Web 3.0: Developments and Directions of the Future Internet Architecture?*

Yuchao Zhang**, Pengmiao Li, Peizhuang Cong, Huan Zou, Xiaotian Wang, and Xiaofeng He

Beijing University of Posts and Telecommunications

Abstract. As a promising emerging technology, Web3.0 has become the focus of more and more manufacturers and researchers. Web3.0 is an integration of network readability, writability, and authenticity. It is not only a new Internet architecture that integrates multiple rising technologies based on decentralization, but also an Internet infrastructure owned and trusted by each individual users. It reshapes the relationship between users and applications, by storing data on the network, rather than on specific servers owned by large service providers, which means that anyone can use this data without creating access credentials or obtaining permission from those monopolistic providers. This vision paper will first review the way the current network services work, then introduce some key technologies closely related to Web3.0, and finally point out the future research directions and potential opportunities, which are expected to give researchers a better understanding of Web3.0.

Keywords: Web3.0, Storage, Transmission, BlockChain, CrossChain, Decentralized Identity, Federated Learning, Security.

1 Introduction

In the past several decades, the Internet has been developing along with the development of storage, computing and transmission. Our current network built based on TCP/IP provides web services via http(s). Service or application providers usually first collect user registration information, and then store all the data (generated from users or required by users) in cloud data centers, they design data transmission algorithms to realize data synchronization across their multiple datacenters, and provide low latency services by using edge caching.

Compared with Web1.0, where users can only passively receive information, Web2.0 service mode allows users to interact with network, but the core of it is

* The work was supported in part by the Key Project of Beijing Natural Science Foundation under M21030, the National Natural Science Foundation of China(NSFC) under Grant 62172054, and the National Key R&D Program of China under Grant 2019YFB1802603.

** Corresponding author: Yuchao Zhang (yczhang@bupt.edu.cn). All the other authors contribute equally to this paper.

strong centralization, including centralized identity authentication and centralized data storage. This leads to efficient network management, but also brings the following drawbacks to the current network: 1) fragile authentication: simple (or encrypted) password checking to confirm user identity and access rights, 2) un-guaranteed user privacy: all personal information and the generated data from users are stored on the central servers of service providers, which is vulnerable to attacks and lead to data leakage, 3) un-guaranteed security: once any of those servers fails, user data stored in that server will be lost and cannot be recovered. 4) strong data isolation: data among multiple applications is difficult (if not impossible) to interact with each other, leading to extremely weak cross application interoperability.

To address the above problems, some emerging technologies are gradually attracting the attention of more and more researchers. Just to name a few, 1) Blockchain, which is a distributed ledger that combines data blocks in a sequential order, and is guaranteed to be non-tamperable and unforgeable equipped with cryptography. 2) Decentralized Identity, which is a globally unique, persistent and tamper proof personal identity, and it can be completely controlled by the owner and does not depend on the centralized platform and identity provider. 3) Distributed Storage, which is a scalable structure that uses distributed servers to share the storage and uses location servers to locate storage information, so that it can improve the reliability, availability, scalability of data systems.

Base on these rising technologies, we are trying to introduce some promising future research directions and some potential solutions, including but not limited to: cross-chain interaction, decentralized storage, multi-client multi-server transmission and security.

Along this line, the remainder of this paper is to first review the state-of-the-art network ecology, and then considers some of the rising technologies of Web3.0, at the developing stage of its conception, followed by some future directions.

2 The SOTA Network Ecology

2.1 Web Service

As the web has evolved, the nature of web services has become more apparent: it's about providing users with higher quality of experience through the use of new technologies. The existing Internet has motivated the rapid development of different areas in order to give users a higher quality of network services, especially in the area of storage and transmission. The essence of web services is to provide users with requested data efficiently. The quality of the user's network service is often reflected in the response time.

Web2.0 services store all content in centralized datacenters far away from users. Still, as the network grew, the number of users and data grew explosively, which inevitably burdened the data center and the network. Service or application providers often build or rent some suitable edge servers or edge nodes close to users to store a portion of frequently accessed contents in order to ensure faster delivery and reduce the redundant transmission of such contents, thus reducing the burden on data centers and networks. Latency can be reduced through proximity storage edge server, but with limited bandwidth resources and multiple transmission paths, there is a need to choose the right transmission path, whether it is at the edge server or edge node or to the data center, to get the data. This enables faster delivery of user-requested content to users and reduces possible network congestion, data loss, and other occurrences. The goal in the storage area is to reduce the back-to-source time by storing the content that users may access in the future through servers closer to the user's edge. In summary, the combination of storage and transport enables a higher quality of service for users in the current network.

2.2 Storage Architecture

Based on the development of edge computing, more edge servers can be fed to service or application providers, moving away from a single data center Web1.0. However, edge servers have limited storage resources and are widely distributed geographically. How to improve cache hit rate, transmission latency, etc., through cache replacement techniques, and deploy these edge servers for each application to reduce transmission latency has been a key research problem in industry and academia. Next, some representative cache replacement strategies and node deployment studies are briefly described.

The cache replacement strategy aims to ensure that the user's future requested content exists in the edge server as much as possible by real-time cache replacement within the limited storage space. To ensure that such content can be transferred to the user via edge servers close to the user instead of the more distant cloud in order to reduce the network transmission time. Traditional cache replacement strategies include LRU, LFU, and their variants [5], which are widely used in the industry due to their simplicity and ease of deployment. However, with the development of applications, these strategies lack performance in some specific scenarios, so intelligent cache replacement strategies [31, 37, 51, 54] are proposed with popular content prediction based on artificial intelligence techniques. For example, Zhang et al. proposed GraphINF [54] as a popular content prediction strategy to obtain highly accurate hot content prediction results by exploiting the attractive geographical propagation characteristics of short videos, which supported the cache replacement strategy. Li et al. proposed the cache replacement strategy, CRATES [31], to solve the problem of low hit rate due to low-access frequency periods in short-video networks by predicting the possible future accessed popular content by exploiting the relationship between popular content and core users. Zhang et al. proposed distributed cache replacement strategy, AutoSight [53], to improve the caching hit rate by analyzing the popular periodicity and the unstable access characteristics in the short video network. And they designed an observation horizon for automatically acquiring popular content to prevent the ebb and flow of popular content from being unknown, thus reducing the obsolete content cached in the server.

