Subject: Re: How to promote U++ ?
Posted by mr_ped on Wed, 11 Jul 2007 09:26:44 GMT
View Forum Message <> Reply to Message

it just occurred to me those two question were not directly answered:
Quote:
1) is it really not difficult to convert mfc or jdk to u++?
2) is it a compiled u++ application running better than those coded in mfc and jdk?

1) I think it may be difficult to convert instead of writing UI part from scratch. The U++ is pretty efficient if written properly, and trying to convert some MFC in little steps may lead to somewhat cluttered U++ code?
So choosing a project with has good modularity and the UI part of it is both small and well disconnected from the rest of project may be good idea, if you don't want to do something really big.

2) depends. Firstly depends what compiler you use in U++. For windows platform the MSVC is pretty much a must if you want to compete against some other project which is compiled with MSVC. On linux everything is gcc.
And than it depends how much out application is efficient. In JDK some JIT optimizations of some algorithms can lead to an advantage over C++ statically compiled code, but most of the time Java is 20-50% slower, so JDK is by definition not on par with C++ (U++). If your Java app is faster/on par, than you screwed up the C++ version.
MFC is just UI. Most of applications performance don't depend too much on UI used, but the U++ looks more lean and mean than MFC, so it will very likely start faster. Otherwise the results should be similar.

And there is the STL vs NTL issue, where NTL should be much faster. So in case the project's performance depends on heave usage of containers, the conversion from MFC/STL to U++/NTL may lead to interesting boost in performance.

conclusion:
In the end you can't say U++ is better.
It is better in many cases. But sometimes it's not.
If you want to make U++ look good, you must choose the project for conversion wisely, because some project would not really benefit from such conversion and the whole effort may backfire than.

And still the main problem remains:
WHO would convert some project?