Solidity Smart Contract Optimization

Learn how to optimize your code to save gas

Daniel (Sơn) PHAM

Solidity Developer Bootcamp

Ngày 3 tháng 8 năm 2023

Outline

1. Why Optimizing Gas Is A Challenge?
2. Solidity Gas Optimization Techniques
3. QnA

Optimizing gas is a challenge

Optimizing gas consumption is a challenge for developers because it requires a deep understanding of how smart contracts work and the intricacies of the Ethereum Virtual Machine (EVM).

Gas optimization is tough competition.

In addition, the cost of gas can vary greatly depending on the complexity of the smart contract and the amount of data that needs to be processed. This can make it difficult to predict a smart contract's gas cost accurately.

Solidity Gas Optimization Technique

- Use immutable and constant
- Pack your variables
- Use mappings instead of arrays if posible
- Use fixed-size arrays
- Avoid large arrays or iterations
- Free up unused storage
- Cache in a memory variable if a storage slot read multiple times
- Store iata in calldata instead of iemory for read only params
- Cache into stack if a memory variable read multiple times
- Use uint8 can increase gas cost
- Remember to Add view and pure to functions
- Batch operations
- Enable the solidity compiler optimizer
- Minimize on-Chain data

Solidity Gas Optimization Technique

- Use unchecked if possible
- Use ‘(uint256) a != 0‘ instead of ‘a > 0‘
- Hard code gas when calling if possible
- Use library
- Remove redundant checks
- Use assembly code
- Keep 1 wei of tokens inside a smart contract

QnA