Subject: Re: Avoiding FOR loops (version googleplex.infinity)
Posted by Tom McGlynn on Fri, 11 Apr 2008 15:04:09 GMT
View Forum Message <> Reply to Message

On Apr 11, 2:15 am, Gaurav <selfishgau...@gmail.com> wrote:
> Dear Tom,
> The trouble is that coming into this group, I always feel I am a
> novice with IDL programming and algorithm designing. Hence, my
> reluctance in asserting my views.
>
> For example, my perusal of IDL help documents has led me to the belief
> that creating temporary arrays help hasten things up while you mention
> something to the contrary. I shall definitely look this up and would
> appreciate pointers in the right direction.
>
> As far as your code generating a result different from mine at the
> edges is concerned, that-of course-is true. In that respect, my code
> produces the same effect as the normal FOR loop, and this has been
> verified.
>
> And as for the bug in my code, I definitely stand corrected for my
> code would fail if all the array elements in the kernel are different.
> Thanks indeed for pointing it out to me.
>
> Thanks indeed,
> Gaurav

Dear Guarav,

I learn something new in almost every discussion I participate on in this group -- and that's ignoring all of the stuff on IDL objects which I promptly forget. Don't assume that just because we're willing to pontificate we really know what we're talking about. With that caveat...

With regard to the use of temporaries, the big savings -- the factors of 100 -- come when you can replace some inner loop with an array expression.
E.g., we don't want to have 100,000,000 iterations over all of the pixels of your 10,000x10,000 images. However computing arrays has some cost in both CPU and memory footprint: so you don't want to create them willy nilly. If could just reuse existing arrays, then maybe that will be your best shot.

One thing that I didn't consider a couple of messages up is that a simple array expression

```
x = y + z
```

may be somewhat faster than a bounded array expression

```
x[1:10] = y[0:9]+z[21:29]
```

so it may be -- I don't know -- that creating temporaries so that you can use a simple expression will be a net benefit. It's the cost of building the new arrays versus the cost of the more complex computation. But I think differences are going to be quite modest -- factors of 2 at best I'd think. I suspect the only way to tell if the benefit of being able to simpler array operations outweighs the cost of building intermediate temporary arrays is to try it out with your hardware and your problem.

Regards,
Tom McGlynn
