Subject: Experimental CUDA support
Posted by mirek on Fri, 14 Feb 2025 13:13:57 GMT
View Forum Message <> Reply to Message

For now, only Windows/Visual C++.

See reference/Cuda

Subject: Re: Experimental CUDA support
Posted by Tom1 on Fri, 14 Feb 2025 14:16:19 GMT
View Forum Message <> Reply to Message

Hi,

Very interesting development! :)

Maybe I should get some NVIDIA hardware to test this.

I wonder if Intel oneAPI could be supported too in a similar way...? That would offer cross-platform compatibility for Windows, Linux (and gradually MacOS) for hardware from NVIDIA, AMD and Intel. Therefore, oneAPI looks like a good fit for u++, but of course I don't know the internals.

Best regards,

Tom

Subject: Re: Experimental CUDA support
Posted by mirek on Fri, 14 Feb 2025 19:02:36 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Fri, 14 February 2025 15:16Hi,

Very interesting development! :)

Maybe I should get some NVIDIA hardware to test this.

I wonder if Intel oneAPI could be supported too in a similar way...? That would offer cross-platform compatibility for Windows, Linux (and gradually MacOS) for hardware from NVIDIA, AMD and Intel. Therefore, oneAPI looks like a good fit for u++, but of course I don't know the internals.

Best regards,

Tom

Uhm, too soon to tell. Quick glance at oneAPI and the SYCL authors does not bring too much confidence.

Originally I tried with Vulkan / GLSL, but that is really unusable - I am not bothered with 500+ lines to run simple kernel, but GLSL is unusable language for anything complex. It is like C without pointers... Meanwhile, CUDA is for years full blown C++ with hardly any limitations. Impossible for oneAPI / SYCL to catch up with something incompatible IMO.

Amd HIP on the other hand - that one looks promising.

Right now I am a bit unhappy that CUDA requires Visual C++ compiler, but maybe a bit of effort I might be able to persuade it to use mingw-clang instead...

Subject: Re: Experimental CUDA support
Posted by zsolt on Sat, 15 Feb 2025 22:08:22 GMT

View Forum Message <> Reply to Message

I'm absolutely not an expert, but AFAIK nvidia-cuda-toolkit on Ubuntu can be used in Clang or G++ projects.

Subject: Re: Experimental CUDA support
Posted by mirek on Sun, 16 Feb 2025 08:54:52 GMT

View Forum Message <> Reply to Message

zsolt wrote on Sat, 15 February 2025 23:08I'm absolutely not an expert, but AFAIK nvidia-cuda-toolkit on Ubuntu can be used in Clang or G++ projects.

In Linux yes. In Win32, no way - you need to use Visual C++. At least that is Nvidia's position.
https://stackoverflow.com/questions/68726422/is-it-possible-to-integrate-nvcc-with-gcc-g-in-windows

Now nvcc seems to work by separating cuda code and host code, then compiles cuda code and creates a new c++ file which combines host code and compiled cuda code (as arrays like

unsigned long long fatbinData[]= {0x00100001ba55ed50ull ....
)

Then this file is compiled by host compiler. There are explicit checks in CUDA headers for MSC++ compiler (#ifdef MSC_VER). There might also be a problem with CUDA runtime library, but I doubt that...

I believe that if I take clang-cl and give that to nvcc, with some additional trucks like -D MSC_VER it should work. Do not have time right nor energy now to test right now...

Subject: Re: Experimental CUDA support
Posted by zsolt on Sun, 16 Feb 2025 13:41:23 GMT

View Forum Message <> Reply to Message

Ah, I see.