Software as a Research Tool

Connections
Outline

- State of the Madagascar project
- Madagascar philosophy
- Madagascar fundamentals
"Abandoning the habit of secrecy in favor of process transparency and peer review was the crucial step by which alchemy became chemistry. In the same way, it is beginning to appear that open-source development may signal the long-awaited maturation of software development as a discipline."  

*Eric Raymond*
In a Nutshell, Madagascar…
…has had 11,993 commits made by 83 contributors representing 874,752 lines of code
…is mostly written in C with an average number of source code comments
…has a well established, mature codebase maintained by a very large development team with stable Y-O-Y commits
…took an estimated 239 years of effort starting with its first commit in May, 2003 ending with its most recent commit about 3 days ago
Contributors

Stable Versions

- 0.9 – 06/2006
- 1.0 – 07/2010
- 1.6 – 05/2014
- 1.7 – 01/2015
Madagascar Schools

- 2006 – Vancouver, Canada
- 2007 – Austin, USA
- 2008 – Golden, USA
- 2009 – Delft, Netherlands; Bahia, Brazil
- 2010 – Houston, USA
- 2011 – Beijing, China
- 2012 – Austin, USA
- 2013 – Melbourne, Australia
- 2014 – St. Petersburg, Russia
- 2015 – Harbin, China
Commits per Month

- 2006
- 2009
- 2012
- 2015
about Madagascar

Contributors per Month
Austin-2013 (Working Workshop)
Migration Gallery
Houston-2014 (Working Workshop)
Madagascar Website Visits
Madagascar Website Visits
Outline

- State of the Madagascar project
- Madagascar philosophy
- Madagascar fundamentals
Research Pyramid

- Papers
- Workflows
- Examples
- Programs
Research Pyramid

- 1,000 Programs
- 500 Workflows
- 5,000 Figures
- 150 Papers

Languages and Tools:
- LaTeX
- Python
- SCons
- Unix
- C
What is Science?
“Science is the belief in the ignorance of experts.”

Richard Feynman

What is Science? (1966)
What is Science?

Science is the systematic enterprise of gathering knowledge about the universe and organizing and condensing that knowledge into testable laws and theories. The success and credibility of science are anchored in the willingness of scientists to expose their ideas and results to independent testing and replication by other scientists. This requires the complete and open exchange of data, procedures and materials.
Fig. 1. The seismic experiment, conducted over a simplified subsurface with one dipping reflector. Applying the law of cosines to triangle $s'sr$, one may express the travel time $t$ from source $s$ to receiver $r$ in terms of zero-offset time $t_0$, half-offset $h$, velocity $v$, and dip $\theta$. The result is equation (3) in the text, the

Defining

$$A \equiv \frac{dt_n}{dt_0} = \frac{t_0}{t_n} = \left[ 1 + \frac{(\Delta t_0)^2}{(\Delta y)^2} \frac{h^2}{t_n^2} \right]^{1.2},$$

and using equation (10) to replace $p_0(\sqrt{t_n^2 + (\Delta t_0/\Delta y)^2}h^2$, $y, h) = p_n(t_n, y, h)$, the Fourier transform becomes

$$P_0(\omega_0, k, h) = \int dt_n A^{-1} e^{i\omega_0 t_n} \int dy e^{-iky} p_n(t_n, y, h). \quad (12a)$$
“An article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship. The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures.”
(Buckheit and Donoho, 1995)
Reproducible Research

“It is a big chore for one researcher to reproduce the analysis and computational results of another [...] I discovered that this problem has a simple technological solution: illustrations (figures) in a technical document are made by programs and command scripts that along with required data should be linked to the document itself [...] This is hardly any extra work for the author, but it makes the document much more valuable to readers who possess the document in electronic form because they are able to track down the computations that lead to the illustrations.”

(Claerbout, 1991)
NEWS

REPRODUCIBLE RESEARCH

ADDRESSING THE NEED FOR DATA AND CODE SHARING IN COMPUTATIONAL SCIENCE

By the Yale Law School Roundtable on Data and Code Sharing
Reproducibility in Computational & Experimental Mathematics @ ICERM 2012
Reproducibility in Computational and Experimental Mathematics @ ICERM

- “It is important to promote a culture change that will integrate computational reproducibility into the research process.”
- “Journals, funding agencies, and employers should support this culture change.”
- “Reproducible research practices and the use of appropriate tools should be taught as standard operating procedure in relation to computational aspects of research.”
Objectives

- To make computational research efficient
- To make it easy to share computational results
- To promote an open community
Outline

- State of the Madagascar project
- Madagascar philosophy
- Madagascar fundamentals
Madagascar Fundamentals

- RSF file format
- Command-line usage
- Introduction to SCons/Python
Madagascar Design Principle

- Data arrays are file objects on disk

- “Write programs that do one thing and do it well. Write programs to work together. Write programs to handle text streams, because that is a universal interface.”

Doug McIlroy
• “To design a perfect anti-Unix, make all file formats binary and opaque, and require heavyweight tools to read and edit them.”

• “If you feel an urge to design a complex binary file format, or a complex binary application protocol, it is generally wise to lie down until the feeling passes.”

