BIG DATA : ENJEUX, STOCKAGE ET EXTRACTION

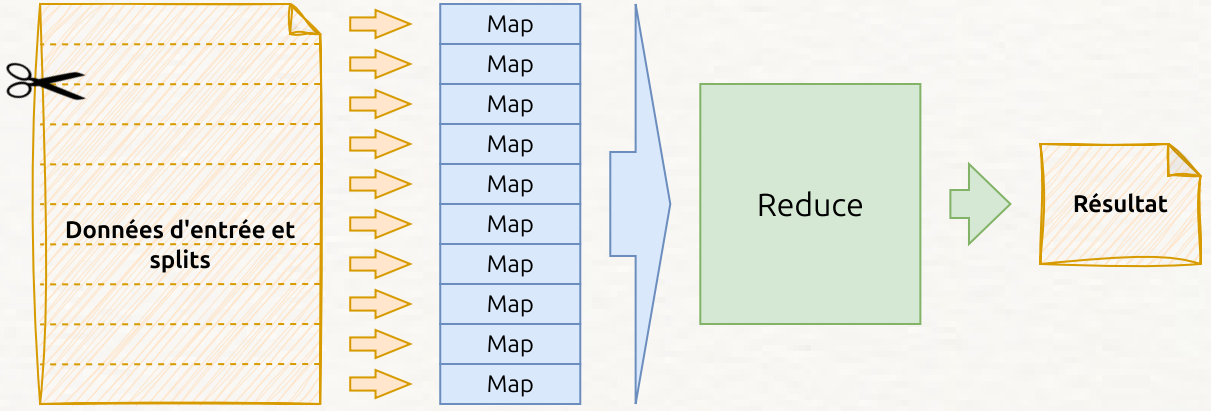
TP 2 : MAPREDUCE

Objectif : Comprendre l'algorithme MapReduce, brique fondatrice des traitements Big Data.