The large number and wide distribution of edge servers are deployed as storage servers for service or application providers to store content. However, choosing which edge servers to use as storage servers is crucial because it directly affects the response time to user content requests and the cost to the provider. Data analysis revealed that some providers deploy many storage servers for services or applications to ensure lower response times. However, the frequency of requests from these servers is unbalanced, resulting in the wasting of storage resources and raising of the total cost. To avoid the unwelcome situation, a number of researchers have focused on the problem. For example, li et al. proposed edge storage nodes deployment strategy Frend [30], which presented a frequency-based transmission latency criterion by analyzing data and, using this criterion presented a deployment strategy that ensures both qualities of service and reduces the number of nodes.

Both the cache replacement strategy and the edge storage node deployment strategy have a role in improving the quality of user experience for Web2.0.

2.3 Transmission Mechanism

Based on decades-old network technologies and service architectures, Web1.0 placed limited requirements on network transmission. However, along with the developments of the network and the flourishing of services, Web2.0 put forward requirements on network transmission in terms of latency, bandwidth, packet loss rate, jitter, and more. To this end, some specialized technologies have been studied to optimize diversified services. Here, in this section, we introduce some classic works in network transmission optimization.

Multipath Transmission Control Protocol (MPTCP): Multi-home hosts are widespread, such as servers under Fat-tree network topology in the data center, or some smart devices with 5G/WIFI/Bluetooth multi-connectivity [26]. And such multi-home hosts will become more common with the deployment of IPv6. Traditional TCP can only exploit multiple connections by establishing multiple TCP connections since it only supports a single channel for an individual connection. To obtain the benefits of multi-connected network resources of the multi-home hosts, the proposed MPTCP supports the reverse multiplexing of redundant channels, which can increase the overall data transmission rate to the sum of all available channels [38]. Furthermore, in wireless network environments, MPTCP enables links to be added or dropped when clients enter or exit the network coverage, without breaking the end-to-end TCP connection. Thus, the problem of link switching can be solved at the endpoint instead of using any special handling mechanisms at the network or link level. Quick UDP Internet Connection (QUIC): The protocols of the transport layer mainly include TCP and User Datagram Protocol (UDP). The lightweight UDP is more efficient than TCP, which has been widely employed in many services, such as online games, streaming media, etc. But it is unable to provide reliable connections as TCP. To address the requirements of low connection latency and high reliability at the transport layer and application layer, Google proposed QUIC, a UDP-based protocol that incorporates the features of TCP, TLS, and HTTP/2 [25, 29]. When the client connects to the server for the first time, QUIC only needs a delay of 1 Round Trip Time (RTT) to establish a reliable and secure connection, which is faster than 1-3 RTTs of TCP+TLS. After this connection, the client can cache the encrypted authentication information locally and establish a connection with the server again with 0 RTT connection establishment latency. As QUIC is based on UDP, it can reuse the multiplexing feature of the HTTP/2 protocol while avoiding the HTTP/2 Head-of-Line Blocking problem. Additionally, QUIC runs in the user space instead of the kernel space, which enables a fast update and deployment.

To meet the needs of Web2.0 where different services tend to prefer different performance paths, such as low latency, high bandwidth, low packet loss, etc., network layer protocols are not only limited to meet reachability but also ought to customize the routing paths for different types of flows. In this context, some learning-based routing algorithms have been proposed in recent years. For example, Cong et al. [13,15] proposed a multi-constraint reinforcement learning-based routing strategy by model fusion to provide different routing paths for different types of flows to fully utilize the available network resources; Zhang et al. [14] proposed a cross-domain routing decision mechanism assisted by intra-domain information based on homomorphic encryption technology, which can provide a good performance cross-domain routing path by leveraging intra-domain information.

3 Web3.0: Rising Technology

3.1 BlockChain

Blockchain technology emerged as the basis for crypto-currencies Bitcoin [36], has been widely applied to many frontiers with its characteristics of decentralization, tamper-resistant, traceability and anonymity. Blockchain is a chain of blocks that can be described as an immutable distributed database which records traceable transactions through cryptographic algorithms. It holds a shared distributed ledger without relying on a common trusted third party and is maintained by a group of nodes.

Depending on the degree of decentralization and openness, there are three types of blockchain: public blockchain, consortium blockchain and private blockchain. Public blockchains(also called permissionless blockchains) allow participants to access the network without any authentication. Two prominent examples are Bitcoin and Ethereum [9]. Consortium blockchains and private blockchains can be deemed permissioned blockchains where identity authentications are required when enrolling in the network. Permissioned blockchains are more common for organizations and enterprise demands and examples include Quorum [11], Corda [7], Hyperledger Fabric [4] and Tendermint [8] .

Blockchains deploy smart contract to ensure ledger updating immutable and irreversible. Smart contract is a set of automated programms which are executed in virtual machines. It makes blockchains programmable and can be used to extend the state machines. For example, the Ethereum Virtual Machine (EVM) is used to store and execute smart contracts in Ethereum and for decentralized applications (DApps). Within smart contracts, blockchains can execute transactional workloads which have so far been handled almost exclusively by databases. Compared with traditional contract, smart contract executes itself without the involvement of third parties and uses cryptography to prevent random modifications of the ledger [6].

