

PyPy status talk

Holger Krekel Maciej Fijalkowski
Merlinux GmbH

PyCon 2009 - Rosemont

March 28 2009



What this talk is about

- Why we work on PyPy?
- What you can run on top of PyPy
- How fast is PyPy?
- Details about coming 1.1 release
- Questions and Answers

PyPy - user motivation

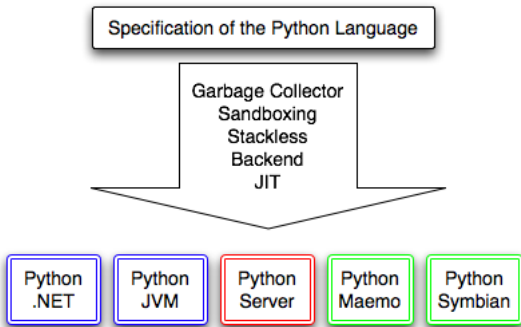
- use Python rather than C for performance
- have a more speedy, resource efficient interpreter
- support more programming paradigms

PyPy - motivation

- CPython is nice, but not flexible enough
- IronPython, Jython - bound to the specific VM
- Psyco and Stackless Python hard to maintain

PyPy: generating Python Interpreter

- **high level Python specification!**
- layer GCs, JIT, Stackless atop the spec
- **generate interpreters** for targets



Brief history of PyPy

- first sprint 2003, about 30 more by now
- CPython/Psyco/Jython/Stackless developers participating
- MIT-License, more sprints
- EU Research project 2004-2007
- 2007-now - open source project
- some google sponsoring

Getting Production ready

- we worked a lot on running existing applications on top of PyPy
- sometimes requiring to change applications slightly
- especially refcounting details tend to be a problem

```
open('xxx', 'w').write('stuff')
```

CTypes

- official way to have bindings to external (C) libraries for PyPy
- can handle i.e. pysqlite-ctypes, pyglet, pymunk or Sole Scion, almost whatever....
- contribution to original ctypes (better errno handling, bugfixes, tests...)
- part of google sponsoring
- note: a bit slow

Sqlite

- part of cpython stdlib since 2.5
- we use Gerhard Haering's CTypes version
- works reasonably well after some fixes

Django

- we run unmodified Django 1.0
- only sqlite DB backend for now

<http://www.djangoproject.com>

<http://code.djangoproject.com/wiki/DjangoAndPyPy>

Pylons

- worked almost out of the box once eggs were working (1 day)
- no SQLAlchemy yet, obscure problems ahead
- unmodified passes all tests
- <http://pylonshq.com/>

Twisted & Nevow

- twisted works (60/4500 tests failing)
- nevow works
- we don't support PyCrypto nor PyOpenSSL and we won't anytime soon (if nobody contributes CTypes or rpython versions)
- <http://twistedmatrix.com/>

Stackless

- We support stackless
- tasklets, frame pickling, greenlets
- fully cross-platform

Other software

- pure python should just work
- BitTorrent
- PyPy translation toolchain
- py lib
- sympy
- various smaller things, templating engines

Obscure details that people rely on

- non-string keys in `__dict__` of types
- exact naming of a list comprehension variable
- relying on untested and undocumented private stuff
- exact message matching in exception catching code
- refcounting details

Conclusion on Compatibility

- lessons learned: There is no feature obscure enough for people not to rely on it.
- pypy-c interpreter probably the most compatible to CPython 2.5
- main blocker for running apps will be missing external modules

Speed - comparison with CPython

- we're something between 0.8-4x slower than CPython on various benchmarks without JIT
- our JIT is a huge leap ahead
- pypy-c has fastest Interpreter startup

Speed - JIT generator

- 20-30x faster on small examples
- nice proof of concept
- a bit of time needed to speed up large python programs
- probably won't make it to 1.1
- completely separated from the interpreter

Memory - comparison with CPython

- PyPy has smaller Python objects
- user class instances often 50% of CPython size!
- PyPy has pluggable Garbage Collection

Threading / Stackless

- currently using GIL
- free threading? “it’s work”
- pypy-c has software threading / stackless
- no modifications to interpreter involved

Other backends

- PyPy-CLI runs! Still needs a bit more integration with .NET
- PyPy-JVM runs, little integration so far
- general speed improvements
- both backends are progressing - very slowly though
- contributors wanted!

pypy-c on small devices

- cross-compilation
- startup time
- security
- RAM usage
- share interpreter state across processes
- pypy approach a very good fit!

1.1 release

- compatible to Python 2.5.2
- well tested on win/linux 32 bit
- running major packages unmodified
- easy_install/distutils working
- help e.g. by writing ctypes modules
- hopefully released next month

Contact / Q&A

holger krekel, Maciej Fijalkowski at <http://merlinux.eu>

PyPy: <http://codespeak.net/pypy>

Blog: <http://morepypy.blogspot.com>

