Subject: Re: Cyclic array interfaces
Posted by cedricl on Tue, 13 Jul 2004 12:59:04 GMT
View Forum Message <> Reply to Message

Since you're aiming for optimization, I might as well point out an obvious one:

> A=[[1,2,3],[4,5,6]]
> B=[7,8,9]
> A[*,1]=B

Use A[0, 1] = B instead, since the elements are contiguous in the first dimension. You might also want to take a loot at "temporary" if you haven't already.

> (e.g. Does IDL actually rewrite the entire array, regardless of how many
> elements are being changed,

No.

> or does it only change the particular elements
> and hence the first example should be about twice as fast as the second)?

It doesn't work that way, like the previous poster has shown. Your performance mostly depends on the type of access that you're doing. If you're accessing memory in a continuous fashion (the first dimension varies), it's almost sure IMO that it's going to be faster.
Furthermore, consider the next example:

```
a = fltarr(5)
; Option 1: Create a new array
a = [3, 4, 5, 6, 7]
; Option 2: Rewrite the array
a[0] = [3, 4, 5, 6, 7]
```

Option 1 and 2 are equivalent, code-wise, but I would argue (and of course, profile, were I not lazy) that option 2 is faster, because it doesn't have to deallocate the array that was already contained in a, and allocate a new one. Of course, there are many other optimizations that IDL could do that would make 1 faster than 2, but considering RSI's approach to compile-time optimizations, I'm confident in my analysis.

Cedric