Subject: Re: Large Arrays in IDL
Posted by Eric Deutsch on Sat, 13 Jan 1996 08:00:00 GMT
View Forum Message <> Reply to Message

Saeid Zoonematkermani wrote:
>
> This is a question that concerns astronomers once in a while. I have
> wondered what the limit is to the size of the arrays that IDL can create.
> Some times IDL fails to read a FITS image complaining:
>
> % Unable to allocate memory: to make array.
> not enough core
>
> This is not so uncommon if one has to deal with even moderate size cubes.
> I have tried to use .SIZE to increase the default value for data area but
> it doesn't seem to work. So the question is if there is a way to open a
> very large image by IDL. IRAF seems to have no problem with this. Does
> this problem arise from lack of RAM on the computer?

The "not enought core" error appears when you try to create an array for which there is not enough room in all of the memory space of the machine you're working on. For most machines, this is the total virtual memory available, which is not usually related to the amount of physical RAM installed in the box. You don't mention what platform you use, so I can't tell you how to check it exactly. On SunOS machines, try 'pstat -T' to get 421092/601888 swap which indicates 421MB of possible 601MB in use. On Solaris, try vmstat. On a VMS machine, you can try 'show mem'. There is an added complication on VMS that individual processes are usually given a certain working set limit, so even if the machine has plenty of virtual memory, you may be limited to using only a certain portion. To use large arrays in IDL, you'll need to increase the amount of virtual memory available to you (either your working set or the whole machine's available memory). This usually just involves setting aside more disk space for virtual memory purposes, not buying more RAM chips. It usually isn't very hard.

More efficient programming can help out, too. You can make use of the IDL TEMPORARY() function or other tricks to avoid many copies of large arrays. Or, you can do things the way IRAF does things: IRAF never really holds image arrays in memory. Instead of loading image A and loading image B and adding them and then writing out image C (which requires memory space for three arrays plus temporary storage space), IRAF usually does a piece-by-piece approach, i.e. in a loop, read small chunks of A and B, add, and then write out the result to C. It never reads the whole multi-megabyte images at once, but rather makes heavy use of temporary disk images and works on parts of files. IRAF was designed to be able to work with large image arrays without much memory. You can write IDL routines to do this, too, but it is usually much simpler to give your

computer a huge chunk of virtual memory, with 1 GB disks costing well under $300 these days...

Eric

--

---------- ----------

Eric Deutsch email: deutsch@astro.washington.edu
Department of Astronomy Voice: (206) 616-2788
University of Washington FAX: (206) 685-0403
Box 351580 WWW: http://www.astro.washington.edu/deutsch
Seattle, WA 98195-1580 Physics/Astronomy Bldg., Room B356F