Traditional fault-tolerant consensus protocols have been adpoted to blockchains for reaching a unified agreement on the state of the network in a decentralized way [10]. Normally the consensus needs a balance between resource consumption and security, since high degree of trust means high energy-intensive consensus. Bitcoin uses Proof-of-Work(PoW) as the consensus protocol. Mining nodes in Bitcoin compete on sloving a cryptographic puzzle that is easy to verify to get the ledger writing right. Once a node finds the solution to the puzzle, it can propose a valid block and append it to the ledger. However, PoW consensus mechanism has some issues, for example, the 51% attack risk and large resource consumption. Ethereum2.0 changes from Proof-of-Work to Proof-of-Stake(PoS) and effectively improves the throughput. The new block is yieled by validators who are elected according to the stake size or coin age, instead of miners. Compared with PoW, PoS consumes lower resource and is more robust to 51% attack. Recent blockchians like Tendermint and Hyperledger Fabric emphasize more on the security. Since there may be malicious nodes in the network, blockchain systems ought to be Byzantine fault-tolerant(BFT) [16]. They perform BFT state machine replication for deterministic state machines which supports up to one-third faulty replicas. PBFT is the first practical BFT protocol to work in an untrustworthy environment and tolerate Byzantine failures. PBFT consensus creates agreement on the global ledger state in the presence of Byzantine faults, while PoW and PoS only attain only crash fault-tolerant. However, since PBFT is a partially synchronous protocol, the value of the timer used to control the latency boundary is normally hard to set appropriately , which can then cause performance degradation.

3.2 Decentralized Identity

In most designs of Web3.0 network, a decentralized, verifiable and self-sovereign identity system is expected to be part of the Web3.0 infrastructure. Decentralized identity (DID), also called self-sovereign identity (SSI), is considered a key technology to realize the above requirements. In the traditional Internet, identity information is collected and handled by different big firms and organizations. An user cannot use the same social account on different platforms, which not only brings inconvenience to users, but also increases maintenance costs for web service providers. In addition, the right to disclose identity information is not controlled by users, thus there is a huge risk of user privacy leakage. The design goal of DID is to enable users to have full control over their identities, to use the same verifiable identity on different platforms, and to selectively expose or withhold their identity information.

So far, there have been some efforts to establish DID standards and design the specific implementation of DID. In the DID specification published by

W3C [1], DIDs are defined as URIs that associate DID subjects with DID documents. DID documents provide cryptographic materials, verifiable methods and services for interaction with DID. In addition, W3C also defines verifiable credential (VC) [2], which is a machine-readable credential bound to a specific DID and provides a claim of a series of attributes associated with DID. A VC is issued by an entity called issuer to another entity called holder, and verified by entities called verifiers. For example, a university can issue a VC for the DID attributes about the degree the student has earned, and the student can hand over the VC to a company for proof when applying for a position. However, some definitions are still vague and need to be further specified in implementations. In implementation, DID is often closely related to distributed ledger technology, especially blockchain. Dunphy et al. [19] analyzed three representative DID platforms based on distributed ledgers at that time: Sovrin, uPort, and OneName. They believe that these platforms have defects such as dependence on centralized authorities, ad hoc trust and lack of usable user key management. Hyperledger Indy [3] is a DID implementation based on distributed ledgers. Indy Nodes jointly maintain a ledger in a decentralized manner to store identity records related to each DID. Organizations and individuals acquire the right to put transactions on the ledger by getting the role of Trust Anchor. CanDID, proposed by Maram et al. [34], solves the bootstrapping problem of existing standards and implementations by offering legacy-compatibility. It constructs user credentials and performs key recovery based on the user's existing web service account. It also offers more safety guarantees like sybil-resistence and accountability based on multiparty computation.

Based on the above works, we believe that any implementation of DID needs to have three important characteristics: decentralization, verifiability and privacy. However, there are still many trade-offs for DID to be widely used: How to ensure the authenticity and uniqueness of user registration while ensuring decentralization? How to ensure that malicious users can be audited while ensuring the privacy of other users? How to combine a DID system with existing blockchains and other distributed systems? These questions need to be further answered in future works.

3.3 Distributed Storage

Blockchains have been used as distributed storage system in many scenarios to achieve tamper-resistant storage, secure data access and robust data sharing. The features of traceability, immutability and auditability in blockchain are suitable for distributed data storage [33]. Authenticated data structure like Merkle Tree in the blockhead can be used to ensure the integrity of a query on the distributed ledger, which usually does not exist in the traditional database. Besides, fault-tolerant consensus protocol in blockchain detects potential misbehaviour and ensures the reliability of database operations without requiring the involvement of a central trusted party. BigchainDB [35] is the first decentralized database system based on blockchain which leverages some effective blockchain features to construct a shared database in a distrusting environment while avoiding the drawbacks brought from blockchain. FalconDB [39] proposed a blockchain-based database which provides verifiable and integral query results and prevents undesired operations through incentive mechanism.

