Subject: Re: a=a(*,*,[4,1,2,3,0]) efficiency
Posted by menakkis on Wed, 15 Jul 1998 07:00:00 GMT
View Forum Message <> Reply to Message

Ray <muzic@uhrad.nospam.com> wrote:
> I am wondering about the efficiency of the following
>
> ; read data from file into a which is an integer array 128x128x5
> ; open, ..., read a, ... close,...
>
> ; reorder data
> a=a(*,*,[4,1,2,3,0])
>
> Does IDL make a temporary copy of a when size of the left
> hand side (a) is the same as the right hand side a(*,*,[4,1,2,3,0]) ?
> If so, is there a better way to reorder my data? In my application
> the last dimension of a is typically much greater than 5 (e.g. 300).

I believe that there's actually a bit of a "temporary variable happening" (over and above a single copy of your array) when you reorder a dimension of an array like this. And if you have hundreds of these 128*128 image bands, it might be worth your while to do something about it, depending on your program's purpose (e.g., research- or production-orientated).

When it comes to checking out memory issues like this, I favour the direct approach - run a simple test case on a platform that has a tool for monitoring memory usage. Win95 is good - you can follow relative changes in the "Memory Manager - Allocated memory" output of the System Monitor (which is in Programs . Accessories . System Tools). Use quite a large INT matrix (e.g., 128*128*320 = 10MB) so that you can track things easily (slow enough and big enough). If you don't have a Windows IDL licence, a demo installation works fine for this sort of test. (BTW, a little test I ran on your example showed about a 3* temporary variable overhead.)

Anyway, given that the memory-wastage problem is of concern to you, here are four alternatives that come to mind:

1. Find another way to deal with the problem - one that DOESN'T rely on the whole image being in memory.

2. Set up and work with an array of pointers with one image band per pointer. (If you need fast access to the image as a whole, e.g., to pull spectra out of the last dimension, then this won't do, of course.)

3. Do the reordering more explicitly, making your own "temporary" copy, viz. B=A &FOR I=0,N-1 DO B[0,0,I]=A[*,*,reordered[i]] &A=0 &A=TEMPORARY(B) This WILL be more efficient.

4. Write a C routine to do the reordering. This will only require an overhead of 128*128 INTs.

Cheers

Peter Mason

-----== Posted via Deja News, The Leader in Internet Discussion ==-----
http://www.dejanews.com/rg_mkgrp.xp Create Your Own Free Member Forum