

Python performance



- CPython performances are limited by the Python language and the C API
- PyPy made great progress on cpyext performance, but still has a few minor compatibility issues
- C API blocks some CPython enhancements as well

Stable ABI



- Stable API? Not really.
- Stable ABI is not tested: multiple regressions
- Stdlib doesn't use the stable ABI/API.
- Default remains the full API, in practice the stable ABI is useless for users.
- Gilectomy uses a different C API in exchange of real parallelism.

What is the C API?



- API used by CPython core
- API used by C extensions
- API used by low-level debuggers
- All header files are in the same Include/ directory
- => Split Include/ into subdirectories?

2x faster?



- CPython 3.7 is still as fast as CPython 2.7 (released in 2010) on most benchmarks
- How to become 2x faster?
- JIT compiler for CPython? Unladen Swallow, Pyston and Pyjion not successful yet.
- Promote PyPy and deprecate CPython?

Ideas



- New language similar to Python but stricter, similar to Hack for PHP?
- Compilation Ahead-of-Time (AoT) using guards checked at runtime (ex: my FAT Python project)
- Modern Javascript VM has multi-stage JIT compilers
- <http://faster-cpython.readthedocs.io/>