Subject: Large array memory problem.
Posted by Pitufa on Tue, 22 Nov 2005 12:39:18 GMT
View Forum Message <> Reply to Message

Hi,

I am running IDL 5.5 in a computer that has 2GB of RAM and a paging file size of 4GB. However it seems that IDL will not give more than about 1.2GB for array storage per IDL session. Unfortunately, I need to do an array multiplication where the arrays are double precision complex 11000 by 11000 square arrays, and IDL will not even give me enough memory for a single dcomplexarr(6500,6500) array!

Am I being too ambitious? Could anyone please tell me if it is possible to store such a large array?

I though IDL was the best language for large array manipulations, is it true? or are there other languages that are better suited for this?

Many thanks for your time,

Carolina.

Subject: Re: Large array memory problem.
Posted by Marius Hagen on Tue, 22 Nov 2005 22:16:10 GMT
View Forum Message <> Reply to Message

Carolina wrote:

> Unfortunately, I need to
> do an array multiplication where the arrays are double precision
> complex 11000 by 11000 square arrays, and IDL will not even give me
> enough memory for a single dcomplexarr(6500,6500) array!

I often find myself working with large arrays. Usually, these are sparse, and so I can make use of sparse storage techniques to make matrix multiplication and the like tractable. However, if the matrices are not sparse, then one method I've made use of in the past is to break the matrix multiplication up into pieces. That is, rather than store the matrices in full, you can store in IDL just a single row/column vector of each of the matrices, then apply the multiplication, save the result, and loop over the next row/column. It's a little slow, but there are quite a number of operations taking place in each iteration of the loop, so it's faster than you might expect. If accessing the next row/column vector requires reading a large file over and over again, you might see the loop taking an enormous amount of time. On the other hand, if you can actually generate the matrix elements on the fly, then you can generate the row vector and column vector inside the matrix multiplication loop.

> I though IDL was the best language for large array manipulations, is it
> true? or are there other languages that are better suited for this?

Of course, C/C++ can do this stuff a lot better than IDL, but you have to deal with a lot of programming hassles and a rather unfriendly programming environment (compared to IDL) that really slows down the process of writing routines. I have some experience doing this in Matlab and Mathematica, but Matlab generally is more limited than IDL as far as speed of operation and array size limitations go. Mathematica is even more so, and can sometimes crash with surprisingly small matrices. Not to mention Mathematica's well-known tendency to corrupt datasets and program files.

- Marius

Subject: Re: Large array memory problem.
Posted by Pitufa on Wed, 23 Nov 2005 18:12:19 GMT
View Forum Message <> Reply to Message

Marius Hagen wrote:
> I often find myself working with large arrays. Usually, these are
> sparse, and so I can make use of sparse storage techniques to make
> matrix multiplication and the like tractable.

Does anyone know a clever trick to compute inverse of large sparse matrices?

Thanks for all the replies,

Carolina.

Subject: Re: Large array memory problem.
Posted by news.qwest.net on Wed, 23 Nov 2005 19:19:06 GMT
View Forum Message <> Reply to Message

"Carolina" <c.c.calderon@gmail.com> wrote in message
news:1132769539.270480.61990@o13g2000cwo.googlegroups.com...
>
> Marius Hagen wrote:
>> I often find myself working with large arrays. Usually, these are
>> sparse, and so I can make use of sparse storage techniques to make
>> matrix multiplication and the like tractable.
>

> Does anyone know a clever trick to compute inverse of large sparse
> matrices?
>
> Thanks for all the replies,
>
> Carolina.
>

IDL has some functions for sparse arrays.
For example see the IDL help:
http://idlastro.gsfc.nasa.gov/idl_html_help/mathematics15.ht ml

Depending on your situation you may need to look at some other
code (netlib.org for example)
http://netlib.org/sparse/index.html
or I've heard matlab has some good sparse array routines as well.

Cheers,
bob

Subject: Re: Large array memory problem.
Posted by Marius Hagen on Wed, 23 Nov 2005 20:43:08 GMT
View Forum Message <> Reply to Message

Carolina wrote:
> Does anyone know a clever trick to compute inverse of large sparse
> matrices?

While I'm no expert on the subject, I do spend a lot of time working with large matrices, so maybe I can say a few things that might help here. In most instances getting the inverse of a large matrix is a *bad idea*, since, in floating-point math, you can easily wind up amplifying very small errors in the input into HUGE errors in the output. For an array with 10^8 elements, even in double-precision, most direct matrix inverses are so corrupted with these amplified errors as to be almost useless. Instead, what most people try is to use either a direct solver or an indirect solver of the *forward* matrix problem.

That is, if you are trying to calculate Ax = b (A=huge matrix, x and b long vectors), then rather than solving this as x = A^-1 b, you can use QR decomposition to change the form of A such that it is easy to solve by back-substitution. This can be rather slow, so an alternative is to obtain an approximate solution for x by use of an iterative solver. These are much faster, and typically they can obtain as much accuracy as you want in a fraction of the time that a direct solver requires. And these techniques do not have the extreme amplification of small errors.

Having said that, my own work is entirely in matrices which do not have
an inverse at all, and which have 10^15 elements or so, and so
iterative solvers are my only choice. Choosing which algorithm to use
here is an art --- generally each application wants something
different. But if you have access to Matlab, its support for sparse
matrix routines and for direct or indirect matrix solvers is
impressive.

- Marius

Subject: Re: Large array memory problem.
Posted by Nigel Wade on Mon, 28 Nov 2005 11:40:53 GMT
View Forum Message <> Reply to Message

Reimar Bauer wrote:

> Rick Towler wrote:
>
>> On a related note, is anyone using IDL x86-64 for linux?
>>
>> -Rick
>
> Yes,
> we have several dual opterons using it in 64 bit.
>
> By one of the last security updates of the kernel I believe we can't start
> idl with the -32 option on these machines.
> But this doesn't matter. 64 bit is great.
>
>
> cheers
> Reimar
>

Which kernel/security update?

We require 32bit IDL (for the time being, until some extensive external code is
made 64-bit safe).

--
Nigel Wade, System Administrator, Space Plasma Physics Group,
University of Leicester, Leicester, LE1 7RH, UK
E-mail : nmw@ion.le.ac.uk
Phone : +44 (0)116 2523548, Fax : +44 (0)116 2523555