Partie 1 : Pour aller plus loin

Quels sont les éléments à considérer pour faire évoluer votre code afin qu'il puisse gérer de grosses volumétries de données (fichiers de plusieurs To ou millions de fichiers par exemple) ?

Nous avons mis en place un pipeline de données sur Python utilisant Pandas. Ce pipeline ingère quelques fichiers de quelques lignes chacun. Dans l'optique où le besoin de traitement changerait et la volumétrie de données venait à devenir importante il faudrait prendre en compte plusieurs éléments

I. L'ingestion des données

Une augmentation de la volumétrie des données pose logiquement la question de la façon dont on va ingérer nos données. Cela peut se faire en streaming, soit en temps réel, ou alors sous forme de lots de données qu'on appelle des batch.

II. Le temps de traitement

Dans le cas actuel, nous faisions le traitement des données sur une machine locale. Cependant avec une augmentation de la volumétrie des données, on engendre une augmentation des ressources utilisées (RAM, GPU, CPU) qui ne serait pas supporté par une machine locale. En effet cela est dû au fait que l'on ne peut pas charger un fichier de plusieurs To sur python en local. De plus, même si on arrivait à séparer le fichier en plusieurs lots avant de le traiter, les temps de traitement seraient tout de même conséquents.

III. Le stockage des données

Pour l'instant les fichiers d'entrée sont stockés sont forme de CSV/JSON, il advient donc de se demander si ce type de stockage est adapté pour l'ingestion de grand volume de données. On pourrait envisager d'utiliser un stockage avec une base de données, mais avec une augmentation du volume de données on perdrait en performance de réponse.

Pourriez-vous décrire les modifications qu'il faudrait apporter, s'il y en a, pour prendre en considération de telles volumétries ?

Comment peut-on donc faire pour ingérer de plus grandes volumétries de données.

I. Augmentation de la capacité de calcul

On a vu qu'avec une machine locale, on devient assez vite limité en capacité de calcul. La première piste à explorer est l'augmentation de la capacité de calcul.
Le choix logique pour améliorer cette infrastructure est le fait d'ajouter des machines similaires à la nôtre afin de former un cluster. On pourrait ainsi distribuer notre stockage et nos calculs entre nos machines.

Une solution assez évidente pour créer notre cluster est donc l'utilisation des services cloud (AWS, GCP, Azure) sur lesquels on peut louer des serveurs afin de créer nos/notre cluster.s. Il est également possible de créer notre cluster sur des serveurs locaux hébergé par nos soins.

II. Utilisation de framework de calcul distribué

Afin d'exploiter notre nouvelle architecture distribuée, il paraît logique d'utiliser un framework de calcul distribué. Plusieurs choix s'offrent à nous : ApacheSpark, ApacheBeam qui sont des framework de calculs distribués permettant de traiter de grand volume de données.