Weekly Report - Decomposing Ethash(Shared Memory Implementation)

MSc Project
Runchao Han

February 3, 2018

1 Progress of the Last Week

1. Decomposed Ethash (Shared Memory Implementation) with the help of a sequence diagram
2. Read paper about optimising CUDA programs

2 Decomposing Ethash

The Ethash has two CUDA implementations, both of which have been commented in detail:

- dagger_shared.cuh[1]
- dagger_shuffled.cuh (CUDA $\geq$ 9.0)[2]

Where the shuffled version is more efficient than the shared memory version.

Ethash can be described as shown in Fig. 1, which starts from hashing the header and a random nonce, then mixes it with random bits in the DAG(approximately 1G) for 64 times, where accessing DAG is the performance bottleneck.

The meaning of every line of the code is commented in the Github project above. At present, the shared memory implementation has been totally figured out. The next step is to improve its performance.

3 How to Optimise a CUDA Program

The shared memory version has several *__syncthreads*(), regarded as barriers. Obviously, this program is not well optimised(The shuffled version has not been well analysed by me yet).

A highly cited paper from UIUC and Nvidia[1] illustrates some principles of optimising a CUDA program:

- Leverage zero-overhead thread scheduling to hide memory latency

[1] `https://github.com/SebastianElvis/ethminer/blob/master/libethash-cuda/dagger_shared.cuh`

[2] `https://github.com/SebastianElvis/ethminer/blob/master/libethash-cuda/dagger_shuffled.cuh`

- Optimize use of on-chip memory to reduce bandwidth usage and redundant execution
- Group threads to avoid SIMD penalties and memory port/bank conflict
- Threads within a thread block can communicate via synchro- nization, but there is no built-in global communication mechanism for all threads

This week the target is to optimise the Ethash shared memory version according to this paper.

4 Next Week's Plan

1. Attempt to optimise Ethash shared memory version
2. Figure out Ethash shuffled version
3. Examine CryptoNight algorithm
4. Import CryptoNight miner to Eclipse

References

[1] Shane Ryoo, Christopher I Rodrigues, Sara S Baghsorkhi, Sam S Stone, David B Kirk, and Wenmei W Hwu. Optimization principles and application performance evaluation of a multithreaded gpu using cuda. In *Proceedings of the 13th ACM SIGPLAN Symposium on Principles and practice of parallel programming*, pages 73–82. ACM, 2008.

Appendices

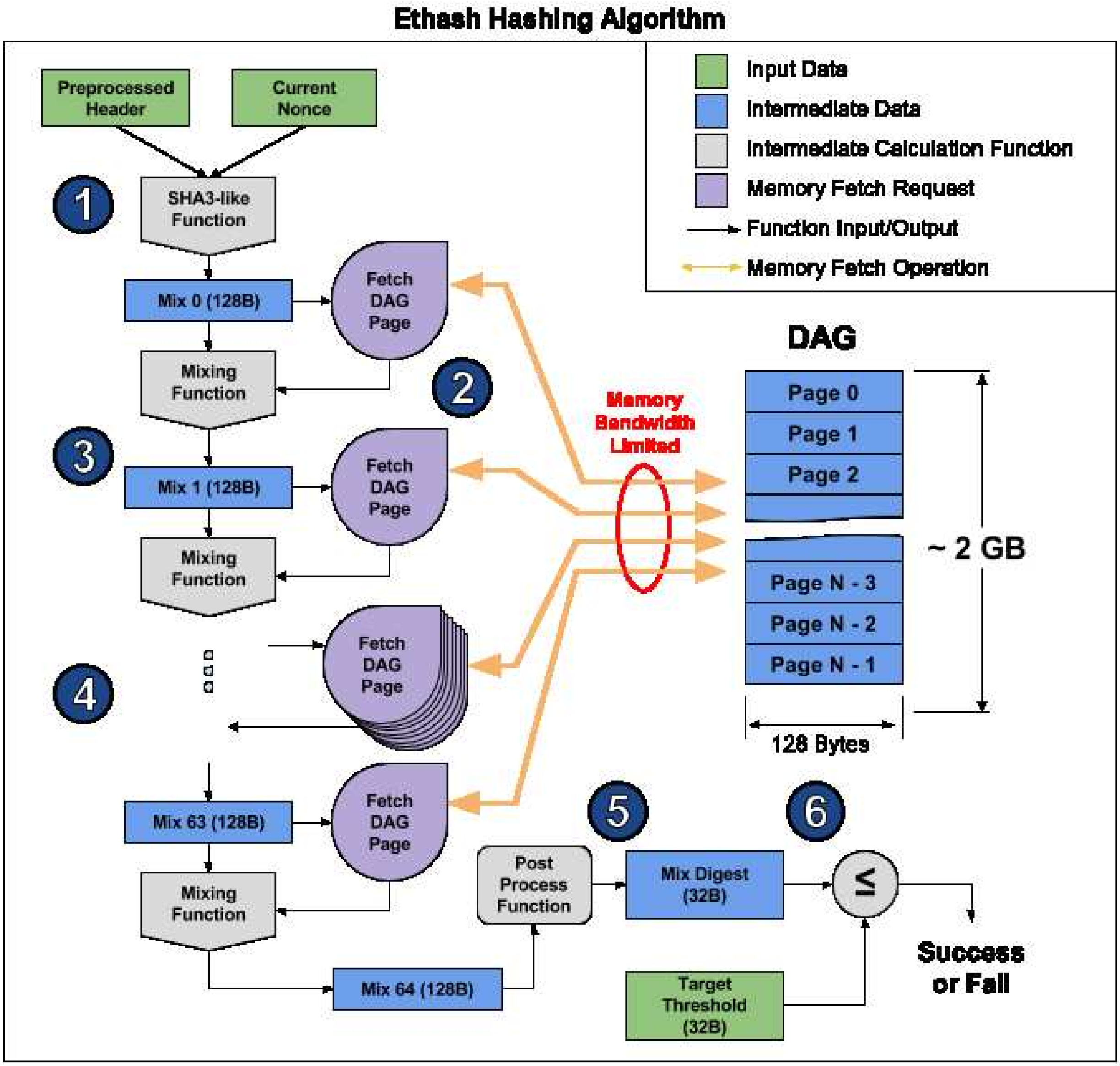

Figure 1: The process of Ethash mining algorithm