ROCm backend for HPX.Compute

Google summer of code proposal

Mentors: Mikael Simberg

Personal Details

- **Student** : Yanlong Li
- **IRC** : DragonMain
- **University** : Weifang University
- **Major** : Computer Science And Technology
- **Degree Program** : University Undergraduate
- **Email** : dragonroot2018@gmail.com
- **Github** : https://github.com/Dragon20180618
- **Time Zone** : UTC+8:00 (Beijing, China)
- **How much will I plan to spend on my GSoC** :

  I think, If you are interested in my work, I will never feel tired and bored.

  I can spend the most of the time to do the project after lectures.
- **Start time** : from now on .

Skills

I am interested in parallel computing. And I have done some easy research in CUDA, Message Passing Interface and OpenMP.
When I found the CUDA project of HPX, I felt exited. If I will be selected, I think it's my honor.
During the study of CUDA, I tried to achieve Molecular dynamics by the cuda speed.

- language :

  C++ : 3 __as like `extern` ,`enum`, `Override`, `class`,`namespace`,

  Git : 4 __before half a year. I code a `Golang` project with a friend, I learned git. 329 contributions in the last year

  CUDA : 3__`stream` `SM` `manage memory` `mem-check`...

  Python: 2 `Pygame` `pyautogui` `pyCUDA`...
- HPX matrix :

  https://github.com/Dragon20180618/GSOC_2020

Project Proposal

- **Problem** : ROCm backend for HPX.Compute

- **Solution** : I have checked your demands. So, The first thing is through and understand your CUDA code. At the same time, I will spend time on ROCm. After that, Exploring the interface or

  the other way is to come true the ROCm. If It is not simple and efficient, I can change `CUDA` to `HIP`. As you know, it will allow a single implementation to be used for both AMD and NVidia GPUs. HIP is very similar to CUDA. In fact, the most of instructions are just different names. For example hipMemcpy, hipMalloc, hipFree. Before that, I had some contact with HIP. This will allow me to complete the task more quickly.

- **Result** : CUDA code will be replaced by HIP code. Dramatically improve code portability. The backend is comparable with CUDA. It can schedule some algorithms, `cuFFT`,`parallel for-loop`. Simplify the HPX-CUDA. Now CUDA 10.2 has updated.

Proposed Milestones and Schedule

It's my first time to have a wonderful adventure in GSoC. So, I learned the Google Schedule carefully.

The Full Program Timeline

- **2020.4** get ROCm message, and Read your CUDA code.
- **2020.5** Community Bonding. Fully understand what is HPX and what can it do.

  learning more about HPX organization’s community.

Coding

week 1

Perform the basic CUDA-ROCm connection. I will convert the direct conversion program to ROCm. The ones that are not directly convertible are pending. Recording the advantage of ROCm.

week 2

I will schedule some easy algorithm in ROCm, as like `for-loop`. And I will guarantee the original backward compatibility feature.

week 3

Converting the main CUDA program to HIP. Based on the accumulated CUDA to ROCm experience, this time I turn the CUDA to HIP will be faster. Good code is changed. If I find that the code in the CUDA source program needs to be updated, I will modify the source code to some extent, such as making multiple GPU calls through OMP. cudaStream_t multi-stream processing.

week 4

Completing remaining the CUDA to HIP. And I have checked your hpx cuda file, as like get_cuda_targets.cpp. and your examples such as cublas reality. Do I need to copy the newer CUDA include files to include/hpx/compute/ ?

Maybe `libs_compute_cuda_api` is the better choice.

week 5

Adding algorithm to HIP, `parallel for-loop` test.

week 6

Comparing the reality of ROCm with the reality of HIP. Get the best reality or we can create the third way of HPX-CUDA. Debug them.

week 7

Adding implementation of advanced algorithms, as like `cuFFT`, `cuBlas`, `Thrust`. Give `scoped_active_target.hpp` one optimization. One by one streams.

week 8

Final test.

- **2020.8** Submit Code and Final Evaluations

Here are the awards I have received related to this project

- Blue Bridge Cup Regional 5th
- Blue Bridge Cup National Programming Competition second prize.

> > I think two months for coding maybe get a better software. Not only the basic power, may be we will let hpx-cuda be much better.