

Received January 20, 2022, accepted March 15, 2022, date of publication March 23, 2022, date of current version April 6, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3161744

RBaaS: A Robust Blockchain as a Service Paradigm in Cloud-Edge Collaborative Environment

ZHENGONG CAI¹, GUOZHENG YANG², SHAOYONG XU¹, CHENG ZANG²,
JIAJUN CHEN², PINGPING HANG^{1,3}, AND BOWEI YANG^{1,3}

¹School of Software Technology, Zhejiang University, Hangzhou 310012, China

²China Zhesang Bank Company Ltd., Hangzhou 311200, China

³School of Aeronautics and Astronautics, Zhejiang University, Hangzhou 310012, China

Corresponding author: Bowei Yang (bowei@zju.edu.cn)

This work was supported by the China Zhesang Bank-Zhejiang University Joint Research Center.

ABSTRACT As a decentralized distributed ledger, blockchain is endowed with immutability, traceability, anonymity, and transparency, which has got rapid development in cryptocurrency and production as a new trend. In the meantime, the occurrence of Blockchain-as-a-Service (BaaS) helps in mitigation of the complexity and difficulty in deployment and management of blockchain systems and makes it easier to concentrate on business logic implementation for developers. However, most existing BaaS systems are hosted in the environment of cloud vendors, which has also incurred vendor lock-in risk and impairs the inherent trustless characteristic of blockchain. Though present BaaS systems own a level of availability by using cloud computing or edge computing as infrastructure, the availability is limited, considering the availability of network connections and the data center itself. In this article, a novel cloud-edge collaborative BaaS paradigm is proposed. With the assistance of redundant blockchain node candidates, leader election, and edge network self-healing components, the proposed BaaS system is equipped to extend BaaS to the on-premises edge or private cloud, and realize high availability of blockchain systems in edge-autonomy and cross-data-center scenarios. Through testing a simplified system in a simulated environment in cloud servers, this proposed BaaS system has an acceptable throughput under specific transaction sending rates without much performance degradation due to containerization and data synchronization compared with the original blockchain on-premises deployment method.

INDEX TERMS Blockchain, blockchain as a service, cloud computing, cloud-edge collaboration, edge computing, high availability.


I. INTRODUCTION

The past few years have seen the ever-growing development of emerging technologies such as blockchain, edge computing, and cloud computing. Blockchain was first conceptualized by Satoshi Nakamoto [1] as a data structure for bitcoin in 2008. With the help of its decentralized distributed cryptographic ledger and append-only data structure, blockchain has been endowed with features like decentralization, immutability, traceability, transparency, and a level of anonymity. The popularization of cryptocurrency and employment in commerce have further fueled the popularity of the blockchain technique.

Despite the many benefits of blockchain, the complexity and difficult maintenance of blockchain systems have

stopped developers from concentrating on their business logic or even the employment of blockchain. To address these challenges, Blockchain-as-a-Service (BaaS) has been put forward, serving as a new infrastructure to simplify blockchain deployment, monitoring, and maintenance. Many tech giants have their own BaaS platforms, including, but not limited to, Amazon AWS BaaS, Microsoft Azure BaaS, Oracle BaaS etc. [2] There are also open-source BaaS projects like BlockForm.

Nonetheless, problems remain to be addressed in the present BaaS platforms. For one thing, BaaS offers an integrated blockchain framework to simplify blockchain deployment. For another, BaaS platforms have also incurred vendor lock-in risk [3], new trust concerns regarding the BaaS providers who are presumed to be trustable, and the data of blockchain usually stored in the cloud. Data localization is possible but not an easy task for BaaS. [4]

The associate editor coordinating the review of this manuscript and approving it for publication was Chakchai So-In .

In addition, as far as we know, the present BaaS platform is not targeted for the cloud-edge collaborative scenario, and the BaaS system is usually hosted in a single cloud, including blockchain data [4]. The cloud-edge scenario means such an unstable network that the network connection between them may be interrupted for some time. For a Kubernetes-based BaaS system, the conventional Kubernetes framework would evict pods in worker nodes located at the edge when the network disconnection between public cloud and edge nodes lasts for a specific period.

