Final Project Proposal – Parallel Computing 2020-Spring (Plan A)

Name: 王元廷/ ID:106062119

0. Topic: Accelerating blockchain implementation using parallel computing techniques

1. Motivation

In the current implementation of blockchain, one of the most significant disadvantages is its efficiency issue. Take the bitcoin blockchain as an example, a block is designed to be mined once every ten minutes, which makes its throughput much less than Visa, the mainstream online payment system around the world at present. The main bottleneck of the Bitcoin blockchain is the Nakamoto consensus, or the Proof-of-Work consensus, for which researchers has been struggling recently to find an alternative that is as decentralized and secure as PoW but with a better throughput of transaction per second.

Replacing the mining protocol of blockchain is not closely related to the domain of parallel computing. In fact, it is the major goal of my undergraduate project. However, there are other parts in blockchain which could be boosted using parallel computing concepts, which has been instructed throughout this semester. For example, HW4 boosted the performance of mining through spawning the tasks of finding nonce into threads. I have identified some parts that could also be improved by applying parallel computing concepts:

i. The part of finding UTXOs (Unspent Transaction Output) when initializing and verifying transactions.

ii. The part of calculating merkle tree root/ merkle patricia tree when constructing block headers.

iii. The part of getting latest blocks/block headers by sending *getblock()* messages to peer nodes in the blockchain P2P network.

2. Implementation Plans

After deliberation, I think the first two parts mentioned above are feasible for me to take as the term project of the course, mainly because the due date is only roughly a month later. Thus, I would focus on parallelizing the part of finding UTXOs (to get the remaining balance of a wallet) and calculating the root of the merkle tree (or merkle patricia tree for my project’s implementation) in block headers.

Firstly, I would implement the sequential version of the two parts, and then test their performance. Afterwards, I will try to distribute the tasks in the view of parallel computing. Then again, I would implement the parallel version of them and also test their performance. Last but not least, I will compare the results and try to further optimize the code based on the performance tradeoff experiments.

3. Tentative Schedule

By Jun. 6th: Sequential version of code

By Jun 10th: Decomposition and assignment of tasks

By Jun 20th: Parallel version of code

By Jun 30th: Comparison and discussion of result

4. Expectations

The expectations are different between the two parts.

For the first part, since the UTXOs are harder and harder to find when the blockchain gets longer and longer, I expect that the parallelized version of this part should perform significantly well when the scale gets larger.

For the second part, due to the upper bound of transaction counts implicitly set by block size, I do not expect this part to boost in a human-distinguishable way. Instead, I simply expect it to perform better than O(n^2), which the sequential code performs.