Subject: Re: FOR loops removal
Posted by Jeremy Bailin on Wed, 20 Aug 2008 11:50:27 GMT
View Forum Message <> Reply to Message

On Aug 19, 9:43 am, Wox <nom...@hotmail.com> wrote:
> On Tue, 19 Aug 2008 05:38:50 -0700 (PDT), loebasboy
>
> <stijn....@gmail.com> wrote:
>> FOR l = 0, n*2 DO BEGIN
>> temp = 0
>> FOR i =0,max_y-1 DO BEGIN
>> FOR j=0,max_x-1 DO BEGIN
>> jtemp = j + l
>> jtemp2 = j + n
>> temp = temp + (arr[i,jtemp] * arr [i,jtemp2])
>> ENDFOR
>> ENDFOR
>> output[l] = temp/(max_x*max_y)
>> ENDFOR
>
> The code below is a start. Does this processing have a name? It feels
> familiar somehow. Btw, in IDL the first index of an array is the
> column and the second is the row. So in your case y are the columns
> and x are the rows. No problem with that off course, just check
> whether this is how you intended it.
>
> n = 8
> max_x = 5
> max_y = 5
> output = fltarr(2*n+1)
> arr = findgen(max_y, 2*n+max_x) +1
>
> arr2=arr[0:max_y-1,n:max_x-1+n]
> FOR l = 0, 2*n DO $
> output[l] = total(arr[0:max_y-1,l:max_x-1+l]*arr2)
> output/=max_x*max_y

Following on that last version, I think we can *completely* get rid of
the loop... though at the expense (as usual) of memory:

```
n = 8
max_x = 5
max_y = 5
arr = findgen(max_y, 2*n+max_x) +1

max_area = max_x*max_y
output = total( arr[rebin(lindgen(max_area),max_area,2*n+1) +
  max_y*rebin(reform(lindgen(2*n+1),1,2*n+1),max_area,2*n+1)] *
```

rebin((arr[*,n:max_x-1+n])[*], max_area,2*n+1), 1) / max_area

Whether that's actually faster will depend on how big max_x, max_y and n are, of course... it ends up internally storing a couple of max_x*max_y*(2*n+1) arrays, so if that is going to take you into swap then you're best off sticking with Wox's version. If that stays in physical memory, though, I bet this will win.

-Jeremy.
