Ethereum: A Blockchain Platform with smart contract support for Distributed Application Development

Tinu N.S[#1]

[#1]*Department of Computer Science and Engineering , New Horizon College of Engineering*

[1]tinuns@newhorizonindia.edu

***Abstract*— Blockchain technology is gradually getting integrated with multidisciplinary domains due to many of its features like append only distributed ledger, immutability, verifiability, consensus, cryptocurrency and support of smart contract. Ethereum is one of the mile stone development in the field of blockchain technology as it supports for smart contract functionality and native cryptocurrency called ether. Ethereum uses its decentralized virtual machine called the Ethereum Virtual Machine (EVM) inorder to execute the smart contracts (business logics).This paper focus on the details of ethereum platform and its features and capabilities.**

***Keywords*— Ethereum, distributed ledger technology, EVM, smart contract, ether, gas.**

I. INTRODUCTION

Ethereum was proposed by Vitalik Buterin, who is a cryptocurrency researcher and programmer, during late 2013. The main idea with ethereum was to incorporate smart contracts, which form the business logics, in to blockchain so as to improve its functionalities. Smart contracts in ethereum are executed with the help of ethereum virtual machine (EVM), that converts the business logic into byte codes. Ethereum supports computations which are turing complete and it also follows transaction-based state transitions, which means it accepts inputs in the form of transactions and these transaction inputs invoke the functions in smart contracts which can further lead to updation of state of the ledger, once after validation.This feature of smart contract enabled ethereum to address several limitations of the Bitcoin's scripts. The initial block of any blockchain network is known as the genesis block, which corresponds to the start state in the state transition stage. As initial transactions executes , the state transition happens from genesis block to any final state, which then becomes the current state of ethereum as shown in fig.1.

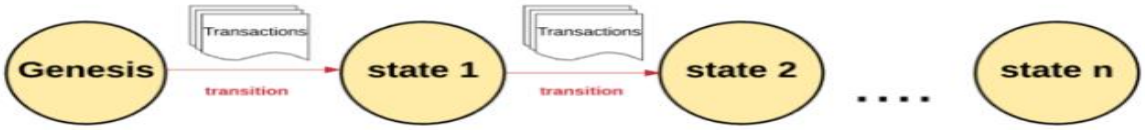

Fig.1 Transaction-based state transitions

The current size of ethereum blockchain is 280.70 GB[1]. The predecessor of ethereum was the bitcoin blockchain which mainly concentrated on keeping track of ownership of digital currency, while the Ethereum blockchain, on the other hand, focuses on execution of the programming code or smart contract for any decentralized applications, that have multi-part involvement.In the Ethereum blockchain, once the transaction or set of transactions are submitted, the miners start the process of validation inorder to form the block and include it in the blockchain network. As miners work on the validation process using PoW or PoS consensus algorithm, they can earn Ether(ETH) as incentives, which is the native currency that fuels the network. Using ether peers can pay transaction fees and for other services on the network. Before ethereum, there where specialized protocols for specific applications like Bitcoin, Namecoin, Colored Coins.While bitcoin concentrated on its cryptocurrency and served mainly as the e-wallet, ethereum on the other hand became the open source permissionless blockchain with an intend to build generalized distributed applications. The main idea behind the concept of ethereum platform model is to support a native programming language, rather than explicitly supporting specific applications.

II. ETHEREUM PLATFORM MODEL

Ethereum supports a built-in turing complete scripting or programming language, which is essentially a hybrid between standard virtual machine architecture and bitcoin scripts etc.. for executing smart contracts. Anyone can create application with any rules by defining it as a contract. In ethereum environment, users can write programs/codes in the script/contract and compile it, put these compiled scripts in transactions and send transactions to blockchain. The transactions gets confirmed and an address gets generated.

The Ethereum state consists of accounts, each account has a 20-byte address and state transitions. The world state is a mapping between addresses and account states [2]. There are two types of accounts in ethereum, user accounts (controlled by private keys) that has balance and address, and contracts(controlled by EVM code) that has code and private storage. Both of these accounts have equal privileges.

