Subject: Re: How does IDL do ...
Posted by Thomas A. McGlynn on Mon, 05 May 1997 07:00:00 GMT
View Forum Message <> Reply to Message

Peter Webb wrote:
>
> Eric Williams (ewilliams@wesleyan.edu) wrote:
>
> : I am also curious as to how IDL does matrix calculations. A simple example:
>
> : If you want to operate on an 2D array with FORTRAN you need to write
> : nested loops to work through the rows and columns and work with each
> : element.
>
> : In IDL you apply a function or WHERE statement to a whole array in one command.
> : I am wondering if IDL is still doing the nested loops anyway, and
> : therefore not really any faster at doing the job?
>
> Indeed, IDL is doing the nested loops internally (how could it not,
> unless you have more than one processor).

Actually this is not really correct and if you assume it you may
fall into a trap that I've hit more than once. E.g.,
suppose you have something like:

```
x = intarr(6) + 1
y = intarr(6)
y(x) = y(x) + 1
```

If you are thinking of IDL as simply hiding loops you may expect
to get y(1)=6 from these statements. In fact you get y(1)=1. IDL acts as if
the vector operations are occuring in parallel, even on a non-parallel
machine. In terms of computing efficiency, I essentially agree
with the statements that follow, though I imagine that there are cases
where IDL's loop operations will be a little faster than Fortran's.

> Its speed on such operations
> is comparable to the explicit loop code in C or FORTRAN. If you
> do the explicit loop in IDL, it will be very slow.
>
> So, you don't get to go faster than FORTRAN. On the other hand, you
> don't often go much slower. And your program will be working a week
> before that other guys!
...
> Peter

Regards,
Tom McGlynn
tam@silk.gsfc.nasa.gov