Cuddy Network: A Peer-to-Peer WebCluster

version 1.0.0

Authors:
Jakub Wierzcholski
wierzcholski@protonmail.com
Wiktor Jezior
lakewik@lakewik.pl

Contributors:
Maciej Buczowski
maciejbuczowski@gmail.com

Abstract. Cuddy is a fully decentralized cluster for web application without a single point of failure. Based on a peer to peer network with proof of stake as a reward system. Applications based on traditional networks are threatened by the policies and control of countries and various units. Cuddy aids this problem, we believe that people should have the freedom of speech and sharing of information. Applications based on Cuddy can be accessed from standard browsers - including mobile devices , without the need for external applications. The network uses standard ports and routing to provide resources from each side from every nodes in the network.

1. Contract for container

Each user can make a contract in the network which serves as the container for the user's application. The Web Container can hold resources such as static content, but also executable code. Any Contract can be modified anytime and will be simultaneously updated on each node which is contains it. Each contract, created from one wallet, will be connected to this particular wallet. Any given contract has a few main values: value of contract which holds 3 main parameters (holding value, transfer value and execution value), all of them will be fixed to difficulty of contract by nodes. The following parameters are: the expiration date, the content (code or resources) and finally the checksum. The checksum provides proof that a contract is submitted or updated by a person with access to the right wallet. It is based on the container content and private key of the wallet. Each time a contract is processed -- network will transfer funds to the address of the appropriate miner.

2. Blockchain for tokens

The blockchain will contain a whole list of all transfers, no need to explain basic cryptography as bitcoin standard will stay the same. Also in the blockchain will be saved

informations about contracts and nodes which storing resource defined in contract - it will make the network more reliable and integral. Also to the blockchain in periodic time will be saved informations about nodes reputation.

3. The Ledger of contracts

Each node contains the whole Ledger of contract details in the decentralized network. The Ledger includes every contract that has not expired. The Ledger contains basic statistical data about each contract, such as: the number of active copies in the network, transfer used by contract, process rates and the complexity of the contract. Each change to the node is analysed by the network's proof of work algorithm and the nodes' trust Ledger.

4. The Ledger of trust

Each node contains a Ledger with ratings of all nodes. The rating is based on a system of random verification of resources of other nodes and by checking of the uptime. A node that is under control is checked by nodes nearby and they verify if the resources are the same as it is the case in other nodes. If a node doesn't respond than the rating of uptime is automatically decreased. Every node is verified and checked with other nodes for verification regarding data manipulation. If data was manipulated, the corresponding node's reputation rating is decreased. If a node significantly reduces its reputation it will be blocked by the network. For each operation consensus is needed. Nodes are checked by network in 2% of cases by each other.

5. Contract processing

Each node can have resources to share such as: space, processing time, etc. Any Node can set the parameters and it will join up with node networks of miners. Nodes calculate the complexity of a contract (based on execution and basic metrics like size). The Node can take a contract which doesn't have enough copies yet. Any Node can check if a contract needs more copies by calculating the following:

$$\sum_{i=0}^{nodes} Rate^2 = 5$$

$$Rate - uptime\ time\ percent\ (last\ day)$$

Node sorts all contracts based on metrics of profitability

$$M = \frac{StorageFee}{Size}$$

$$M - Metric\ of\ profitability,\ StorageFee - fee\ set\ in\ contract,\ Size - size\ of\ contract$$

The same metrics are applicable for processing time and transfer fee.
When a node decides to execute/store a contract it goes to the Ledger of contracts and starts seeding. Nodes get rewards every 5 minutes, based on executions and storage rates.

$$Reward = Rate^2 * (StorageFee + ExecutionRates * ExecutionFee + TransferFee)$$

6. Tokens distribution

Tokens are distributed based on proof of stake concept and basic economic consensus. Each node has the right of speek based on reputation and ownership of tokens. Tokens can be sent between each other in standard blockchain structure.

7. Communication, webcontainer

The Webcontainer can contain resources and can be updated simultaneously, each node has 80 port and 9336 port open. In 80 port the users can list all contracts.
The standard type of getting resources from webcontainer will be as follows:
$255.255.255/tx/contractid$
$contractid - sh2\ checksum\ eg.\ e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855$

Connecting with each node from the network will enable getting resources from the contract. Resources will be dynamically loaded over the nearest node. Future updates predict usage of a connection based on DNS settings in the network. This kind of CNAME connection is allowed by a lot of domain distributors. Port 9336 will allow for node communication regarding consensus and other metrics.

8. Proof of stake

Consensus in network in case of transfer is based on “Proof of stake” algorithm, each online node which have funds can vote with power of vote correspond balance of tokens on node addresses. Each consensus is accepted with 50% approval. This kind of consensus has an economic basis, tokens in network which try to manipulate data will lose value, each user with higher balance are determined to have online nodes and work in good faith.

Each node with tokens can earn by holding them online in full-node, this reward is sent for all nodes which has been signing consensus proportional for token amount which node is holding.

9. Anti - fraud mechanisms

Each node possesses a predetermined positive reputation. Every action within the network is registered and a randomized system of node verification detects anomalies in the network such as forged requests. Users have several metrics that determine the correctness of the requests. The first one is a correct and verified request (this kind of request was checked by a checksum by other nodes). The second one is the so called ‘no request’. It means that there was a lack of resources or the time dedicated for fulfilling a request was exceeded. The third one is an incorrect execution or manipulation of the request. After two random

confirmations of the resources related to the incorrect or manipulated request in question, there node is banned from the network. The nodes are unable to forge a request in order to pass the verification. Every request is the same so a node can’t decide if a request was made by a user or a tunnel by node to user or was made by a different node from the network in order to check for correctness.

Reputation based on random checkings:

$$R = 0,\ Uincorrect > 2$$

$$R = \frac{Uup}{Uup + Udown}$$

R − *Reputation*

Uup − *Request that has been checked with other nodes and has the same checksum*

$Udown$ − *Lack of resources or time out*

$Uincorrect$ − *Manipulated / Incorrect data*

10. Summary

Our project has huge potential, even static resources can be used as static content delivery. Basic usage can be "Content Delivered Network". Storing data on private nodes is cheaper than that kept in data centers -- even those of the largest private companies. (The StorJ project has reached better rates than dropbox, etc. in just a few weeks). A P2P based CDN maintains a 100% uptime by concept. Additionally, it is independent from any policies implemented by governments. Content delivery by P2P networks is about 20-40% cheaper than that provided by huge hosting companies, moreover, it can't be taken down.