Research on accelerating GPU of pollution tracking system under block chain technology

Ying Hou\textsuperscript{1,,*}, Tingdong Yu\textsuperscript{2,\textasteriskcentered}, Yining Wang\textsuperscript{3,\textasteriskcentered}

\textsuperscript{1}School of Water Resources and Environment, China University of Geosciences (Beijing), Beijing 100083, China
\textsuperscript{2}School of Information Engineering, China University of Geosciences (Beijing), Beijing 100083, China
\textsuperscript{3}School of Government, Beijing Normal University, Beijing 100875, China

*Corresponding author e-mail: 1005162117@cugb.edu.cn,
\textsuperscript{a}1006151220@cugb.edu.cn, \textsuperscript{b}wangyining_nsh@hotmail.com

Abstract. Nowadays, there are more and more environment problems around us. It will be a hard work to track pollution with computer, because of its huge calculated amount. To speed up this process, we use GPU instead of CPU to compute, and use block chain technology. With the two powerful tools, we can significantly speed up the computation speed.

1. GPU Acceleration Technology

GPU, graphics processing unit, popularly speaking, is the core of computer graphics, which is a processor designed for highly parallel computing with high density computing. Compared with CPU, GPU has a large number of transistors to process massive data, but GPU weakens the data cache and stream processing \cite{1-2}. Most of the tasks are controlled by CPU, but computed by GPU, in other words, CPU is leader, and GPU is worker.

Among all GPU unit, there is a core unit called GPGPU. GPGPU, or general-purpose graphic processing unit, is used for tasks other than computer graphics processing, such as physical computing, encryption and decryption, scientific computing, and the generation of virtual currencies such as bitcoins. Because graphics cards are built for large-scale parallelism, even the most powerful CPU still has a low computational speed compared with GPU in the face of a large number of parallel tasks. So far, the most high-end GPU has thousands of computing cores \cite{3-4}, but CPU has only a dozen cores. The GPGPU model is shown in Figure 1.
As shown in Figure 1, there are several SMs in the GPGPU model, namely Streaming Multiprocessor, which can be regarded as a core of the GPU. Each stream multiprocessor contains a large number of cells, a small number of instruction caches, and shared memory [5-6]. The stream multiprocessor communicates with host data through global memory, constant memory or texture memory.

2. Pollution tracking system based on block chain technology

2.1. Block Chain Architecture
In order to better understand block chain technology, this section introduces the architecture of block chain. As shown in Figure 2, it is a five-tier architecture of block chain technology. From bottom to top are data layer, network layer, consensus layer, incentive layer and intelligent contract layer. These five layers will be briefly introduced below.

The data layer is at the bottom level, which is the foundation and one of the core contents of the block chain technology [7-8]. The second layer is the upper layer of the data layer, which mainly provides the connection and communication functions between network nodes. The third level consensus layer is the core part of the block chain technology. The fourth level is incentive layer, as the name implies, it mainly promotes the development of the system in a better direction through the issuance and distribution mechanism of block chains. The fifth level is intelligent contract layer. It
mainly provides the programmable features of the block chain, encapsulating all kinds of algorithms and scripts and intelligent contracts [9].

2.2. Technical Characteristics of pollution tracking system
Block chain technology mainly presents four characteristics, which will be briefly introduced in this section. First, decentralization. In block chain technology, the system does not have a centralized management organization or hardware facilities. Any problem of any node in the system will not affect the normal operation of the system. Each node is equally spaced. Second, distrust. there is no need to establish trust rules when data exchange between each network node in the system since the whole system operates according to open and transparent rules. Third, maintenance collectivization. There are many nodes in the system with maintenance function, so the data in each block of the system is jointly maintained by these nodes, and it is not maintained by a maintainer with special permission. Fourth, system stabilization. Block chain ensures that every node in the system can obtain a complete copy of the database through distributed storage, which can avoid the risk of misuse of database by single node. The more nodes in the system, the higher the stability of the system [10].

3. Practical Byzantine fault-tolerant algorithm
BFT refers to Byzantine fault tolerance [11]. The Byzantine hypothesis is a modeling of the real world, where computers and networks may behave unexpectedly due to hardware errors, network congestion or interruptions, and malicious attacks. As the occurrence of malicious attacks and software errors will be increasingly frequent, resulting in the failure of nodes to generate arbitrary behavior. Early Byzantine fault-tolerant algorithms were either based on the assumption of synchronous systems or were inefficient because of their low performance. Therefore, PBFT (Practical Byzantine Fault Tolerance) was proposed. PBFT algorithm solves the inefficiency of the original Byzantine fault-tolerant algorithm, reduces the complexity of the algorithm from exponential level to polynomial level, and makes the Byzantine fault-tolerant algorithm feasible in practical system applications [11]. The PBFT algorithm flow is shown in Figure 3. A complete consensus shall undergo five stages: request, pre-prepare, prepare, commit and reply.

4. Design of PBFT Algorithm Based on GPU Acceleration
First step: To set the environment variables of GPU kernel and initialize the GPU. It is necessary to consider information such as the number of stream processors and the size of memory.

Second step: PBFT parallel processing. After receiving the processing node from the host end, the GPU allocates one thread for fault tolerance processing for several nodes via the thread execution manager. One of these nodes is selected, and the rest is generated from the node, which controls the generation of new blocks. In fault tolerant processing, each thread is treated internally by the following steps:
1) All nodes are equally divided according to the number of nodes in the block, and a primary node is selected from each part after the equalization.

