PYPY

OPTIMIZING PYTHON PROGRAMS, PYPY TO RESCUE

6. Oct. 2016, Cape Town

RICHARD PLANGGER

MORE "GENERAL" PYPY TALK

Goals:

- An approach to optimize Python programs
- Examples
- How not to start optimizing
- What is PyPy up to now?

PYPY IS A ...

... fast virtual machine for Python

developed by researchers, freelancers and many contributors.

```
$ python yourprogram.py
$ pypy yourprogram.py
```

PYPY IS NOT JUST THAT

Experiment with new ideas

- Python written in Python
- RPython
- JIT compiler
- VMProf
- PyPy STM
- ...

ABOUT ME

Working on PyPy (+1,5y)

Master thesis → GSoC 2015 → PyPy

living and working in Austria

SPEEDY PYTHON PROGRAMS?

When is your Python program fast enough?

When it gets a speeding ticket because it is too fast?

or when PyPy's benchmark suite reaches 10x faster on average?

Neither

Run your program an measure your **criteria**

FOR EXAMPLE?

- CPU time
- Peak Heap Memory
- Requests per second
- Latency
- ...

Dissatisfaction with one criteria of your program!

SOME THEORY ...

COMPLEXITY

Big-O-Notation

Classify e.g. a function and it's processing time

Increase input size to the function

- `a = 3 # O(1)`
- `[x+1 for x in range(n)] # O(n)`
- `[[x+y for x in range(n)] \`
  `for y in range(m)] # O(n*m) == O(n) if n > m`

Bubble sort vs Quick Sort

O(n**2) vs O(n log n)

COMPLEXITY

Yields the most gain, independent from the language

E.g. prefer O(n) over O(n**2)

ONLY OPTIMIZE A ROUTINE IF ...

you know that the complexity cannot be stripped down

LET'S START FROM THE BEGINNING

with a small example

READING LOG FILES!

JITLOG (facility to observe PyPy's JIT internals)

- Written in Python
- Moved to vmprof.com
- Log files can easily take up to 40MB uncompressed
- Takes ~10 seconds to parse with CPython
- Complexity is linear to input size of the log file

THANKS TO PYTHON

+ Little development time

+ Easy to test

- Takes too long to parse

- Parsing is done each request

Our criteria: CPU time to long + requests per second

(Many objects are allocated)

SUGGESTION

Caching

Reduce CPU time

Let's have both

Caching - Easily done with your favourite caching framework

Reduce CPU time - PyPy seems to be good at that?

