Subject: Re: Array indexing: what is IDL doing?
Posted by Jean H. on Fri, 06 Jun 2008 21:15:38 GMT
View Forum Message <> Reply to Message

> cx = n_elements(wx)
> cy = n_elements(wy)
> nx = n_elements(bigarr[*,0])
> ind = transpose(rebin(wy,cy,cx))*nx + rebin(wx,cx,cy)
> res = bigarr[ind]

You don't have to use nx.
res = bigarr[rebin(bigx,75,10000),rebin(bigy,10000,75)]

Note that you must have the same number of elements in every subscripted dimension.

Read http://www.dfanning.com/code_tips/asterisk.html

Jean

Subject: Re: Array indexing: what is IDL doing?
Posted by Jeremy Bailin on Sat, 07 Jun 2008 13:12:26 GMT
View Forum Message <> Reply to Message

> So I want to pull out a small subsection from within a larger idl
> array. I want to do something like this:
>
> bigarr = findgen(500,100000)
> bigx = [indgen(50)+100,indgen(25)+200]
> bigy = indgen(10000)+50000
>
> res = bigarr[bigx,bigy]

In the particular case above, I would expect that this would be most efficient:

res = [bigarr[100:149,50000:59999], bigarr[200:224,50000:59999]]

Of course, that's only going to work when you know a priori that everything will be in a relatively small number of chunks.

-Jeremy.

Subject: Re: Array indexing: what is IDL doing?
Posted by Conor on Mon, 09 Jun 2008 12:56:28 GMT

On Jun 7, 9:12 am, Jeremy Bailin <astroco...@gmail.com> wrote:
>> So I want to pull out a small subsection from within a larger idl
>> array. I want to do something like this:
>
>> bigarr = findgen(500,100000)
>> bigx = [indgen(50)+100,indgen(25)+200]
>> bigy = indgen(10000)+50000
>
>> res = bigarr[bigx,bigy]
>
> In the particular case above, I would expect that this would be most
> efficient:
>
> res = [bigarr[100:149,50000:59999], bigarr[200:224,50000:59999]]
>
> Of course, that's only going to work when you know a priori that
> everything will be in a relatively small number of chunks.
>
> -Jeremy.

Interesting, thanks!

Subject: Re: Array indexing: what is IDL doing?
Posted by Conor on Mon, 09 Jun 2008 13:02:19 GMT

On Jun 6, 5:15 pm, Jean H <jghas...@DELTHIS.ucalgary.ANDTHIS.ca>
wrote:
>> cx = n_elements(wx)
>> cy = n_elements(wy)
>> nx = n_elements(bigarr[*,0])
>> ind = transpose(rebin(wy,cy,cx))*nx + rebin(wx,cx,cy)
>> res = bigarr[ind]
>
> You don't have to use nx.
> res = bigarr[rebin(bigx,75,10000),rebin(bigy,10000,75)]
>
> Note that you must have the same number of elements in every subscripted
> dimension.
>
> Readhttp://www.dfanning.com/code_tips/asterisk.html
>
> Jean

On second thought, I'm now thoroughly confused. I don't understand

this bit:

res = bigarr[rebin(bigx,75,10000),rebin(bigy,10000,75)]

obviously rebin(bigx,75,10000) and rebin(bigy,10000,75) give you two arrays with exactly opposite dimensions, so I don't understand how you can index bigarr with these two arrays. Also, I'm not even sure what IDL is doing when you index an array like that. My first thought was that this was the equivelent of:

res = bigarr[[rebin(bigx,75,10000),rebin(bigy,10000,75)]]

But that obviously can't be the case because since they have different dimensions, you can't concatenate the two arrays. So how does IDL pull out an array index in your above example?

