Subject: Re: IDL performance and FFTs (was: call external speed)
Posted by steinhh on Thu, 17 Sep 1998 07:00:00 GMT
View Forum Message <> Reply to Message

In article <stevenj-1609981832310001@wetelectron.mit.edu>
stevenj@alum.mit.edu (Steven G. Johnson) writes:

[...]
> [steinhh@ulrik.uio.no (Stein Vidar Hagfors Haugan) wrote:]
>> 1. RSI could produce a wrapper for FFTW and make it available
>> through their web site so the user could make a dynamically loadable
>> module (and the user would have to fetch the FFTW separately). AFAIK
>> this would mean that FFTW is not "sold as a part of IDL".
>
> Nope, you can't get around the GPL in this way. To quote R. Stallman of
> GNU, "A GPL-covered plug-in that is designed to be combined with [only] a
> non-free master program is a form of combined work, and a violation of the
> GPL." (The original authors can make an exception allowing their code to
> be used in such a plugin, but no one else can do so.)
>
> (Otherwise, the GPL would essentially devolve to the LGPL--you could link
> any GPL'ed code you wanted into a non-free program just by making it a
> "plugin." For more info, do a search on Dejanews for: ~a
> (rms@santafe.edu) & ~g (gnu.misc.discuss) & "Plug-ins")
>
> You can make such a plugin for your own use, but you can't distribute it.
> (In any case, you are probably right in that such a plugin wouldn't be
> widely useful unless it came with IDL.)

I see the point that RSI cannot supply an operational plugin. Such a
plugin would appear as much a part of IDL as e.g. the jpeg/hdf/cdf
support routines, which are actually situated in dynamically loaded
modules (plugins), which *could* have been written by third parties.

What I was suggesting was (the unlikely scenario) that RSI
would write a short piece of C code that takes care of the type
checking, extracting the array sizes etc, before calling
FFTW functions (that are not supplied by RSI).

The user (or his/her system manager) would have to fetch the FFTW
code from the original web site (http://theory.lcs.mit.edu/~fftw/),
and make sure it got linked together with the piece of code supplied
by RSI.

I.e., RSI would only give instructions on, *how* to make a plugin
out of the GPL-covered code. The user would make the plugin,
by combining the instructions and the wrapper code.

This may be splitting hairs, but I could imagine myself writing such wrappers for FFTW (and indeed it appears that Karl Krieger has already done so) or other GPL-covered libraries.

Would I really be breaking the GPL licence by giving away *only* this:

1. C file containing calls to GPL-covered libraries
2. Instructions on how to get the GPL-covered library
3. Makefile that links my C file and the GPL-covered library into a plugin for IDL.

If so, I've nearly done a bummer with my regular expression DLM, since it's possible to use the GPL-covered regex package to provide the regcomp, regexec, regerror and regfree routines.

(And thanks for the tip on Fortran/C optimizations)

Regards,

Stein Vidar
(*No* expert on GPL!)