  Eric Raymond
RSF File Format

- Principles
- Header and data files
  - Datapath
  - Type
  - Form
  - Hypercube
- Compatibility with other formats
  - SEPlib
  - SEGY
  - SU
Datapath Rules

- `datapath=` parameter on the command line.
- DATAPATH environmental variable
- `.datapath` file in the current directory.
- `.datapath` file in the user home directory.
- Current directory.
Data Type

- uchar,
- char
- int
- float
- complex
- short
- double
Data Form

- native
- XDR
- ASCII
Hypercube

- $n_1, n_2, \ldots$
- $d_1, d_2, \ldots$
- $o_1, o_2, \ldots$
- $label_1, label_2, \ldots$
- $unit_1, unit_2, \ldots$
- label
- unit
Outline

- RSF file format
- Command-line usage
- Introduction to SCons/Python
What is SCons?

- Build system (Software Construction)
- Written in Python
  - Configuration files are Python scripts
- Built-in support for different languages
- Dependency analysis
- Parallel builds
- Cross-platform
Evolution of Build Systems

- **Make (1977)**
  - “Sendmail and `make` are two well known programs that are pretty widely regarded as originally being *debugged into existence*. That's why their command languages are so poorly thought out and difficult to learn. It's not just you - everyone finds them troublesome.”
  - *Peter van der Linden*

- **GNU Make (1988)**

What is Python?

- Dynamic programming language
- Clear, readable syntax
  - “friendly and easy to learn”
- Full modularity
  - “batteries included”
- Integrates with other languages
  - “plays well with others”
- Free and open-source
Who uses Python?

- "Python has been an important part of Google since the beginning, and remains so as the system grows and evolves. Today dozens of Google engineers use Python, and we're looking for more people with skills in this language." Peter Norvig

- “Python is fun, free, runs on a broad range of platforms and has a large library of sophisticated modules, including numerical. It meets all our criteria for a first language.” John Scales & Hans Ecke
Python is the Top First Language

Number of top 39 U.S. computer science departments that use each language to teach introductory courses

Analysis done by Philip Guo (www.pgbovine.net) in July 2014
Python in 5 Easy Steps

1. Variables and strings
2. Lists and dictionaries
3. For loop
4. If/else, indentation
5. Functions and modules
1. Variables and Strings

```python
>>> a = 'Harbin'
>>> a[0]
'H'
>>> a[:3]
'Har'
>>> b = a + "is cool"
>>> print b
'Harbin is cool'
>>> a+2
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: cannot concatenate 'str' and 'int' objects
>>> a+str(2015)
'Harbin2015'
```
2. Lists and Dictionaries

```python
>>> a = ['Harbin', 'China']
>>> a[0]
'Harbin'
>>> len(a)
6
>>> a.append(20)
>>> a
['Harbin', 'China', 20]
>>> b = ('Harbin', 'China')
>>> b.append(20)
Traceback (most recent call last):
  File "<stdin>" , line 1, in <module>
AttributeError: 'tuple' has no attribute 'append'
>>> c = { 'city' : 'Harbin', 'year': 2015 }
```
3. For loop

```python
>>> a = (‘Harbin’, ‘Beijing’)
>>> for city in a:
...     print city, len(city)
Harbin 6
Beijing 7
>>> for k in range(2):
...     print k, a[k]
0 Harbin
1 Beijing
>>> c = {‘city’: ‘Harbin’, ‘year’: 2015}
for key in c.keys():
...     print c[key]
    ‘Harbin’
    2015
```
4. If/else, indentation

```python
>>> for k in range(4):
    if k < 2:
        print k
    else:
        print 'no'
0
1
no
no
>>> try:
    a = 'Harbin' + 2015
except:
    print 'error'
error
```
5. Functions and modules

```python
>>> def add_5(a):
...     'Add 5 to input'
...     return 5+a
>>> a = add_5(3)
>>> a
8
>>> def add_b(a,b=5):
...     'Add b to a'
...     return b+a
>>> add_b(a)
13
>>> import math
>>> math.sqrt(add_b(a,8))
4
```
SCons for Program Compilation

env=Environment()
env.Program(“program”, “file.c”)
Madagascar Processing: rsf.proj

- **Fetch(‘filename’, ‘dirname’)**
  - A rule for downloading files from a server
- **Flow(‘target’, ‘source’, ‘command’)**
  - A rule for making target from source
- **Plot(‘target’, ‘source’, ‘command’)***
  - Like Flow but generates a figure file
- **Result(‘target’, ‘source’, ‘command’)***
  - Like Plot but generates a final result
Parallel Processing with pscons

- Figures out the number of nodes/CPUs and runs `scons -j`
- Use `split=` and `reduce=` to split data for simple data-parallel processing
Madagascar Fundamentals

- **RSF** file format is simple, represents a multidimensional array as text + binary
- Madagascar programs are building blocks, can run on the command line and pipe Unix-style
- **SCons** is a powerful and convenient build system adopted by Madagascar for data processing and reproducible documents
  - Configuration files are **Python** scripts