Subject: Re: help on avoiding a FOR loop
Posted by Jeremy Bailin on Tue, 19 Jan 2010 17:35:42 GMT
View Forum Message <> Reply to Message

On Jan 18, 9:45 am, steffenh <hlde...@gmx.de> wrote:
> Hi there,
>
> maybe someone can help me on avoiding this FOR loop:
>
> ha=histogram(myBin.event, binSize=1, reverse_indices=r_acc, min=0)
>
> FOR i=0L, n_elements(ha)-1 DO BEGIN
> IF (ha[i] ne 0) THEN BEGIN
> idx=reverse_indices(ha, r_acc, i)
> myBin2[idx].energy=total(myBin[idx].energy, /CUMULATIVE)/
> total(myBin[idx].energy)
> ENDIF
> ENDFOR
>
> To explain: I have a dataset, which contains multiple energy entries
> which can be linked to individual events. The energies should be
> comulatively summed for each event. Each event spreads over roughly
> 10-100 energy entries and I am looping in excess of 1 Mio. events. For
> all what I know this is "sub-optimal" in IDL, since I'm actually doing
> very little processing in each loop-iteration.
>
> Cheers and thanks in advance,
>
> Steffen

How's this:

```
h1 = histogram(myBin.event, binSize=1, reverse_indices=ri, min=0)
nh1 = n_elements(h1)
t1 = total(myBin.energy[ri[ri[0]:ri[nh1]-1]], /cumulative)
startpts = total(h1,/cumulative,/int) - 1
t1_droppts = t1[startpts] - [0,t1[startpts]]
h2 = histogram(total(h1,/cumulative)-1, bin=1, min=0,
reverse_indices=ri2)
nh2 = n_elements(h2)
i2 = ri2[0:nh2-1]-ri2[0]
denom = t1_droppts[i2]
enew = myBin[ri[ri[0]:ri[nh1]-1]].energy / denom
enew[startpts[0:nh1-2]+1] -= 1.
myBin2[ri[ri[0]:ri[nh1]-1]].energy = total(enew, /cumulative)
```

Explanation: You're really just taking a cumulative sum that gets
reset every bin and dividing it by a different denominator in every

bin. So I use the histogram chunk indexing trick to create an array of denominators, and then subtract 1 at the beginning of every bin so that the cumulative total effectively gets reset.

Incidentally, this should also solve Wox's problem of a few days ago.

-Jeremy.
