Blockchain programming - Lab 04

Ometita Radu - Adrian

March 2021

1 Review of lecture and assignment

Answer any questions about the lecture or assignment.

2 Questions

1. Is there a limit to the number of contracts being accessed by a single contract?

2. Why can't you have contracts inside of contracts?

3. A centralized system can be stopped at one poit. But blockchains, given that they are decentralized, can they? Can a blockchain be killed, stopped?

4. I believe that in future, proof of stake will take over proof of work. There are already multiple methods of PoS like multi-round voting, coin aging, delegate or random selection. The problem with PoS is that the user with more stakes has more chances of writing the next block. But probably in future there will be an algorithm that will be more fair and not so costly as PoW. What is your opinion?

5. Was the concept of "gas" introduced in Ethereum only for making sure that the users won't exhaust the blockchain or has it another purpose?

6. In the third CryptoZombies course, specifically in the lesson #5, programmers are encouraged to encode timestamps in 32-bit (which Solidity does by default unless you specifiy otherwise) instead of 64-bit

to "save gas". With the year 2038 and the bug that will break all systems that run 32-bit timestamps approaching, isn't this like practically begging your whole framework/programming language/contract to become obsolete earlier than it should? Does a 64-bit timestamp have that much of an impact on gas cost? Why doesn't Solidity encode timestamps in 64-bit by default or choose another epoch to start from (like OpenVMS) or maybe take a different approach to encoding (like Javascript does)?

7. In the idea of combining blockchains with AI: training a model will use lots of resources thus costing lots of money to run. Could you maybe train a model on your own machine and then only deploy the weights on the blockchain?

8. Is it possible to apply zero knowledge proofs to the Ethereum protocol without requiring significant changes to the transaction structure? The end goal is to essentially anonymize the 'to', 'from' and 'value' fields of a transaction. Can this be done without changing the datatype of these fields or introducing new ones? (potentially requiring a hard fork)

9. In the course it states that a contract must be executed by every node from the blockchain. Does that mean that each computer which run the blockchain has to run each contract? Or a node is something else?

10. What benefits can we have if we run an Ethereum node on our machine? Does a regular computer have the enough data processing power to run a node?

3 Final assignment

Discussing ideas for your final assignment.