LET'S RUN IT...

```
$ cpython2.7 parse.py 40mb.log
```

~ 10 seconds

```
$ pypy2 parse.py 40mb.log
```

~ 2 seconds

CACHING

Requests really feel instant after the log has been loaded once

Precache

THE LAZY APPROACH OF OPTIMIZING PYTHON

VMPROF

```
$ pip install vmprof
```

```
$ python -m vmprof --web parse.py
```

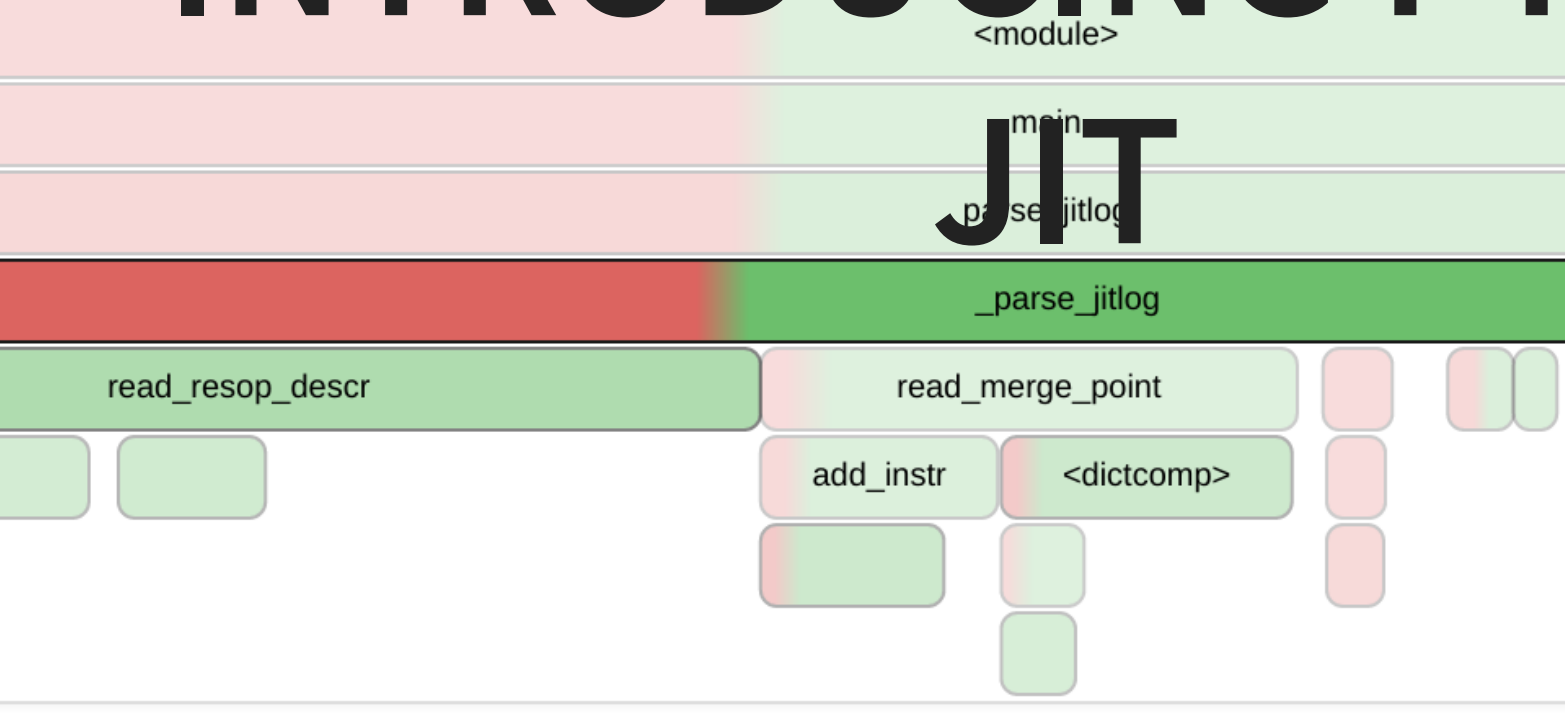
→ link

(100.00%)

00% ↓ 0.00% / 0.00%

JIT: 57.74% GC: 0.00%

INTRODUCING PYPY'S JIT



HOT SPOTS

Loops / Repeat construct!

What kind program can you build without loops?

A SIMPLIFIED VIEW

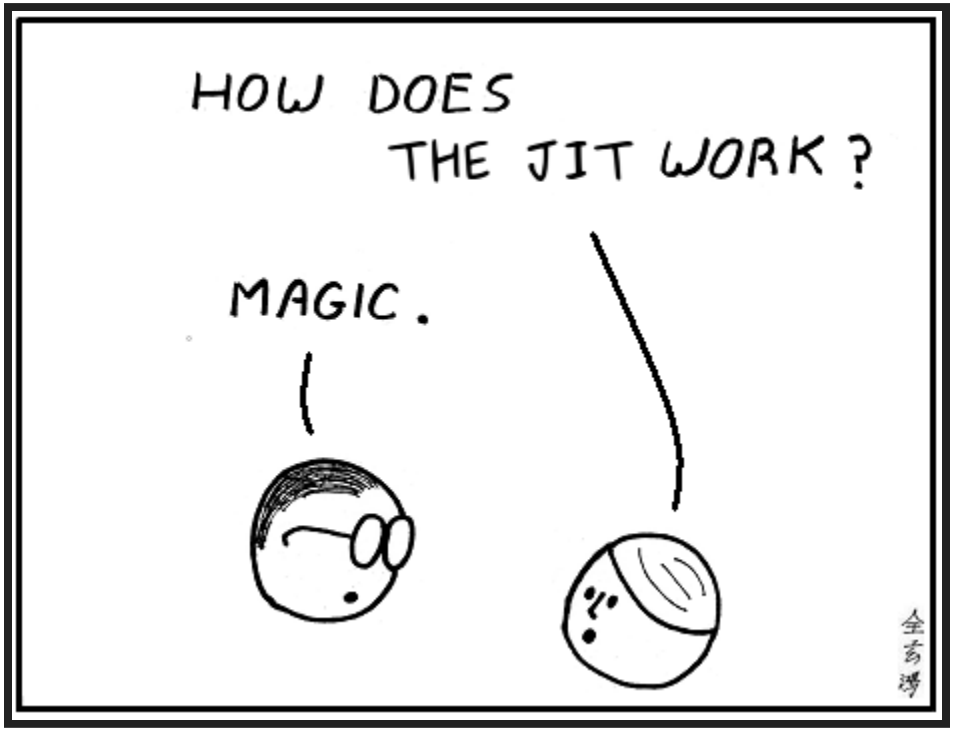
1. Start interpretation
2. Loops trigger recording
3. Optimization stage
4. Machine code generation

