Subject: Re: Pick overloaded by Rvalue?
Posted by mr_ped on Mon, 24 Nov 2008 08:05:14 GMT
View Forum Message <> Reply to Message

I think as long as you stay "KISS" with your SW, C++ will be always superior in terms of performance.
If you let your code grow in complexity, it may be that "higher" language like Java/C# would help cut some complexity out, and protect you from some bloat, so the performance of C# vs C++ would be similar then. But that's not C++ failure or C# win, that's solely failure of code design.

Quote:So what we need is something that compiles to machine code without the complexity of C++. What is there now? What is coming up that merges the benefits of both C++ and interpreted languages?

The only thing forcing to use all the complexity of C++ are the underlying libs. If you use simple enough libs, or well encapsulated and well working, so you don't have to dig into them, then just your source decides the level of C++ complexity you have to manage. I mean, you can write very simple (and dumb) source in C++. But it may get quite larger/bloated, than the same functionality written with more complex C++.

Then at some point you have to adopt some source from someone else... and there you go, a) it's easier to write code, than read, b) he will certainly use some C++ features, with which you are not familiar enough and don't feel comfortably using it.

Also I think big part of that "C# is less complex" is strong platform API which helps you more times than C++ stdlib. Surely it's the language definition too, but when I write some simple GUI app with U++, it looks insanely simple. Thank to underlying U++ lib which hides lot of complexity.