Subject: Re: compile a routine wich inlude a commun
Posted by JD Smith on Tue, 24 Jan 2006 21:13:23 GMT
View Forum Message <> Reply to Message

On Tue, 24 Jan 2006 02:04:01 -0800, Maarten wrote:

> JD Smith wrote:
>
>> I think he means pointers are a kludge for extensible arrays.
>
> I think you can read minds, that is what I meant.

Not mind reading, just shared annoyance.

>> As one of the perpetrators of that page, I have to agree, these
>> examples (and many of the IDL Way) are not terribly obvious. Some
>> have maintenance concerns, to be sure. But, they enable you, in a
>> typeless language, to obtain the kind of speed of operation on large
>> (many MB to many GB) piles of data that are simply otherwise unheard
>> of.
>
> I think we disagree where the balance between readability and execution
> speed lies.

The balance, I find, depends quite sensitively on the reader.

>> Moreover, a elegant Python Drizzle would probably run 10x slower even
>> than the straightforward loop-based IDL drizzle. At some point, you
>> give up and write it quite simply in C, spending 95% of the time and C
>> code figuring out how to communicate the results back with IDL. So, I
>> agree with the original poster that the algorithms mentioned, among
>> many others in IDL, are not at all transparent, while simple versions
>> of the same are not at all fast. However, in my experience, this is
>> the price you pay in the tradeoff of elegance and raw speed.
>
> I prefer my "normal" everyday use to be elegant, and when I really need
> the speed (after profiling, no need for premature optimization), an
> easy route to C, Fortran, (assembly, no, I'm not that nuts) is
> appreciated.

Elegance is also fairly subjective, especially when it comes to IDL programming. After you immerse yourself in vector-based programming for long enough, the sight of a loop is highly unsettling, not just for the implicit speed penalty it may well entail, but in terms of perceived elegance as well. Whether anyone else in the world would share that aesthetic is debatable.

JD