CYB: IMMORTAL ROBOT THAT USES CYBER PROTOCOL WHICH ANSWERS QUESTIONS.

BUILDING INNOVATIVE SYSTEMS (UCS757)

Submitted By:

PARAG NASSA

101983021

4 COE-22

Live Link: https://cyb.ai/

GitHub Link: https://github.com/paragnassa/cyb

Novelty Of Work As Well As Methodology

1. As it is Web based Project, it requires mainly the original protocols such as: TCP/IP, DNS, URL etc.

2. Protocols brought the web to a stale point, where it is located as of now. Considering all the benefits that these protocols have produced for the initial development of the web, along with them, they have brought significant obstacles to the table. Globality, being a vital property of the queb is under a real threat since its inception. The speed of the connection keeps degrading while the network itself keeps growing due to ubiquitrus government interventions. The latter causes privacy concerns as an existential threat to human rights.

3. Data Set: The dataset in this project is a lots and lots of information connected through various website links.

4. Cyber protocol:

    - In its core the protocol is very minimalistic and can be expressed with the following steps:

    - Compute the genesis of cyber protocol based on the defined distribution

    - Define the state of the knowledge graph 3 Gather transactions using a consensus computer

    - Check the validity of the signatures

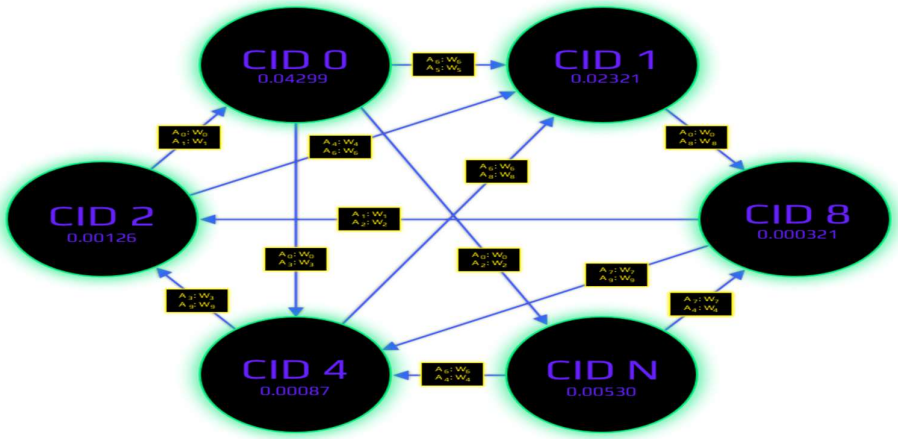
    - Check the bandwidth limit

    - Check the validity of CIDS

5. Methodology

Knowledge Path:

- We represent a knowledge graph as a weighted graph of directed links between content addresses. Aks, content identificators, CIDs, PFS hashes c simply-IPFS links.



By using content addresses to build the knowledge graph we gain the so much needed IPFS-like superpowers of pop protocols trees for

Search engine:

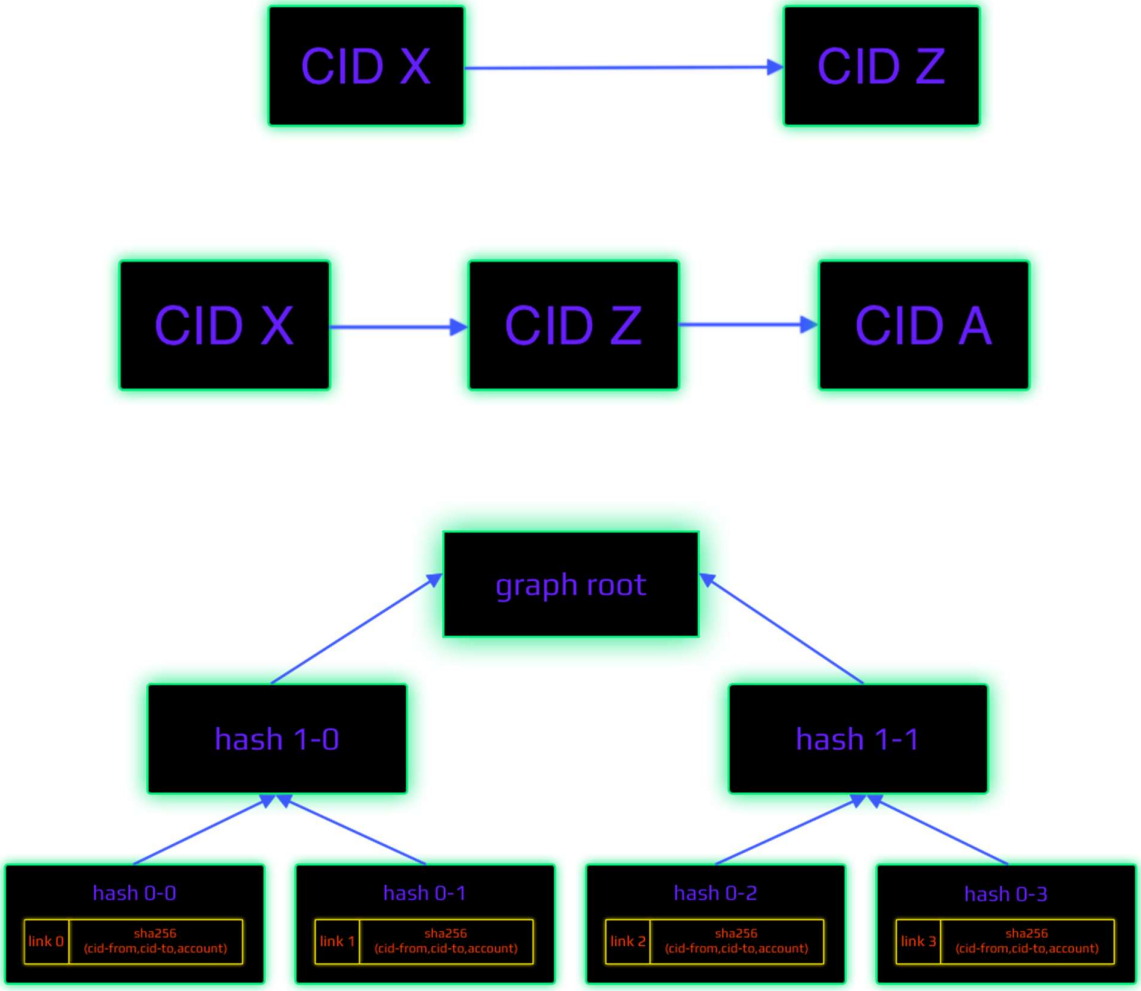
- mesh-network future-proof
- interplanetary accessibility
- censorship resistance
- technological independence

