

PHP interpreter using PyPy technology





October 4, 2013

introduction

- me - Maciej Fijałkowski, PyPy core developer
- technology - PyPy
- project - PHP interpreter

Wait, what???!!!1

- PHP - by far the most popular language on the web
- PyPy - proven technology for speeding up Python
- examples who can gain: wikipedia, facebook, wordpress....

Current landscape

- Zend - a simple, bytecode-based interpreter
- HipHop - PHP to C++ compiler, facebook project
- HHVM - successor to HipHop, JIT-based, also facebook

Current benchmarks landscape

- benchmarks are hard
- a set I've seen is mostly numeric or language shootout
- not very representative
- HipHop is 2-4x faster than Zend, HHVM 40% faster than hiphop
- no real-world PHP benchmark suite (a la speed.pypy.org)

- fast interpreter for a python language
- but also, a toolchain for constructing interpreters
- comes with a just-in-time compiler

More about PyPy

- implementation language of PyPy is RPython
- RPython is a subset of Python
- RPython can be compiled statically to C
- ... but also can have just in time compiler generated for

More about PyPy

- implementation language of PyPy is RPython
- RPython is a subset of Python
- RPython can be compiled statically to C
- ... but also can have just in time compiler generated for

More about RPython

- a great language for writing interpreters
- a horrible language with tons of tricks
- great results in good enough time
- http://tratt.net/laurie/blog/entries/fast_enough_vms_in_fast_enough_time

More about RPython

- a great language for writing interpreters
- a horrible language with tons of tricks
- great results in good enough time
- `http://tratt.net/laurie/blog/entries/fast_enough_vms_in_fast_enough_time`

Introducing hippy

- PHP interpreter written in RPython
- bug-to-bug compatible with Zend
- interpreter + just in time compiler (for free)
- preliminary study sponsored by facebook
- good preliminary performance results

PHP is hard

- crazy standard library
- function calls by name
- `*args` equivalent, `apply` equivalent, etc.
- crazy reference semantics
- copy-on-write and refcounting

PHP is hard

- crazy standard library
- function calls by name
- `*args` equivalent, `apply` equivalent, etc.
- crazy reference semantics
- copy-on-write and refcounting

... but getting it fast is easier

- we **do** have a JIT as soon as we write an interpreter
- PyPy has really good technology

Basic construction

- interpreter loop
- standard library
- typical stuff

Basic construction

- interpreter loop
- standard library
- typical stuff

Web server integration

- PHP is very request based with throwing out all the data in between
- we want to persist as much as possible
- various optimizations possible

We're hiring

- want to work on something obscure and challenging?
- with smart people?
- talk to me or Armin

- any questions?
- fijall at gmail
- `http://baroquesoftware.com`