EduGroupe
Accompagner pour réussir

Dask : mise en oeuvre, programmation

Référence : **PYCB042** Durée : **3 jours (21 heures)** Certification : **Aucune**

Connaissances préalables

- Bases de la programmation python

Profil des stagiaires

- Chefs de projet, Data Scientists, Développeurs, Architectes...

Objectifs

- Savoir mettre en oeuvre Dask pour paralléliser des calculs en Python

Certification préparée

- Aucune

Méthodes pédagogiques

- 6 à 12 personnes maximum par cours, 1 poste de travail par stagiaire
- Remise d'une documentation pédagogique papier ou numérique pendant le stage
- La formation est constituée d'apports théoriques, d'exercices pratiques et de réflexions

Formateur

- Consultant-Formateur expert Bigdata

Méthodes d'évaluation des acquis

- Auto-évaluation des acquis par le stagiaire via un questionnaire
- Attestation des compétences acquises envoyée au stagiaire
- Attestation de fin de stage adressée avec la facture

Contenu du cours

1. Introduction

- Présentation de Dask, fonctionnalités, apports.Comparaison avec d'autres environnements : yarn, spark
- Calculs parallèles en environnements distribués, ou sur un seul serveur
- Les composants de Dask : scheduler, collections BigData

2. Premiers pas avec Dask

- Différentes méthodes d'installation : Anaconda, pip, depuis les sources
- Exemple d'atelier : installation, et création d'objets Dask, choix des méthodes et tâches, visualisation des graphes d'exécution.
- Exécution par le scheduler

3. Elements de base

- Array: cas d'usages, compatibilité NumPy, définition de chunks, exemples, bonnes pratiques
- Atelier : création, stockage de Dask Array
- Bag : définition, limites
- Exemple d'atelier : exemple de création, stockage, calcul sur des Dask Bags
- Dask Dataframes : regroupement de dataframes pandas, stockage sur disque ou dans un cluster, critères de choix par rapport aux dataframes pandas, bonne pratiques, compatibilité avec Parquet, intégration de tables SQL
- Exemple d'atelier : mise en oeuvre de dask.dataframes et comparaison avec pandas
- Delayed ou Futures : une exécution stockée dans un graphe d'actions, ou en temps réel, critères de choix

4. Fonctionnement avancé

- Gesion des performances
- Configuration du scheduler
- Les graphes d'exécution
- Utilisation du dashboard
- Outils de debugging
- Exemple d'atelier : tests de performances et debugging

5. Dask.distributed

- Fonctionnalités : exécution dans un environnement distribué ou en local, outils de diagnostic et de suivi des performances, utilisation de l'API Futures pour des calculs en temps réel
- Architecture : dask-scheduler et dask-worker
- Exemple d'atelier : mise en oeuvre de dask.distributed : installation, configuration, initialisation d'un client
- Présentation du dashboard
- Analyse des performances
- Limites de Dask.distributed
- Bonnes pratiques

6. Dask-ML

- Apports : utiliser les outils classiques de machine learning comme scikit-learn dans un environnement Dask
- Exemples d'utilisation : modèles complexes, volumes de données importants
- Présentation de Dask-ML et principe de fonctionnement
- Intégration scikit-learn, PyTorch, Keras / Tensorflow
- Exemple d'atelier : Installation et exemples avec scikit-learn

Notre référent handicap se tient à votre disposition au 01.71.19.70.30 ou par mail à referent.handicap@edugroupe.com pour recueillir vos éventuels besoins d'aménagements, afin de vous offrir la meilleure expérience possible.