Native Implementation of OpenMP For Python

Introducing openmpy

OUAKLI Dorian

HPC, Big Data & Data Science Devroom
FOSDEM 2025

February 2, 2025

A built-in library

- A Thread class
- Functions to gather data about the current thread as well as others.
- Local memory
- Synchronization: Locks, Semaphores, Events, Barriers...
- Only one single thread can run at a given moment.

Some use cases

- Parallelizing Input/Output.
- Functions implemented in other programming languages.
- Event-driven architecture.

But PyOMP and Pythran?!

- Extension of Numba
- Compiled on the fly into LLVM IR, then in native machine code.
- Requires a specific implementation of libraries.

Pythran?

- Python source code to be compiled to generate an executable.
- Built-in support for OpenMP.
- Python support is highly limited to mathematical usage.

Parallelization in Python

- Multiprocessing
- Concurrent.futures
- AsyncIO

Nogil

- Sam Gross, Engineer at Meta, co-author of PyTorch.
- Fork of Python, without the GIL.
- One version in Python3.9
- One version in Python3.12
- Has applications in Machine Learning.

PEP 703

- Optional removal of the GIL in CPython.
- Has been implemented in Python 3.13 (October 2024)
- Compilation option `--disable-gil`
- Environnement variable `PYTHON_GIL=0`

omp

- Implements OpenMP.
- Usage similar to OpenMP in C.
- Python's peculiarities.
- Use only standard libraries.

Syntax

- Call `OpenMP(“directive”)` for directives.
- `OpenMP(“directive”)` context manager using `with` for constructs.
- Primitives with `omp.<primitive>`.

Multithreading

- Built-in library `threading`.
- **Function** as an entry-point.
- Extend the Thread class.
- Introduce a Team class.
- Support for `multiprocessing`?

parallel

- Shared variables by default.
- `private` clause support.

for

- Dynamic distribution by default.
- Makes the iteration variable private.
- `nowait`, `private`, `reduction` and `schedule` clauses support.

parallel for

- Syntax sugar for nested `parallel` and `for` constructs.
- `nowait`, `private`, `reduction` and `schedule` clauses support.

barrier

- Syntax sugar for a `threading.Barrier` shared across a Team.

critical

- Protects the block with a `threading.Lock` that is shared across a Team.
- `private` clause support.

single

- Protects the block with a Team-shared `threading.Lock`.
- `private` and `nowait` clauses support.