Software Carpentry IV: Profiling & Debugging

Luís Pedro Coelho

Programming for Scientists

February 12, 2009

University of Pittsburgh

Carnegie Mellon

Today's Lecture

- Debugger
- Profiler

Debugger

Your code will have bugs.

Three Ways to Debug

1. Run in debugger.
2. Debugger at “crash” (exception in Python).
3. Print statements.

Print Statements

Learn about the `logging` module.

```
import logging
# Setup logging

logging.basicConfig(
    # FIXME: This should be read from
    # os.env['PYTHON_DEBUG'] or some variation thereof
    level=logging.DEBUG,
    format='%(asctime)s %(levelname)-8s %(message)s',
    datefmt='%a, %d %b %Y %H:%M:%S',
    #filename='slif.python.log',
    #filemode='w'
    )

logging.info('Starting up')
...
logging.debug('Executing this, that, and the other')
```

Performance doesn't matter until it matters.
—Python Lore

“We should forget about small efficiencies, say about 97% of the time: premature optimization is the root of all evil.”
— Donald Knuth

Making it faster

97/3 Rule

Your code will spend 97% of its time in 3% of the code.

“We should forget about small efficiencies, say about 97% of the time: premature optimization is the root of all evil. Yet we should not pass up our opportunities in that critical 3%.”
—Donald Knuth

Profiler

Where are those 3%?

Profiler

Where are those 3%?

You don’t (generally) know where the speed-bottle necks will be!

Profiler

```
import profiler
...
profiler.run(code)
```

or (inside ipython):

```
%prun code
```

Performance Bugs

At one time, I wanted to optimise a bit of my code.

Original Version

```
labeled,N = ndimage.label(protproc > 0)
for obj in xrange(1,obj+1):
    objimage = croptobbox(protproc * (labeled == (obj+1)))
    ...
```

Original Version

```
labeled,N = ndimage.label(protproc > 0)
for obj in xrange(1,obj+1):
    objimage = croptobbox(protproc * (labeled == (obj+1)))
    ...
```

Faster Version

```
labeled,N = ndimage.label(protproc > 0)
objects = ndimage.find_objects(labeled)
for i,slice in enumerate(objects):
    objimage = protproc[slice] * (labeled[slice] == (i+1))
    ...
```