Subject: Re: chunk indexing like
Posted by rogass on Sat, 09 Jan 2010 09:37:18 GMT
View Forum Message <> Reply to Message

Hi, Paolo indirectly shows that loops are sometimes not evil ;) I
tried this one:

```
pro cr_test_chunk_indexing,n=n,verbose=verbose
n= keyword_set(n)? n : [2,3,1,0,5]

if keyword_set(verbose) then print,n

t0=systime(1)
l1=((l=lindgen(((nm=n_elements(n)>(max(n)))),nm) mod nm))[where((l lt
transpose(rebin(n,nm,nm,/sample))))]
print,'CR-Rebin/mod-approach: ',systime(1)-t0,' seconds
for ,',n_elements(n), ' elements'
if keyword_set(verbose) then print,l1

t0=systime(1)
h=histogram(total(n>0,/CUMULATIVE,/int)-1,/
BINSIZE,MIN=0,REVERSE_INDICES=ri )
nh=n_elements(h)
chinkind=ri[0:nh-1]-ri[0]
ind2=where([1,chinkind[1:*]-chinkind[0:nh-2]] ne 0)
l1=lindgen(nh)-ind2[chinkind]
print,'Histogram-approach: ',systime(1)-t0,' seconds for ,',n_elements
(n), ' elements'
if keyword_set(verbose) then print,l1

t0=systime(1)
res=intarr(total(n))
indexarr=findgen(max(n))
i2=0
FOR i=0,n_elements(n)-1 DO BEGIN
  IF n[i] GE 1 THEN BEGIN
    res[i2:i2+n[i]-1]=indexarr[0:n[i]-1]
    i2=i2+n[i]
  ENDIF
ENDFOR
print,'Loop-approach: ',systime(1)-t0,' seconds for ,',n_elements(n),
' elements'
if keyword_set(verbose) then print,res

end
```

...and got this one:

```
IDL> cr_test_chunk_indexing, n=round(((i=10))*randomu(seed,i)),/
verbose
          9          6          3          0          6
1          2          8          7          5
Rebin/mod-approach:       0.00000000 seconds for ,          10
elements
          0          1          2          3          4
5          6          7          8          0
1          2          3          4          5
          0          1          2          0          1
2          3          4          5          0
0          1          0          1          2
          3          4          5          6          7
0          1          2          3          4
5          6          0          1          2
          3          4
Histogram-approach:       0.00000000 seconds for ,          10
elements
          0          1          2          3          4
5          6          7          8          0
1          2          3          4          5
          0          1          2         -6         -5
-4         -3         -2         -1         -1
-2         -1         -8         -7         -6
         -5         -4         -3         -2         -1
-7         -6         -5         -4         -3
-2         -1          0          1          2
          3          4
Loop-approach:       0.00000000 seconds for ,          10 elements
      0      1      2      3      4      5      6      7
8      0      1      2      3      4      5      0
1      2      0      1      2      3      4
      5      0      0      1      0      1      2      3
4      5      6      7      0      1      2      3
4      5      6      0      1      2      3
      4
IDL> cr_test_chunk_indexing, n=round(((i=100))*randomu(seed,i))
Rebin/mod-approach:     0.0010001659 seconds for ,         100
elements
Histogram-approach:       0.00000000 seconds for ,         100
elements
Loop-approach:       0.00000000 seconds for ,         100 elements
IDL> cr_test_chunk_indexing, n=round(((i=1000))*randomu(seed,i))
Rebin/mod-approach:      0.050000191 seconds for ,        1000
elements
Histogram-approach:      0.017000198 seconds for ,        1000
elements
Loop-approach:     0.0049998760 seconds for ,        1000 elements
```

IDL> cr_test_chunk_indexing, n=round(((i=10000))*randomu(seed,i))
Rebin/mod-approach: 3.9480000 seconds for , 10000 elements
Histogram-approach: 2.0690000 seconds for , 10000 elements
Loop-approach: 0.46099997 seconds for , 10000 elements

So, sometimes histogram method fails. Loop is here the fastest one and my one is the slowest but right one without loops.

JD? Any improvements? :)

Regards

CR
