Subject: Re: Impressive improvement in stl::vector when dealing with raw memory.
Posted by Lance on Mon, 14 Nov 2022 13:52:57 GMT
View Forum Message <> Reply to Message

Why is operating on raw bytes a big deal? Isn't Upp still doing a lot better on Upp::Moveable objects like Upp::String? Well, it's just a matter of teaching stl::vector to treat Upp::String(and Upp::Moveable as a whole) as raw bytes and it will catch up or even outperform.

Well it all starts with testing Upp code for C++20 compliance. let's try theide first. The ide compiles fine on both GCC and CLANG with -std=c++20 option, except some complaints on capturing this by default is deprecated in C++20, which are easy to fix or safe to ignore for now. But it's a total different story with MSC. with standard set to C++17, MSC rejects a bunch of stuff like

return somecondition? "a literal string" : AString;

These are also easy to fix if you don't mind your local version is slightly different from the main stream.

When standard is set to C++20 or c++latest, Upp::Moveable AssertMoveable0() is start to causing compilation failure, this one seems to be quite difficult to fix.

I was thinking it's just a mechanism to communicate to the compiler that it can treat object of this class as raw bytes, maybe we can do it differently with so much more facilities available in more recent c++ library.

So I start to do some experiment.