From: **David S Lawrie** dlawrie@usc.edu
Subject: Re: Philox and CUDA suggestion
Date: January 30, 2015 at 11:36 AM
To: Moraes, Mark Mark.Moraes@DEShawResearch.com

Hi Mark,

We had been in contact previously about a problem I had come across using Philox and I wanted to let you know that I think I have solved it. To remind you what it was, I want to instantiate Philox in a function that was both for the host and device. I got 32-bit Philox working, but had trouble with 64-bit. The problem is that the CUDA instrinsic __umul64hi is device only and the GCC version wouldn't work because NVCC doesn't recognize __uint128_t, so R123_USE_GNU_UINT128 had to remain 0. Thus, R123_USE_PHILOX_64BIT was being set by compilerfeatures.h to 0 as no methods to do MULHILO64 were available.

Thus I turned `R123_USE_MULHILO64_ASM` on in the event that Philox was compiled on a host function through NVCC. This allows Philox32 and Philox64 to be compiled when called in both host and device functions using the CUDa intrinsic when compiled on the device and ASM when compiled on the host. This allows me to generate some random numbers on the CPU and some on the GPU which gives me greater flexibility for my simulation and less need to transfer numbers back and forth between the CPU and the GPU, streamlining and accelerating my code. This may be a capability that other researchers would be interested in. Attached is the new nvccfeatures.h file and my test program.

Cheers,
David


nvccfeatures.h


test_philox.cu

On Oct 9, 2014, at 1:47 PM, David S Lawrie <dlawrie@usc.edu> wrote:

> No worries, I think it would be neat functionality to add to the package.
>
> Cheers,
> David
>
> On Oct 7, 2014, at 1:31 PM, Moraes, Mark <Mark.Moraes@DEShawResearch.com> wrote:
>
> > Er, still on my to-do list, sorry. I'll try to catch them this week.
> >
> > ---
> >
> > **From:** David S Lawrie
> > **Sent:** Friday, October 03, 2014 7:50:24 PM
> > **To:** Moraes, Mark
> > **Subject:** Re: Philox and CUDA suggestion
> >
> > Hi Mark,
> >
> > I was wondering if your CUDA gurus ever got back to you?
> >
> > Cheers,
> > David
> >
> > On Sep 10, 2014, at 12:59 PM, David S Lawrie <dlawrie@usc.edu> wrote:
> >
> > > Hi Mark,
> > >
> > > The problem was that I could not instantiate a Philox instance inside a host function when the program was compiled with nvcc. For the simulation I am writing, I wanted to create some random numbers on the host and others on the device. The attached code should not compile with the standard nvccfeatures.h but should compile with mine. However, I have noticed with my nvccfeatures, 64-bit Philox no longer compiles (compiler does't recognize it), so I've broken something unfortunately. While I'm only using 32-bit Philox and so it doesn't affect me, something about my ifdefs is not quite right and should probably be fixed. There may be more bugs caused by them, but that's one I found. My guess is the handling of `R123_USE_MULHILO64_CUDA_INTRIN` is not right, I needed to make sure it is only defined when compiled for the device since the host function cannot call the device function intrinsic, but now the compiler says Philox4_64 is undefined if I try to substitute it in for Philox4_32.

Cheers,
David

<test_philox.cu>

On Sep 10, 2014, at 9:29 AM, Moraes, Mark <Mark.Moraes@DEShawResearch.com> wrote:

> Hi,
>
> If I understand you correctly, you're saying that the problem is that the current ifdefs don't allow you to include any of the Random123 headers in a host-side program that's compiled with nvcc? That probably escaped my test cases because I tend to just compile all host-side files with gcc and only compile device source files with nvcc and then link the whole lot, so we never noticed this issue before. Hmm, with a couple of different versions of CUDA, I tried
>
> $ nvcc -O -I../include kat_c.c -o kat_c_nvcc
> $ ./kat_c_nvcc
> 4 test vectors of type aesni4x32 skipped
> 3 test vectors of type ars4x32 skipped
> Perform 60 tests.
> PASSED 60 known answer tests
>
> so it seems nvcc is ok if it just sees host-side definitions for the R123 generators. Do you have a small code example that won't compile with the current header files?
>
> The extra ifdefs that you added sound sensible to me, I'll consult with our CUDA gurus, give them a spin and if they pass all the test environments, will put them or something like them in the next release.
>
> Thanks,
> Mark.
>
> -----Original Message-----
> From: David S Lawrie [mailto:dlawrie@usc.edu]
> Sent: Friday, September 05, 2014 10:11 PM
> To: Desres Random123 Feedback (World)
> Subject: Philox and CUDA suggestion
>
> Dr. Salmon/Moraes,
>
> I've been using your Philox random number generator in a CUDA program and I needed to use the rng on both on the host and the device in the same program. I noticed that when compiled by the nvcc compiler, only a device-side rng was permitted. So I modified the nvccfeatures.h file (modified file attached) with #ifdef __CUDA_ARCH__ ... #endif strategically placed. This allows the compiler to only instantiate those commands if compiled for the device and if not, fall back on the gcc and mvcc compiler definitions. It seems to work (generating random numbers correctly on both the GPU and CPU), but I've only tested it for Philox4 32-bit. This seems like something you might consider including to allow for more flexible usage in CUDA? Do you have any suggestions to improve my changes? or is there a reason not to what I have done at all?