Kudos: A Peer-to-Peer Discussion System Based on Social Voting

Luca Matteis

Sapienza University of Rome
matteis@di.uniroma1.it

Abstract. We describe a peer-to-peer discussion system that uses a blockchain for tracking users' points which are mapped to specific content, and a distributed hash table for storing/serving content. Points are minted through the mining process which achieves consensus on the chain to accept. Peers only accept transactions (aka votes) of a *universal constant amount*. Fees for the miners are also constant to prevent self-promotion. Users earn points either by mining or by posting relevant content that others are willing to upvote. Highest blocks in the chain contain references to the latest most relevant content, which is republished more frequently by peers to enable quicker response times. The system has the property of allowing peers to discuss openly and freely, without the fear of being blocked or banned. It also allows relevant content to stand out without the need of a central tracker or index. Peers gain reputation through the points they earn but also through the irrevocable proof of the things they said available in the blockchain.

1 Introduction

Discussion on the internet happens through a variety of mediums ranging from client-server systems to trustless peer-to-peer networks. Recently a number of services have begun using the concept of *social voting* or rating to allow relevant content to rank higher in the viewing order. This allows content of higher quality to stand out and be viewed by more people, and results in a pleasing experience for users which can find themselves consuming relevant content, rather than having to browse and pick content from a sea of information. One service that makes this their main functionality is the famous discussion website REDDIT. Their front page is full of remarkably high-quality content ranging a large variety of topics.

A problem with these social voting-based services is that they rely on a central entity or server for them to function. This has the effect that the central entity has control over what can be posted or discussed on the site, and can remove content as it sees fit. Another problem is that the reputation that a user has acquired could be manipulated by the controlling entity to favor specific types of users. Furthermore, the service could go offline, leaving no facility for others to verify someone's reputation. What is needed is a decentralized discussion system that works without a single-point-of-failure and where anyone can prove someone's reputation without having to trust a central service.

KUDOS, the system we designed, tries to achieve these things using a combination of two overlay networks: (i) a blockchain for tracking users' points which are mapped to specific content, and (ii) a distributed hash table (DHT) for storing and serving content.

In this paper we describe such system and evaluate certain novel aspects of it to show its feasibility.

1.1 System description

The overall idea is that users submit transactions to the blockchain to showcase their vote towards specific content. Transactions contain a reference to content stored in the DHT network[1], and the recipient of the transaction is the publisher of such content. To calculate the relevance of specific content, we add up all the transactions in the blockchain which contain its reference. Similarly, to calculate the points a user has earned, we add up all the transactions where the user appears as the recipient.

Our system takes the same general approach as cryptocurrencies such as Bitcoin for generating points. Points are hard to obtain and therefore have the valuable property of being sybil-resistant: one cannot create several fake personas to send many points to a user without either mining or earning them. Newly mined blocks contain a *pointbase* transaction to create points out of thin air and therefore allow a fair distribution of points.

A strong departure point of Kudos compared to cryptocurrency systems is that both the amount and fee of transactions are of a *universal constant amount* which is the same for all transactions in the network. This ensures that a user with a certain amount of points cannot upvote content posted with another of their accounts indefinitely; because of a constant fee they will eventually run out of points. This also makes it expensive for peers to upvote the same content multiple times.

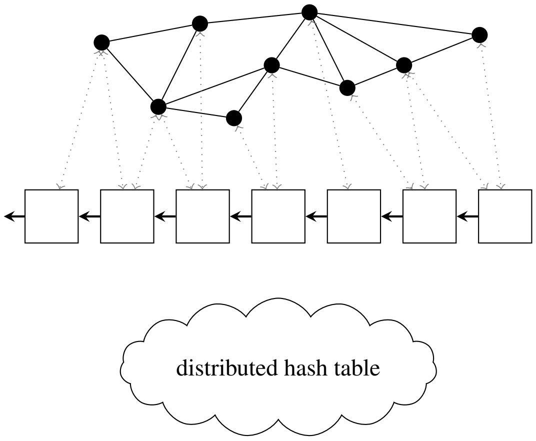


Fig. 1. Peers learn about relevant DHT keys through the blockchain

In Figure 1 we see the general functioning of the system where the blockchain is used as a reference point for content contained in the DHT. Once the reference is found

[1] One can imagine storing content using other systems such as BitTorrent or HTTP.

in the blockchain, peers request it directly from the DHT network. If peers like specific content, they can announce their interest by submitting a transaction to the blockchain, effectively giving a point to the publisher of the content.

