

---

Subject: Re: for loops and calls to functions/objects  
Posted by [Conor](#) on Tue, 15 Apr 2008 18:59:44 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

On Apr 15, 11:11 am, Charles at AER <cpax...@aer.com> wrote:

> Dear Newsgroup,  
>  
> A program I am optimizing loads a series of images from netcdf files,  
> the user may select an arbitrary number of images to display. Within  
> a for loop, the program calls a netcdf reader that initializes a  
> netcdf object, gets lat and lon info and then grabs an image. If the  
> user wants to look at 10 images, the read net cdf is called 10 times,  
> and the time to initialize the netcdf object takes 0.1 seconds, and to  
> get the image takes 0.2 seconds. Here's the rub, if the user wants  
> to investigate 70 images, the netcdf initialization is 10 times longer  
> - about 1 second, similarly, the get method loads the image in 2  
> seconds. So to read 10 images takes 3 seconds, and to read 70 images  
> takes 210 seconds. The exponential increase is untenable. Does any  
> one understand what is happening, and are there work-arounds? Thanks.  
>  
> Sincerely,  
>  
> Charles

I'm not sure if there's really enough info here to go on, but I'll hazard a guess. I don't actually know anything about netcdf files, but I suspect that isn't the problem. After all there is no reason why the time it takes to load an image should change from call to call, assuming all the images are the same size. My guess then is that you are simply running out of memory. If the first 50 images fill up your memory then for the next 20 images your computer is going to be reading and writing to a swap directory, which is a VERY slow process. Might that be the problem?

---

Subject: Re: for loops and calls to functions/objects  
Posted by [Charles at AER](#) on Tue, 15 Apr 2008 19:29:42 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

On Apr 15, 2:59 pm, Conor <cmanc...@gmail.com> wrote:

> On Apr 15, 11:11 am, Charles at AER <cpax...@aer.com> wrote:  
>  
>  
>  
>> Dear Newsgroup,  
>  
>> A program I am optimizing loads a series of images from netcdf files,  
>> the user may select an arbitrary number of images to display. Within

>> a for loop, the program calls a netcdf reader that initializes a  
>> netcdf object, gets lat and lon info and then grabs an image. If the  
>> user wants to look at 10 images, the read net cdf is called 10 times,  
>> and the time to initialize the netcdf object takes 0.1 seconds, and to  
>> get the image takes 0.2 seconds. Here's the rub, if the user wants  
>> to investigate 70 images, the netcdf initialization is 10 times longer  
>> - about 1 second, similarly, the get method loads the image in 2  
>> seconds. So to read 10 images takes 3 seconds, and to read 70 images  
>> takes 210 seconds. The exponential increase is untenable. Does any  
>> one understand what is happening, and are there work-arounds? Thanks.

>

>> Sincerely,

>

>> Charles

>

> I'm not sure if there's really enough info here to go on, but I'll  
> hazard a guess. I don't actually know anything about netcdf files,  
> but I suspect that isn't the problem. After all there is no reason  
> why the time it takes to load an image should change from call to  
> call, assuming all the images are the same size. My guess then is  
> that you are simply running out of memory. If the first 50 images  
> fill up your memory then for the next 20 images your computer is going  
> to be reading and writing to a swap directory, which is a VERY slow  
> process. Might that be the problem?

Hi Conor,

Thank you very much. Indeed I believe you are correct. I wrote a  
separate  
code to isolate the module in the intervening hours, and I found  
memory leaks.

I've cleaned them all seems reasonable now. Each file now takes a  
similar amount  
of time to process, whether I churn over 10 or 100.

Regards,

Charles Paxson  
AER