Subject: Re: Migrate away from idl?
Posted by Maarten[1] on Thu, 19 Nov 2009 12:25:41 GMT
View Forum Message <> Reply to Message

On Nov 19, 12:29 am, Brian Larsen <balar...@gmail.com> wrote:

> I have a quick poll for the experts out there. I recently learned
> about (and was wowed) scipy and all the other python stuff that
> basically make it a viable substitute for idl. I am wondering if
> anyone has made the switch (if so why are you still on this usenet?)
> or came away from python or has any advice at all?

For a new project we have/will switch(ed) to Python + scipy + numpy +
matplotlib + pytables + pyhdf + ...

> Here is a gallery of plots and code from python
> matplotlibhttp://matplotlib.sourceforge.net/gallery.html

In some cases the plots from matplotlib require less fudging to become
publication quality. Changing the graphs interactively, and then
produce pdf/eps at a later moment really help. Perhaps the same can be
done with the itools, but I can't be bothered, matplotlib is just
easier. The IDL plots require quite a bit of work to get to
publication standards (mine anyway - no Hershey fonts for me).

> I am seeing some cool advantages and some disadvantages too, ill list
> a few to try and start a bit of discussion,
>
> Sames:
> array based a = a+1 for arrays
> multiplatform
> command line to test things out
> all the same plots are there (or at least all I do)

There are some plots that I would have to think hard to produce, where
matplotlib produces them with ease (a box plot comes to mind:
http://en.wikipedia.org/wiki/Box_plot).

iPython is easily better than the IDL commandline

> +idl:
> I already know it
> code library that I am familiar with
> Coyote's Guide to IDL Programming

The latter is very valuable. Without it I would have gone python quite
some time ago.

> +python:
> no licensing fee
> easier scriptable (like into web pages etc)
> interactive plots that make iplot look like 1980
> super easy export of plots to png, pdf, ps whatever even interactively
> from the plot
> seems like its up and coming

Python is big, way bigger than IDL. Using SWIG it is generally easy to include C-code in python, easier in my limited experience than it is to do the same with IDL. Since for loops aren't quite as devastating for performance as they are in IDL, you generally end up with code that is easier to maintain (human-readable code).

Python is also a general programming language, with neat objects and fully dynamic arrays and fully automated memory management (no object or pointers to forget). Oh, the scripts and the run-time environment use the same rules, no more for i=0,10 do & begin & j=j+i^2 & endfor on the command line, and something subtly different in the script, and python is smart about end of lines, so no
file_list = file_search(dir, $
pattern)

Docstrings are probably the best idea: have your documentation ready for interactive use.

One disadvantage of Python: to install all the science code you need, you'll be busy for a while, and some are just hard to get running (scipy is one example, there are others).

Maarten