Fusion between distributed storage system and blockchains is an upward trend since it is possible to apply techniques in traditional distributed database to the blockchain. For example, an significant storage scalability issue in blockchain originates from the full-replication data storage scheme, that is, a full node maintains a record of the whole block data of the ledger. BFT-store [41] utilizes the storage partition approach which is adopted in distributed storage system scale out blockchain. It uses erasure coding to divide a block into several chunks and assign these chunks to each node together with some parties for storage. Fan et al. [20] proposed a group storage mechanism which allows multiple nodes jointly maintain a complete copy of the ledger to reduce per-node storage overhead. Distributed database applies crash fault-tolerant consensus protocol like Raft for state replication while blockchains can use Byzantine fault-tolerant protocol like PBFT to prevent malicious operations. However, it improves security at the expense of performance. Concurrency control techniques adopted in the distributed database system are being used to enhance the performance of blockchains [44]. Hyperledger Fabric supports concurrent transaction execution and uses optimistic concurrency control to improve parallelism. Moreover, sharding has been proved to be an effective way to improve scalability while maintain high security [18]. Examples of recent sharded blockchains include Brokerchain [24], Pyramid [23] and Monoxide [47]. By leveraging sharding protocols in traditional distributed system as a technique to reduce cost of consensus protocols, the transactional throughput increases at scale.

4 Future Directions

4.1 Cross-Chain Technology

Cross-chain technology, also called blockchain interoperability, refers to protocols or platforms that enable homogeneous or heterogeneous blockchains to communicate with each other in a verifiable manner. While there have been many different blockchain platforms aim to serve Web3.0, the lack of a unified cross-chain scheme hinders these platforms from working together as a generic Web3.0 infrastructure. Due to the heterogeneity of existing blockchain platforms, including differences in consensus algorithms, smart contract languages, and access rights, these platforms cannot interoperate through a unified protocol or interface, thus becoming data silos in the Web3.0 world. Liu et al. [32] proposed that a secure interoperability platform is one of the three key enablers of Web 3.0 (the other two are independent blockchains and federated or centralized platforms that provide verifiable states for blockchains). As cross-chain has become an increasingly concerning research topic in both Web3.0 and blockchain scalability, some existing works have tried to design cross-chain platforms or protocols from different aspects. In this section, we try to analyze the characteristics and shortcomings of some representative works, and finally propose a possible cross-chain framework.

Cross-chain communication first appeared between homogeneous blockchains to improve the scalability of the overall system. Sidechain is one of the earliest cross-chain technologies. It mainly refers to the expansion of public blockchains, such as Bitcoin and Ethereum, in the form of another chain [21]. Asset transfer is carried out between the main chain and the sidechain through a certain protocol, a representative example of which is a two-way peg [45]. When a portion of the asset is to be transferred from the main chain to the side chain, a certain amount of tokens on the main chain needs to be sent to a special address and then locked, and tokens of the same value are created on the side chain. This method improves the overall scalability of blockchain without affecting the performance of the main chain, but the application is limited to the asset exchange of homogeneous blockchains. A more general form of cross-chain than sidechains is the cross-shard protocol. Sharding, as one of the most commonly used horizontal expansion methods in traditional distributed systems, is also used by some blockchains to improve their scalability [18,27,50]. Transactions between different shards, i.e. homogeneous blockchains, are carried out in the form of cross-shard transactions. Existing cross-shard protocols usually focus on the safety and atomicity of cross-shard transactions. For example, Omniledger [27] relies on clients to assist with cross-shard transactions. RapidChain [50] splits a cross-shard transaction into multiple intra-shard transactions based on the UTXO model. AHL [18] designs a 2PC and 2PL protocol based on a reference committee to guarantee atomicity and isolation. Although the application scope of the cross-shard protocols has expanded from token exchange to more general transactions compared to sidechains, it can only achieve communication between homogeneous blockchains, which is not sufficient to meet the requirements of Web3.0.

Cross-chain communication between heterogeneous blockchains is another issue for interoperations. Relay technology is one solution by constructing another chain or a relay structure between two chains to verify the validity of the cross-chain transactions and forward them from one chain to the other. Examples include Cosmos [28], and Polkadot [49] solve cross-chain issues by using the Cosmos Hub or Polkadot Relay Chain to provide interconnections. However, the above two solutions has poor security guarantee and fail to consider the active status of nodes, which is not conducive to the efficient execution of the system. He et al. [22] proposed a nested blockchain architecture and dynamically select the high efficient nodes to construct the relay chain for stable and low-latency cross-chain communication. Hashed time-locks is implemented through the Hashed TimeLock Contract (HTLC) [48] to build a bidirectional payment channel within a certain period of time. The lightning network [40] is a typical hashed time-locks project fund on top of the Bitcoin. It has an assumption that the amount of the single payment is small enough. It ensures a small loss of one party in the transaction even if one party defaults. Hashed time-locks allows atomic swap between heterogeneous blockchains which means the valid cross-chain transactions must be executed simultaneously on both chains. Morever, it usually supports only micro-payments and the atomic swap may lead to high

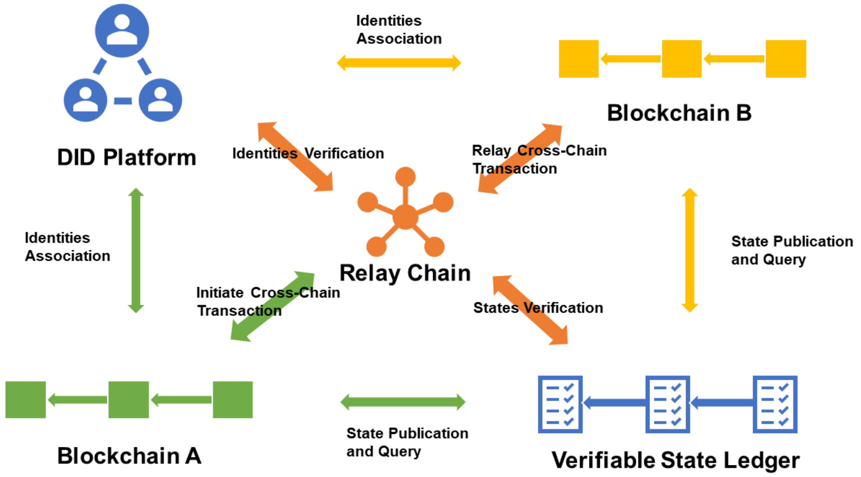

Fig. 1. A general cross-chain framework abstraction

waiting time, which limits its use in large-scale applications. The Notary mechanism [42] uses a third party to propose transactions and exchange data between two chains which does not require the authentication of transaction participants' identity. It is a more resonable and secure approach since the system security is enhanced when some nodes are injected by malicious one or crash errors occur. However, it may involve unverified nodes operating in a dishonest manner, since the identity verification during cross-chain communication is critical for making blockchains interoperable.