Transactions are combined to from blocks, which acts as a journal, that records a series of transactions together with the previous block and an identifier for the final state. A newly formed block is mined by the specific peer nodes inorder to add it to the blockchain. The miners gets incentives in the form of ethers. The smallest denomination of ether(ETH)is Wei. One Ether is defned as being 10^{18} Wei. The other denominations are

TABLE I
ETHEREUM CRYPTOCURRENCY DENOMINATIONS

Name	Multiplier
Wei	10^0
Szabo	10^{12}
Finney	10^{15}
Ether	10^{18}

The computational work done by miners is known as PoW.These blocks are chained together using a cryptographic hash, thus forms the blockchain. Any disagreement among nodes regarding which block to be added next may lead to fork in the network.

III. ETHEREUM STATE, TRANSACTIONS AND BLOCKS

A) World State:The world state is a mapping between addresses and account states .An account is an object in the world state.An Ethereum account state is made of four fields: nonce, ether balance, contract code hash, and storage root [2],[3].
Nonce: A scalar value equal to the number of transactions sent from this address or, in the case of accounts that have associated code, the number of contract-creations made by this account.
Balance: A scalar value that is equal to the number of Wei owned by this address.
StorageRoot: A 256-bit hash of the root node of a Merkle Patricia tree. It encodes the storage contents of the account.
CodeHash: Is the Keccak-256 hash of Ethereum Virtual Machine (EVM) code of the account, which is executed if an address receives a message call.This code is immutable and cannot be modified once it is deployed. Similar code fragments are stored in the state database in the form of hashes for later retrieval.
As mentioned earlier, there are two types of accounts in ethereum, user accounts or externally owned account that is controlled by private keys and contracts(controlled by EVM code).Both of these accounts have equal privileges and has got a 20 bytes address. Contracts also have code(static) and storage that has key/value pairs that can only be read or written by parent account.

B) Transaction
A transaction is a single cryptographically-signed instruction constructed by the user external to the scope of ethereum. There are two types of transactions:

- that result in message calls
- that result in the contract creation, which means creation of new accounts with associated code.

Both transaction type consist of a number of common fields: nonce, gasPrice, gasLimit,to,value,(v,r,s), init

nonce: A scalar value that denotes the number of transactions sent by the sender.
gasPrice: A scalar value equivalent to the number of Wei to be paid per unit of gas for all computation costs incurred as a result of the execution of this transaction
gasLimit: A scalar value equivalent to the maximum amount of gas that should be used in order to execute this transaction. This is paid up-front, before any computation is done and may not be increased later.
to: The 20 bytes address of the message call's recipient or, for a contract creation transaction.
value: A scalar value equivalent to the number of Wei to be transferred to the message call's recipient or, in the case of contract creation, as an benefaction to the newly created account.
v, r, s: ECDSA signature values corresponding to the signature of the transaction and is used to determine the sender of the transaction.
init: An unlimited size byte array that specifies the EVM-code for the account initialization process. init is the part of the EVM-code, and it returns the body, a second part of code that executes each time when the account receives a message call (either through a transaction or due to the internal execution of code). init is executed only once during account creation and gets discarded immediately thereafter.

data: An unlimited size byte array specifying the input data of the message call, in a message call transaction.

Each transaction consist of the address of the recipient of the message, a signature identifying the sender, amount of Ether to be sent, an optional data field, STARTGAS, and GASPRICE values . STARTGAS and GASPRICE fields play a vital role in dealing with attackers on the network. "Gas" is the fundamental unit that represents computation, as each transaction involve certain amount of computations. STARTGAS field denotes the maximum number of computational steps that the transaction is allowed to consume. Usual price is 1 gas per 1 computational step and also fixed additional price of 5 gas for every byte in the data area, but this value can be greater and is specified in GASPRICE field. The miners in the blockchain network will be interested in processing taransactions with higher GASPRICE, thus the sender has to choose carefully the GASPRICE value if he wants his transaction to be processed faster. In addition, miners also specifies some minimal GASPRICE under which they refuse to process a transaction .

The transactions in ethereum acts as a state transition function, that changes states of the sender as well as the recipient, once the transaction is executed. First the correctness of the transaction (the signature of sender is valid and the nonce matches the nonce in the sender's account) is verified. If credentials are correct, then the transaction fee is calculated as STARTGAS * GASPRICE, it subtracts this value from sender's account balance, and increments his nonce. Once it is done, the fee is paid per byte in the transaction and the requested amount of Ether is transferred to the recipient. The receiving account is created if it doesn't already exist, and if it is a contract, then the contract's code is executed. If the sender doesn't have enough amount of Ether for transaction or the code execution spend all the gas, the state transition function reverts all state changes except the payment fees for the miners. One contract can send a message to the other in the Ethereum network. The message resembles the transaction, but the only difference it is produced by a contract. Same like with transactions, the message induces the recipient account to run its code.

