
Subject: Re: IDL 8.2.2 released

Posted by [lecacheux.alain](#) on Wed, 20 Mar 2013 16:07:23 GMT

[View Forum Message](#) <> [Reply to Message](#)

Le mercredi 20 mars 2013 10:58:17 UTC+1, Tom Grydeland a écrit :

> On Thursday, February 7, 2013 12:09:57 AM UTC, Mark Piper wrote:

>

>> I love the fact that people care enough about IDL (like I do) to post comments on this thread!
Please keep them coming. I'll try to respond to each comment in a critical and responsible manner.

>

>>

>

>>

>

>

>

> Hello,

>

>

>

> I heard that plotting speed had improved in 8.2.2 and decided to give it a go.

>

>

>

> Building up plots incrementally is still painfully slow. The routine below creates an illustration of a phased array antenna where all elements are aligned radially or tangentially. On a reasonable computer, building the plot takes an unreasonable 44 seconds to complete. (linux x86_64 m64)

>

>

>

> I am profiling the call, and getting some very interesting statistics:

>

>

>

> Hit count Time self(s) Time+sub(s) Sys Routine

>

> 292577 1.0121676922 1.0121676922 1 IDLGRPLOT::GETPROPERTY

>

> 1311191 2.1339576244 2.1339576244 1 IDLITCOMPONENT::GETPROPERTY

>

> 116646 1.2261884212 10.1981632710 0 IDLITVISPLOT::GETXYZRANGE

>

> 560278 2.1338310242 2.1338310242 1 IDL_CONTAINER::GET

>

> 619133 0.7483308315 0.7483308315 1 MIN

>

> 297796 1.9745259285 3.3823873997 0 _IDLITCONTAINER::GET

```

>
> 259440 5.5053386688 11.2065567970 0 _IDLITCONTAINER::_GETIDENTIFIERS
>
> 256828 1.5982770920 1.9980626106 0 _IDLITVISUALIZATION::GETPROPERTY
>
> 118096 3.9485795498 14.2965276241 0 _IDLITVISUALIZATION::GETXYZRANGE
>
> 122931 1.0249559879 1.3674829006 0
_IDLITVISUALIZATION::SEEKPIXELATEDVISUALIZATION
>
> 233776 4.5677740574 5.8650910854 0
_IDLITVISUALIZATION::_ACCUMULATEXYZRANGE
>
>
>
> So the time spent per routine call isn't bad, but the hit counts ...
>
>
>
> No less than 26 routines are being called more than 90 thousand times each.
>
>
>
> 12 routines take up more than a second of accumulated time each, totalling over 28 seconds
between them.
>
>
>
> We're talking about 121 points and 240 lines here. Not exactly staggering.
>
>
>
>> Thanks,
>
>> mp
>
>> IDL Product Manager
>
>> mark.piper@exelisvis.com
>
>
>
> Cheers,
>
>
>
> --T
>

```

```

>
>
>
>
>
>
>
>
> function radial_antenna, n, length=length
>
>
> if n_elements(length) eq 0 then length = 0.9
>
>
>
> ii = indgen(n)-(n-1.)/2
>
> xx = ii[* ,ii]
>
> yy = transpose(xx)
>
> cg = plot(xx[*], yy[*], 'D', aspect_ratio=1)
>
> ;; cg.refresh, /disable
>
>
>
> nx = n_elements(xx)
>
> if nx mod 2 then begin
>
>   ;; if there is an element in the centre, eliminate it
>
>   tmp = [indgen(nx/2), nx/2+1+indgen(nx/2)]
>
>   xx = xx[tmp]
>
>   yy = yy[tmp]
>
> endif
>
> th = atan(yy, xx)
>
>
>
> for ii=0L, n_elements(xx)-1 do begin
>

```

```

>   cx = length/2*[-1, 1]*cos(th[ii])
>
>   cy = length/2*[-1, 1]*sin(th[ii])
>
>   !null = plot(xx[ii]+cx, yy[ii]+cy, 'k', /current, /overplot)
>
>   !null = plot(xx[ii]-cy, yy[ii]+cx, 'r', /current, /overplot)
>
>   endfor
>
>
>
>   cg.refresh
>
>   return, cg
>
> end
>
>
>
>
>
> ;; main routine
>
> profiler, /reset
>
> profiler, /system
>
> profiler
>
>
>
> cg = radial_antenna(11)
>
>
>
> profiler, /report, filename='report.txt'
>
>
>
> end

```

You should not call the NG plot function in a loop. The POLYLINE function with using CONNECTIVITY keyword is much better in your case.
 For instance, replacing your:

```

for ii=0L, n_elements(xx)-1 do begin
  cx = length/2*[-1, 1]*cos(th[ii])
  cy = length/2*[-1, 1]*sin(th[ii])
  !null = plot(xx[ii]+cx, yy[ii]+cy, 'k', /current, /overplot)
  !null = plot(xx[ii]-cy, yy[ii]+cx, 'r', /current, /overplot)
endfor

```

by :

```

cx = fltarr(2,n_elements(xx)) & cy = cx
conn = lonarr(3,n_elements(xx))
for ii=0L, n_elements(xx)-1 do begin
  cx[* ,ii] = length/2*[-1, 1]*cos(th[ii])
  cy[* ,ii] = length/2*[-1, 1]*sin(th[ii])
  conn[* ,ii] = [2,2*ii,2*ii+1]
endfor
nx2 = 2*N_elements(xx)
tic
!null = polyline(reform([1,1]#xx + cx, nx2), reform([1,1]#yy + cy, nx2), 'k', CONNECTIVITY=conn,
TARGET=cg, /DATA)
!null = polyline(reform([1,1]#xx - cy, nx2), reform([1,1]#yy + cx, nx2), 'r', CONNECTIVITY=conn,
TARGET=cg, /DATA)
toc

```

the execution time goes from 36.1 sec. on my machine down to 0.031 sec.
A factor 1000 which has to be explained by Exelis...

alain.
