Subject: IDL new graphics memory leak?
Posted by belkaraza on Tue, 04 Oct 2016 11:56:49 GMT
View Forum Message <> Reply to Message

Hey,

I have some enourmous problem with the amound of memory (virtual) IDL is using. I believe they are resulting from my extensive use of plots and images using the plot/image functions from IDL. If I omit them from my code my memory usage is on a normal level (1-8gb). If I want to include plots and images then my usage of memory will grow till it overloads the cluster I am working on (200-400 gb). Right after saving them I destroyed the objects via obj_destroy. This didn't solved my problem so I used heap_free. Still no improvment. Is there anything I am missing here? I am using IDL 8.3 on linux 64 bit.
rough sketch of my program structure:

```
read file (big image)

For
 For
   data analysis
   Img=image(...)
   img.save,....
   destroy img
   plot=plot()...
   ...
endfor
endfor
end
```

Thanks in advance!
B.R.

Subject: Re: IDL new graphics memory leak?
Posted by Helder Marchetto on Tue, 04 Oct 2016 12:39:10 GMT
View Forum Message <> Reply to Message

On Tuesday, October 4, 2016 at 1:57:12 PM UTC+2, belk...@web.de wrote:
> Hey,
>
> I have some enourmous problem with the amound of memory (virtual) IDL is using. I believe they are resulting from my extensive use of plots and images using the plot/image functions from IDL. If I omit them from my code my memory usage is on a normal level (1-8gb). If I want to include plots and images then my usage of memory will grow till it overloads the cluster I am working on (200-400 gb). Right after saving them I destroyed the objects via obj_destroy. This didn't solved my problem so I used heap_free. Still no improvment. Is there anything I am missing here? I am using IDL 8.3 on linux 64 bit.

```
>  rough sketch of my program structure:
>
>  read file (big image)
>
>  For
>   For
>     data analysis
>     Img=image(...)
>     img.save,....
>     destroy img
>     plot=plot()...
>     ...
>  endfor
>  endfor
>  end
>
>
>  Thanks in advance!
>  B.R.
```

Did you try img.close instead of obj_destroy? Did this give the same result?
Also, if I do a lot of images, I would change the loop to something like this:

```
Img=image(...) ;can also be empty or use dist(100) or whatever...
plt = plot(...)
For
 For
   data analysis
   Img->setData, ...
   img.save,....
   plt->setData...
   ...
endfor
endfor
destroy img
end
```

This should also speed things up, but probably your bottleneck is not the the call to image(), but the "data analysis" before that.
Notice that you can call setData also pass x and y (as arrays).

Cheers,
Helder

Subject: Re: IDL new graphics memory leak?
Posted by Markus Schmassmann on Tue, 04 Oct 2016 13:36:43 GMT

On 10/04/2016 02:39 PM, Helder wrote:

```
> On Tuesday, October 4, 2016 at 1:57:12 PM UTC+2, belk...@web.de wrote:
>> I have some enourmous problem with the amound of memory (virtual)
>> IDL  is using. I believe they are resulting from my extensive use of plots
>> and images using the plot/image functions from IDL. If I omit them from
>> my code my memory usage is on a normal level (1-8gb). If I want to
>> include plots and images then my usage of memory will grow till it
>> overloads the cluster I am working on (200-400 gb). Right after saving
>> them I destroyed the objects via obj_destroy. This didn't solved my
>> problem so I used heap_free. Still no improvment. Is there anything I am
>> missing here? I am using IDL 8.3 on linux 64 bit.
>> rough sketch of my program structure:
>>
>> read file (big image)
>>
>> For
>>  For
>>   data analysis
>>   Img=image(...)
>>   img.save,....
>>   destroy img
>>   plot=plot()...
>>   ...
>> endfor
>> endfor
>> end
> Did you try img.close instead of obj_destroy? Did this give the same result?
> Also, if I do a lot of images, I would change the loop to something like this:
>
> Img=image(...) ;can also be empty or use dist(100) or whatever...
> plt = plot(...)
> For
>  For
>   data analysis
>   Img->setData, ...
>   img.save,....
>   plt->setData...
>   ...
> endfor
> endfor
> destroy img
> end
>
> This should also speed things up, but probably your bottleneck is not
> the the call to image(), but the "data analysis" before that.
> Notice that you can call setData also pass x and y (as arrays).
```

hopefully Helder's comments are sufficient to reduce your memory problems, if not,

```
help, /heap_variables
help, /shared_memory
help, /memory
```

might give you some hint on where the problem is.

Markus

Subject: Re: IDL new graphics memory leak?
Posted by Phillip Bitzer on Wed, 05 Oct 2016 13:51:29 GMT
View Forum Message <> Reply to Message

> On 10/04/2016 02:39 PM, Helder wrote:
>> Did you try img.close instead of obj_destroy? Did this give the same result?
>> Also, if I do a lot of images, I would change the loop to something like this:
>>

This is exactly the problem. A MWE shows it explicitly:

```
help, /heap, /brief ;to start

img = IMAGE(/test)
img.close
help, /heap, /brief ;anything on the heap?

img = IMAGE(/test)
OBJ_DESTROY, img
help, /heap, /brief ;Oops!
```

Further, the setData method Helder suggested will be very handy as well!