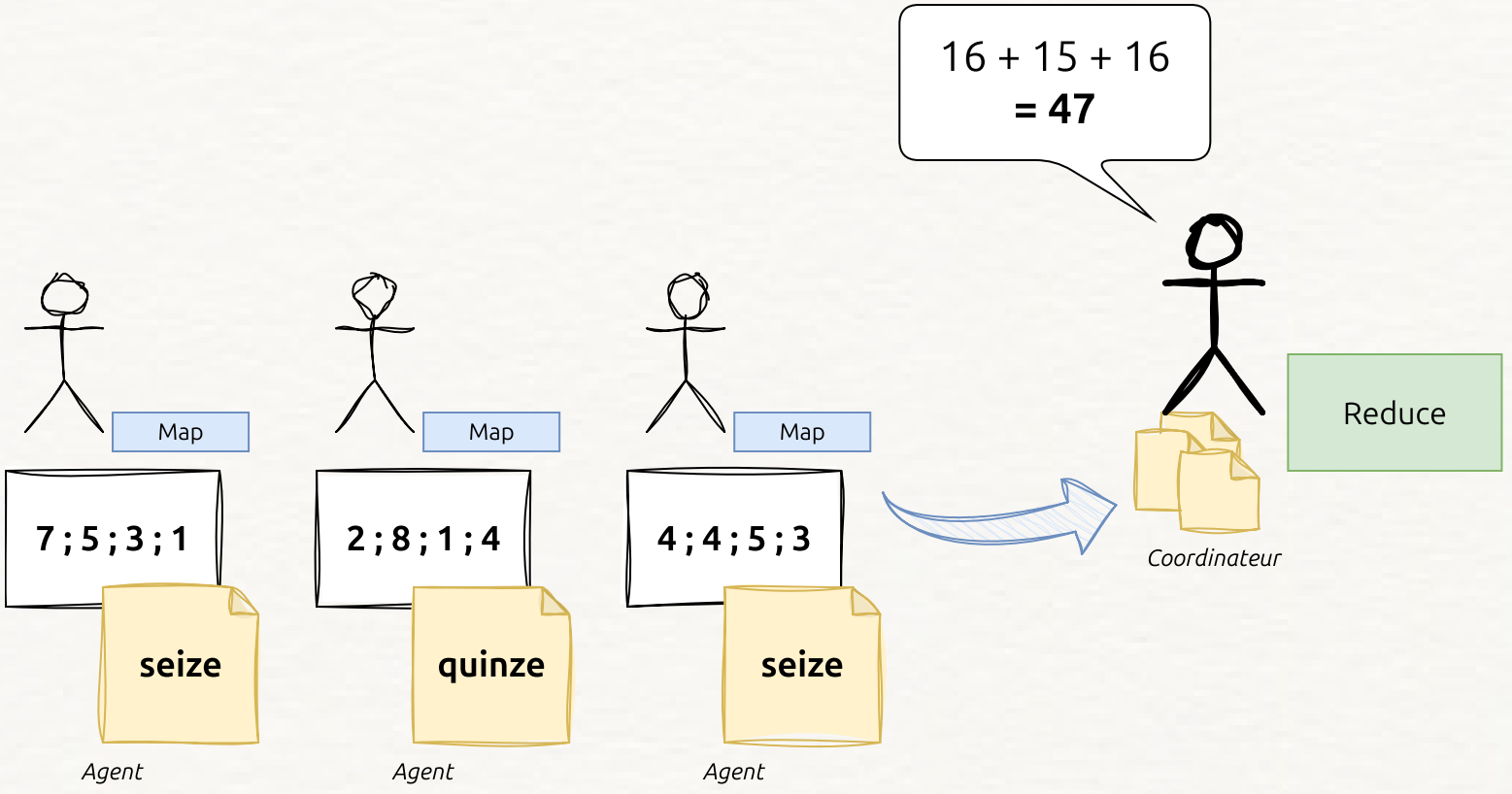
PRÉSENTATION DE MAPREDUCE

Principe : MapReduce est un algorithme de traitement parallèle sur des données volumineuses. C'est aussi un cadre de programmation assez strict, raison pour laquelle on ne le programme pas directement.

Les données à traiter sont découpées en *splits*. Les splits sont traités en parallèle lors d'une phase de **Map**. Une phase de **Reduce** (en général non parallèle) combine les résultats intermédiaires pour produire le résultat.



MAPREDUCE EN MODE MANUEL



QUE S'EST-IL PASSÉ ?

OÙ AVONS-NOUS PERDU DU TEMPS ? (1)

Dans la distribution initiale des fiches, avant le début des calculs

Un traitement distribué a un coût incompressible lié au démarrage des agents d'exécution par le coordinateur.

QUE S'EST-IL PASSÉ ?

OÙ AVONS-NOUS PERDU DU TEMPS ? (2)

Dans la transmission des Post-It vers le coordinateur

L'échange des informations a aussi un coût, proportionnel à son volume.

En pratique, il s'agit principalement des entrées/sorties : réseau, disque, ...

QUE S'EST-IL PASSÉ ?

OÙ AVONS-NOUS PERDU DU TEMPS ? (3)

Dans l'écriture et la lecture des nombres en toutes lettres sur les Post-It

La transmission d'informations nécessite une mise en forme : c'est la **sérialisation/désérialisation**.

Elle a un coût proportionnel au volume d'informations, et à la complexité de la représentation.

QUE SE SERAIT-IL PASSÉ SI...

... nous avions été 2 fois plus nombreux pour traiter le même volume (2 nombres par fiche au lieu de 4) ?

⇒ Plus de cerveaux n'auraient pas suffi à compenser le temps de démarrage supérieur et les échanges plus nombreux. Le temps de calcul efficace est trop faible pour que ça en vaille la peine.

⇒ On aurait même augmenté les échanges et donc perdu plus de temps.

QUE SE SERAIT-IL PASSÉ SI...

... il y avait eu 400 nombres par fiche à additionner ?

⇒ C'est le calcul qui aurait été le goulet d'étranglement, à cause du volume à traiter par chacun(e). Les échanges ne seraient plus significatifs.

⇒ Le temps de calcul total serait devenu à peu près proportionnel au volume de données à traiter.

⇒ Attention, pour des très gros calculs on se remet à perdre du temps, pour des points de reprise par exemple (échanges avec un système de stockage).

QUE SE SERAIT-IL PASSÉ SI...

... l'un d'entre nous avait dû répondre à un appel urgent en plein calcul ?

⇒ Une partie du traitement aurait été mise en pause, ralentissant l'arrivée du résultat final.

⇒ Le coordinateur peut décider de donner la tâche interrompue à un autre agent (rejeu), si le premier ne revient pas assez vite.

⇒ La probabilité d'un tel événement augmente avec le nombre d'agents.

LE TP DE PROGRAMMATION

Récupérer le notebook (`TP2.ipynb`) et les données (`data.zip`) du TP sur https://github.com/tvmbl/BUT-SD-R6.01/tree/main/tp-2.

Dézipper `data.zip` dans le même répertoire que le notebook.

Ouvrir le notebook dans Jupyter et suivre les instructions.