
Subject: Re: IDL vs PV-WAVE

Posted by [thompson](#) on Fri, 26 Aug 1994 14:35:18 GMT

[View Forum Message](#) <> [Reply to Message](#)

tonyg@hdos.hac.com (Tony Gruszczak) writes:

> ... Many people I know use C or Fortran to do their heavy data processing
> (which is performed much faster in a compiled program than when done in IDL or
> PV-WAVE) then pass it on to the graphics package for its display capabilities.

(rest deleted)

It's not really **quite** true that C and Fortran are much faster than IDL or PV-Wave. It really depends on the problem. If it can be expressed in array notation, then IDL and its cousin PV-Wave perform quite competitively with C and Fortran. For example, I find no perceptible difference between IDL and Fortran in doing non-linear least-squares fits, even on slow machines like a MicroVAX II.

The kicker is usually loops. Procedures in IDL that require looping generally perform slowly because it's an interpretive language. I've often found that IDL novices who are more used to compiler languages use loops to do things that can be done more elegantly in IDL without them (I did it myself when I started out.) Admittedly, some of the tricks one can use in IDL to avoid loops are not exactly obvious, although most are quite straightforward and often easier to understand than the loops in compiler languages.

I haven't tried RPC, but I have used the reverse, which is to use `CALL_EXTERNAL` to call C and Fortran routines from IDL. That way, I can do most of the programming in IDL, which is easier than C or Fortran, and can make use of nice features like rapid prototyping and the IDL widget interface, and still use C or Fortran for when one has to do real number crunching. It makes more sense to me to have IDL as the front end, and the compiler languages as the back end. Of course, if one has a heritage system that one needs to integrate with a package like IDL, then it does make sense to use RPCs to do it the other way around.

Maybe the confusion has to do with the platform being used. RPCs, and their inverses `CALL_EXTERNAL` and `LINKIMAGE`, have been around for some time on many Unix and VMS platforms, I believe that it wasn't supported under Windows until more recently. Also, some platforms, like DEC Ultrix workstations, still don't support things like `CALL_EXTERNAL`.

Bill Thompson
