Subject: Re: Any cross-platform IDL alternatives?
Posted by George N. White III on Sat, 13 Oct 2001 13:08:49 GMT
View Forum Message <> Reply to Message

On Wed, 10 Oct 2001, Eric Williams wrote:

> Since RSI/Kodak is killing IDL for the Mac (and others) I am wondering
> if anyone has suggestions for trying out some other cross-platform
> software. I am not really worried too much about Windows but the
> Mac/*nix world is where I work. We use IDL mostly for processing
> astronomical data and writing procedures to play with the data
> visually. So a tool with a simple language and good graphing
> capabilities is what we really need.

There are open source tools whose capabilities substantially overlap those of IDL. Many of these tools can easily be ported to MacOS-X. Even if MacOS-X is heavily adopted in the technical community there will, for many people/tasks, be viable alternatives to IDL. Thus it is far from certain that sales of IDL for OSX would generate the required return on investment.

It is also worth noting that the cost of supporting multiple platforms involves more than just the cost of porting code. A big part of IDL's advantage over open source tools is performance, and that often means tuning algorithms to suit the hardware. If you want to support a wide range of hardware you end up either using more generic approaches that affect performance on all platforms or developing platform-specific implementations of the core algorithms. In the long run, IDL may need the sort of major transformation to core algorithms that was done by Mathworks when they moved from LINPACK to LAPACK for the linear algebra core.

I don't think any open source package compares with IDL in all aspects, but there are certainly packages with comparable capabilities for many specific tasks. With open source, however, there is no advertising budget, so you may have to invest considerable effort to track down an open source package that meets a particular set of requirements.

For "a simple language and good graphing capabilities" candidates include the R language: http://www.ci.tuwien.ac.at/~hornik/R/R-FAQ.html, Python: http://www.python.org/topics/scicomp/plotting.html, and the various Matlab clones.

> BTW, if anyone from RSI cares, I only have two Mac licenses but I have
> brought 31 Solaris licenses to my campus since coming here, with
> plenty more potential on the horizon. Once I have an alternative tool
> I will be recommending against purchasing anymore licenses and
> dropping maintenance. Too bad because I have been a huge evangelist
> for IDL over the years. I have turned more than 50 students and

> scientists onto IDL in the last few years which has resulted in most
> likely hundreds of licenses if not many more as they have moved onto
> other sites. That will all stop now and I will try to do the opposite,
> and probably save some money too.

Would Kodak/RSI and the current IDL user base be better off if the IDL command line interpreter was converted to open source? Kodak/RSI could sell systems like ENVI written in IDL, while the user community would port IDL (at least the command-line portions) to new platforms.

Organizations selecting tools for a large group of users were likely to favor allocating more resources to software (e.g., by purchasing big commercial packages like Matlab and IDL) in order to use less expensive hardware.

In general, people who use open source tools tend to buy hardware that provides the required performance using tools chosen for portability, while users of commercial tools are more likely to look for the tool that provides the required performance using more generic hardware (i.e., because a commercial vendor has invested considerable effort into optimizing algorithms for the target hardware).

Matlab and IDL are suffering from similar limitations in the range of platforms supported (e.g., Intel uniprocessors and a few unix platforms). This means the technical computing community is becoming fragmented between groups with problems that require high performance hardware and parallel processing, and groups that can use hardware that runs commercial tools. As a result, there are fewer opportunities to share code and algorithms between the communities.

Companies who understand open source can use it to their advantage by actively promoting third party ports of kernels to the full range of platforms, and can use these kernels to support high level commercial products that will be available to both communities.

--
George N. White III <gnw3@acm.org> Bedford Institute of Oceanography

---