2) The master node numbers the requests from the host and sends the pre-prepared type information to all the slave nodes. After all the nodes have received the information, the number of the request is ready if agreed.

3) If the slave node agrees with the master node request number, a prepare type message is sent to the master node and other slave nodes. Otherwise, it means disagreement.

4) After entering the prepared state from the node, a COMMIT type message is sent to all other nodes to tell them that it has a prepared authentication certificate. If the slave node receives 2F+1 items of submitted information, it proves that the slave node has entered a committed state.

5) Finally, the above-mentioned nodes in the commit state are sent back to the host side, which can submit new blocks and trade them to the local block chain and state database.

Third step: Call the GPU kernel. The kernel of GPU needs to be called on the CPU side, and each thread in the kernel will have a unique identification. In the parameters of the kernel function, you can specify the thread block dimension and the size of each dimension to determine the number of parallel threads. The number of parallel threads has an upper bound, the computation speed will be slow down if overflow. Generally speaking, the size of a common thread block is an integer multiple of the GPU stream processor, thus avoiding the idle flow processor.

5. Experimental result
The development environment of this experiment is mainly based on a PC, and the experimental environment is based on one server. The server contains three GPU, one of which is selected here, and the basic configuration is shown in Table 1.

| Development Environment | Operating system | Editor |
|-------------------------|------------------|-------|
| CPU Intel (R) Core (TM) i5-4590 | 8GB RAM | 100G | Ubuntu14.04 LTS | Parallel Nsight 2.1 |

| Testing environment | GPU GeForce GTX 1070 | Memory size | Stream processor number | SMX quantity | Maximum thread count in Block |
|---------------------|-----------------------|-------------|-------------------------|-------------|------------------------------|
| | 8GB | 1920 | 15 | 1024 |

This experiment compares the execution speed of traditional BFT algorithm, PBFT algorithm and PBFT algorithm after GPU acceleration, as shown in Figure 5.
As shown in Figure 5, the processing speed of the three is not significantly different when the number of nodes is 100. With the increase of the number of nodes, the processing time of BFS algorithm increases exponentially, while the PBFT algorithm optimized on the basis of BFS algorithm consumes less time than BFS algorithm, but the effect is still unsatisfactory. The PBFT algorithm based on GPU acceleration proposed in this paper has obvious advantages after the number of nodes reaches a certain number, and the programs running with GPU are nearly 30 times faster than those without.

6. Conclusion
At present, block chain technology has developed rapidly in a few years since it was put forward. Apparently, it has been developed from concept to maturity and has been applied to many fields such as banking, medical and insurance companies. For block chain technology, the practical Byzantine fault-tolerant algorithm of the consensus layer in the architecture is accelerated by the powerful parallel computing characteristics of GPU in this paper. It can reduce the execution time of the algorithm so as to optimize the overall performance even though the whole implementation process is complicated. In addition, relevant scholars can also accelerate the asymmetric encryption algorithm, which will be the subsequent research work in the future.

References
[1] J. van der Geer, J.A.J. Hanraads, R.A. Lupton, The art of writing a scientific article, J. Sci. Commun. 163 (2000) 51-59.
[2] Beck R, Czepluch J S, Lollike N, et al. BLOCKCHAIN – THE GATEWAY TO TRUST-FREE CRYPTOGRAPHIC TRANSACTIONS [C]// Twenty-Fourth European Conference on Information Systems. 2016.
[3] CCID Consulting. Structure And Application Of Block Chain Development. Software And Integrated Circuits, 2017 (10):20-24.
[4] Vuduc R., Chandramowlishwaran, A., Choi, J., Guney, M., & Shringarpure, A. (2010). On the limits of GPU acceleration. Usenix Conference on Hot Topics in Parallelism (pp.13-13).
[5] Han T D, Abdelrahman T S. hiCUDA: High-Level GPGPU Programming [J]. IEEE Transactions on Parallel & Distributed Systems, 2010, 22 (1):78-90.
[6] Kosba A, Miller A, Shi E, et al. Hawk: The Block Chain Model Of Cryptography And Privacy Preserving Intelligent Contract [C]// Security and Privacy. IEEE, 2016: 839-858.
[7] Al-Akhras M T, Aleidi B, Albesher A. Using Blockchain Technology to Validate the Integrity and Confidentiality of Backups Versions on the Cloud [C]// Al Yamamah Forum on Telecommunications and Information Technology. 2018.
[8] Ge Hui, Yang Qilu, Liang Yingchun, et al. Field Test And Analysis Of Subway Track Structures At Different Speeds [J]. Railway Standard Design, 2017, 61 (1): 11-15.

[9] Wang, Maoning, M. Duan, and D. O. Information. "The Second-preimage Attack to Blockchain Based on the Structure of Merkle Hash Tree." Netinfo Security (2018).

[10] Li He, Sun Jifei, Yang Yong, et al. A Preliminary Study of Etaifang Based on Block Chain 2.0. China Financial Computer, 2017 (6): 57-60.

[11] Yuan Yong, Wang Fei Yue. Development Status And Prospect Of Block Chain Technology [J]. Acta automatization Sinica, 2016,42 (04): 481-494.