codon | Compiling Python for 100x performance gains

Star 12,614

Why Codon?

- Python is used across all industries and is becoming the *lingua franca* of computing
- But Python is slow and can't scale, meaning code is often rewritten in C++, Rust or another low-level language
- Codon leverages novel compiler technology to combine the simplicity of Python with the performance of C
- 10-100x faster (or more!) than Python on a single thread

Codon: a *next-generation* programming platform

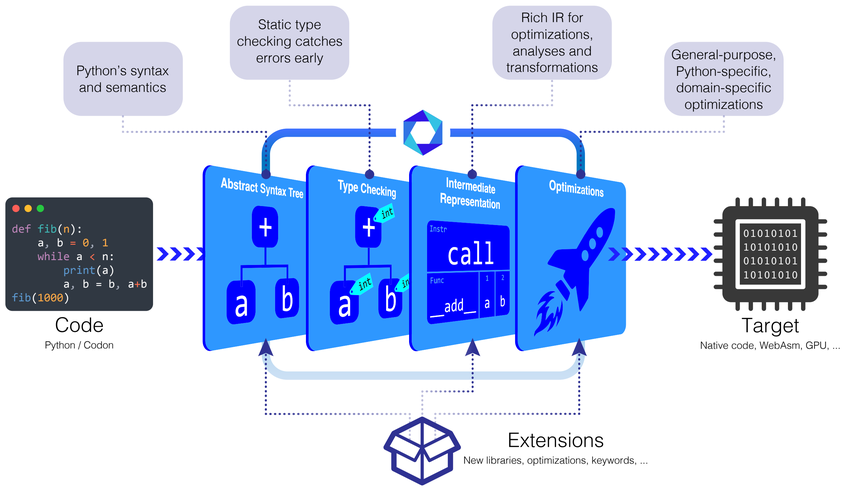
- **Python syntax** – zero learning curve
- **Native speed** – fast as C/C++, and sometimes faster!
- **Multithreading** – OpenMP backend for parallelism
- **GPU backend** – write GPU kernels in plain Python
- **Runnable in the browser** *(coming soon)*
- **Cloud connected** *(coming soon)*

Find Codon online

GitHub
github.com/exaloop/codon

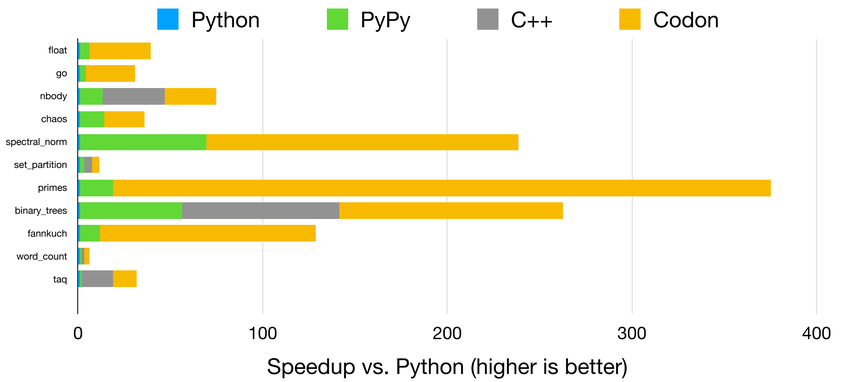
Docs & Info
exaloop.io

Under the hood



Codon uses ahead-of-time type checking techniques to compile Python code to native executables *without* any runtime overhead, and uses a custom optimization framework to apply a range of domain-specific and general purpose optimizations.

Benchmarks



Timed on Apple M1. Benchmarks "primes" and "fannkuch" use Codon's multithreading features. See exaloop.io/benchmarks.

Using Codon

- Codon is an end-to-end compiler and can generate executables, object files, shared libraries, LLVM IR and more.
- Have an existing Python codebase? You can use Codon on a per-function basis via the `@codon.jit` decorator, to compile & optimize just the performance-critical functions.
- Codon can also compile to *Python extension modules* that can be imported and used from Python while still leveraging all of Codon's features (à la Cython).
- Codon supports calling *any* Python library through its Python interoperability features. Codon-native versions of popular libraries like NumPy are currently under development.
- Codon can call C libraries with zero overhead. Just import them like you would any Python module!
- Docs, examples and guides available at docs.exaloop.io.

Applications

Codon for bio – Codon's bio module makes working with genomic data seamless and efficient. Scale to terabytes of data with the comfort of Python.

Codon for finance – Avoid costly rewrites and re-engineering for converting Python to C++: stay in Python *and* get all of C's performance benefits.

Codon for AI/ML – Solve data pre- and post-processing bottlenecks with Codon, and avoid under-utilization of hardware by leveraging GPUs throughout the pipeline.

Codon is developed by

exaloop

Reach us at info@exaloop.io to learn more!