The proposed BaaS model in this paper is based on Kubernetes. To extend the original Kubernetes to the cloud-edge collaborative scenario, a simple way to realize this is to utilize edge computing frameworks designed for this scenario, such as Openyurt and KubeEdge. For instance, Openyurt is a project under the Cloud Native Computing Foundation (CNCF), one of whose objectives is to address the challenges of cloud-edge orchestration under unstable cloud-edge networks and maintain a non-invasive architecture. Using it makes it possible to deploy Kubernetes worker nodes at the edge where lots of edge nodes work in private networks without their own external public IP addresses and communicate with the world wide web via a router.

Nevertheless, there are still problems remaining in this framework. Using tools like Openyurt or KubeEdge, the pods deployed in edge nodes will not be evicted when the cloud control plane (specifically Kubernetes apiserver component) loses network connection with the edge nodes for a moment. However, what if some edge nodes crash or pods in an edge node work abnormally when the connection between cloud and edge breaks down temporarily? In this scenario, since the network connection between cloud and edges is broken, in no way can cloud nodes collect failure information from edge nodes, such as liveness probe or readiness probe information from “kubenet” components in edge nodes. It is not just a bug of tools like Openyurt. Instead, it will be a common problem hard to solve because in no circumstances can the cloud clearly distinguish between a remote edge server crash and a network connection failure without extra assistance.

It is acceptable for consortium blockchain to work when one consortium blockchain node fails as long as the chain has a quorum. For instance, when Practical Byzantine Fault Tolerance (PBFT) algorithm [5] is used as the consensus algorithm, provided that more than two-thirds of the blockchain nodes work normally, the blockchain system will function. However, the secret, certificate, and blockchain data in the edge pod would be lost along with the failure, and the user assigned with the blockchain node would be unable to get access to it. Even worse, a consortium blockchain system may fail if the private data center itself breaks down.

A robust BaaS paradigm, called RBaaS with cloud-edge collaborative capability and high availability is proposed in this paper, especially for the on-premises edge or private cloud of possibly a different cloud vendor, and the block data is stored at the edge zones. For accuracy, the edge may be referred to as edge nodes or edge zones hereafter and edge

zones refer particularly to edge data centers in this paper. Besides, edge nodes correspond to servers located in edge data centers in private cloud or on-premises. In the end, some performance tests have been taken on a simplified RBaaS system in different circumstances, and the results indicate that this platform gets an acceptable throughput without much performance degradation under experiment conditions.

A. RESEARCH CONTRIBUTIONS

This paper proposes a cloud-edge collaborative, highly available, privacy-preserving BaaS paradigm that utilizes cloud computing, edge computing, and blockchain technique to localize blockchain data and endow BaaS with edge autonomy capability. The main contributions of this paper are as follows.

- This paper proposes a paradigm of BaaS architecture with cloud-edge coordination capability and high availability, which can serve as a stepping stone towards further research and the potential incorporation of blockchain and edge computing in hybrid cloud.
- The deployment method and mechanism of the proposed highly available BaaS architecture are also introduced in this research.
- A performance comparison between a simplified proposed BaaS system and a traditional blockchain system deployed on-premises is presented in this paper through tests.

B. PAPER STRUCTURE

The rest of this paper is laid out as follows. Section II presents preliminaries of the proposed RBaaS system, including the necessity of extending BaaS to the edge and some assumptions. Section III introduces the related works of the combination of blockchain and edge computing. Section IV illustrates the architecture of the highly available blockchain system presented in the paper. Section V gives a performance evaluation of this system compared to the original blockchain system on-premises and analyzes performance influence factors. In the end, we conclude this paper and present future work.

II. PRELIMINARIES

A. NEED FOR BaaS IN CLOUD-EDGE COLLABORATIVE ENVIRONMENT

Based on the present survey, the necessity of BaaS in a cloud-edge collaborative environment can be summarized below.