Our knowledge graph is generated by the awesome masters. Masters add themselves to the knowledge graph with the help of a single transaction, **Cyber Link**. Thereby, they prove the existence of their private keys for content addresses of their revealed public keys. By using there mechanics, a consensus computer could achieve provable differentiation between subjects and objects on a knowledge graph.

6. Cyberlinks:

To understand how cyberlinks function we need to understand the difference between a URL link (aka, a hyperlink) and between an IPPS link. A URL link points to the location of the content, whether an IPFS link points to the content Itself. The difference between web architectures based on location links and content links is radical and requires a unique approach.

Cyberlink is an approach to link two content addresses, or IPFS links, semantically:



Using cyberlinks, we can compute the relevance of subjects and objects on the knowledge graph. But we need a consensus computer,

Transformation Problem into supervised and unsupervised learning: Labeled data

1. The main distinction between the two approaches is the use of labeled datasets. To put it simply, supervised learning uses labeled input and output data, while an unsupervised learning algorithm does not.

2. In supervised learning, the algorithm “learns” from the training dataset by iteratively making predictions on the data and adjusting for the correct answer. While supervised learning models tend to be more accurate than unsupervised learning models, they require upfront human intervention to label the data appropriately. For example, a supervised learning model can predict how long your commute will be based on the time of day, weather conditions and so on. But first, you’ll have to train it to know that rainy weather extends the driving time.

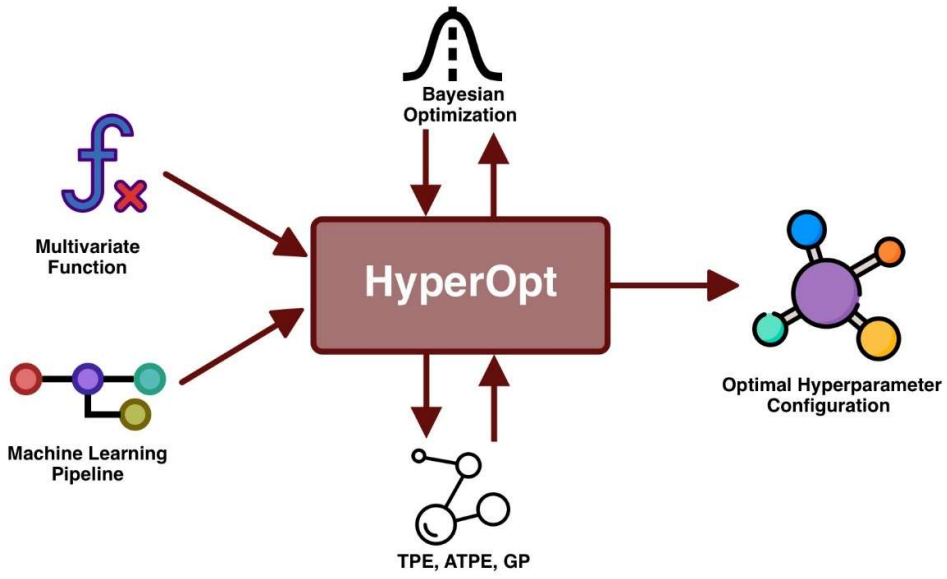
3. Unsupervised learning models, in contrast, work on their own to discover the inherent structure of unlabeled data. Note that they still require some human intervention for validating output variables. For example, an unsupervised learning model can identify that online shoppers often purchase groups of products at the same time. However, a data analyst would need to validate that it makes sense for a recommendation engine to group baby clothes with an order of diapers, applesauce and sippy cups.

