Python, programmation parallèle et calcul distribué

Cours Pratique de 4 jours - 28h
Réf : PYP - Prix 2024 : 2 390€ HT

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.

OBJECTIFS PÉDAGOGIQUES

À l'issue de la formation l'apprenant sera en mesure de :

Acquérir les concepts de la programmation parallèle

Savoir identifier les portions d'un programme qui sont parallélisables

Posséder une vision claire de l'écosystème de calcul parallèle pour Python

Développer des applications parallélisées (programmation asynchrone, multithreading, multiprocessing, calcul distribué)

Savoir exécuter des calculs sur les GPU des cartes graphiques

Savoir exécuter un workflow de tâches dans le Cloud

MÉTHODES PÉDAGOGIQUES

70% du temps est consacré à la mise en pratique des concepts et librairies présentées. L'utilisation des notebooks Jupyter et l'exécution de code dans le Cloud apportent une réelle interactivité.

PARTICIPANTS

Développeurs, data scientists, data analysts, chefs de projets.

PRÉREQUIS

Bonnes connaissances du langage Python et si possible de ses librairies scientifiques Numpy, Scipy et Pandas.

COMPÉTENCES DU FORMATEUR

Les experts qui animent la formation sont des spécialistes des matières abordées. Ils ont été validés par nos équipes pédagogiques tant sur le plan des connaissances métiers que sur celui de la pédagogie, et ce pour chaque cours qu'ils enseignent. Ils ont au minimum cinq à dix années d'expérience dans leur domaine et occupent ou ont occupé des postes à responsabilité en entreprise.

MODALITÉS D'ÉVALUATION

Le formateur évalue la progression pédagogique du participant tout au long de la formation au moyen de QCM, mises en situation, travaux pratiques…
Le participant complète également un test de positionnement en amont et en aval pour valider les compétences acquises.

MOYENS PÉDAGOGIQUES ET TECHNIQUES

• Les moyens pédagogiques et les méthodes d'enseignement utilisés sont principalement : aides audiovisuelles, documentation et support de cours, exercices pratiques d'application et corrigés des exercices pour les stages pratiques, études de cas ou présentation de cas réels pour les séminaires de formation.
• À l'issue de chaque stage ou séminaire, ORSYS fournit aux participants un questionnaire d'évaluation du cours qui est ensuite analysé par nos équipes pédagogiques.
• Une feuille d'émargement par demi-journée de présence est fournie en fin de formation ainsi qu'une attestation de fin de formation si le stagiaire a bien assisté à la totalité de la session.

MODALITÉS ET DÉLAIS D'ACCÈS

L'inscription doit être finalisée 24 heures avant le début de la formation.

ACCESSIBILITÉ AUX PERSONNES HANDICAPÉES

Vous avez un besoin spécifique d'accessibilité ? Contactez Mme FOSSE, référente handicap, à l'adresse suivante psh-accueil@orsys.fr pour étudier au mieux votre demande et sa faisabilité.

LE PROGRAMME

dernière mise à jour : 12/2018

1) Le parallélisme et son écosystème Python

- Les différentes formes du parallélisme et ses architectures (CPU, GPU, ASIC, FPGA, NUMA, OpenMP, MPI...).
- Contraintes et limites.
- L'écosystème de calcul parallèle pour Python.

Travaux pratiques : Profiling d'un programme (cProfile, Kcachegrind et pyprof2calltree). Compiler un programme C avec les instructions SIMD. Bien installer Numpy : comment obtenir un gain de vitesse x40.

2) Les bases : programmation asynchrone, multithreading et multiprocessing

- Programmation asynchrone : générateurs et asyncio.
- Multithreading : accès concurrents, verrous…
- Limites du multithreading en Python.
- Multiprocessing : mémoire partagée, pools de process, conditions...
- Premier cluster de calcul distribué avec les Managers et Proxy.

Travaux pratiques : Réalisation d'une même chaîne de traitement de données avec chaque modèle et d'un cluster de calcul distribué entre les machines des participants.

3) Calcul distribué : Celery, Dask et PySpark

- Concepts et configuration.

- Mise en œuvre de chaque librairie.
Travaux pratiques : Plusieurs exercices seront abordés (calcul matriciel, traitement d'image/texte, Bitcoin, Machine Learning...). Utilisation des notebooks Zeppelin.

4) Calculer sur GPU
- Les architectures GPU : kernels, mémoire, threads...
- Les librairies OpenCL et CUDA.
- Mise en œuvre des librairies Scikit-cuda, PyCUDA et Numba.
Travaux pratiques : Calcul matriciel et traitement d'images. Machine Learning avec la librairie mxnet : Neural Art. Compilation Just In Time.

5) Autres librairies de programmation parallèle
- Message Passing Interface avec MPI4py.
- PyOpenCL : implémenter un code avec des systèmes hétérogènes.
- Joblib : Les pipelines légers.
- Greenlets : vers un meilleur multithreading.
- Pythran : Compiler vos programmes Python sur architectures multicœurs et vectorisées.
Travaux pratiques : Exercices de base avec chaque librairie.

6) Créer des workflows de tâches
- Les primitives disponibles avec Celery, Dask et PySpark.
- Créer et superviser des workflows avec les librairies Luigi et Airflow.
Travaux pratiques : Création de pipelines de traitements de données avec chaque librairie.

7) Exécuter des calculs dans le Cloud
- Panorama de l'offre Internet pour le Cloud.
- Administrer un cluster avec Ansible.
Travaux pratiques : Effectuer des calculs dans le Cloud.

LES DATES

CLASSE À DISTANCE
2024 : 27 mai, 16 juil., 08 oct.

PARIS
2024 : 09 juil., 01 oct.