- Better privacy and decentralization: BaaS provides a better way to build up a full-fledged blockchain system. However, in a traditional BaaS system, data of blockchain is usually stored at the cloud of service vendor [4], which is a violation of the intrinsic trustless nature of blockchain and presumes the cloud vendor to be a trustable third party. By localizing the block data on-premises, better privacy and decentralization of a blockchain system can be achieved.
- Reuse of on-premises resources and high cost-efficiency: Though more and more enterprises choose to

migrate to the public cloud, it is not an easy decision on cloud migration considering underlying cloud adoption challenges like vendor lock-in and security risks [6]. There are still some organizations unwilling to migrate their systems to the cloud [7]. Besides, cloud migration may not be economical without proper initiatives [8]. By extending BaaS to the edge, enterprises can obtain a third choice apart from deploying a blockchain platform totally in the cloud or on-premises entirely. They can even deploy their blockchain system based on open-source BaaS programs without so much dependence on cloud vendor.

- Reduction of latency and better quality of service: Extending the BaaS to the edge means better network stability, together with lower delays between blockchain nodes and users.
- Higher Availability: By extending BaaS to distributed edge data centers with proper configuration, blockchain ledger data is stored in distributed edge data centers, which provides the blockchain system with extra availability and reliability and avoids a single point of failure in a single cloud. It is especially beneficial for three data centers in two cities deployment as a disaster recovery scheme.

B. SCENARIO ASSUMPTIONS

The targeted scenario of the RBaaS system presented in this article is based on three assumptions.

Firstly, the network connection between the cloud control plane and edge nodes is unstable, which means the edge nodes may not be able to communicate with the cloud control plane from time to time. Thus cloud control plane may not be able to collect information from edge nodes timely. In worst conditions, the cloud control plane may lose connection totally with edge nodes for minutes, hours, or even days.

Secondly, edge nodes are located inside some local area networks, which share a public IP address and connect to the cloud control plane via a router for every edge zone. Therefore, the cloud control plane cannot connect to the edge nodes simply by their internal IP addresses or router IP addresses as in traditional Kubernetes.

Lastly, edge nodes are assumed to communicate with each other through internal IP addresses. In addition, low network latency between edge nodes at the same edge is assumed. Dedicated networks are assumed to be used among different edge data centers in this research. Besides, it is taken into account that a network partition may occur among edge data centers sometimes. As far as we know, there are some CNI plugins for Kubernetes still under development, which can get rid of dedicated network cable and realize the communication among pods located in different edge data centers. However, it is out of the scope of this research, and we will not discuss it in this paper.

III. RELATED WORKS

Our work focuses on a blockchain-as-a-service platform with cloud-edge collaboration capability to enhance privacy and

trustability, and integration of blockchain and edge computing techniques. In this section, research on the integration of blockchain and edge computing is presented first. Then work related to BaaS systems is discussed.

A. THE INTEGRATION OF BLOCKCHAIN AND EDGE COMPUTING

With lots of advantages, edge computing and blockchain on their own still have many limitations. On the one hand, as the number of heterogeneous edge devices, including some IoT devices, climbs up, the management, privacy, and security of edge computing are becoming more and more challenging. On the other hand, blockchain technique in practice is confronted with limitations such as low throughput and resource exhausting like storage capacity. [9] These limitations can be mitigated with the incorporation of blockchain and edge computing.

There has already been some research trying to combine blockchain technology with edge computing. Sharma *et al.* [10] present a novel blockchain-based distributed cloud architecture integrating software-defined networking, edge computing technique, and blockchain to process ever-growing raw data from IoT devices at the edge side. Blockchain is used for distributed cloud and SDN controllers at the edge in this research. The study of [11] proposes a distributed authentication system to address the challenge of isolated information among IoT platforms by integration of blockchain and edge computing. A consortium blockchain using an optimized practical Byzantine fault tolerance consensus algorithm is designed to store authentication information in the research. There are also various incorporation scenarios of blockchain and edge computing, such as smart grid [12], distributed control system [13], cooperative edge computing [14], smart vehicle [15]. The research mainly focuses on utilizing blockchain techniques to promote edge computing capability. Nevertheless, deploying and maintaining a blockchain system for a specific application often takes great pains due to the complexity of blockchain technology itself.

B. CURRENT BAAS SYSTEMS

To help alleviate the difficulties of blockchain application and reduce management overheads, Blockchain-as-a-Service comes into being and is getting more and more mature both in academia and industries. Besides, BaaS systems have been used in many fields, such as essential transaction services, Internet of Things (IoT) [16], [17], Software Defined Networking (SDN) [18], [19].