Based on the above works, we propose a simple but universal cross-chain framework design, as shown in Fig. 1. We believe that a general cross-chain framework requires three key pieces of information: global user identities, the ledger states of different blockchains, and the network and service information of different blockchains. Therefore our proposed framework consists of the following three components: a DID platform, a verifiable state ledger, and a relay chain. The DID platform provides global verifiable identity credentials, and each user associates their identities registered in different chains with their DID. Different blockchains register state information that requires external verification in the cross-chain process on the verifiable state ledger. For example, a consortium blockchain member can publish the content and signature of a specific block that is otherwise inaccessible to the outside world through the verifiable ledger. The relay chain is responsible for registering the network addresses and service interfaces of different blockchains, forwarding transactions to different blockchains and verifying cross-chain transactions with the help of the DID platform and verifiable state ledger. Here we do not specify the implementation forms of these three components, but we commend adopting blockchain using hybrid consensus to ensure performance and decentralization at the same time. In addition, each component can be implemented on the same blockchain together with other components, or a component can be composed of multiple blockchains. We leave more specific implementation details to future works, and hope this cross-chain framework abstraction can be helpful to researchers in this field.

4.2 AI-based caching and storaging

In Web 2.0, due to the centralization of data storage, it is only necessary to store the content attributes according to their value. In Web 3.0, due to the decentralization of data storage, the storage of content across nodes, and the unique identification of user IDs and other characteristics, it has changed to mainly measure the value of user attributes for storage. Compared with Web 2.0, web 3.0 storage considerations are more complex; at this time, using AI technology is also a potential solution. Compared with Web 2.0, web 3.0 storage considerations are more complex, so solving storage problems under challenging situations through the advantages of AI technology may be a potential solution. Decentralized storage is one of the critical technologies of web 3.0, which faces many challenges in future implementation. These are potential research directions, such as cross-node data placement, user request content retrieval, hit rate improvement of storage node, and availability guarantee of storage content.

Cross-node data placement. Web 3.0 is inherently decentralized in content storage and does not require the creation of data center nodes for each application. So Web 3.0 rarely deploys applications that run on a single server (node) or store data in a single database. But, this does not mean that nodes are not needed to store data. That is, Web 3.0 nodes store mixed types of data associated with users from various applications, i.e., data generated by the same application need to be placed across nodes based on user attributes. Therefore, placing the user-requested data across nodes by selecting appropriate nodes among many nodes ensures low response time and high space utilization of stored content.

User request content retrieval. The data is placed across nodes, which means that not only a few or even one service or application's data is stored in each server node. This also means that each service or application may occupy all the available storage servers to store the data. Additionally, due to the vast amount of requested data, how to retrieve the requested content and efficiently find the storage location where cache the requested content is one of the potential future research directions.

Hit rate improvement of storage node. In order to ensure the quality of user experience, it is theoretically necessary to store the user's future requested content on the storage nodes closer to the user to reduce the response time. Although the total number of nodes is large, the number of nodes adjacent to users is smaller than the number of users. Additionally, the edge server storage space is limited, so storing the content of all users required through adjacent nodes is difficult. So it is impossible to place the content of future user requests in the neighboring nodes. Furthermore, because of the variety of applications and the vast amount of data content, it is impossible to know which content will be requested by the user in the future. If the stored content is not what the user will access in the future, this will seriously affect the quality of the user's

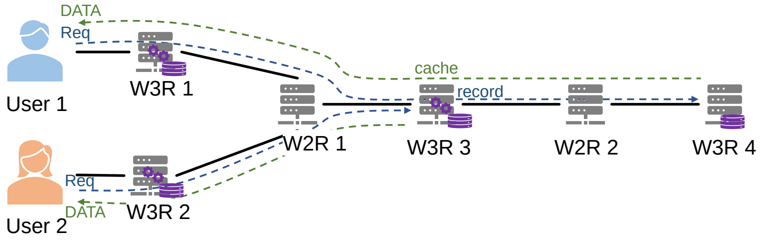

Fig. 2. Incrementally deployable content-based addressing architecture

experience. Therefore, how to improve the hit rate that the content stored in nodes is the content requested by users in the future is a research focus of node storage in the web3.0 storage model.

Availability guarantee of storage content. In daily use, the storage node is unavailable due to the aging of the equipment and other irresistible factors, which affects the use of the data stored on the node. When the node is out of the network, how to ensure that the data inside the unavailable node is still available is an inevitable future research direction.

4.3 Web 3.0 transmission

Compared with the centralized features of web 2.0, web 3.0 aims to establish a user-owned and user-constructed decentralized network ecology. Accordingly, web 3.0 architecture has two main characteristics: content-driven addressing and multi-client and multi-server transmission. Hence, for these two features, we analyze the challenges and propose potential solutions in transmission perspective in this paper.

Content Driven Addressing In contrast to existing network architecture, a key feature of web 3.0 transmissions is content-driven. Given the developments from IPv4 to IPv6, incremental deployment is particularly important in the network architecture evolution. Hence, we propose an incrementally deployable content-based addressing architecture.

