Subject: Re: HELP: Memory allocation problem
Posted by offenbrg on Tue, 09 Nov 1993 18:46:28 GMT
View Forum Message <> Reply to Message

<FUHRER@IAP.unibe.ch> writes:

> I am posting this for Markus Heuer, who has not yet access to
> Internet / Usenet.

> "I am using IDL procedures/functions TRIANGULATE and TRIGRID with huge
> amounts of sample data on VAX/VMS (IDL Verson 3.0.0, VMS Version 5.5,
> machine: VAX 4000/90). To reduce the required array sizes, I split the
> rectangular resampling region into smaller subrectangles and collect the data
> lying in them. However, the arrays inevitably contain several 100'000 elements.
[Trimming]

> Questions:

> - Is the problem due to VAX system parameter settings? (The system manager
> already tuned system parameters).
> - Could it be a memory leakage problem in IDL procedures TRIANGULATE and
> TRIGRID (results of HELP, /MEMORY inserted in the loop above suggest this,
> although the discrepancies are small)?
> - Might there be problems with the VAX implementations of the C functions
> malloc and free presumably used in the implementations of TRIANGULATE
> and TRTGRID?
> - Is there a workaround for the memory allocation failure, APART FROM
> reducing large array sizes (which I really need in the application at hand)?

I'm not quite sure what you are trying to do, but it sounds like you are
running into the old IDL memory-fragmentation bugaboo. Essentially, if
you create the array:

x = fltarr(1000,1000)
and then you stuff and then you
x = 0
all of the memory allocated in the original 1000x1000x4B array is not freed.
As a result, when you next try to create another array:
y = fltarr(1000,1000)
IDL can not use the block freed up when you erased X. After a while of
manipulating large arrays, IDL finds itself with a lot of memory, but not
in large enough chunks to make a new array. When faced with this situation,
IDL reports an error. (It could "collect garbage," that is group together
the data in memory to make larger regions of free memory, but this is
tricky programming and even then won't necessarily work).

The only way around this is to break the arrays up into smaller bits.
They'll be more likely to find a home in your memory...what you can do

is break things up into smaller arrays when manipulating them and then
put them back together later.

I'm not sure I'm expressing this as well as I might, but I'm willing
to rephrase things if I'm not making myself clear.

Joel
--
Joel D Offenberg | "Aren't you in the position of a life-long
Hughes STX/ UIT Science Team | vegetarian giving us your recipe for steak
offenbrg@fondue.gsfc.nasa.gov | and kidney pie?" - Rumpole of the Bailey
I get paid to stare into space. |