Well-known BaaS systems in commercial fields include, but are not limited to, Microsoft BaaS, IBM BaaS, Amazon BaaS, and Alibaba BaaS. In addition to the BaaS platforms of tech giants in industries, there is also much BaaS research in academia. Most of it concentrates on endowing BaaS systems with new features. Wan *et al.* [20] believing that a centralized BaaS system will impair the intrinsic decentralization and trustless mechanism of blockchain, proposed a novel BaaS paradigm (NBaaS) to help reduce some

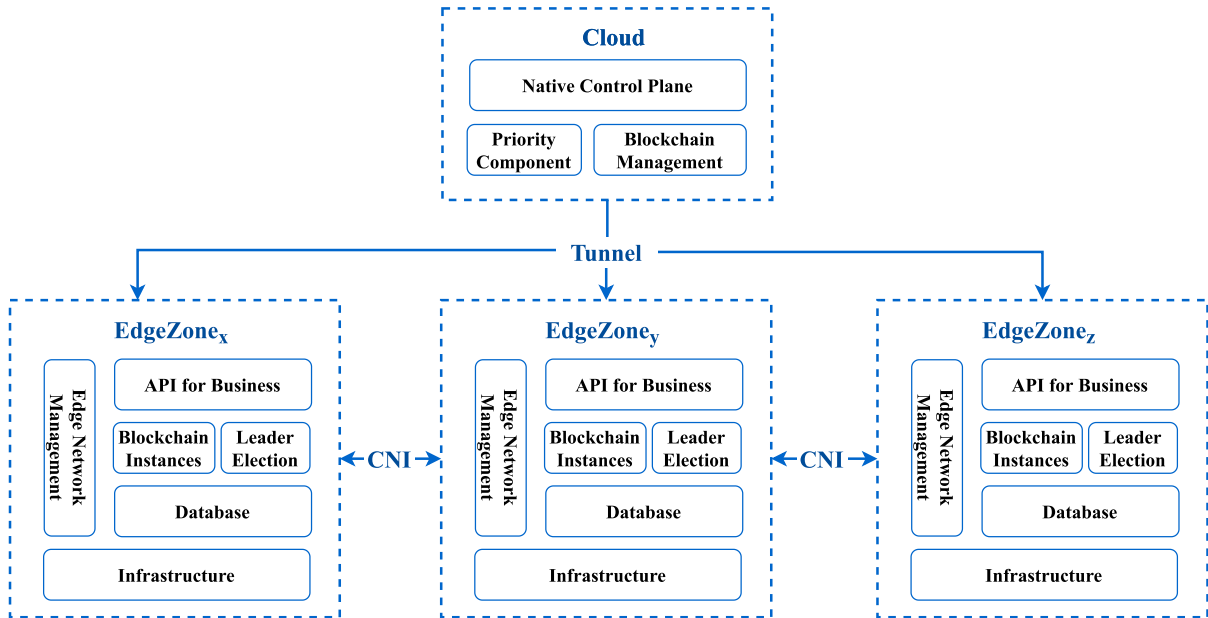


FIGURE 1. RBaaS framework.

limitations of PaaS-based BaaS. By integrating deployable components to BaaS, users are capable of reusing the components generated for their blockchain instances in other computing environments. Chen *et al.* [21] argue that the choice between a private blockchain and consortium blockchain from the outset will bring users into a dilemma for long-term use. They introduce a Full-Spectrum Blockchain as a Service (FSBaaS) to combine private blockchain runtime with consortium blockchain runtime through a unified interface. There are also many other BaaS systems focusing on different aspects of a service promotion, such as security and vulnerability detection [22], Function-as-a-Service integration with lighter implementation [23].

Nevertheless, these proposed BaaS systems, either industrial or academic, mainly put their concentrations on a single cloud, without considering the possibility of hybrid deployment in a cloud-edge collaborative environment, which can help promote the privacy and decentralization of BaaS systems. To the best of our knowledge, the proposed BaaS system is a relatively novel attempt to extend BaaS to the edge side.