In the following sections we describe and evaluate the main components of our system. In Section 2, we outline the core concepts and relate them to existing work. In Section 3 we explain the structure of voting transactions and how points are calculated with minimal network and storage overhead. In Section 4 we show how constant amounts make self-promoting and collusion expensive. Section 5 describes how to use the height of the blockchain as a measure to keep content alive in the DHT network. Finally we end with discussion and future work in Section 6.

2 Core concepts & related work

Kudos is based on the same concepts as Bitcoin for generating and distributing tokens fairly to the network. One strong departure point is the use of *universal constant amounts*. As discussed in Section 1, several important differences, that we outline below, make Kudos inefficient to use as a currency, but effective for social voting.

2.1 Trustless reputation

One core concept of our decentralized discussion system Kudos, is the use of scarce tokens (such as bitcoins), not for economical exchange, but for upvoting and giving reputation to other users. This new field of *trustless reputation* has been studied in academia and implemented in many industry and open-source projects. One prominent project is Ripple [] which uses a consensus algorithm based on trusting peers, and allows peers to build reputation models around the behavior of other peers. The main difference between Ripple and our Kudos system is that we do not rely on past behavior of peers, rather a peer's reputation is based off how many points they have and the things they said. Also our system's goals are inherently different because we use user's reputation to guide discussions and not to achieve consensus on the state of transactions.

Another project that deals with trustless reputation is CoinAwesome []. Their sense of reputation is more connected to concepts such as tipping and is related to our work: peers show their appreciation of someone's content by tipping them with a certain amount of coins. However, coins in CoinAwesome have an economical value and can be exchanged with other currencies. Kudos on the other hand builds strong measures to avoid using points for economical trading through universal constant amounts.

2.2 Decentralized discussion

Decentralized discussion systems are common in P2P networks. Usenet [], probably the oldest of this kind, allows peers to discuss and comment using message boards which are not controlled by central servers, rather, each peer in the network has a copy of all the boards. Another recent decentralized discussion system is Bitmessage [] which main feature revolves around anonymity. Twister [], a decentralized Twitter clone, also provides discussion capabilities through the use of blockchain and DHT technologies.

Kudos, compared to these systems, adds the novel feature of allowing discussion posts and comments to be upvoted and ranked in a fully decentralized manner.

2.3 Decentralized ranking

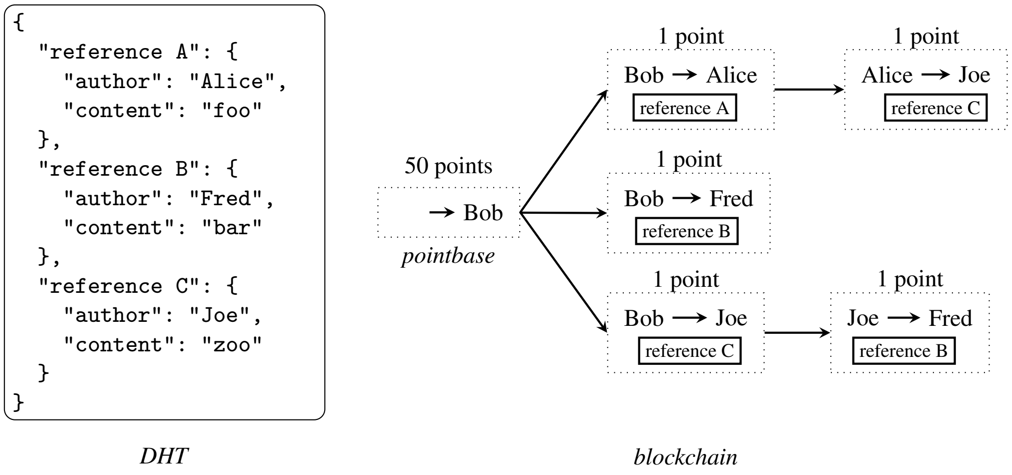
Ranking things that exist in the digital world has always been something that requires a central index. Whether it is a search-engine such as Google, an e-commerce website such as Amazon, or a social voting system such as Reddit, they still require a centrally controlled server (or servers) for them to function. Being able to rank digital objects in a way that doesn't require a single-point-of-control could unleash new ways of thinking about trust and reputation on the internet. For instance, being able to rank torrents so that only top-rated torrents appear as browsable list to users, is only achievable via a torrent search-engine which can be easily shut down by authorities. Kudos provides this ranking functionality via a blockchain that works differently than what is used in Bitcoin. Users can upvote digital objects, which in our system are objects contained in the DHT network, based on how many tokens they have in the system. People can only earn tokens via proof-of-work mining (which requires expenditure of energy) or by posting relevant digital objects that others are willing to upvote.

