Subject: Re: How to speed up large FOR loops?
Posted by Fabzi on Tue, 10 Apr 2012 16:17:26 GMT
View Forum Message <> Reply to Message

Hi,

Do you concatenate arrays in your loop?

e.g: out = [out, tmp]

On 04/10/2012 05:48 PM, Saurav Dhital wrote:
> Hi,
>
> By necessity, I have to use a long FOR loop (N>~ 300,000). I have
> made the operations inside the loop efficient enough that the first
> 40,000 iterations are relatively fast (~30 mins). Then it slows down
> to a crawl, with the entire process taking>24 hours. I would
> appreciate any advice how to speed this up.
>
> The operation inside the loop involves matching between two (large)
> FITS files and assigning the matches to another (previously created)
> FITS structure. There are no new variable created per iteration, so
> there should be no memory being used (each loop writes over the
> variables anyways). I even create small substructures (e.g.,>
> sub_mystruct = mystruct[0:10]) so that I don't have to index the large
> structures.
>
> Any tip or advice would be much appreciated,
> ~Saurav

Subject: Re: How to speed up large FOR loops?
Posted by Saurav Dhital on Tue, 10 Apr 2012 16:26:27 GMT
View Forum Message <> Reply to Message

Yes. More specifically, I reform arrays and assign them to the new
FITS structure. This happens a lot.

Subject: Re: How to speed up large FOR loops?
Posted by Fabzi on Tue, 10 Apr 2012 16:53:29 GMT
View Forum Message <> Reply to Message

On 04/10/2012 06:26 PM, Saurav Dhital wrote:
> Yes. More specifically, I reform arrays and assign them to the new

> FITS structure. This happens a lot.

This is probably the problem. Doing out = [out, INDGEN(10)] is very inefficient in memory.

If you have IDL 8+ I suppose you could use lists. If not (like me), you may want to use the useful MGcoArrayList from Michael Galloy:

http://docs.idldev.com/idllib/collection/mgcoarraylist__defi ne.html

Cheers,

Fab

Subject: Re: How to speed up large FOR loops?
Posted by David Fanning on Tue, 10 Apr 2012 17:13:52 GMT
View Forum Message <> Reply to Message

Saurav Dhital writes:

> Yes. More specifically, I reform arrays and assign them to the new
> FITS structure. This happens a lot.

Are you setting the OVERWRITE keyword on your REFORM command? You should be!

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: How to speed up large FOR loops?
Posted by Saurav Dhital on Tue, 10 Apr 2012 17:23:31 GMT
View Forum Message <> Reply to Message

David, I am NOT using the REFORM routine, which I now realize could be my problem. Here is what I do:

;; combine two 2-d arrays ([N1,N2]) such that the output is

[2,N1,N2]
FUNCTION
form_3Darray,a1,a2
return, transpose([[[transpose(a1)]],
[[transpose(a2)]]])
END

I call this as:
>> x = [[x1],[x2],[x3]] & y = [[y1],[y2],[y3]]
> z = form_3Darray([[x]],[[y]])

I have to do this so I can store z in a single field in the FITS
structure.

Thanks,
~S

On Apr 10, 1:13 pm, David Fanning <n...@idlcoyote.com> wrote:
> Saurav Dhital writes:
>> Yes. More specifically, I reform arrays and assign them to the new
>> FITS structure. This happens a lot.
>
> Are you setting the OVERWRITE keyword on your REFORM
> command? You should be!
>
> Cheers,
>
> David
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:http://www.idlcoyote.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: How to speed up large FOR loops?
Posted by Craig Markwardt on Tue, 10 Apr 2012 18:21:50 GMT
View Forum Message <> Reply to Message

On Tuesday, April 10, 2012 12:26:27 PM UTC-4, Saurav Dhital wrote:
> Yes. More specifically, I reform arrays and assign them to the new
> FITS structure. This happens a lot.

This is a problem. A pattern like this,
L = [0]
for i = 0, 1000000-1 do L = [L, i]

will run quickly at first and then slow down as L gets longer and longer.

The trick is to pre-allocate your lists like L. If the size of L is not known in advance, then pre-allocate a reasonable fixed size to begin with, and then grow it by factors of 2 as needed (up to a point).

At the end, you will need to trim the unused portion of L.

I do this kind of pattern all the time, and it can be quite efficient.

Craig

Subject: Re: How to speed up large FOR loops?
Posted by Saurav Dhital on Tue, 10 Apr 2012 19:39:29 GMT
View Forum Message <> Reply to Message

Hi Craig, yup, that is a big problem, which I had to correct a while back. I create a FITS structure of a fixed size at the beginning and fill it within the FOR loop and trim it at the end.

I only concatenate arrays so as to put them into fields in the FITS table (for e.g., x = [[x1],[x2],[x3]] and struct.field = x).

One more thing that I don't do: I don't even use the where() function in my matching as where() searches through the entire array EVERY single time. If you are matching large arrays, where() ends up being the biggest bottleneck.

~Saurav

Subject: Re: How to speed up large FOR loops?
Posted by Matt Francis on Fri, 13 Apr 2012 02:41:23 GMT
View Forum Message <> Reply to Message

Have you looked at PROFILER to see where the bottlenecks are? You might need to temporarily turn parts of the code inside the loop into function, just to see which bits are using causing problems. I find the IDL PROFILER remarkably useful, at least compared to the equivalent tools in other languages I work with. David Fanning (see comment 5) has a very good PROFILER tutorial on his site if you are unfamiliar with its use.

Subject: Re: How to speed up large FOR loops?

Posted by Saurav on Fri, 20 Apr 2012 16:48:40 GMT
View Forum Message <> Reply to Message

Thanks! I will use th profiler, probably David's as I already use a lot of his code.