IV. SYSTEM STRUCTURE

A. ARCHITECTURE DESIGN OVERVIEW

Since the primary purpose of this research is not to implement a full-fledged BaaS system, the implementation details of a BaaS system are out of the scope of this research, and not all components of a mature BaaS system are instantiated. Instead, focusing on integrating BaaS, cloud-edge collaboration capability, and high availability, we realize a simplified RBaaS model for test and performance evaluation.

The BaaS paradigm described in this paper is based on Kubernetes and Openyurt, which can be divided into cloud and edge parts, as illustrated in Fig. 1. The cloud comprises

mainly three parts. Native Control Plane in the cloud means native Kubernetes control plane components and Openyurt addons including etcd database, apiservers, schedulers, and controller managers. The Blockchain Management component serves to manage blockchain tenants, including, but not limited to, certificate management and organization management. The Priority Component aims at offering topology priority information for leader election of blockchain nodes. As for the edge side, six parts are included in every edge data center. From the bottom up, they are the infrastructure layer, multi-master database, blockchain instances, leader election component, API for Business, and Network Management, respectively. The database here is used to store blockchain world state and ledger data. In this research, the blockchain type we use is FISCO BCOS, an enterprise-level open-source financial consortium blockchain platform, using PBFT as its consensus algorithm. At the edge, the Leader Election component provides extra high availability for the blockchain nodes. Since the network management of native Kubernetes is impaired when the cloud-edge network connection is unstable, we implement a new network component to assist in repairing the network at the edge side. Besides, we expose native FISCO BCOS interfaces, like JSON-based RPC interface, via Kubernetes “Service” resource as API for business. These interfaces can be further unified and extended in future work.

B. CLOUD-EDGE COLLABORATIVE PLATFORM

As shown in Fig. 1, establishing a cloud-edge collaborative platform is a fundamental step for the RBaaS to work. Before the basic steps of establishing a consortium blockchain using BaaS, the on-premises edge nodes first should join the cloud using Openyurt tools. Afterward, a network tunnel is established between the cloud side and edge sides, granting the

cloud control plane capability to control the edge nodes and assign Kubernetes resources to the edge sides. With the assistance of an appropriate CNI plugin, the pods at the edge side can communicate with each other through assigned pod IP addresses.

C. THE HIGHLY AVAILABLE ARCHITECTURE

Despite many advantages of the extension of BaaS to the edge, accompanying problems have to be considered. In the conventional BaaS systems using cloud computing technique, when a blockchain node turns abnormal, the cloud control plane can likely detect the failure and repair the pods on the node. However, since the cloud-edge network connection is unstable in RBaaS architecture, the cloud may remain unconscious when a blockchain node fails at the edge side, leaving the failure unrepaired. In addition, even the traditional BaaS system in a single cloud cannot resist a breakdown of the data center itself. Therefore, we design a new redundant architecture to realize high availability mechanism.

1) BLOCKCHAIN

In the RBaaS architecture, FISCO BCOS consortium blockchain is chosen as blockchain technology, which supports a multi-group structure that isolates blockchain nodes into different groups. Ledger data are isolated between groups without the need to create a new chain, and a blockchain node can join different groups simultaneously. This feature promotes scalability and lowers the complexity of the proposed RBaaS architecture. Every blockchain node is deployed as a Kubernetes “StatefulSet” resource with three replicas distributed to different edge zones in parallel, one of which will take part in blockchain consensus along with the other two serving as candidates. Note that the “candidate” here is not the same concept as in the Raft algorithm [24].

We design a priority component in the cloud, which keeps monitoring the status of edge nodes (especially topology information from labels) and blockchain nodes replicas, then offer priority information to the blockchain nodes replicas according to the status information. In this research, the priority is mainly related to the location of edge zones, and the more frequently visited zone can be set with higher priority by configuration.

