Subject: Compilation speed concerns
Posted by cbpporter on Tue, 11 Mar 2008 17:38:41 GMT
View Forum Message <> Reply to Message

After using C/C++ compilation model for all this time and experience with different sizes projects, I think that C and especially C++ scales horribly.

I know that BLITZ speed up things a lot, but the numbers in general don't look that great. I've tested a small application (about 25KB code) written in U++, not during first compilation, but during repeated compilations while altering only parts. I got two case, which alternate during development: one in which compilation lasts for about 3 seconds and linking for about 3.5, and one where compilation is 1 second and linking is 1.5. It must be because of incremental linking or other optimizations. But once in a while, you will get the big number, and if BLITZ kicks in for multiple files, you will get some extra time here also. I've also tested compiling TheIDE, and compilation short times are 4 seconds compilation, 7 seconds link, and long times scale up nicely, up to 25 seconds link.

I know that these numbers don't mean much and are far from a real benchmark. And even 30 seconds compilation + link is not that much. But as programs grow, the numbers grow also. For example, I'm working now on something which takes between 5 and 10 minutes to build on any trivial modification. Of course, modifying a header file which is included in most packages will result in over an hour build time. This leads to a nice fear of modifying header files, which is done only on a real need basis, leading to some poor design decisions, abstraction breaking, etc.

The same conclusion can be drawn while compiling open source software. I remember waiting 2-3 hours for my K3b, only because there was some problem with the official binary for my distro. This is total time, and I don't know how much it would take for a minor modification + link, but still enough, I wager.

This big difference was easy to notice recently, because in order to learn a new language/library, I generally start porting something that I have written. This approach really works for me. And while I was porting a Delphi application, I noticed that even though my current code base is about 5 times smaller, the minimum compilation time under C++ (one file changed) is a lot longer than the maximal compilation time under Delphi (full rebuild after clean).

Also, I've written my college graduation thesis's application in D, and full compile + link after clean was done instantly (well not really instantly , but fast enough that you barely managed to read the compilation message). Also, compiling only the std part of the standard D library (870KB of code) happens on the same machine in about 0.7 seconds, which is faster that the recompilation of a C++ file in the before mentioned 25KB test case.

Such examples make me really doubt the scalability of the C language family compiler performance.

Subject: Re: Compilation speed concerns
Posted by unodgs on Tue, 11 Mar 2008 19:59:20 GMT

View Forum Message <> Reply to Message

Yes, you're right. I remember writing applications in turbo pascal. After pressing Alt+F9 no matter how big the project was my app was run almost immediately. And I've got 486sx 25 then. C++ compilation is simply PITA. Most of the time I wait for the end of compilation and linking process. I'm trying to write as many as possible LOCs at once but it's not so easy in most cases. And yes D builds fast, but Mirek (me too frankly) dosn't like GC approach. The real pity is nothing's gonna change in the future about compilation speed in c++. C+0x is getting more complicated than was before and it sill have header <-> source split (I hate writing prototypes twice..). Fortunately 8 cores CPU are comming

Subject: Re: Compilation speed concerns
Posted by Mindtraveller on Wed, 12 Mar 2008 08:28:22 GMT
View Forum Message <> Reply to Message

Frankly speaking, C++ has sometimes too complicated syntax, which makes compilation of it a disaster for compiler. And, yes, D is simpler in that, more - it has a number of features which would fit well in C++. But also D has a number of disadvantages which make it no real opponent to C++ when we talk about maximizing efficiency and speed of code execution (to mention - GC makes D app behavour undetermined). You can read siscussion here:
http://www.ultimatepp.org/forum/index.php?t=msg&th=2873& amp;start=0&

So, I would vote for D, if it`s efficiency would be no worse than C++, but this isn`t so.

Subject: Re: Compilation speed concerns
Posted by cbpporter on Wed, 12 Mar 2008 08:52:29 GMT
View Forum Message <> Reply to Message

Yes, I know the reasons why C++ is so slow to compile, and have a good ideea why other languages mentioned behave so nicely.

I was not suggesting to switch over D. I remember the discussion. And unfortunately D is not ready. Only the experimental D 2.0 would have the right features to implement the same concepts and abstractions that are so common in U++ and the same efficiency level, and 2.0 is a mess right now. Maybe in 1-2 years.

And anyway, I was not proposing a switch for U++ or something like that. I was just expressing my frustration with this issue. I'm seriously thinking about a small professional reorientation towards anything which is serious work (no webdev or test scripts) and compiles fast .

Subject: Re: Compilation speed concerns
Posted by waxblood on Wed, 12 Mar 2008 19:50:04 GMT
View Forum Message <> Reply to Message

I'm looking forward for clang http://clang.llvm.org/ . As a C compiler seems very good and efficient, too bad they say "we don't expect to have respectable C++ support for another 2 years or so."

Still it is possible to use the LLVM compiler on which clang is based to compile C++ code via the gcc parser. LLVM structure is much more open and hackable than the gcc one and, as the name suggests, it is even possible to run intermediate code produced by LLVM on a virtual machine to perform runtime code analysis and post-run code optimization.

David