2.4 Blockchain & DHT

Another core concept of Kudos is the combination of a blockchain with a distributed hash table, to enable relevant content to stand out from the DHT network. Blockstore [] is a project that uses this combination of blockchain and DHT in a way that is similar to Kudos: actual data is stored in the DHT, while the blockchain only keeps track of the references. However, Blockstore is used to store key-value information (such as DNS information) and therefore uses the blockchain only as a way to keep track of the owners of such key-value pairs. On the other hand, Kudos uses the blockchain to track users points and not for transferring ownership of each content in the DHT network. Although both systems use the combination of blockchain and DHT, the goals of each system are different: one is a decentralized DNS-like system, the other is a decentralized discussion system.

3 Voting transactions

As discussed in Section 1, transactions made within the system are used to showcase an upvote towards specific content. The structure of transactions is very similar to Bitcoin: each transaction has an input, referencing other transactions the user can spend, and an output, to assign who can claim those points. Because our system cares mainly of transferring single atomic values (which are constant), we do not care about splitting and combining values as Bitcoin does. For that reason transactions are much simpler and only need to specify a single input and output. An important addition to transactions is what we call a *transaction reference*, which contains a reference for content contained in the DHT network.



```
{
  "reference A": {
    "author": "Alice",
    "content": "foo"
  },
  "reference B": {
    "author": "Fred",
    "content": "bar"
  },
  "reference C": {
    "author": "Joe",
    "content": "zoo"
  }
}
```

reference	*points*
reference A	1
reference B	2
reference C	2

content ranking

user	*points*
Bob	47
Alice	0
Fred	2
Joe	0

user points

user	*points*
Bob	0
Alice	1
Fred	2
Joe	2

user reputation

Fig. 2. The blockchain acts as a ranking system for content in the DHT.

Figure 2 shows an example of the contents of the DHT and the blockchain. We can see that although the DHT can store information in a decentralized fashion, ranking such information would require a trusted centralized index that can be easily manipulated and shut down. The blockchain acts as way to rank information in the DHT through the scarce tokens that are generated via mining.

The various tables in Figure 2 explain how things can be ranked, and how reputation could be calculated. For instance, Bob finds himself with many more points than others because he mined the current block. However, in terms of reputation he is the only one to have 0, because nobody has yet upvoted any of his content.

3.1 Calculating content & user points

To calculate how many upvotes a post has received we add up all the transactions which have a reference equal to the hash of such post. One does not need to lookup all the blocks in the chain; inspecting the latest *n* blocks should provide enough assurance regarding the popularity of posts.

Calculating user's points requires one to keep track of the unclaimed outputs for that specific user. This is a more cumbersome process that requires to store more information. However, using methods such as bloom filters already implemented in Bitcoin will provide capabilities for lightweight clients to calculate someone's reputation.

Hereunder we evaluate the size of a simulated Kudos blockchain, comparing it to that of Bitcoin, to understand how much data one would require to download to take part in the network. An important difference is that in Kudos, users are meant to have a single address. This is the address that is tied to people's identity in the system, and there is no reason one would want to create several addresses unless they want several identities in the system. On the other hand, Bitcoin wants users to have as many different addresses as possible, through HD wallets, because it preserves people's privacy. This simple but important difference makes the amount of data needed to be stored in a Kudos node, drastically smaller than Bitcoin.