A leader election component residing in each blockchain node replica is designed in RBaaS. The election components of one same blockchain node will communicate with each other and elect a replica as the leader to participate in the blockchain consensus using the Raft algorithm. If a network partition occurs among edge datacenters or one datacenter crashes, the Raft algorithm can guarantee only a single leader for a blockchain node at all edges zones. As illustrated in Fig. 2, once a leader is elected, it will leverage the priority information obtained from the priority component at the cloud to ensure the leader replica is deployed at the edge zone with the highest priority. Privilege to access nodes topology is not given to blockchain nodes replicas in case of security problems. The leader will try to provoke leadership

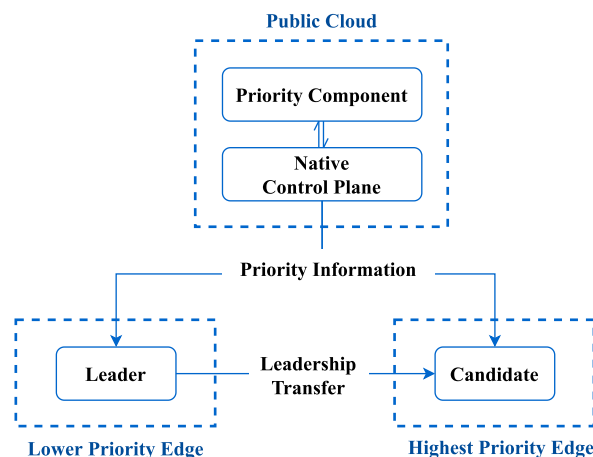


FIGURE 2. Leadership transfer process.

transfer to candidate blockchain node replica with higher and highest priority, if any. The leaders of the blockchain nodes communicate with each other through Kubernetes “Service” resources, which function as a reverse proxy. Therefore, the visiting traffic can be routed to the proper blockchain backend. Besides, the leadership information is stored in a multi-master database described below.

2) DATA REDUNDANCY

To realize the high availability of the blockchain, the redundancy of blockchain data is necessary. We use MariaDB Galera as a multi-master database to store the blockchain world state and ledger data, also deployed as Kubernetes StatefulSet with three replicas distributed in different regions. The database also supports high availability and recovers automatically so long as the remaining database nodes have a quorum. Note that this database is not going to have very high performance when a noticeable delay between database nodes exists. We will discuss it in Section IV-C4. However, it is enough for the RBaaS model under a moderate transaction sending rate right now. We use GlusterFS as distributed storage for the MariaDB Galera database in this system. The storage in every edge zone is independent of each other without synchronization. Theoretically, other network-attached storage (NFS) is acceptable as long as it can be used as the persistent volume in Kubernetes as database storage.

The blockchain nodes connect to their database instance through Kubernetes Services with the service-topology feature, which means blockchain nodes connect to the nearest database replica by appropriate configuration. The network delay can be further reduced in this way.

3) EDGE NETWORK MANAGEMENT

In a traditional cloud computing environment, the control plane will repair network failure and modify network forwarding rules according to relevant Kubernetes resources. With OpenYurt, the cloud connect to edge servers by reusing initial cloud-edge connections [25]. However, the network at the edge side is out of control when the cloud-edge

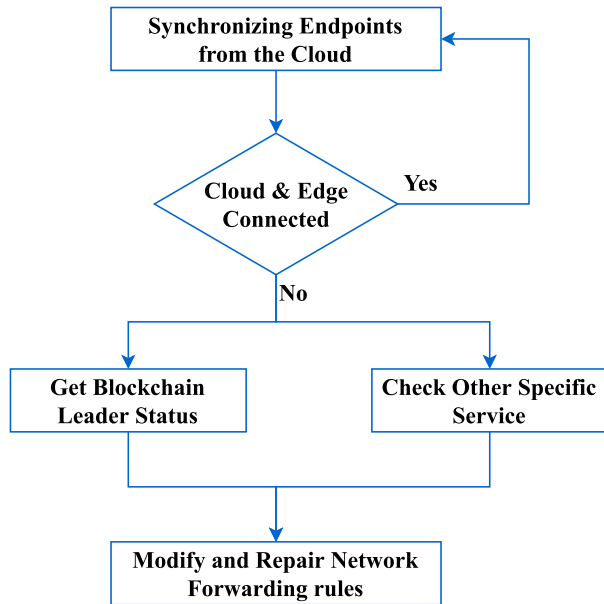


FIGURE 3. Edge network management component framework.

connection breaks down. Even when the network between cloud and an edge zone is fine, in no circumstances can the cloud side clearly distinguish between an edge server crash and network connection failure without extra assistance.

