

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 time do I plan to spend on my GSoC :
In my opinion, If you are interested in your work, you will never feel tired and bored.
I can spend the most time to do the project after lectures.
- **Start time** : from now on .

Background Information

I am interested in parallel computing, so , I have done some Research in

CUDA, Message Passing Interface, OpenMP

When I found the CUDA project of HPX, I felt exited. I think it's my honor. It's belong to me.

During the study of CUDA, I tried Molecular dynamics by 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 which you kindly ask:
https://github.com/Dragon20180618/GSOC_2020

Project Proposal

- **Problem** : ROCm backend for HPX.Compute
- **Solution** : I have check your demands. So, The first thing is look through and understand your CUDA code. At the same time, I will spend time on ROCm. After that, explore the interface or other way to come true ROCm. If It is not simple and efficient. I will check HPI. change `CUDA` to `HIP`. As you know, it will allow a single implementation to be used for both AMD and NVidia GPUs.
- **Result** : 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 the wonderful adventure of 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 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 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 CUDA to HIP. And I have Check your hpx cuda file, as like get_cuda_targets.cpp. and your examples for example: cublas reality. And In my opinion, If I need to copy the newer CUDA include files to include/hpx/compute/ ?

Maybe `libs_compute_cuda_api` is the better choice.

week 5

add 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

Add 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 coding maybe get a better software. Not only basic power, may be we will let hpx-cuda be much better.