Subject: Re: idl speed question
Posted by Michael Galloy on Sat, 14 Mar 2009 22:28:21 GMT
View Forum Message <> Reply to Message

oxfordenergyservices@googlemail.com wrote:
> I have the following strange result
>
> a=double(1.0)
> for i=0,10000 do begin
> for j=0,10000 do begin
> a=a+1.0
> endfor
> endfor
>
> takes 13 seconds whereas
>
> a=double(1.0)
> b=double(1.0)
> c=double(1.0)
> d=double(1.0)
> e=double(1.0)
> for i=0,10000 do begin
> for j=0,10000 do begin
> a=a+1.0
> b=b+1.0
> c=c+1.0
> d=d+1.0
> e=e+1.0
> endfor
> endfor
>
> takes 60 seconds? I thought the overhead with IDL was in the loops
> rather than the computing?

The real killer for speed is number of statements (each one has to be be interpreted). Loops are bad only because they could execute a statement a possibly large number of times. So in your example, the first case has 10001 * 10001 statements while the second has 5 * 10001 * 10001 statements. So if the statements are doing the same amount of work, one would expect the second to take about 5 times more time.

The conclusion: try to do more work per statement.

Mike
--
www.michaelgalloy.com
Associate Research Scientist
Tech-X Corporation

Subject: Re: idl speed question
Posted by oxfordenergyservices on Mon, 16 Mar 2009 10:39:21 GMT
View Forum Message <> Reply to Message

On 14 Mar, 22:28, Michael Galloy <mgal...@gmail.com> wrote:
> oxfordenergyservi...@googlemail.com wrote:
>> I have the following strange result
>
>> a=double(1.0)
>> for i=0,10000 do begin
>> for j=0,10000 do begin
>> a=a+1.0
>> endfor
>> endfor
>
>> takes 13 seconds whereas
>
>> a=double(1.0)
>> b=double(1.0)
>> c=double(1.0)
>> d=double(1.0)
>> e=double(1.0)
>> for i=0,10000 do begin
>> for j=0,10000 do begin
>> a=a+1.0
>> b=b+1.0
>> c=c+1.0
>> d=d+1.0
>> e=e+1.0
>> endfor
>> endfor
>
>> takes 60 seconds? I thought the overhead with IDL was in the loops
>> rather than the computing?
>
> The real killer for speed is number of statements (each one has to be be
> interpreted). Loops are bad only because they could execute a statement
> a possibly large number of times. So in your example, the first case has
> 10001 * 10001 statements while the second has 5 * 10001 * 10001
> statements. So if the statements are doing the same amount of work, one
> would expect the second to take about 5 times more time.
>
> The conclusion: try to do more work per statement.
>
> Mike
> --www.michaelgalloy.com
> Associate Research Scientist
> Tech-X Corporation

0k, thanks Mike, that makes sense!