtrage Simple:
  - **Cas d'étude** : fichier contenant tous les posts des utilisateurs sur un forum
  - **Filtre** : Retenir les posts les plus courts, contenant une seule phrase
    - ✓ Une phrase est un post qui ne contient aucune ponctuation de la forme: .!?, ou alors une seule à la fin.

```
def mapper():
    reader = csv.reader(sys.stdin, delimiter='\t')
    writer = csv.writer(sys.stdout, delimiter='\t', quotechar='"'
                                                , quoting=csv.QUOTE_ALL)
    for line in reader:

        for i in line:
            #print('-',i)
            if len(i) == 0:
                continue
            if "!" in i[:-1] :
                continue
            if "." in i[:-1]:
                continue
            if "?" in i[:-1]:
                continue
            else:
                writer.writerow(line)
```

Patrons de Filtrage - Exemple 2

Aiway
Your way to AI

- Exemple: Top 10
  - Trouver parmi les différents posts des forums, les 10 posts les plus longs
  - Dans une Base de données Relationnelle:
    - ✓ Trier les données
    - ✓ Extraire les 10 premières
  - Map-Reduce (de la même manière qu'une sélection sportive)
    - ✓ Chaque Mapper génère une liste Top-10
    - ✓ Le Reducer trouve les Top 10 globaux

```
def mapper():
    a = []
    b = []
    reader = csv.reader(sys.stdin, delimiter='\t')
    writer = csv.writer(sys.stdout, delimiter='\t', quotechar='"',
                                                    quoting=csv.QUOTE_ALL)
    for line in reader:

       for i in line:
            if len(i) == 0 :
                continue
            else:
                a.append(line)

    a.sort(key=lambda a: (int)(a[4]), reverse=True)

    for i in range(0,10):
        b.append(a[i])
    b.sort(key=lambda b: (int)(b[4]))

    for b1 in b:
        writer.writerow(b1)
```

Patrons de Récapitulation

Aiway
Your way to AI

- Permettent de vous donner une idée de haut niveau de vos données
- On distingue deux types :
  - Index : Tels que les index à la fin d'un livre, ou les index par google pour représenter les pages web
  - Récapitulation (ou résumé) numérique :
    - Chercher des chiffres, des comptes
    - Min et Max
    - Premier et dernier
    - Moyenne
    - Etc

Patrons de Récapitulation - Index

Aiway
Your way to AI

- Les index permettent une recherche plus rapide
- Dans un livre : pour chaque mot donné, indiquer les différentes pages où se trouve ce mot
- Dans un site web : on trouve des liens vers des pages web à partir d'un ensemble de mots clés

Exemple (1/2)

Aiway
Your way to AI

- **Exemple** : Indexation des mots dans les posts d'un forum
- Mapper ➔

```
import sys
import csv
import re

firstLine = 1

reader = csv.reader(sys.stdin, delimiter='\t')
writer = csv.writer(sys.stdout, delimiter='\t', quotechar='"', quoting=csv.QUOTE_ALL
)

for line in reader:
        if firstLine == 1:
                #Si on se trouve dans la premiere ligne (celle des titres), sauter c
ette ligne
                firstLine = 0
                continue
        body = line[4]
        node = line[0]
        words = re.findall(r"[\w']+|[.,!?;]",body)
        for word in words:
                if word not in ('.','!',',','?','#','$','[',']','/','\"','<','>','='
,'-',':',';','(',')'):
                        print "{0}\t{1}".format(word,node)
_
```

Exemple (2/2)

Aiway
Your way to AI

- **Exemple** : Indexation des mots dans les posts d'un forum

- Reducer ➔

```
import sys

nbTotal = 0
oldWord = None
listNodes = []


for line in sys.stdin:
    data_mapped = line.strip().split("\t")
    if len(data_mapped) != 2:
        continue

    thisWord, thisNode = data_mapped

    if oldWord and oldWord.lower() != thisWord.lower():
        listNodes.sort(key=lambda listNodes:(int)(listNodes))
        print oldWord, "\t", nbTotal, "\t", listNodes
        oldWord = thisWord.lower();
        nbTotal = 0
        listNodes = []

    oldWord = thisWord.lower()
    nbTotal = nbTotal + 1
    if thisNode not in listNodes:
        listNodes.append(thisNode)


if oldWord != None:
     listNodes.sort(key=lambda listNodes:(int)(listNodes))
```

Récapitulations Numériques

Aiway
Your way to AI

- Peuvent être :
  - Le nombre de mots, enregistrements…
    - Souvent la clé est l'objet à compter et la valeur est égale à 1
  - Min et Max / Dernier et premier
  - La moyenne
  - La médiane
  - Écart-type
  - Etc
- Exemple de question :
  - Y a t-il une relation entre le jour de la semaine et la somme dépensée par le client ?
  - <jour_de_la_semaine, somme_dépensée>

Mélangeur

Aiway
Your way to AI

- Possibilité d'utiliser un mélangeur (*combiner*) entre les mappers et les reducers

- Permettent de réaliser la réduction en local dans chacun des DataNodes AVANT de faire appel au reducer

Mélangeur - Exemple

Aiway
Your way to AI

- Exemple : calcul de la moyenne des ventes par jour :
  - Sans combiner
    - Les mappers parcourent les données et affichent l'étiquette
    - Pour chaque jour, le reducer conserve une somme et un compteur
    - Il divise à la fin la somme par le compteur
  - Avec combine
    - Chaque noeud réalise une réduction où les moyennes locales sont calculées
    - Le reducer final regroupe ces moyennes et synthétise la moyenne finale
    - Nombre d'enregistrements envoyés au réducteur significativement réduit
    - Temps nécessaire pour la réduction diminué

Patrons Structurels

Aiway
Your way to AI

- Utilisés quand les données proviennent d'une base de données structurée
- Plusieurs tables, donc plusieurs sources de données, liées par une clé étrangère
- Les données des différentes tables sont exportées sous forme de fichiers délimités

Patrons Structurels

Aiway
Your way to AI

- Le mapper aura comme tâche de
  - Parcourir l'ensemble des fichiers correspondant aux tables
  - Extraire de chacune des entrées les données nécessaires, en utilisant comme clé la clé étrangère joignant les deux tables
  - Afficher les données extraites des différentes tables, chacune sur une ligne, en créant un champs supplémentaire indiquant le source des données
- Le reducer
  - Fera l'opération de jonction entre les deux sources, en testant la provenance avec le champs supplémentaire

Atouts de Hadoop

Aiway
Your way to AI

- La gestion des défaillances : que ce soit au niveau du stockage ou traitement, les noeuds responsables de ces opérations durant le processus de Hadoop sont automatiquement gérés en cas de défaillance. Nous avons donc une forte tolérance aux pannes
- La sécurité et persistance des données : grâce au concept “Rack Awareness”, il n’y a plus de soucis de perte de données
- La garantie d’une montée en charge maximale
- La complexité réduite des traitement de données massives
- Le coût réduit : Hadoop est open source, les données même massives sont traitées efficacement et à faible coût

Inconvénients de Hadoop

Aiway
Your way to AI

- Difficulté d'intégration avec d'autres systèmes informatiques : le transfert des données d'une structure Hadoop vers des bases de données traditionnelles n'est pas trivial
- Administration complexe : Hadoop utilise son propre langage. L'entreprise doit donc développer une expertise spécifique Hadoop ou faire appel à des prestataires extérieurs
- Traitement de données différé et temps de latence important : Hadoop n'est pas fait pour l'analyse temps réel des données