Subject: Re: FOR loops and efficiency
Posted by JDS on Fri, 29 May 2009 20:39:16 GMT
View Forum Message <> Reply to Message

On May 28, 1:38 am, Craig Markwardt <craig.markwa...@gmail.com> wrote:
> On May 26, 5:51 pm, JDS <jdtsmith.nos...@yahoo.com> wrote:
>
>
>
>>> I still stand by my rule of thumb. The problem with FOR loops is the
>>> amount of time spent doing loop overhead stuff. If you run your loop
>>> but *take all the calculations out*, and the total execution time is
>>> not perceptible, then you probably won't gain by optimizing/
>>> vectorizing.
>
>> I find that analysis lacking for a few reasons. Consider this
>> example:
> ...
>> IDL> t=systime(1) & for i=0L,10000000L-2 do a[i+1]+=a[i] &print,systime(1)-t
> ...
>> IDL> t=systime(1) & for i=0L,10000000L-2 do begin & end & print,systime(1)-t
>> 0.12700295
> ...
>> Loop overhead is one reason to avoid FOR loops with high iteration
>> count, but it is by no means not the *only* reason. ...
>
> I agree with everything you said. I still stand by my guideline as
> rule of thumb to know when optimization is important. Note that the
> rule of thumb didn't involve trying to compare the execution time of
> an empty loop and a full loop. :-)
>
> By the way, if you put a simple dummy statement like this,
>
> t=systime(1) & for i=0L,10000000L-2 do begin & dummy = 0 & end &
> print,systime(1)-t
>
> Then the execution time is more like 0.5 seconds. While I agree that
> this is not the same as 2.2 seconds, it is definitely more
> comparable.
>
> Craig

Right. Now that I read your rule of thumb more carefully, I see your point is really "keep it to a small number of iterations." The problem is, if this is in a function which itself (perhaps later) gets called millions of times, it will be of no solace that each function call performs only 100 iterations. But your rule of thumb is actually useful for all types of performance optimization. Should I optimize?

Only if it takes too long.

JD