Subject: Re: Sacrilegious but genuine question
Posted by Liam E. Gumley on Tue, 08 Jan 2002 18:16:44 GMT
View Forum Message <> Reply to Message

Francis Burton wrote:
> How easy would it be to obtain the functionality of IDL by
> using Python as the underlying language glue, supplemented
> by hard-coded modules for image, signal & file processing,
> graphical I/O etc.? It seems that it can already do that in
> some problem domains - e.g. Python Imaging Library.

http://nickbower.com/computer/pydl is a first cut.

> Put another way: What are the advantages of IDL compared to
> such an open source framework?

It depends on what your goal is.

If your goal is to read, analyze, and visualize data for the purpose of earning a living or obtaining a degree, then I'd use IDL.

If your goal is to have fun writing new code for which you don't get paid or earn any credits, then I'd roll your own open source solution in Python (if you can get someone to pay you for it, well done).

Cheers,
Liam.
Practical IDL Programming
http://www.gumley.com/

Subject: Re: Sacrilegious but genuine question
Posted by Craig Markwardt on Tue, 08 Jan 2002 19:38:02 GMT
View Forum Message <> Reply to Message

"Liam E. Gumley" <Liam.Gumley@ssec.wisc.edu> writes:

> Francis Burton wrote:
>> How easy would it be to obtain the functionality of IDL by
>> using Python as the underlying language glue, supplemented
>> by hard-coded modules for image, signal & file processing,
>> graphical I/O etc.? It seems that it can already do that in
>> some problem domains - e.g. Python Imaging Library.
>
> http://nickbower.com/computer/pydl is a first cut.
>
>> Put another way: What are the advantages of IDL compared to
>> such an open source framework?

>
> It depends on what your goal is.
>
> If your goal is to read, analyze, and visualize data for the purpose of
> earning a living or obtaining a degree, then I'd use IDL.
>
> If your goal is to have fun writing new code for which you don't get
> paid or earn any credits, then I'd roll your own open source solution in
> Python (if you can get someone to pay you for it, well done).

I've thought about this kind of question before too. I think there is another question:

If your need is to use one of the many library functions written in IDL, either already available in the IDL distribution, or available on the internet, then use IDL. Use and compatibility of existing libraries is a huge issue for me.

Craig

--
-- --------------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
-- --------------

Subject: Re: Sacrilegious but genuine question
Posted by Nigel Wade on Wed, 09 Jan 2002 10:28:20 GMT
View Forum Message <> Reply to Message

Francis Burton wrote:

> How easy would it be to obtain the functionality of IDL by
> using Python as the underlying language glue, supplemented
> by hard-coded modules for image, signal & file processing,
> graphical I/O etc.? It seems that it can already do that in
> some problem domains - e.g. Python Imaging Library.

That might provide some functionality for image processing. But what about all the other graphics/visualization capabilities of IDL? To emulate those you need to find other Python add-ons, each with its own peculiarities, build and install them, configure them to do what you want and maybe they'll come somewhere near what IDL can do out of the box.

>
> Put another way: What are the advantages of IDL compared to

> such an open source framework?

Support and functionality

(unless you own a Mac - but we're not going there again are we?)

What do you do when the author of your favorite Python add-on gets bored
with Python and decides to go on to do something else?

--

Nigel Wade, System Administrator, Space Plasma Physics Group,
University of Leicester, Leicester, LE1 7RH, UK
E-mail : nmw@ion.le.ac.uk
Phone : +44 (0)116 2523568, Fax : +44 (0)116 2523555

Subject: Re: Sacrilegious but genuine question
Posted by Mark Fardal on Wed, 09 Jan 2002 19:51:53 GMT

View Forum Message <> Reply to Message

Francis Burton <F.Burton@biomed.gla.ac.uk> writes:

> How easy would it be to obtain the functionality of IDL by
> using Python as the underlying language glue, supplemented
> by hard-coded modules for image, signal & file processing,
> graphical I/O etc.?

Not very easy. But easier than say, writing an open-source
operating system, and look what happened there...

Just don't try doing it by yourself. The Numpy and SciPy mailing
lists are places to start. At least on the Numpy list, the topic of
IDL emulation and translation has come up recently.

Mark

Subject: Re: Sacrilegious but genuine question
Posted by Andy Loughe on Wed, 09 Jan 2002 20:06:06 GMT

View Forum Message <> Reply to Message

Mark Fardal wrote:
>
> Francis Burton <F.Burton@biomed.gla.ac.uk> writes:
>
>> How easy would it be to obtain the functionality of IDL by

>> using Python as the underlying language glue, supplemented
>> by hard-coded modules for image, signal & file processing,
>> graphical I/O etc.?
>
> Not very easy. But easier than say, writing an open-source
> operating system, and look what happened there...
>
> Just don't try doing it by yourself. The Numpy and SciPy mailing
> lists are places to start. At least on the Numpy list, the topic of
> IDL emulation and translation has come up recently.
>
> Mark

Mmmm... maybe you should ask someone who has done it!

http://nickbower.com/computer/pydl

--
Andrew Loughe ===
NOAA/OAR/FSL/AD R/FS5 | email: loughe@fsl.noaa.gov
325 Broadway | wwweb: www-ad.fsl.noaa.gov/users/loughe
Boulder, CO 80305-3328 | phone: 303-497-6211 fax: 303-497-6301

Subject: Re: Sacrilegious but genuine question
Posted by Mark Fardal on Wed, 09 Jan 2002 21:45:42 GMT
View Forum Message <> Reply to Message

Andy Loughe <loughe@fsl.noaa.gov> writes:
> Mark Fardal wrote:
>> Just don't try doing it by yourself. ...
>
> Mmmm... maybe you should ask someone who has done it!
>
> http://nickbower.com/computer/pydl

Pydl does NOT give you "the functionality of IDL". It implements a
few commonly used IDL commands in Python. Probably 90% of all IDL
commands are missing, and those that are there are not necessarily
implemented completely. If you want to calculate and plot a few
things with a free package, without learning an entirely new syntax,
Pydl could be useful. If you want something as powerful as IDL, there
is a lot of work yet to be done.

