Subject: Re: Writing an efficient array operation in IDL
Posted by Ammar Yusuf on Thu, 03 Feb 2011 03:37:18 GMT
View Forum Message <> Reply to Message

On Feb 2, 8:14 pm, Rony K Varghese <ronykvargh...@gmail.com> wrote:

```
>  Dear ALL,
>
>  Suppose, I have to create an output array B filled with value 8 as
>  follows, from the input 5x5 array A.
>
>  A  = 0 0 4 0 0
>       0 4 0 4 0
>       4 0 0 0 4
>       0 4 0 4 0
>       0 0 4 0 0
>
>  B =  0 0 8 0 0
>       0 8 8 8 0
>       8 8 8 8 8
>       0 8 8 8 0
>       0 0 8 0 0
>
>  The logic is to fill the whole region bounded by value 4 in the input
>  array, with the value 8 in the output array.
>  How i can do this efficiently in IDL, like using array operations?
>  I need to apply for a large sized array.
>  Thanks in advance..
>
>  - Rony
```

So what you're doing is going through each row and finding the first 4
and replacing it with 8 until you hit the next four right?

I really can't think anything that will do it one vectorized call but
you can try this I guess.

```
for each row
  use temp = where(to find all elements gt 0 or eq 4)
  for that row arr[i,*] do this
  if n_elements(temp) eq 1 then arr[i,temp] = 8
  arr[i,temp[i]:temp[n_elements(temp-1)] = 8
  go to next row.
endfor
```

I think that might work if you understand it. I would try it but I
don't have IDL the computer I'm using.

The other pseudocode algorithm I thought of but will definitely be slower is this:

Make a double for loop and for each row find the first index and after you find
that then start from the back of the row and find the last index. Use some logic there
when the case is you only have one element in the row. Use those two indicies to
replace the values.

Hope that helps.