4. We assume that the proposed algorithm does not guarantee high-quality knowledge by default. Just like a new born, it needs to acquire knowledge to develop further. The protocol itself provides just one simple tool: the ability to create a sxkedink with an agents stake between two content addresses.

5. Analysis of the semantic core, behavioural factors, anonymous data about the interests of agents and other tools that determine the quality of search, can be achieved via smart contracts and off-chain applications, such as: web3 browsers, decentralized social networks and content platforms. We believe, that it is in the interest of the community and the masters to build the initial knowledge graph and to maintain it. Hence, for the graph, to provide the most relevant search results.

Optimizaion of Hyperparameters:

1. Hyperparameter optimization or tuning is **the problem of choosing a set of optimal hyperparameters for a learning algorithm**. A hyperparameter is a parameter whose value is used to control the learning process. By contrast, the values of other parameters (typically node weights) are learned.

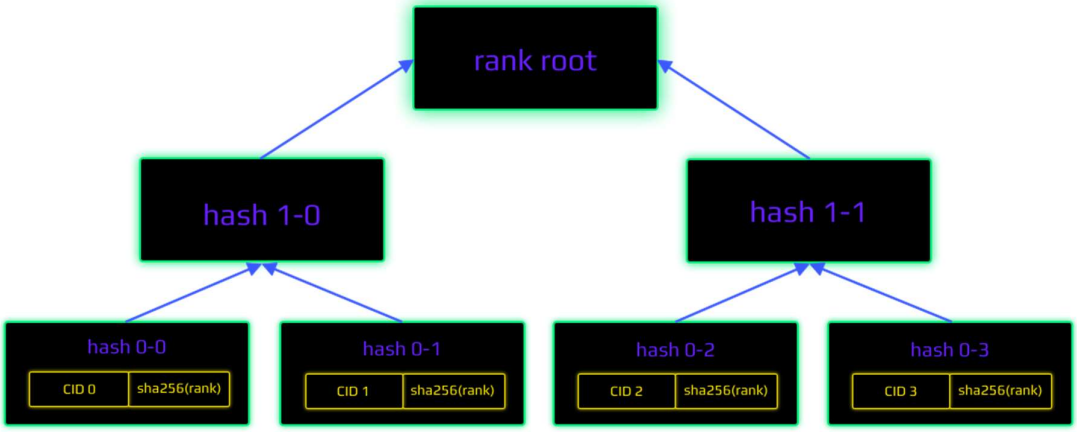


- Cyber^Rank: Ranking using a consensus computer can be challenging, as consensus computers have serious resource constraints.

$$\begin{gathered} \text{CIDs } V, \text{cyberlinks } E, \text{Agents } A \\ agents(e) : E \to 2^A \\ stake(a) : A \to \mathbb{R}^+ \\ rank(v, t) : V \times \mathbb{N} \to \mathbb{R} \\ weight(e) = \sum_{a \in agents(e)} stake(a) \\ rank(v, t+1) = \frac{1-d}{N} + d \sum_{u : (u,v) \in E} \frac{weight(u, v)}{\sum_{w : (u,w) \in E} weight(u, w)} rank(v, t) \\ rank(v) = \lim_{t \to \infty} rank(v, t) \end{gathered}$$

2. Proof of Relevence:

- We have designed the network under the assumption that with regards to search, such a thing as malicious behaviour, does not exist. This can be assumed as no malicious behaviour can be found in the intention of finding the answers. This approach significantly reduces any surface attacks.



3. Speed

We require instant confirmation time to provide users with the feeling of a conventional web-application. This is a powerful architectural requirement that shapes the economical topology and the scalability of the cyber protocol. The proposed blockchain design is based on the Tecdesmit consensus algorithm with 146 validators and has a quick, 5 second ty finality time. The average confirmation time is closer to 1 second and could make complex blockchain interactions almost invisible to agents
We are aware of certain mechanisms to make this function order of magnitudes faster:

- optimization of the consensus parameters
- better parallelization of rank computation
- better clock ahead of consensus

.

4. Scalability:

We require an architecture which will allow us to scale the idea to the significance of the likes of Google. Let us assume, that node implementation, which is based on Cosmos-SDK can process 10k transactions per second. This would mean, that every day, at least 8.64 million masters will be able to submit 100 syberlinks each, and impact the search results simultaneously. This is enough to verify all the assumptions out in the wild, but not enough to say that it will work at the current scale of the Internet. Given the current state of the art research done by our team, we can safely state that there is no consensus technology in existence, that will allow scaling a particular blockchain to the size that we require. Hence, we introduce the concept of domain-specific knowledge graphs.