Subject: Re: Compiling IDL ... ever likey ?
Posted by thompson on Thu, 25 Jan 1996 08:00:00 GMT
View Forum Message <> Reply to Message

steinhh@amon.uio.no (Stein Vidar Hagfors Haugan) writes:

> The key to improving performance is declaring the type and
> dimensionality of the data that are to be manipulated. Very often,
> IDL subroutines are made to deal with very specific data,
> but there's no way to tell IDL about this -- it has to do all
> the checks all the time. In the survey about the future of IDL
> I suggested the possibility of having "pseudocode blocks", where
> all the data to be manipulated are declared in the beginning.
> If some of the input data do not match the declaration, a
> runtime error occurs.

Yeah, but then it wouldn't be IDL. You might as well write it in FORTRAN at that point, IMHO.

Almost all the IDL code that I write expects to be able to ingest data in a variety of data types and dimensionality. That's what I like about IDL, and a good part of why I use it.

People generally ask for IDL compilers for two reasons:

1. To be able to distribute IDL code without having to require other people to buy IDL. It was that possibility I was considering in my previous post. I think that it is perfectly possible to do this, and still let IDL be IDL.

2. To speed up execution time on tasks that cannot easily be vectorized (or which are not efficiently written). I don't see anyway of doing this without making fundamental changes in the way IDL works.

Just my $0.02 worth,

Bill Thompson