

RE: /CLR floating point performance, inter-assembly function call performance

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.vc/2004-05/0211.html>

From: Yan-Hong Huang[MSFT] (yhhuang_at_online.microsoft.com)

Date: 05/06/04

Date: Thu, 06 May 2004 06:09:57 GMT

Hello Bern,

Generally speaking, the v1 JIT does not currently perform all the FP-specific optimizations that the VC++ backend does, making floating point operations more expensive for now. That may be why managed->managed is more expensive than managed->unmanaged in your test.

So for areas which make heavy use of floating point arithmetic, please use profilers to pick the fragments where the overhead is costing you most, and keep the whole fragment in unmanaged space.

Also, work to minimize the number of transitions you make. If you have some unmanaged code or an interop call sitting in a loop, make the entire loop unmanaged. That way you'll only pay the transition cost twice, rather than for each iteration of the loop.

By looking into ILCode, we can see that when InterOping, there are some extra IL instructions. So minimizing the number of transitions can save many IL instructions and improve performance.

For some more information, you can refer to this chapter online:

"Chapter 7 ;^a Improving Interop Performance"

http://msdn.microsoft.com/library/en-us/dnpag/html/scalenetchapt07.asp?frame=true#scalenetchapt07_topic12

Hope that helps.

Best regards,
Yanhong Huang
Microsoft Community Support

Get Secure! "C www.microsoft.com/security

This posting is provided "AS IS" with no warranties, and confers no rights.