Subject: Re: simple vectorizing problem
Posted by dpm314 on Wed, 10 Sep 2008 23:53:43 GMT
View Forum Message <> Reply to Message

On Sep 9, 4:58 am, FÖLDY Lajos <fo...@rmki.kfki.hu> wrote:
> On Tue, 9 Sep 2008, Chris wrote:
>> Looks like the memory allocation overhead associated with
>> vectorization is worse than looping:
>
>> vectorization 1 time: 5.0360808 s
>> vectorization 2 time: 4.8453941 s
>> looping itme: 4.0875931 s
>
>> The vectorization performances become comparitvely worse as the array
>> grows (both reform() and total() create temporary, large arrays).
>
>> chris
>
>> pro test
>
>> nimage = 1500
>> im = fltarr(512,512,nimage)
>
>> t0 = systime(/seconds)
>> mean=total(reform(im,512L*512L, nimage), 1)/(512.*512)
>> t1=systime(/seconds)
>
>> mean=total(total(im,1),1)/(512.*512.)
>
>> t2 = systime(/seconds)
>
>> for i=0, nimage-1, 1 do $
>> mean[i]=total(im[*,*,i])/(512.*512)
>
>> t3=systime(/seconds)
>
>> print, t1-t0
>> print, t2-t1
>> print, t3-t2
>
>> end
>
> try this:
>
> t0 = systime(/seconds)
> mean=total(reform(im,512L*512L, nimage, /overwrite), 1)/(512.*512)
> ; ^^^^^^^^^^^^
> t1=systime(/seconds)

>
> regards,
> lajos

Hi everyone and thanks a lot for your suggestions so far. That is a cute trick with the Total function, and I was just not aware that you could specify to total only rows or columns like that. This will work, and I have tried it but didn't get a chance to do the timing test.
However, my first post must not be clear, because this doesn't really answer my question.

I have these image arrays and need to perform several functions on them which return a scalar (or a 6 element array) for each image in a large stack. Is there *in general* a way to so something like:

results = fltarr(num_images)
results = myFunctThatDoesSomeAnalysis(image)
;now results holds an array of scalars for the result of myFunctThatDoesSomeAnalysis() on each image

without either using a loop here or inside the analysis function iterating over each frame in the image?

again, I've tried many things like:

a = indgen(num_image)
results(a) = myFunctThatDoesSomeAnalysis(image(*,*,a))
which I thought would work but it does not.

It is necessary to use a loop like this then in general :
for i 0, num_images-1 do results(i) =
myFunctThatDoesSomeAnalysis(image(*,*,i))

???

I thought the trick with an indexing array ('a' in the example above) would work because if I say
b = indgen(101)/(!pi/100)
print, cos(b)

I get not one number out but an array of cos() evaluated from 0 to pi

I guess I don't understand how the cos() example gives me an array back, and
myFunctThatDoesSomeAnalysis(image(*,*,a)) gives me just one number.

If this question makes no sense I apologize, and I could provide more

code to illustrate the problem if that would help. I am still trying
to wrap my head around vectorizing code, which (I at least) find quite
different from purely procedural languages like c and Fortran which
I've written in for years.

Thanks,
David M.