Mark

Subject: Re: Sacrilegious but genuine question
Posted by rmw092001 on Thu, 10 Jan 2002 02:55:57 GMT
View Forum Message <> Reply to Message

"Liam E. Gumley" <Liam.Gumley@ssec.wisc.edu> wrote in message
news:<3C3B378C.F85C497E@ssec.wisc.edu>...
> Francis Burton wrote:
>> How easy would it be to obtain the functionality of IDL by
>> using Python as the underlying language glue, supplemented
>> by hard-coded modules for image, signal & file processing,
>> graphical I/O etc.? It seems that it can already do that in
>> some problem domains - e.g. Python Imaging Library.
>
> http://nickbower.com/computer/pydl is a first cut.
>
>> Put another way: What are the advantages of IDL compared to
>> such an open source framework?
>
> It depends on what your goal is.
>
> If your goal is to read, analyze, and visualize data for the purpose of
> earning a living or obtaining a degree, then I'd use IDL.
>
> If your goal is to have fun writing new code for which you don't get
> paid or earn any credits, then I'd roll your own open source solution in
> Python (if you can get someone to pay you for it, well done).
>
> Cheers,
> Liam.
> Practical IDL Programming
> http://www.gumley.com/

It's a pity RSI don't make a decent, inexpensive 'IDL Limited Edition'
- just file read/writing, numbercrunching, plotting etc - no widgets,
object graphics or industrial stuff. IDL 'student edition' restricts
array sizes so much it's almost useless. If pydl effectively becomes
IDL-LE, that would be great.

Richard

Subject: Re: Sacrilegious but genuine question
Posted by Francis Burton on Thu, 10 Jan 2002 10:15:29 GMT
View Forum Message <> Reply to Message

Andy Loughe wrote:
>
> Mark Fardal wrote:

>>
>> Francis Burton <F.Burton@biomed.gla.ac.uk> writes:
>>
>>> How easy would it be to obtain the functionality of IDL by
>>> using Python as the underlying language glue, supplemented
>>> by hard-coded modules for image, signal & file processing,
>>> graphical I/O etc.?
>>
>> Not very easy. But easier than say, writing an open-source
>> operating system, and look what happened there...
>>
>> Just don't try doing it by yourself. The Numpy and SciPy mailing
>> lists are places to start. At least on the Numpy list, the topic of
>> IDL emulation and translation has come up recently.
>>
>> Mark
>
> Mmmm... maybe you should ask someone who has done it!
>
>   http://nickbower.com/computer/pydl

Thank you for the pointer to PYDL, and to Liam Gumley who also mentioned it a couple of days ago.

While I think it is interesting and worthwhile project, it is really only a start (it's at version 0.1b2). It offers very limited functionality, the documentation looks nice but is riddled with errors, and some of the comments therein made me wince a bit. E.g. talking about READ_ASCII "If you don't use white-space delimiters, then use the UNIX command tr to create a white-space delimited version. Live with it." There is no reason why trivial misfeatures like that can't be fixed. And they probably will be if enough people who care about quality get involved.

What I think PYDL in its current state is valuable for is to show what is possible. What I have seen so far is encouraging.

Let me make it clear that I have nothing against IDL - apart from its high cost (putting it beyond the reach of individuals like me), the irksome licensing issues, the uncertainty about which platforms will be supported, the fact that it is rather difficulty to add functionality of one's own (see below), and its rather odd syntax (e.g. the comma between command name and arguments). I =do= like its very respectable performance and the fact that all the functionality is provided in a single package - though I don't see why PYDL (or similar framework) couldn't at some point be provided as a consolidated bundle.

I mentioned extensibility. What I would =really= like to be
able to do in IDL is have windows which allow much greater
interaction with plotted data. For instance, I have fairly
large (50,000-500,000 point) digitized signals which I would
like to 1) plot in an efficient manner (i.e. min-max for each
x pixel rather than drawing a line between every point) and
2) create and overlay various mousable widgets which return
values to IDL code. For example, I'd like to be able to click
an arbitrary number of points under the trace to define a
baseline for subtraction - and be able to adjust the position
of already defined nodes, but constrain them to be monotonically
increasing in x. In this case, the widget (if that is the right
name for it) would return a list of (x,y) values. Then the IDL
code could do the subtraction and replot the adjusted trace.

The trouble is that I have no idea how to do this in IDL. I'm
not even 100% sure that it is possible - at least, I believe
the effort required would be prohibitive. On the other hand,
a well designed framework of the type exemplified by PYDL
=could= make such extension quite easy.

Anyway enough rambling. Thanks to everyone who responded.

Francis