To address the network challenges we encounter in the aforementioned temporary breakdown of cloud-edge network connection, we have developed a network management component called EdgeChecker, deployed in every edge node as Kubernetes “DaemonSet” resource to help cope with network problems when an edge node crashes or pods in an edge node malfunction.

As Fig. 3 demonstrates, the EdgeChecker component keeps monitoring the connection between the cloud control plane and the located edge node and synchronizing service endpoint information of Kubernetes. When the cloud and edge node connection is interrupted, the EdgeChecker component checks the status of blockchain leaders and the health of backends of other specific services set in the configuration, for example, database service. Suppose the EdgeChecker component finds a service backend is problematic. In that case, the network of the specific edge node will be repaired by forwarding relevant traffic to the replica of the service backend, if any. A simplified process of EdgeChecker is shown in Algorithm 1.

4) FURTHER DISCUSSION

From a financial perspective, the proposed RBaaS model can be cost-saving for enterprises users when used for on-premises edge. Enterprises can reuse their original on-premises servers as edge servers of the RBaaS model, and cloud vendors can sell BaaS service without the necessity of maintenance of on-premises edge servers.

As for security issues, the proposed RBaaS model is a relatively secure platform. Blockchain nodes in the same edge

Algorithm 1 Process of Edge Network Management

```

func synchronize()
define endpoints
loop
  synchronize endpoints from the cloud periodically
  if successful then
    continue
  end if
  try access Kubernetes Apiserver /healthz
  if failed or unhealthy then
    goroutine EDGECHECK(endpoints)
    repeat
      try access Apiserver /healthz
    until successful and healthy
    stop goroutine EDGECHECK
  end if
end loop

func edgecheck(endpoints)
for endpoint in endpoints do
  check endpoint health
  if not healthy then
    modify networking rules (iptables)
  end if
end for
  try access database
  if successful then
    get blockchain leadership from the database
    check network forwarding rules
    if endpoint is outdated then
      update network forwarding rules (iptables)
    end if
  end if

```

zone communicate with each other inside local area networks and edge servers in difficult edge zones are supposed to communicate through dedicated networks as illustrated in SectionII-B, which means traffic between edge servers will not be exposed to the world wide web users and malicious attacks from the world wide web are difficult to hijack the network traffic between blockchain nodes. It further promotes the security of the proposed RBaaS model.

To enhance availability, the proposed model distributes three replicas of each blockchain node to different edge data centers, which is unnecessary if not concerning about single data center collapse. Therefore, the edge zones in the RBaaS system can be just different parts of a single data center, for example, racks.

In addition, MariaDB Galera has been used in the proposed model, which supports virtual synchronous replication. Though usually faster than traditional synchronous database replication, the performance of virtually synchronous replication is still influenced noticeably compared with asynchronous replication. Since there is only one leader among

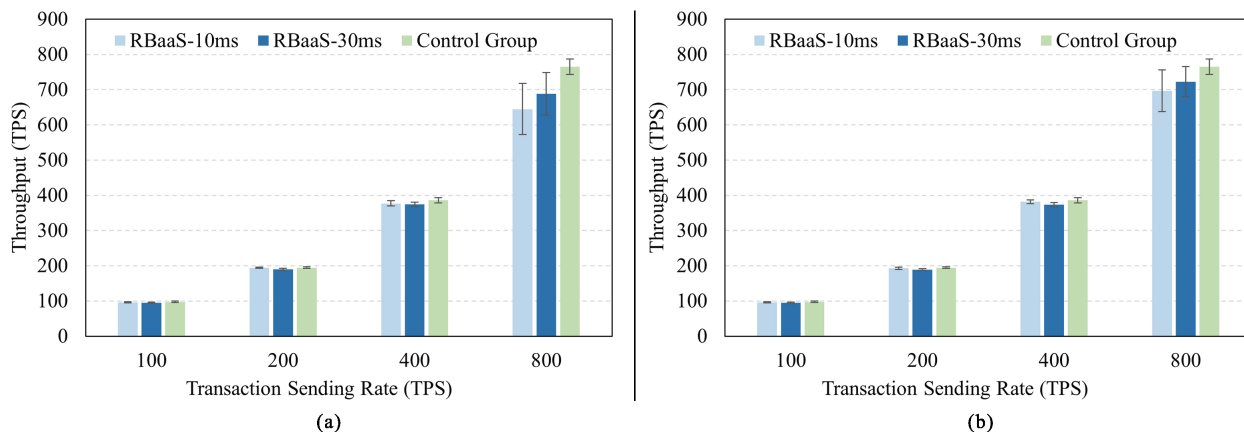


FIGURE 4. Throughput under different transaction sending rate: (a) using glusterfs storage; (b) using local host storage.