Subject: Re: Array indexing: what is IDL doing?
Posted by Heinz Stege on Mon, 09 Jun 2008 15:34:29 GMT
View Forum Message <> Reply to Message

On Fri, 6 Jun 2008 09:50:33 -0700 (PDT), Conor wrote:

> So I want to pull out a small subsection from within a larger idl
> array. I want to do something like this:
>
> bigarr = findgen(500,100000)
> bigx = [indgen(50)+100,indgen(25)+200]
> bigy = indgen(10000)+50000
>
> res = bigarr[bigx,bigy]
>
> However IDL doesn't like that and says:
>
> % All array subscripts must be same size. Var = BIGARR
> % Execution halted at: $MAIN$
>
You may want to try this command:
  res = bigarr[bigx,bigy,0]

IDL accepts arrays of different size as subscripts, if at least one of the subsripts is a scalar. Here the "0" addresses an "extra" dimension. Since this extra dimension is of size 1, it is degenerated in the result. Array res has two dimensions, 75x10000.

Heinz

Subject: Re: Array indexing: what is IDL doing?
Posted by Jean H. on Mon, 09 Jun 2008 17:12:49 GMT
View Forum Message <> Reply to Message

Conor wrote:
> On Jun 6, 5:15 pm, Jean H <jghas...@DELTHIS.ucalgary.ANDTHIS.ca>
> wrote:
>>> cx = n_elements(wx)
>>> cy = n_elements(wy)
>>> nx = n_elements(bigarr[*,0])
>>> ind = transpose(rebin(wy,cy,cx))*nx + rebin(wx,cx,cy)
>>> res = bigarr[ind]
>> You don't have to use nx.
>> res = bigarr[rebin(bigx,75,10000),rebin(bigy,10000,75)]
>>
>> Note that you must have the same number of elements in every subscripted
>> dimension.
>>
>> Readhttp://www.dfanning.com/code_tips/asterisk.html
>>
>> Jean
>
> On second thought, I'm now thoroughly confused. I don't understand
> this bit:
>
> res = bigarr[rebin(bigx,75,10000),rebin(bigy,10000,75)]
>
> obviously rebin(bigx,75,10000) and rebin(bigy,10000,75) give you two
> arrays with exactly opposite dimensions, so I don't understand how you
> can index bigarr with these two arrays. Also, I'm not even sure what
> IDL is doing when you index an array like that. My first thought was
> that this was the equivelent of:
>
> res = bigarr[[rebin(bigx,75,10000),rebin(bigy,10000,75)]]
>
> But that obviously can't be the case because since they have different
> dimensions, you can't concatenate the two arrays. So how does IDL
> pull out an array index in your above example?

Ah, I guess you are right... I was typing too fast (and the no-error
didn't warn me..)

it should have been:
IDL> help, bigarr[rebin(bigx,75,10000),transpose(rebin(bigy,10000,75))]

So, rebin(bigx,75,10000) replicates your 75 entries 10 000 times.
rebin(bigy,10000,75)) replicates the 10 000 entries 75 times. Now,
transpose(rebin(bigy,10000,75)), well, transposes it so the two
subscript now have the same dimensions. At last, for each corresponding

index in both subscripts, the value of bigArr is returned.

Was i posted was working because there are the same number of elements
in both subscripts... though the combination was wrong and therefore it
returns the wrong result..

Sorry for the confusion!
Jean

Subject: Re: Array indexing: what is IDL doing?
Posted by Conor on Tue, 10 Jun 2008 14:16:41 GMT
View Forum Message <> Reply to Message

On Jun 9, 11:34 am, Heinz Stege <public.215....@arcor.de> wrote:
> On Fri, 6 Jun 2008 09:50:33 -0700 (PDT), Conor wrote:
>> So I want to pull out a small subsection from within a larger idl
>> array. I want to do something like this:
>
>> bigarr = findgen(500,100000)
>> bigx = [indgen(50)+100,indgen(25)+200]
>> bigy = indgen(10000)+50000
>
>> res = bigarr[bigx,bigy]
>
>> However IDL doesn't like that and says:
>
>> % All array subscripts must be same size. Var = BIGARR
>> % Execution halted at: $MAIN$
>
> You may want to try this command:
> res = bigarr[bigx,bigy,0]
>
> IDL accepts arrays of different size as subscripts, if at least one of
> the subsripts is a scalar. Here the "0" addresses an "extra"
> dimension. Since this extra dimension is of size 1, it is degenerated
> in the result. Array res has two dimensions, 75x10000.
>
> Heinz

Hmm... this does work! It seems very strange though. Seems like a
rather arbitrary rule...
