Hi,

Subject: Re: Doing Nothing Takes Longer Than Doing... Nothing?
Posted by Ken Knapp on Wed, 11 Feb 2004 13:51:38 GMT
View Forum Message <> Reply to Message

Not sure if this is much help, but it was fun to compile and run. Here's
the output from:
IDL Version 6.0, Microsoft Windows (Win32 x86 m32). (c) 2003, Research
Systems, Inc.

```
Average1  =   3.3160019e-006 s
Average2  =   3.1799984e-006 s
```

IDL Code Profiler reports:

```
Module          Type  Count     Only(s)   Avg.(s)     Time(s)   Avg.(s)
TESTROUTINE1    (U)  250000    0.417654  0.000002    0.417654  0.000002
TESTROUTINE2    (U)  250000    0.416161  0.000002    0.416161  0.000002
```

Tim Robishaw wrote:
> pathology: no matter which modules I race and no matter which order I
> race them in (e.g., TESTROUTINE1 vs. TESTROUTINE2 or TESTROUTINE2 vs.
> TESTROUTINE1 or *even* TESTROUTINE1 vs. TESTROUTINE1) I find that the
> module inside the first loop is always FASTER. I threw in a 3rd
> routine that does NOTHING as well... same deal. I have a feeling this
> may have something to do with the fundamentals of computer science, or
> magic. Any help here would be appreciated! Best -Tim.

Looks like they're the same to me.

Have fun with your puzzle.
-Ken

--
***** to reply remove the _REMOVE_ *****

Ken Knapp   Ken.Knapp@_REMOVE_noaa.gov
Remote Sensing and Applications Division
National Climatic Data Center
151 Patton Ave
Asheville, NC 28806
828-271-4339 (voice) 828-271-4328 (fax)

Subject: Re: Doing Nothing Takes Longer Than Doing... Nothing?
Posted by justspam03 on Wed, 11 Feb 2004 17:17:10 GMT
View Forum Message <> Reply to Message

Hi,

same on my machine - minimal but consistent time differences.

Differences when calling the same function in both loops are negligible, though.
I found that adding a call to one of the two test functions before the main loop will make the loop that calls this function run faster.
E.g. in your test program below with the added line, the 2nd loop runs faster than the 1st.
Maybe (wild speculation mode on) it's got to do with how IDL internally manages function calls/function lists - calls to functions further up in the list would be faster than calls to functions further down the list? (wild speculation mode off)
Cheers
  Oliver

```
>
>  ;======================================
>  pro testroutine1
>  end
>  ;======================================
>  pro testroutine2
>  end
>  ;=====================================
>  pro testroutine3
>  end
>  ;=====================================
>  pro benchmark
>
>  NREPS=5000L
>  NAVGS=50L
>
>  profiler, /SYSTEM, /CLEAR, /RESET
>  profiler,'testroutine1'
>  profiler,'testroutine2'
>  profiler,'testroutine3'
>
>  ;===== added this line ====
>  testroutine2
>
>  delt1 = dblarr(navgs)
>  delt2 = dblarr(navgs)
>  for j = 0L, navgs-1L do begin
>
>     ; TIME ROUTINE NUMBER 1...
>     tstart = systime(1)
>     for i = 1L, nreps do begin
>        testroutine1
>     endfor
>     delt1[j] = (systime(1)-tstart)
>
>     ; TIME ROUTINE NUMBER 2...
>     tstart = systime(1)
>     for i = 1L, nreps do begin
>        testroutine2
>     endfor
>     delt2[j] = (systime(1)-tstart)
>
>     ; UPDATE OUR PROGRESS...
>     print, 100*j/(navgs-1), format='($,"Progress: ",I4,"%",%"\R")'
>
>  endfor
>
>  ;==============
>  ; GET THE AVERAGE TIMES...
>  avg1 = delt1/double(nreps)
>  avg2 = delt2/double(nreps)
>
>  ;==============
>  ; GET THE MEANS...
>  mnavg1 = total(avg1,/DOUBLE)/double(navgs)
>  mnavg2 = total(avg2,/DOUBLE)/double(navgs)
>
>  ; TELL US ABOUT THE RESULTS...
>  print, mnavg1, format='(2(%"\V"),"Average1  = ",e16.7," s")'
>  print, mnavg2, format='("Average2  = ",e16.7," s",%"\V")'
>
>  ; WHAT DOES IDL CODE PROFILER REPORT...
>  print, 'IDL Code Profiler reports:'
>  profiler, /REPORT
>  profiler, /CLEAR
>
>  end; benchmark
```