Subject: Re: call external speed
Posted by menakkis on Thu, 03 Sep 1998 07:00:00 GMT
View Forum Message <> Reply to Message

curtis volin <cev@U.Arizona.EDU> wrote:
> I use the IDL SVD command very frequently, so I was wondering if it would
> be any faster to create a DLL and call it with call external. I suppose
> that the fundamental question is, are IDL routines optimized for speed?

I haven't done SVD comparisons, but my opinion is that it would not be worth it to replicate this sort of IDL routine in C if you were implementing the *same* algorithm - you might come out worse off.

I assume that you know that SVD is a "native" IDL routine (meaning that it is not written in the IDL programming language, but rather in C already), but that you're wondering how well RSI does this kind of thing. If their array arithmetic is anything to go by, it looks like they are actually pretty good at it. I know of a case where an algorithm coded in C (without "source code level optimisations") compared pretty shabbily against the same thing coded in IDL (using array arithmetic fairly extensively).

Having said that, I do think that there is *some* room for improvement because I have been told that IDL for Windows/i386 is compiled for "pentiums", and if you have a Pentium II or Pro you can compile your stuff for "pentium pros" and realize a benefit. There might also be other compilation optimisations (like the "assume no aliasing" ones - big ones) that you can use, that might not have been used in IDL's build (?)

Funny that you should mention SVD, though. IDL uses the SVD algorithm from Numerical Recipes. There are those who react in a kind of derogatory way when in hearing distance of anyone who carelessly remarks: "I'm using the Numerical Recipes SVD...". Now personally I don't really know the virtues of one SVD implementation versus another (although I do believe that there are a couple of bugs in the NR version that are obscured by that peculiar array indexing they use), but there may be some benefit in coding a *different* implementation to IDL's, in C.

Peter Mason