

---

Subject: Best method for very big array rebinning?  
Posted by [amolins](#) on Wed, 07 Jul 2004 14:10:29 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hello,

I have a program that needs to rebin from one to another coordinate system thus interpolating the actual data in positions already computed and arranged the way INTERPOLATE function demands, so I only have to do INTERPOLATE(array, coords1,coords2,coords3).

I have to rebin a "array" very very big (2GB in disk) of floating-point values, so I made use of SHMMAP function to access to the data fast. But this solution doesn't scale, because IDL complains (both in windows and linux implementations) about the size of the array declared in that way.

My questions are:

- 1) is there any efficient method to perform this without having to code by myself in a FOR loop each acces by POINT\_LUN, READU, and maths for each value to be interpolated?
- 2) is there any way to increase the maximum number of elements in an array in IDL 6.0?

Thanks,  
Antonio.

---

---

Subject: Re: Best method for very big array rebinning?  
Posted by [K. Bowman](#) on Thu, 08 Jul 2004 15:07:19 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

In article <f6d3543d.0407080240.46755e96@posting.google.com>, amolins@mce.hggm.es (Antonio Molins) wrote:

- > Kenneth Bowman <k-bowman@null.tamu.edu> wrote in message
- > news:<k-bowman-72BB68.09413807072004@news.tamu.edu>...
- >> Buy a 64-bit computer with lots of memory (Sun, IBM, SGI, or HP).
- >
- > ok... No more ideas? I can't believe any other method to refine my algorithm
- > exist!

Well, you asked how to make arrays bigger than 2 GB. That can't be done on a 32-bit machine.

The alternative is to break the problem up into chunks, and the approach that you suggested is probably as good as any. If you can do the problem by rows (i.e., in order of memory access), it will probably be relatively efficient.