As shown in the Suppose there are two types of routers, one is the traditional router that supports IP traffic, called *W2R* for convenience, and the other is the content-based addressing-enabled router for web 3.0, called *W3R*. When $User_1$ sends a content request, $W3R_1$ converts the content-based request message into an IP-based packet, where the content index is converted into an IP address and forwarded to $W2R_1$. In this paper, we do not discuss specific techniques of conversion. A strawman way is mapping, whose corresponding massive mapping entries can be mitigated by [12,52]. $W2R_1$ forwards the IP-based content request to $W3R_3$ as per the operation of IP traffic; likewise, the request will be eventually forwarded to $W3R_4$. Assuming that $W3R_4$ can provide the requested contents, then it can deliver the data based on the information in the packet header back to $User_1$. $W3R_3$ will cache the corresponding content when transmitting it in response to the same content requests. When *User*$_2$ proposes the same content request, the cached of the last request of *User*$_1$ will be delivered back by *W3R*$_3$ directly.

Multi-clients and multi-servers transmission In web 3.0, all nodes can act as content producers, and there exists the demand to integrate content from multiple parties, i.e., a content request may need multi-users to respond. However, when multiple parties send back the required data at a short interval will lead to network congestion or even packet loss. Moreover, it is common that multiple requestors to request the same popular content, i.e., the transmission between multiple clients and multiple servers (MCMS) situation, which can aggravate such issues. In this paper, we propose a traffic control strategy for this potential MCMS situation.

Some existing traffic control in network transmission of web2.0 is usually conducted by the sender, which determines whether there are congestion based on some congestion control signals or timestamp information, and then executes corresponding traffic control actions. Based on this, we propose a request-side assisted traffic control strategy. When the congestion ratio of multiple response traffics of a request reaches α (the probability of the congestion caused by this requestor is positively correlated with α), then the requestor will reduce the demand with a probability of $\beta\alpha$ (where $\beta \in [0, 1]$), such as requestor reduces the desired bitrate of the audio or video. Moreover, the requestor will re-diffuse the content request in the network to obtain cached content from other nodes. In this way, the impact of congestion on other users can be mitigated. There still are challenges in implementing this strategy, such as timing synchronization of large-scale networks, incremental deployment, etc., which still require further research and exploration by all network researchers.

4.4 Security issues

Compared with Web2.0, where user data is stored and controlled by service providers to optimize the user's experience on the Internet. The basic goal of Web3.0 is for users to control and manage their own data, and service provider need to apply users for data use, thus more attention is paid to user privacy protection in Web3.0. Therefore, compared with the traditional security protection on the server side, the client side and the data transmission, Web3.0 has stricter security requirements.

Security on the user side In Web3.0, personal digital assets are completely owned and controlled by individuals, and the server no longer has backup of user data. This makes users take on a greater responsibility for their data than in the Web2.0 era, which also means that users are more vulnerable to attacks and may suffer greater losses when attacked. For example, the rapidly developing Crypto Wallet [46], NFT, etc. have great real value (which can be converted into real currency), so they are more frequently attacked by hackers such as private key theft, airdrop scams(NFT) and authorization attacks. Moreover, Whether Web3.0 data will be accepted by other user is also a question while the cyberspace is an untrustworthy environment and the peer identity should be verified. The Web3.0 service provider and user must making a choice about who and how to make the authentication.

Security in transmission Web3.0's emphasis on privacy makes privacy protection algorithms more important. In the Web2.0 system, it's more necessary to protect the user's data from being known by third parties other than the server and the user during the transmission process. In Web3.0, there is also user identity information that needs to be protected to help users ”incognito”. One solution is Mixnet [43], which is a decentralized network arranged in a multi-layered format. The user converts the message packets into encrypted ”Sphinx” [17] packets instead of sending messages directly over the Internet and the ”Sphinx” packets are untraceable, and then shuffles through the Mixnode (mixed network).

Security on the server side The problems on the server side are caused by the design and technology of the Web3.0 system. Even though blockchain is one of the most secure technologies, hackers may get unauthorized access to wallets and other digital assets by exploiting cryptography flaws. And if the breach occurs, it is nearly hard to recover the lost funds or digital assets. Moreover, there is no way to track completed transactions and retrieve lost money. This makes it necessary to provide an effective response at the system fundamental level to assure users of the safety and security of their data and information. Key management is the basis for users to conduct transactions in Web3.0, but at the same time key management is also a very difficult problem. This drives users to choose custodial wallet over non-custodial wallet. However, custodial wallet will lead to the creation of a kind of centralized management wallet application, which is contrary to the fully decentralized direction of Web3.0.

Data decentralization and anonymity are the cornerstones and advantages of Web3.0, but while strengthening user's data privacy also mean that it's difficult to be regulated by the government. Cause Web3.0 to become the platform of many illegal crimes, which has brought great limitations to large-scale promotion of blockchain and Web3.0. In conclusion, before solving the above problems, there is still a long way to go and a lot of content that needs continuous research.

5 Conclusion

Web3.0 is a fully open and decentralized Internet which allows each user to control their data, but there still exist some technical challenges. This papers first introduce the current network architecture, and then analyze several key technologies of web3.0 network, including cross-chain interaction, web3.0 storage, web3.0 transmission and security issues, which we believe will provide reference for relevant researches.

