INITIATION AU CALCUL DISTRIBUÉ AVEC PYTHON

Formation proposée par ALAA EL YASSIR

Python

Le succès de Python pour les applications scientifiques (Data science, Big Data, Machine Learning...) requiert de plus en plus de capacités de calculs. Ce cours vous initie au paradigme du calcul parallèle/distribué, des concepts de base aux techniques et librairies les plus avancées de l'écosystème Python.

◷DURÉE
5 Jour(s)
▣MODALITÉS
Présentiel / Distanciel
♙PARTICIPANTS
Jusqu'à 10 participants
★LANGUE
Français
LIEU
France entière
♙CERTIFICATION
Attestation de formation
▦MISE À JOUR
30/10/2023

VOTRE FORMATEUR



ALAA EL YASSIR
Formateur professionnel

TARIFS INDICATIFS

Inter-entreprise	**Pas de dates planifiées**
Intra-entreprise	**Sur devis**
Individuelle	**Sur devis**
Durée	**5 Jour(s)**

Tarifs indicatifs, sous réserve d'analyse du besoin et de devis.

◎Objectifs pédagogiques

✓• Comprendre les concepts de la programmation parallèle
✓• Savoir identifier quelles parties d'un programme son parallélisables et identifier les facteurs limitant
✓les performances de votre programme
✓• Posséder une vue d'ensemble de l'écosystème de calcul parallèle de Python
✓• Posséder les bases de la programmation parallèle: multithreading et multiprocessing
✓• Savoir distribuer des calculs sur plusieurs ordinateurs
✓• Savoir calculer sur GPU
✓• Savoir utiliser les frameworks modernes dont Dask, Vaex, Xarray, Ray ou Datashader pour le calcul
✓numérique
✓• Savoir créer des worflows de traitements de données avec Prefect
✓Ce cours n'intègre pas de calcul parallèle dans le cadre du machine learning

⌘Compétences visées

Python

Public cible et prérequis

Public cible

- Développeurs Python, adminsitrateurs systèmes souhaitant automatiser des tâches, ingénieurs et data scientists en quête de puissance de calcul, développeurs de sites Internet à forte charge

Prérequis

- Pratique régulière du langage Python et si possible des librairies numpy et pandas.

Programme détaillé

1 Les concepts de la programmation parallèle - 1 jour
• Les différentes formes de la programmation parallèle
- Au sein d'un même programme: multithreading, SIMD, compilation Just In Time
- Sur une même machine: multiprocessing, calcul distribué
- Sur plusieurs machine: calcul distribué, HPC
- Bien comprendre les différences entre multithreading, multiprocessing et programmation asynchrone
• Les bases de la programmation parallèle
- Multithreading: mise en oeuvre, gestion des accès concurrents
- Multiprocessing: mise en oeuvre, mémoire partagée, pools de process
- Présentation des différentes primitives associées au multiprocessing et multithreading: sémaphores, files d'attente, conditions, barrières, ...
- Comprendre les limites du multithreading en Python: Le GIL et savoir le désactiver
• Présentation de l'écosystème de calcul parallèle de Python
- Panorama des différentes librairies par thématique: Just In Time Compilation, Multithreading, Multiprocessing, Calcul distribué, Grille de calcul, Calcul sur GPU...

2 Profiling et Optimisation : 0,75 jour
• Monitorer les performances de votre programme : les différents types de profileurs et leur mise en oeuvre
• Créer son propre profileur
• Identifier ses goulots d'étranglement, savoir mesurer les performances de votre matériel : disque dur, mémoire, cpu
• Pensez à optimiser votre code avant de le paralléliser : techniques et algorithmes
• Compilation Just in Time avec Pypy et numba
• Logger et deboguer vos process avec Eliot

3 Calcul distribué : 0,75 jour
• Calcul distribué avec Celery et dramatiq: exécuter des tâches en arrière plan dans vos programmes Django avec Dramatiq
• Ray: distribuez vos tâches avec simplicité et performance

4 Big Data: 1 jour
• Vaex: manipulez des données tabulaires en utilisant toutes la puissance de votre ordinateur et les algorithmes les plus performants
• Dask: manipulez des données numériques ou tabulaires qui ne tiennent pas en RAM en les parallélisant automatiquement sur votre ordinateur portable, réseau local ou supercalculateur
• XArray: manipulez les fichiers scientifiques et big data ((NetCDF, HDF5, parquet...) avec facilité et en bénéficiant de tous les avantages de Numpy, Pandas et Dask
• Visualiser des données volumineuses avec DataShader et Holoviz

5 Calculer sur GPU : 1 jour
• Comprendre l'architecture matérielle d'un GPU
• Savoir choisir son matériel pour calculer sur GPU avec Python
• Panorama des différentes librairies de Calcul sur GPU avec Python
• Comprendre quand un GPU est préférable à un CPU et quand il ne l'est pas
• Compilation Just In Time sur GPU avec Numba
• Calcul numérique avec Cupy
• Manipuler les dataframes Pandas avec RapidsAI

6 Créer des workflows de tâches avec Prefect : 0,5 jour
• Présentation des concepts: tâches, paramètres, flows...
• Créer un workflow de traitement de données: téléchargement, filtres, transformations, sauvegarde
• Gérer les erreurs et reprises sur incidents
• Superviser l'exécution de votre workflow

▣Méthodes pédagogiques & évaluation

Méthodes pédagogiques

- ✓ Apports théoriques et retours d'expérience
- ✓ Exercices pratiques et mises en situation
- ✓ Travaux de groupe et échanges entre pairs
- ✓ Études de cas concrets

Modalités d'évaluation

- ✓ Quiz de validation des acquis
- ✓ Mises en situation évaluées
- ✓ Projet fil rouge
- ✓ Feuille de route post-formation

♿Accessibilité

Une formation accessible à tous

Ce programme peut être adapté pour les personnes en situation de handicap. Contactez-nous pour étudier vos besoins spécifiques.