

Re: Can Someone Change My Mind About .NET?

Source: <http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.general/2004-05/0025.html>

Olaf.Baeyens_at_skyscan.be

Date: 04/30/04

Date: Fri, 30 Apr 2004 14:21:13 +0200

> > *Windows core is still unmanaged and .NET programs must go through that transition every time. When Longhorn is out, it will be reversed. .NET will be faster than conventional executables now.*

>

> *That would be cool if that was the case. However I don't think that is so.*

> *Win32 is not going away with LH. WinFx classes will still need to call Win32 apis in many cases. Even if a class does not pinvoke a win32 api, why would it be any faster than a comparable method in native win32? Maybe the same, I don't see how faster. That would be nice however if wrong.*

>

Simple, more and more Win32 API is being replaced by .NET variants. That means that .NET programs have less and less transitions from managed to unmanaged to do.

But for existing Win32 executables, they will need Win32 API's that wrap functionality to the .NET core. That means that Win32 executables must go from unmanaged to managed code transitions. So they will slow down.

.NET programs will be faster because the CLR gets better and better. At this moment, conventional exe works at its best performance because of the compiler that optimizes it for the current known processors. If you do not recompile the .NET application and conventional program, you would see that the .NET version gets faster and faster compared to the conventional program because JIT v2.0 will optimize for the most recent processor while the conventional application is still stuck to the Pentium IV optimizations.

Conventional executables are static in nature. all methods and functions are laid out in a certain memory order at the moment you compiled and linked. .NET programs are dynamic in nature. Future JIT compiler might perform statistics what function is used how often, and might relocate it in a different memory block so the most used functions are grouped together

in a small memory address space, ready to be cached more often. Also .NET programs written in pure OOP way is now far more performant compare to the OOP version of a C++ unmanaged. because it does not need to delete small memory memory bloks every time stalling then program. .NET starts deleting the memory when it either needs memory, or if it can find some time to start cleaning up.

Just be aware! .NET is NOT an interpreter, It compiles the complete code to pure win32 code but optimized for your processor. When JIT matures, it ill become even better optimizer.

Another thing that must be taken into mind, is the fact that .NET programs also implements security at the low level standard.

You should compare Win32 functions with also this type of level of security build in. If I remember correctly a web page counter extension for Frontpage 97 could be misused to take over the server. A stupid prgram that only intended to show a bitmap on screen was misused. This is why all programs now created should have this security integrated default. Even if it is a simple screen saver, it could be misused by a worm. (examples enough: e.g. windows help file could be used to take over your computer) Maybe next Solitair....

But this is my opinion? I really believe that this .NET is the way to go, and I admit that I also fight against it to have my code work. ;-)