References

1. W3C: Decentralized Identifiers (DIDs) v1.0, `https://www.w3.org/TR/did-core/`, last accessed 2022/10/15.
2. W3C: Decentralized Identifiers (DIDs) v1.0, `https://www.w3.org/TR/vc-data-model/`, last accessed 2022/10/15.
3. Hyperledger Indy, `https://www.hyperledger.org/use/hyperledger-indy/`, last accessed 2022/10/15.
4. Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., et al.: Hyperledger fabric: a distributed operating system for permissioned blockchains. In: Proceedings of the thirteenth EuroSys conference. pp. 1–15 (2018)
5. Arlitt, M.F., Cherkasova, L., Dilley, J., Friedrich, R., Jin, T.: Evaluating content management techniques for web proxy caches. SIGMETRICS Perform. Evaluation Rev. 27(4), 3–11 (2000), `https://doi.org/10.1145/346000.346003`
6. Atzei, N., Bartoletti, M., Cimoli, T.: A survey of attacks on ethereum smart contracts (sok). In: International conference on principles of security and trust. pp. 164–186. Springer (2017)
7. Brown, R.G.: The corda platform: An introduction. Retrieved 27, 2018 (2018)
8. Buchman, E.: Tendermint: Byzantine fault tolerance in the age of blockchains. Ph.D. thesis, University of Guelph (2016)
9. Buterin, V., et al.: A next-generation smart contract and decentralized application platform. white paper 3(37), 2–1 (2014)
10. Cachin, C., Androulaki, E., De Caro, A., Kind, A., Osborne, M., Schubert, S., Sorniotti, A., Vukolic, M.: Blockchains and consensus protocols. the Wild (2017)
11. Chase, J.M.: Quorum white paper. Accessed: Jan 17, 2019 (2016)
12. Cong, P., Zhang, Y., Liu, B., Wang, W., Xiong, Z., Xu, K.: A&b: Ai and block-based tcam entries replacement scheme for routers. IEEE Journal on Selected Areas in Communications 40(9), 2643–2661 (2022)
13. Cong, P., Zhang, Y., Liu, Z., Baker, T., Tawfik, H., Wang, W., Xu, K., Li, R., Li, F.: A deep reinforcement learning-based multi-optimality routing scheme for dynamic iot networks. Computer Networks 192, 108057 (2021)
14. Cong, P., Zhang, Y., Wang, L., Ni, H., Wang, W., Gong, X., Yang, T., Li, D., Xu, K.: Break the blackbox! desensitize intra-domain information for inter-domain routing. In: 2022 IEEE/ACM 30th International Symposium on Quality of Service (IWQoS). pp. 1–10. IEEE (2022)
15. Cong, P., Zhang, Y., Wang, W., Xu, K., Li, R., Li, F.: A deep reinforcement learning-based routing scheme with two modes for dynamic networks. In: ICC 2021-IEEE International Conference on Communications. pp. 1–6. IEEE (2021)
16. Correia, M.: From byzantine consensus to blockchain consensus. In: Essentials of Blockchain Technology, pp. 41–80. Chapman and Hall/CRC (2019)
17. Danezis, G., Goldberg, I.: Sphinx: A compact and provably secure mix format. In: 2009 30th IEEE Symposium on Security and Privacy. pp. 269–282 (2009)
18. Dang, H., Dinh, T.T.A., Loghin, D., Chang, E.C., Lin, Q., Ooi, B.C.: Towards scaling blockchain systems via sharding. In: Proceedings of the 2019 international conference on management of data. pp. 123–140 (2019)
19. Dunphy, P., Petitcolas, F.A.: A first look at identity management schemes on the blockchain. IEEE security & privacy 16(4), 20–29 (2018)
20. Fan, Y., Qiu, T., Zhang, L., Xu, T., Liu, W., Zhou, X., Wan, Z.: Dlbn: Group storage mechanism based on double layer blockchain network. IEEE Internet of Things Journal (2022)

21. Gaži, P., Kiayias, A., Zindros, D.: Proof-of-stake sidechains. In: 2019 IEEE Symposium on Security and Privacy (SP). pp. 139–156. IEEE (2019)
22. He, X., Zhang, Y., Wang, X.: A scalable nested blockchain framework with dynamic node selection approach for iot. In: 2022 IEEE International Performance, Computing, and Communications Conference (IPCCC). pp. 108–113. IEEE (2022)
23. Hong, Z., Guo, S., Li, P., Chen, W.: Pyramid: A layered sharding blockchain system. In: IEEE INFOCOM 2021-IEEE Conference on Computer Communications. pp. 1–10. IEEE (2021)
24. Huang, H., Peng, X., Zhan, J., Zhang, S., Lin, Y., Zheng, Z., Guo, S.: Brokerchain: A cross-shard blockchain protocol for account/balance-based state sharding. In: IEEE INFOCOM (2022)
25. Kakhki, A.M., Jero, S., Choffnes, D., Nita-Rotaru, C., Mislove, A.: Taking a long look at quic: an approach for rigorous evaluation of rapidly evolving transport protocols. In: proceedings of the 2017 internet measurement conference. pp. 290–303 (2017)
26. Khalili, R., Gast, N., Popovic, M., Le Boudec, J.Y.: Mptcp is not pareto-optimal: Performance issues and a possible solution. IEEE/ACM Transactions On Networking 21(5), 1651–1665 (2013)
27. Kokoris-Kogias, E., Jovanovic, P., Gasser, L., Gailly, N., Syta, E., Ford, B.: Omniledger: A secure, scale-out, decentralized ledger via sharding. In: 2018 IEEE Symposium on Security and Privacy (SP). pp. 583–598. IEEE (2018)
28. Kwon, J., Buchman, E.: Cosmos whitepaper. A Netw. Distrib. Ledgers (2019)
29. Langley, A., Riddoch, A., Wilk, A., et al.: The quic transport protocol: Design and internet-scale deployment. In: Proceedings of the conference of the ACM special interest group on data communication. pp. 183–196 (2017)
30. Li, P., Zhang, Y., Wang, W., Zhao, K., Lian, B., Xu, K., Zhang, Z.: Frend for edge servers: Reduce server number! keeping service quality! In: 2021 IEEE 23rd Int Conf on High Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys), Haikou, Hainan, China, December 20-22, 2021. pp. 107–114. IEEE (2021)
31. Li, P., Zhang, Y., Zhang, H., Wang, W., Xu, K., Zhang, Z.: CRATES: A cache replacement algorithm for low access frequency period in edge server. In: 17th International Conference on Mobility, Sensing and Networking, MSN 2021, Exeter, United Kingdom, December 13-15, 2021. pp. 128–135. IEEE (2021), https://doi.org/10.1109/MSN53354.2021.00033
32. Liu, Z., Xiang, Y., Shi, J., Gao, P., Wang, H., Xiao, X., Wen, B., Li, Q., Hu, Y.C.: Make web3. 0 connected. IEEE Transactions on Dependable and Secure Computing (2021)
33. Maiyya, S., Zakhary, V., Amiri, M.J., Agrawal, D., El Abbadi, A.: Database and distributed computing foundations of blockchains. In: Proceedings of the 2019 International Conference on Management of Data. pp. 2036–2041 (2019)
34. Maram, D., Malvai, H., Zhang, F., Jean-Louis, N., Frolov, A., Kell, T., Lobban, T., Moy, C., Juels, A., Miller, A.: Candid: Can-do decentralized identity with legacy compatibility, sybil-resistance, and accountability. In: 2021 IEEE Symposium on Security and Privacy (SP). pp. 1348–1366. IEEE (2021)
35. McConaghy, T., Marques, R., Müller, A., De Jonghe, D., McConaghy, T., McMullen, G., Henderson, R., Bellemare, S., Granzotto, A.: Bigchaindb: a scalable blockchain database. white paper, BigChainDB (2016)
36. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. Decentralized Business Review p. 21260 (2008)