C) Blocks

The block in Ethereum is the collection of, block header that has relevant information related to the block, comprised collection of transactions, and a set of other block headers ,that are known to have a parent equal to the present block's parent's parent (ommers2). The block header stores several informations: parentHash, ommersHash beneficiary, stateRoot, transactionsRoot, receiptsRoot, logsBloom,difficulty, number, gasLimit, gasUsed, timestamp, extraData, mixHash, nonce as explained in the yellow paper by Gavin Wood[2].

IV. ETHEREUM EXECUTION ENVIRONMENT

Transaction execution is the most difficult part in ethereum protocol. Transactions that are executed must pass the preliminary tests for intrinsic validity, the details of this tests are mentioned in [2].Every transaction specifies a ***to*** address (destination address) that it sends to (unless its creating contract).The ***to*** addresse's code runs and code can send ETH to other contracts, read/write storage, call (ie start execution in) other contracts. Every (full) node on the ethereum blockchain runs under EVM and processes every transaction and stores the entire state, just like bitcoin[4][6][7]. Ethereum Virtual Machine (EVM) is a stack based storage architecture, which includes stack,memory, storage, env variables, logs, sub calling, EVM execution model is depicted in fig.2.The word size of EVM is 256-bit[5]. Maximum size of stack is 1024, though memory is volatile, storage is non-volatile and is maintained as a part of the system state. Both memory and storage locations are initialized to zero.

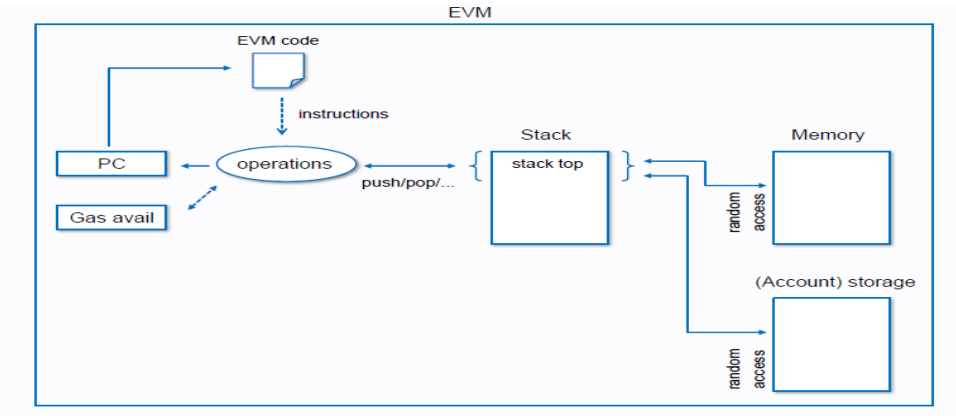

Fig.2 Ethereum Execution Model

V. CONCLUSIONS

Ethereum blockchain was the major contribution happened in blockchain domain after the initial hit of bitcoin crytocurrency. The main feature of ethereum protocol was the support for smart contracts, that can be programmed using languages similar to the general purpose languages. For example serpent similar to python, solidity having features of Javascript. These smart contract codes are finally converted into EVM code and then executed so that the platform can be easily integrated with many distributed applications in addition to the cyptocurrency dependent applications.

REFERENCES

[1] https://bitinfocharts.com/
[2] G. Wood, "Ethereum: a secure decentralised generalised transaction ledger, Byzantium version," 2018, available at: https://ethereum.github.io/yellowpaper/paper.pdf
[3] Y. Sompolinsky and A. Zohar, "Secure high-rate transaction processing in Bitcoin," Financial Cryptography, pp. 507-527, 2015
[4] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: a technical survey on decentralized digital currencies," IEEE Communications Surveys & Tutorials, vol. 18, no. 3, pp. 2084-2123, March 2016
[5] V. Buterin, "Ethereum white paper: a next generation smart contract & decentralized application platform," 2013, available at: http://www.the-blockchain.com/docs/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf
[6] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: a technical survey on decentralized digital currencies," IEEE Communications Surveys & Tutorials, vol. 18, no. 3, pp. 2084-2123, March 2016
[7] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," 2008, available at: https://bitcoin.org/bitcoin.pdf