

PWC Python Course: Discussion

Ramses van Zon

SciNet HPC Consortium

December 11, 2014



Discussion

Below are a few slides to guide the discussion on our experiences and best practices with python.

Linking to external C and Fortran libraries

- There are package for this:
 - ▶ weave
 - ▶ cython
 - ▶ plain python api
- These are hard to get to work, even harder to get to work cross-platform.
- If you have an in-house library that you want to use in python, maybe this is work your while.
- Else, using subprocess to call external programs may just be easier.

Sphinx-like documentation

- Documentation is usually done wrong
- Sphinx is a python system to do documentation
- It itself is poorly documented.

Tip: Don't rely on a documentation system to generate documentation.
Instead:

- Comment your code
- Ensure each function and module has a 'docstring'
- *Write* documentation.
- Use documentation generators only to generate internal reference material.

Error propagation/handling

We talked about this a bit before, but here are some tips:

- Use exceptions, assert, with
- If it's a one-off script, report the error (well) and let it crash.
- If it's an app that need to keep running (e.g. steering an external simulation), catch your exceptions.
- Distinguish program/system errors from workflow events
 - ▶ E.g. An optimization that didn't converge isn't a program error, just mean your workflow failed.
Such cases are best tracked through status variables
 - ▶ But if your results couldn't be written do disk, this is a more serious error.
- Once your script is written, go over it, think about what could go wrong, and make it robust.

List of favorite modules

- SciPy
- NumPy
- mpi4py
- Numexpr
- iPython
- Matplotlib, pylab
- PyTables
- NetCDF4-python
- PySVN
- Mercurial
- nose
- pandas
- subprocess