37. Narayanan, A., Verma, S., Ramadan, E., Babaie, P., Zhang, Z.L.: Deepcache: A deep learning based framework for content caching. In: Proceedings of the 2018 Workshop on Network Meets AI & ML. pp. 48–53. ACM (2018)
38. Nishida, Y., Eardley, P.: Mptcp-multipath tcp. In: WG meeting. vol. 5 (2011)
39. Peng, Y., Du, M., Li, F., Cheng, R., Song, D.: Falcondb: Blockchain-based collaborative database. In: Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data. pp. 637–652 (2020)
40. Poon, J., Dryja, T.: The bitcoin lightning network: Scalable off-chain instant payments (2016)
41. Qi, X., Zhang, Z., Jin, C., Zhou, A.: Bft-store: Storage partition for permissioned blockchain via erasure coding. In: 2020 IEEE 36th International Conference on Data Engineering (ICDE). pp. 1926–1929. IEEE (2020)
42. Qin, K., Gervais, A.: An overview of blockchain scalability, interoperability and sustainability. Hochschule Luzern Imperial College London Liquidity Network (2018)
43. Sampigethaya, K., Poovendran, R.: A survey on mix networks and their secure applications. Proceedings of the IEEE 94(12), 2142–2181 (2006)
44. Sharma, A., Schuhknecht, F.M., Agrawal, D., Dittrich, J.: Blurring the lines between blockchains and database systems: the case of hyperledger fabric. In: Proceedings of the 2019 International Conference on Management of Data. pp. 105–122 (2019)
45. Singh, A., Click, K., Parizi, R.M., Zhang, Q., Dehghantanha, A., Choo, K.K.R.: Sidechain technologies in blockchain networks: An examination and state-of-the-art review. Journal of Network and Computer Applications 149, 102471 (2020)
46. Suratkar, S., Shirole, M., Bhirud, S.: Cryptocurrency wallet: A review. In: 2020 4th International Conference on Computer, Communication and Signal Processing (ICCCSP). pp. 1–7 (2020)
47. Wang, J., Wang, H.: Monoxide: Scale out blockchains with asynchronous consensus zones. In: 16th USENIX symposium on networked systems design and implementation (NSDI 19). pp. 95–112 (2019)
48. Wiki, B.: Hash time locked contracts (2016)
49. Wood, G.: Polkadot: Vision for a heterogeneous multi-chain framework. White Paper 21, 2327–4662 (2016)
50. Zamani, M., Movahedi, M., Raykova, M.: Rapidchain: Scaling blockchain via full sharding. In: Proceedings of the 2018 ACM SIGSAC conference on computer and communications security. pp. 931–948 (2018)
51. Zhang, X., Qi, Z., Min, G., Miao, W., Fan, Q., Ma, Z.: Cooperative edge caching based on temporal convolutional networks. IEEE Trans. Parallel Distributed Syst. 33(9), 2093–2105 (2022), https://doi.org/10.1109/TPDS.2021.3135257
52. Zhang, Y., Cong, P., Liu, B., Wang, W., Xu, K.: Air: An ai-based tcam entry replacement scheme for routers. In: 2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS). pp. 1–10. IEEE (2021)
53. Zhang, Y., Li, P., Zhang, Z., Bai, B., Zhang, G., Wang, W., Lian, B., Xu, K.: Autosight: Distributed edge caching in short video network. IEEE Netw. 34(3), 194–199 (2020), https://doi.org/10.1109/MNET.001.1900345
54. Zhang, Y., Li, P., Zhang, Z., Zhang, C., Wang, W., Ning, Y., Lian, B.: Graphinf: A gcn-based popularity prediction system for short video networks. In: Ku, W., Kanemasa, Y., Serhani, M.A., Zhang, L. (eds.) Web Services - ICWS 2020 - 27th International Conference, Held as Part of the Services Conference Federation, SCF 2020, Honolulu, HI, USA, September 18-20, 2020, Proceedings. Lecture Notes in Computer Science, vol. 12406, pp. 61–76. Springer (2020)