

# Django and PyPy: performant is a word

Alex Gaynor

DjangoCon.eu 2011

6 June 2011



# Django and PyPy: performant is a word

- Django and PyPy core developer
- I like making **your** code faster
- Working at Quora making their codebase run on PyPy, fast.

# What is Django?

- Anyone here know?

# What is PyPy?

- An implementation of Python 2.7.1
- A very fast implementation
- A very compliant implementation

# What is PyPy? (2)

- Python written in Python
- Open source (MIT licensed)
- 8 years old
- Over 150,000 lines of test code (that's more than all of Django)
- A successor to Psyco

# Fast

- Faster than CPython on almost every benchmark we have.
- <http://speed.pypy.org/>
- A very actively developed project:  
<http://bit.ly/pypy-django-bench>

# World's shortest introduction to JITing

- Run interpreter
- Find frequently executed loops
- Turn those loops into efficient assembler, by specializing for the types of variables and other things.



# Case studies

- Production ready
- Real people are using this to speed up their apps.

- Parse the output of `git log` and generate data/reports
- CPython: 63 seconds
- PyPy: 21 seconds

# Some guy on IRC

- Query PostgreSQL and generate reports.
- CPython: 2 minutes
- PyPy: 8 seconds

# Why isn't everyone using PyPy?

- C extensions
- C-API tightly coupled to CPython implementation details

# Solutions

- CPyExt
- Pure Python/`ctypes`
- Cython (GSOC)

# But web apps are I/O bound...

- Eh, maybe they should be, but they often aren't.

# The Wild Wild Web (WWW for short)

- To run a Django site you need a handful of things
- Web server
- Database
- Random other libraries (`PIL`, `lxml`, etc.)

# Web server

- WSGI
- Any pure Python server will do
- I like `gunicorn`, you can use whatever you like
- *Not* `mod_wsgi`



# Database

- Use any database you like, so long as there's an adapter for it that works with both Django and PyPy!

- Standard library, just works!

# PostgreSQL

- RPython `psycopg2` compatible lib, requires compiling your own PyPy
- `pg8000` and tons of other random libraries, Django doesn't work with them, but if they're pure Python they'll work with other stuff (e.g. SQLAlchemy)

# MySQL

- (various expletives censored)
- Nothing that works with Django ATM
- I'm working on a `ctypes` based MySQLdb dropin replacement, hopefully open source soonish.

# Oracle

- We have an RPython `cx_Oracle`
- I know nothing about its status

# Other databases

- There are other databases?
- Uhh, talk to me later?

# Random other libs

- PIL - works under CPyExt
- lxml - doesn't work :(
- Others - how should I know? Others isn't very specific.

# Benchmarking!

- Lies, damned lies, and statistics!
- And benchmarks
- Ignore them, you need to test *your* app.
- But if you need to convince your boss...



# Django template benchmark

- Part of the Unladen Swallow benchmark suite
- PyPy 1.5: almost 10x faster than CPython
- PyPy trunk: almost 12x faster
- <http://bit.ly/pypy-django-bench>

# Rietveld benchmark

- Another part of the Unladen Swallow benchmark suite
- PyPy trunk: about 1.35x faster than CPython

# Tornado web app

- 2x as many requests per second

# Memory

- Mixed bag.
- Some apps use more, some use less.
- Benchmark your own app.

- A better platform for developing Python itself
- A faster Python for your apps

# Recruiting

- We could use some developers/designer to help with our performance tools.
- We have a cool webbased profiling/analyses tool.
- Flask/Jinja/jQuery (sorry)
- Contributors wanted, no compiler experience needed!
- `http://bit.ly/pypy-recruiting`

# Questions?

- `http://alexgaynor.net/`
- `http://pypy.org/`
- I want to make your apps faster, come talk to me!
- Thank you!
- Dank je wel!