Subject: Re: How can I optimize this?
Posted by Spon on Thu, 18 Sep 2008 15:46:13 GMT
View Forum Message <> Reply to Message

On Sep 18, 10:46 am, hlde...@gmx.de wrote:

```
>  Hi,
>
>  I'm looking for a way of optimizing a chunk of IDL code. The following
>  code snippet is part of a larger program. What it does is basically
>  test in some Anti-Coincidence Tables if a certain threshold energy is
>  met. Those AC-Events which meet the condition have their frame times
>  compared with my actual event list. If a AC frame equals an event
>  frame the event frame is thrown away. I've tried doing it with array
>  operations but always end up of having the problem that the AC and
>  frame array have different lengths, so just selecting via WHERE
>  doesn't work. Any help is very much appreciated since this part really
>  slows the whole program
>
>  ;four different AC sets, each with each own threshhold energy
>
>  ACdeleteIDs=[WHERE((myFitsAC.AC EQ 0) AND (myFitsAC.PI GT
>  ACEnergies[i])),WHERE((myFitsAC.AC EQ 1) AND (myFitsAC.PI GT
>  ACEnergies[j])),WHERE((myFitsAC.AC EQ 2) AND (myFitsAC.PI GT
>  ACEnergies[k])),WHERE((myFitsAC.AC EQ 3 AND myFitsAC.PI GT
>  ACEnergies[l]))]
>
>  ;the AC events which are above threshold
>  ACdeletes=myFitsAC[ACdeleteIDs]
>
>  ;keep the original data for the next loop
>  myFitsHED2=myFitsHED
>  myFitsLED2=myFitsLED
>
>  ;filter event list
>  FOR z=0L, n_elements(ACdeletes)-1 DO BEGIN
>
>        del=WHERE(myFitsHED.time EQ ACdeletes[z].time, cnt)
>        IF cnt GT 0 THEN myFitsHED2[del].time=-1
>
>        del=WHERE(myFitsLED.time EQ ACdeletes[z].time, cnt)
>        IF cnt GT 0 THEN myFitsLED2[del].time=-1
>
>  ENDFOR
>
>  myFitsHED2=myFitsHED2[WHERE(myFitsHED2.time NE -1)]
>  myFitsLED2=myFitsLED2[WHERE(myFitsLED2.time NE -1)]
```

Hi,

Firstly, make sure you're not getting -1 values in your array with multiple where-calls.
Making sure you have 'compile_opt strictarrsubs' at the start of your procedure is one way of avoiding this problem.

I would also add something like:

```
>  ACdeleteIDs=[WHERE((myFitsAC.AC EQ 0) AND (myFitsAC.PI GT
>  ACEnergies[i])),WHERE((myFitsAC.AC EQ 1) AND (myFitsAC.PI GT
>  ACEnergies[j])),WHERE((myFitsAC.AC EQ 2) AND (myFitsAC.PI GT
>  ACEnergies[k])),WHERE((myFitsAC.AC EQ 3 AND myFitsAC.PI GT
>  ACEnergies[l]))]
```

```
If Min(ACdeleteIDs) Eq -1 Then $
  ACdeleteIDs = ACdeleteIDs[WHERE(ACdeleteIDs NE -1)]
```

Next, this is how I vectorised your for-loop, which might speed things up a little:

```
; FOR z=0L, n_elements(ACdeletes)-1 DO BEGIN
;
;        del=WHERE(myFitsHED.time EQ ACdeletes[z].time, cnt)
;        IF cnt GT 0 THEN myFitsHED2[del].time=-1
;
;        del=WHERE(myFitsLED.time EQ ACdeletes[z].time, cnt)
;        IF cnt GT 0 THEN myFitsLED2[del].time=-1
;
; ENDFOR

 ; Take the values out of structure fields into
 ; variables for the sake of sanity
ACDTimes = ACdeletes.time
mFHTimes = myFitsHED.time
mFLTimes = myFitsLED.time
NA = n_elements(ACdeletes)
NF = n_elements(mFHTimes)

 ; Make big 2d arrays for vectorisation
ACDTimes = Rebin(ACDTimes, NA, NF)
mFHTimes = Rebin(Transpose(mFHTimes), NA, NF)
mFLTimes = Rebin(Transpose(mFLTimes), NA, NF)

 ; Call WHERE on the 2d arrays
delhi = Where(mFHTimes Eq ACDTimes, cnthi)
dello = Where(mFLTimes Eq ACDTimes, cntlo)
```

```
 ; Replace values into structure fields
If cnthi NE 0 Then Begin
  mFHTimes[delhi] = -1
   ; Get the minimum over the 1st dimension:
   ; if we hit a -1 in any column, use it;
   ; otherwise the column elements will
   ; all be identical anyway
  HiFoo = Min(mFHTimes, Dim = 1)
  myFitsHED2.time = HiFoo
EndIf

If cntlo NE 0 Then Begin
  mFLTimes[dello] = -1
  LoFoo = Min(mFLTimes, Dim = 1)
  myFitsLED2.time = LoFoo
EndIf
```

Hope this helps some,
Regards,
Chris