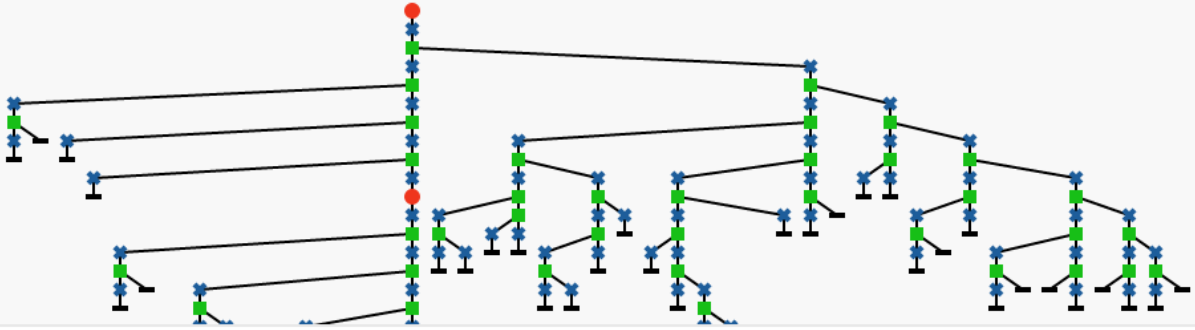
BEYOND THE SCOPE OF LOOPS

Guards ensure correctness

Frequent guard failure triggers recording

PERCEPTION





ges

Recording

py line 351

py line 339

Traced Opt Rewritten Asm

→ link

This trace is a **loop**. Driver name: "" Scope/Function: "schedule"
Filename: ../pypy/rpython/translator/goal/richards.py Lineno: 351

```
 1: increment_debug_counter(9053872) passed 980.8 k times
 2: gc_load_i(p0, 40, -8)
 3: gc_load_r(p0, 64, 8)
 4: gc_load_i(p0, 72, -8)
 5: gc_load_r(p0, 16, 8)
 6: gc_load_r(p0, 32, 8)
 7: gc_load_r(p0, 48, 8)
 8: gc_load_r(p0, 56, 8)
 9: gc_load_r(p24, 16, 8)
10: gc_load_r(p24, 24, 8)
11: gc_load_r(p24, 32, 8)
12: gc_load_r(p24, 40, 8)
```

JITVIEWER

Tool to inspect PyPy internals

Helps you to learn and understand PyPy

Provided at vmprof.com

PROPERTIES & TRICKS

- Type specialization
- Object unboxing
- GC scheme
- Dicts
- Dynamic class creation (Instance maps)
- Function calls (+ Inlining)

ANOTHER REAL WORLD EXAMPLE

MAGNETIC

Marketing tech company

Switched to PyPy 3 years ago

Q: WHAT DOES YOUR SERVICE DO?

A: ... allow generally large companies to send targeted marketing (e.g. serve ads) to people based on data we have learned

Q: PYPY, WHERE WAS IT MOST HELPFUL?

A: ... ~30% speedups immediately from switching to PyPy ...

Q: PYPY ISSUES?

A: ... we had to solve for rolling deploys ... but that's ok, that's fairly easy ...

Q: VALUE TO YOUR COMPANY?

A: Latency speedup was somewhere aroudn 10% ...

But that number is deceiving

It's very valuable for us obviously

But it's only 10%, because even this app that I'm talking about, which is fairly high volume (500,000 QPS), is a WSGI app

So it spends lots of time blocking

TIMEIT

why not use perf?

Try timeit on PyPy

PYTHON 3.5

Progressed quite a bit

async io

Many more small details (sprint?)

C-EXTENTIONS

NumPy on top of the emulated layer

Boils down to managing PyPy & CPython objects

CLOSING EXAMPLE

how to move from cpu limited to network limited

link

QUESTIONS?

morepypy.blogspot.com
software@vimloc.systems
Join on IRC #pypy