Publication date : 07/04/2024

Course : Python, parallel programming and distributed computing

Practical course - 4d - 28h00 - Ref. PYP
Price : 2470 CHF E.T.

4,4 / 5

The success of Python for scientific applications (Data science, Big Data, Machine Learning...) requires more and more computational capacity. This course introduces you to the parallel/distributed computing paradigm, from basic concepts to the most advanced techniques and libraries in the Python ecosystem.

Teaching objectives

At the end of the training, the participant will be able to:

- Acquire the concepts of parallel programming
- Identify which parts of a program can be parallelized
- A clear vision of the parallel computing ecosystem for Python
- Developing parallelized applications (asynchronous programming, multithreading, multiprocessing, distributed computing)
- Know how to perform calculations on graphics card GPUs
- How to run a task workflow in the cloud

Intended audience

Developers, data scientists, data analysts, project managers.

Prerequisites

Good knowledge of the Python language and, if possible, its scientific libraries Numpy, Scipy and Pandas.

Practical details

Teaching methods

70% of the time is devoted to putting the concepts and libraries presented into practice. The use of Jupyter notebooks and code execution in the Cloud provide real interactivity.

Course schedule

PARTICIPANTS

Developers, data scientists, data analysts, project managers.

PREREQUISITES

Good knowledge of the Python language and, if possible, its scientific libraries Numpy, Scipy and Pandas.

TRAINER QUALIFICATIONS

The experts leading the training are specialists in the covered subjects. They have been approved by our instructional teams for both their professional knowledge and their teaching ability, for each course they teach. They have at least five to ten years of experience in their field and hold (or have held) decision-making positions in companies.

ASSESSMENT TERMS

The trainer evaluates each participant's academic progress throughout the training using multiple choice, scenarios, hands-on work and more.
Participants also complete a placement test before and after the course to measure the skills they've developed.

1 Parallelism and the Python ecosystem

- The different forms of parallelism and their architectures (CPU, GPU, ASIC, FPGA, NUMA, OpenMP, MPI, etc.).
- Constraints and limits.
- The parallel computing ecosystem for Python.

Hands-on work

Program profiling (cProfile, Kcachegrind and pyprof2calltree). Compiling a C program with SIMD instructions. Installing Numpy: how to get a x40 speed boost.

2 The basics: asynchronous programming, multithreading and multiprocessing

- Asynchronous programming: generators and asynchrony.
- Multithreading: concurrent access, locks...
- The limits of multithreading in Python.
- Multiprocessing: shared memory, process pools, conditions...
- First distributed computing cluster with Managers and Proxy.

Hands-on work

Realization of the same data processing chain with each model, and of a distributed computing cluster between the participants' machines.

3 Distributed computing : Celery, Dask and PySpark

- Concepts and configuration.
- Implementation of each library.

Hands-on work

Several exercises will be covered (matrix calculation, image/text processing, Bitcoin, Machine Learning...). Use of Zeppelin notebooks.

4 GPU computing

- GPU architectures: kernels, memory, threads...
- OpenCL and CUDA libraries.
- Implementation of Scikit-cuda, PyCUDA and Numba libraries.

Hands-on work

Matrix calculation and image processing. Machine Learning with the mxnet library: Neural Art. Just In Time compilation.

5 Other parallel programming libraries

- Message Passing Interface with MPI4py.
- PyOpenCL: implementing code with heterogeneous systems.
- Joblib: Lightweight pipelines.
- Greenlets: towards better multithreading.
- Pythran: Compile your Python programs on multicore and vectorized architectures.

Hands-on work

Basic exercises with each library.

TEACHING AIDS AND TECHNICAL RESOURCES

• The main teaching aids and instructional methods used in the training are audiovisual aids, documentation and course material, hands-on application exercises and corrected exercises for practical training courses, case studies and coverage of real cases for training seminars.
• At the end of each course or seminar, ORSYS provides participants with a course evaluation questionnaire that is analysed by our instructional teams.
• A check-in sheet for each half-day of attendance is provided at the end of the training, along with a course completion certificate if the trainee attended the entire session.

TERMS AND DEADLINES

Registration must be completed 24 hours before the start of the training.

ACCESSIBILITY FOR PEOPLE WITH DISABILITIES

Do you need special accessibility accommodations? Contact Mrs. Fosse, Disability Manager, at psh-accueil@orsys.fr to review your request and its feasibility.

6 Create task workflows

- Primitives available with Celery, Dask and PySpark.
- Create and supervise workflows with Luigi and Airflow libraries.

Hands-on work
Creation of data processing pipelines with each library.

7 Perform calculations in the cloud

- Overview of Internet offerings for the Cloud.
- Administer a cluster with Ansible.

Hands-on work
Perform calculations in the Cloud.

Dates and locations

REMOTE CLASS
2026 : 16 June, 15 Sep.