PyPy

September 17, 2024

1 Using Pypy instead of CPython

1.0.1 Python interpreters

There are several implementations of Python. We will call the reference implementation *CPython* in this chapter.

- CPython - implemented in C, with its own Virtual Machine, latest Python version **3.11**
- PyPy - implemented in RPython (a subset of Python2.7), with a VM similar to CPython, latest Python version **3.9**
- Jython - implemented in Java for the JVM, latest Python version **2.7**
- Ironpython - Implemented in C# for the .NET VM, latest Python version **2.7**
- Cython - a compiler turning code that is very close to Python into C modules, latest Python version **3.11**

Cpython is between 10 and 30 times slower than C. To make critical code go faster you have these options:

Write C by hand: - Build a C extension by hand. Works with CPython and Pypy, but slow with PyPy. Messy. - Use the *CTypes* standard library module to integrate with C. Works with CPython and Pypy, but slow. - Use *cffi*, a third party module made by the PyPy project. Works with both CPython and PyPy. Fast. Fairly easy.

Write Python: - Use Cython - requires a compilation step. Only works on some Python constructs. May require additional type information to produce fast code. - Use PyPy

1.0.2 The PyPy interpreter

http://pypy.org

PyPy is an advanced computer science project, using state-of-the-art techniques to make Python run faster and to add a number of cool features around the language.

- Highly compatible - only a few well documented differences
- Uses a modern garbage collection technique, rather than reference counting
- Uses a Just-in-Time (JIT) compiler
  - Finds hotspots (tight loops) while interpreting the program
  - Compiles the code of the loop into optimised machine code
  - Measuring and compiling means that there is a warmup time, usually a few hundered milliseconds
  - While **warming up**, PyPy is about **5 times slower** than CPython
  - On **longer runs**, PyPy is on the average **4.4 times faster** than CPython

  - On some selected problems, it is faster than C

Extra features include:

- Stackless mode, for massive parallel microthreads
- A sandbox, for doing restricted computing
  - The only calls you can do outside the interpreter are the ones explicitly listed
- A JIT for regular expressions

2 HPy

The use of C extensions to build Python plugins produces a number of problems:

- Dependency on the reference counting model for garbage collection
- Too much access to CPython internals restricts the refactoring of the CPython code base
- Other Python implementations need a compatibility layer that destroys performance gains

HPy is a joint initiative between some core CPython developers, the Cython project and the PyPy project to alleviate these problems.

- The H in HPy stands for *handle*. By introducing an extra level of indirection, C integration removes the dependencies on the CPython implementation.
- Performance is retained on the various platforms
- There may be tools developed that help porting C extensions to HPy

As of august 2023, HPy is mature enough for experimental use. Several large projects are porting their libraries including Numpy and Matplotlib.

[ ]: