Subject: FOR loops removal
Posted by loebasboy on Tue, 19 Aug 2008 12:38:50 GMT
View Forum Message <> Reply to Message

Dear all,

Some weeks ago I've noticed that IDL is rather slow with FOR loops the hard way...after reading about it and not believing it, offcourse.
Last week I started removing some FOR loops in my code, apart from some embarasing ones like:

```
sum = 0
FOR i=0,max_y-1 DO BEGIN
  FOR j=0,max-1 DO BEGIN
    sum_int = sum_int + data[i,j]
  ENDFOR
ENDFOR
```

which has a straightforward solution. However I have some other FOR loops which aren't that obvious at all. Like for instance this one:

```
    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
```

which I could alter, not that straigthforwardly into:

```
    z = size(arr)
    arr = reform(in_arr, z[1]*z[2], /overwrite)
    endt = (max_y*max_x)-1
    FOR l = 0, n*2 DO BEGIN
      temp  = 0
      FOR i=0,endt DO temp = temp + (arr[i+l*max_y] * in_arr [i
+n*max_y])
      output[l] = temp/(max_x*max_y)
    ENDFOR
    in_arr = reform(in_arr, z[1],z[2], /overwrite)
```

where 1 FOR loop is removed. However there is hardly any time profit at all. It is even so that the following code is faster than both,

which is a very straightforward alteration of the first loop:

```
      FOR l = 0, n*2 DO BEGIN
        temp  = 0
        FOR i =0,max_y-1 DO FOR j=0,max_x-1 DO temp = temp + (arr[i,j
j + l] * arr [i, j + n])
        output[l] = temp/(max_x*max_y)
      ENDFOR.
```

With the following variables set and the for loops repeated with another FOR loop of i= 0,10000 (to see a time difference, and in the full program it is repeated about that many times too, but with even larger arrays):

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

I have for the first for loop : 1.6279998 s
the second : 1.6060002 s
the third : 1.2749999 s

I measured the times with SYSTIME, /SECONDS command.

(the full program takes 22,5 h for a standard image, with the alterations I have allready done, I've came up with 18.1 h, which is still 17 h to go to make it workable, i've used the last loop in the above example so far...)

Can anybody tell me why removing one loop doesn't help in this case or what i'm doing wrong?

thnx
Stijn Van der Linden
