Subject: Re: An optimisation question
Posted by Yngvar Larsen on Tue, 27 Mar 2012 09:35:37 GMT
View Forum Message <> Reply to Message

On Tuesday, 27 March 2012 04:33:08 UTC+2, Bogdanovist wrote:

> Here is a test code snippet isolating this conundrum. In this case the
> REFORM approach is only 5 times slower, so not as bad as my example,
> but why is it slow at all? Surely the FOR loop is not the optimal
> approach here!
>
> pro test_foo_1,basis,lat,lon,ret
> ret = (reform(basis[0,*,*]))[lat,lon]
> end
>
> pro test_foo_2,basis,lat,lon,ret
> for i=0,9999 do $
> ret[i]=basis[0,lat[i],lon[i]]
> end
>
> pro test_foo
> Basis = fltarr(10,1000,1000)
> lat = fltarr(10000)
> lon = fltarr(10000)
>
> ret = fltarr(10000)
>
> test_foo_1,basis,lat,lon,ret
> test_foo_2,basis,lat,lon,ret
>
> end

Hm. Interesting. I've noticed before that one-liner 1D loops are quite fast in IDL. I also cannot explain why your TEST_FOO_1 is so slow since REFORM seems not to be the culprit. However, you can do it much more efficiently like this:

```
pro test_foo_1,basis,lat,lon,ret,slice
  ret[0] = basis[slice+lonarr(n_elements(lat)),lat,lon]
end
```

Profiler report:

```
IDL> profiler & profiler, /system
IDL> for n=0,99 do test_foo
IDL> profiler, /report
Module          Type  Count     Only(s)   Avg.(s)     Time(s)   Avg.(s)
FLTARR           (S)    400    1.646489  0.004116    1.646489  0.004116
LONARR           (S)    100    0.001611  0.000016    0.001611  0.000016
```

```
N_ELEMENTS      (S)     100    0.000101  0.000001    0.000101  0.000001
PROFILER        (S)       2    0.000193  0.000097    0.000193  0.000097
TEST_FOO        (U)     100    0.001783  0.000018    1.917648  0.019176
TEST_FOO_1      (U)     100    0.015138  0.000151    0.016837  0.000168
TEST_FOO_2      (U)     100    0.252603  0.002526    0.252603  0.002526
```

PS: In your test programs, LAT and LON are used as index arrays, but declared as floating point arrays. Not that it matters here though.

--
Yngvar
