Subject: Re: find max in 3D array -- slow
Posted by Gray on Sun, 11 Apr 2010 12:29:15 GMT

View Forum Message <> Reply to Message

On Apr 11, 7:04 am, Maxwell Peck <maxjp...@gmail.com> wrote:
> On Apr 11, 8:44 pm, Maxwell Peck <maxjp...@gmail.com> wrote:
>
>
>
>
>
>> On Apr 11, 2:03 am, "Timothy W. Hilton" <hil...@meteo.psu.edu> wrote:
>
>>> Hello IDL users,
>
>>> I have a 1200x1200x2900 array of floats. The dimensions correspond to
>>> latitude x longitude x time. I need to find the maxium at each
>>> location -- that is, I need the 1200x1200 array containing the max
>>> along the 3rd dimsion. IDL takes almost 3 minutes to do this on my
>>> system. This seemed slow. I compared it with Matlab, which took ten
>>> seconds. Is there a better way to search for the maxima using IDL?
>
>>> The demo code I used to compare IDL and Matlab is below (with output).
>
>>> I'm wondering if I ought to switch to Matlab. I just spent a couple
>>> of days writing IDL code to read my data, so I'd rather not.
>
>>> Many thanks,
>>> Tim
>
>>> --
>
>>> Timothy W. Hilton
>>> PhD Candidate, Department of Meteorology
>>> The Pennsylvania State University
>>> 503 Walker Building, University Park, PA 16802
>>> hil...@meteo.psu.edu
>
>>> ========
>>> scratch.pro:
>
>>> foo = randomu(0, 1200, 1200, 2920)
>>> PRINT, systime()
>>> foo_max = max(foo, DIMENSION = 3)
>>> PRINT, systime()
>>> END
>
>>> IDL> .run scratch

>>> % Compiled module: $MAIN$.
>>> Sat Apr 10 10:44:44 2010
>>> Sat Apr 10 10:47:36 2010
>>> IDL>
>
>>> ========
>>> scratch.m:
>
>>> foo = rand(1200,1200,2920);
>>> fprintf('%s\n', datestr(now()));
>>> foo_max = max(foo, [], 3);
>>> fprintf('%s\n', datestr(now()));
>
>>>> > scratch
>
>>> 10-Apr-2010 10:42:45
>>> 10-Apr-2010 10:42:55
>
>> I don't know if it's any quicker, and it will sure chew some memory
>> but sort_nd might be worth a try.
>
>> http://www.dfanning.com/programs/sort_nd.pro
>
>> Just sort it along the 3rd dimension and then pull the slice of the
>> last band...
>
> I think sort_nd is going to be much slower on testing. Max in IDL does
> seem to be a lot slower than what i'd expect in IDL.

For an array of this size, you're certainly running into memory issues. My suggestion is to use a loop and find the max over a number of subsets, then find the max of those maxima. You can tune the subset size/number to find a balance between the cost of the loop and the memory issues.