L05_FortranCInterface

December 15, 2015

1 Python as Glue

Kevin Stratford kevin@epcc.ed.ac.uk Emmanouil Farsarakis farsarakis@epcc.ed.ac.uk
Contributing authors: Neelofer Banglawala Andy Turner Arno Proeme

www.archer.ac.uk support@archer.ac.uk

[Python as Glue] Scientific Workflows

- Data analysis / Simulation / Numerical computation
  - becoming larger amd more complex
  - can be time critical
  - consuming ever-increasing amounts of resource
- There is a need to
  - manage data, executable programs; and dispose of output
  - develop flexible applications rapidly
  - concert (administratively or geographically) disparate resources

[Python as Glue] Horses for Courses

- "Tradiational" languages (C/C++/Fortran)
  - provide large existing pool of standard libraries / standalone executables
  - are good as expressing numerical algorithms
  - use compilation, admitting fast code
- Python
  - provides a fast development cycle
  - Good at dealing with "messy" unstructured data
  - Allows easy interaction with data / OS / Internet

[Python as Glue] The Best of Both Worlds

- Broadly two approaches:
  - Loosely coupled: interact via operating system / external service
  - Strongly coupled: interact at the level of code API

[Python as Glue] Loose Coupling I

- Standard library includes
  - Data handling: strings, data types

  - Data persistence, compression archiving
  - Data base functionality

- Interact with “outside world” including

  - Operating system
  - Internet protocols
  - HTML/XML support

https://docs.python.org/2/library/
[Python as Glue] Loose Coupling II

- General workflow:

  - Marshal input data / control parameters
  - Launch executable program via OS
  - Review output
  - [Refine and repeat?]

- Care with

  - portability issues
  - interaction with queue systems
  - parallelism

[Python as Glue] Strong Coupling I

- Call one language from another

  1. python calls target language
  2. target language calls python
  3. “two-way coupling”

- We will consider case (1)

  - (2) is possible via C native API, but probably not the model we want
  - (3) likewise, probably undesirable here

[Python as Glue] Strong Coupling II

1. Existing code in target language

   - should have well-defined API (i.e., is a library)
   - may need to be “re-entrant”
   - may be parallel

- Exact approach depends on

  - target language
  - whether a clean separation of python/target language is required
  - what the python interface should look like

[Python as Glue] Fortran I

- Python style is typically to have

  - functions with dummy arguments that remain unchanged
  - result object(s) returned via return list

- Fortran

- functions with intent(in) arguments ok
- subroutines with intent(inout) arguments?

[Python as Glue] Fortran II

- How to proceed
  - numpy supplies, as standard, f2py
  - tool to create python interface directly from Fortran
  - Servicable for external subroutines (a la Fortran 77)
- In practice, for "modern" Fortran
  - Number of other tools have been developed
  - Most (e.g., pyfort) have no active development
  - Best option appears to be f90wrap

https://github.com/jameskermode/f90wrap
[Python as Glue] Fortran III

- What f90wrap does
  - command line tool (python)
  - operates on Fortran source (module.f90)
  - generates a simplified Fortran interface module
- Uses native compiler to
  - compile module.f90 and the simplified interface

[Python as Glue] Fortran IV

- Uses f2py to generate
  - python extension module describing python interface
  - module.py
  - a shared object module.so
- From python (script or shell)
  - interface available via import module

[Python as Glue] C/C++ I

- Python to C/C++
  - is a more natural "match"
  - more interfacing alternatives available
- Some provide a clean separation
  - C foreign function interface (CFFI)

https://cffi.readthedocs.org/en/latest/
[Python as Glue] C/C++ II

- Others allow patching code into python
  - Weave (part of scipy)
- Other still require "intermediate" language
  - Cython - C externsions to python

- A superset of python

http://docs.scipy.org/doc/scipy-0.14.0/reference/tutorial/weave.html http://cython.org/
[Python as Glue]   Summary

- Many possibilities
  - many are “work in progress”
  - some will fall by the wayside
- Care required
  - choosing what to do in the first place
  - identifyying sustainable solutions