Figure 1 shows a simulated comparison between Kudos and Bitcoin with 100,000 physical users. In Kudos every user has 2 addresses (still a large amount for a realistic scenario). In Bitcoin every user has 100 addresses. As we can see, the number of unclaimed outputs is drastically smaller in Kudos. In fact, by assuming the transactions trends are the same in both networks, a Bitcoin node needs to store 100mb of data over a 6 year timestamp, while a Kudos node only 5mb.

4 Universal constant amount

Social voting is different from transacting currency. In currency transactions users should be able to send any amount to any amount of addresses with as little cost as possible. In social voting on the other hand, one should not be able to upvote specific content multiple times and therefore should not be able to send any amount they want to users.

Universal constant amounts is our proposal for making multiple transactions expensive to make. Instead of being of arbitrary value, the network allows only transactions of a specific amount, that is agreed by the entire network. Furthermore, fees are also constant. Our intuition is that not only does this make points uninteresting to use for economical transactions, but also provides benefits for dealing with self-promotion and collusion.

4.1 Self promotion

We outline a scenario where a user with multiple addresses decides to send transactions between each of their addresses to promote their own content. As long as each address contains some points, the user could do this indefinitely and have their content stand out multiple times in a row. To combat this kind of behavior each transaction must contain a fee of constant amount which is claimed by the miners.

Figure 2 draws a comparison of how this type of self-promotion would be accomplished without the use of constant fees. On the right we see that each transaction costs users to lose a certain amount of points and, even though a malicious user could do this a certain number of times, they will eventually run out of points.

4.2 Collusion

Upvoting the same content multiple times is also an unwanted behavior for a social voting system. One way to combat this is to simply not allow duplicate transactions. However, dishonest nodes could still send a portion of their points to other addresses they own, and use these secondary addresses to upvote their target content. Universal constant amounts make this behavior expensive because it would require malicious users to break the amount of points they want to send into equal portions, each containing a fee. Figure 3 shows how quickly malicious users could lose points if they want to upvote content multiple times.

5 Keep alive

As the blockchain grows in size one cannot expect the DHT to hold content from old blocks as it would require too much effort. Instead we rely on the idea that only relatively recent content is retrievable from the DHT. Peers should therefore republish to the DHT, the references of transactions contained in the latest blocks. This would allow recent content to be fetched with greater speed. The reference of old content is still available in the blokchain permanently and there can be historical nodes that can make available this content. Peers can prove for themselves, with only the block headers, that the content retrieved from an historical node is in fact part of the chain.

To keep content alive in the DHT, peers retrieve the latest n blocks, download all the content that is being referenced from the DHT, and at specific intervals perform a DHT push of such content. Our intuition is that latest content is fetched more frequently and therefore needs a higher republish rate by the peers in the network. The republish rate can therefore be taking into account the size of the blockchain s, the amount of unique content being referenced in the chain c, and a constant l representing the latest number of blocks one is interested in.

To test whether our keep alive method allows relevant content to be retrieved in a relatively quick manner, we simulated a chain of 100,000 blocks, 100 addresses and incremental transaction growth. Results that changing l dynamically based on the the amount of transactions in a block, allows for best performance-time trade off.

5.1 Groups subscriptions

6 Discussion and future work

One main issue with our system is that in order to vote, users must possess points. This might sponsor the idea of hoarding points so one can promote their own content. Although, as we explained in this paper, it would be expensive for users to promote their own content multiple times, for a greedy user it would still be better to promote their own content rather than others. A solution to this issue might be in how reputation is calculated. The variety of addresses could be a variable in calculating the reputation points for users: the more users upvote different content, the higher their reputation.

Another problem comes with the transparency of transactions. This allows virtually anybody to see which content someone upvoted. To combat this issue one could employ the same measures that have been discussed for cryptocurrencies. Specifically CoinJoin and Zerocash provide ways to hide the way transactions are linked.

Also we have only described the concept of upvoting, while many social voting systems allow downvoting as well. This should be trivial to implement through simple rules in the protocol to be able to deduct amounts from people's accounts, rather than only add up amounts.