Subject: Re: Array has too many elements?
Posted by Jonathan Greenberg on Mon, 22 Dec 2003 21:35:55 GMT
View Forum Message <> Reply to Message

I'm working with a fuzzy set classifier I found an article on that subdivides a dataspace into subspaces of smaller and smaller divisions. So, for instance, if you have 2 attributes and 2 divisions per attribute, you have an array that has 2^2 elements. Most datasets really need about 25 subdivisions per attribute, and, say, 6 attributes to start getting good classification, so this problem blows up VERY quickly (e.g. 25^6 elements for the problem I just described). One of the issues is that I need to extract a maximum value from the 25^6 array, which, it now looks like, i'll have to do in stages (e.g. subdivide the array into fixed size subsets, check each subset for max value and write to a new array, and then determine the max value for this new array). Doable, but obviously involves rewriting a lot of code.

This, by the way, is why I brought up the supercomputer thread -- manipulating arrays (even if I do get the programming bugs sorted out) of this size will be silly to do with a desktop PC. I hope IDL modifies how they deal with large arrays in the future -- since remote sensing images are getting bigger as the spatial, extent and spectral resolution gets higher, this problem is only going to get worse.

--j

"Jamie" <jamiedotwheeleratoxacuk@dummy.com> wrote in message
news:Pine.LNX.4.44.0312221805210.13640-100000@moriarty.atm.o x.ac.uk...
>
> On Fri, 19 Dec 2003, Jonathan Greenberg wrote:
>
>> So, any suggestions for the best way of getting around these limitations
(I
>> mean, without having to buy a 64-bit machine) -- how about chopping the
>> array up into smaller blocks and performing for-next loops -- processing
>> part of the array, writing the results, and then processing the next
part?
>> Are there better ways than this?
>
> Nope. Frankly, once you start working with arrays that consume 1GB of
> memory, you are in for a whole world of trouble. IDL is a flexible,
> non-compiled language which means that commands are expanded into a
> working stack where copies are often made. Working with big arrays also
> means becoming proficeint in using the NOZERO keyword, the
> REPLICATE_INPLACE procedure, the NO_COPY keyword, and the TEMPORARY
> function. You haven't really told us what exactly you are trying to do...
> I hope that you don't have too many zeros in your arrays ;)

>
> As far as I know, the array size limit in IDL on *all* platforms is still
> 2^31-1 (2 GB). Overcoming this is more-or-less impossible even with a 64
> -bit machine. In short, you need to break up large arrays and be careful
> with your indexing. The best case would be if you can effectively reduce
> the volume of the data as you loop.
>
> Keep in mind that IDL stands for "interactive data language." While, it
> is a capable programming environment for visualizing data, working with
> arrays that consume 1-2 GB is not quite mainstream yet. In another 2
> years, this probably won't be such an issue...
>
> Jamie
>

---