TABLE 1. Leader pods topology.

No.	Scheme	Zone-x	Zone-y	Zone-z	One-way Delay
1	RBaaS	2	2	0	10 ms
2	RBaaS	2	2	0	30 ms
3	Reference	4	0	0	<0.1 ms

three replicas of a blockchain node, theoretically, it is possible for the database to use asynchronous replication. However, extra modifications have to be made to enable automatic recovery and guarantee data consistency.

V. PERFORMANCE EVALUATION

The key features of the proposed BaaS system are cloud-edge collaboration and high availability. Therefore, in this section, we have made a performance evaluation of a simplified RBaaS system in Aliyun ECS Servers. The blockchain management component in the cloud is simplified since it will not influence performance once blockchain nodes have already been deployed at the edge. The purpose of this test is to evaluate the performance degradation of this system, inevitably caused by containerization and data synchronization between edge zones.

A. EXPERIMENT SETUP

An Aliyun ECS server (4vCPU 8GiB) as Kubernetes master node and sixteen ECS servers as worker nodes (2vCPU 4GiB) are utilized to run the tests, among which fifteen are used as OpenYurt edge nodes (OpenYurt version v0.5.0). These servers all run CentOS 7.9, with CPU type of Intel(R) Xeon(R) Platinum 8369HC CPU @ 3.30GHz, 50 GiB disk of ESSD PL1 type, and 100Mbps peak bandwidth. These servers are located in the same zone with a relatively low latency of less than 0.1ms between each other on average. Blockchain nodes and the database will be deployed in different Kubernetes nodes to ensure no mutual interference.

Among fifteen edge nodes, we use the Linux traffic control tool to divide them into three edge zones, among which the

same delays are added for a single test. Different delays among three edge zones have been used in different tests and transaction sending rates (transactions per second, i.e., TPS), and each test runs ten times. The evaluation result uses the average value of throughput.

With a moderate transaction sending rate in this test, we use just one shared database for all blockchain nodes just for simplicity, and FISCO BCOS uses database sharding for different blockchain nodes. Different databases should be used for different blockchain nodes in production environments. Metric Server, Prometheus, and Grafana are used to monitor the utilization of different resources, whose main components are deployed at the cloud side. FISCO BCOS v2.7.2 is adopted as the blockchain runtime. A concurrent payment transfer contract is used for blockchain workload, with ten times the transaction sending rate of blockchain accounts created to reduce concurrent exclusion.

To simplify, we use a blockchain with just four nodes as a group and manually give those blockchain nodes replicas regional information to simulate different scenarios. The topology of edge zones is the same as Fig. 1. For any blockchain node, there are three replicas, one leader replica and two candidate replicas, distributed in different edge zones. The number of leaders of those four blockchain nodes in different edge zones is shown in Table 1. Group No.1 and Group No.2 simulate the scenario when some leader pods fail in Zone-x and leadership is transferred to pods in Zone-y.

Group No.1 and No.2 use the RBaaS platform with the same storage type per edge zone as the storage of MariaDB Galera. Considering the potential impact of database storage, we use two different storage types. One is the GlusterFS distributed volume without replicas, serving as an example of network-attached storage, and the other is the local disk. Since the local disk is of the same type, using a local disk can eliminate possible influences of the network between databases and NFS disks. Group No.3 is set as a reference group deployed in a conventional way, not containerized, with a local MYSQL database. With no need for redundancy, we just put them in a single zone without adding extra delays.

B. RESULT EVALUATION

Fig. 4(a) and Fig. 4(b) illustrate the throughput of blockchain parallel transfer using GlusterFS storage and local disk, respectively. We will take Fig. 4(a) as an example and illustrate it in the following part. As demonstrated in Fig. 4(a), RBaaS obtains a roughly equal throughput compared with the reference group under transaction sending rate of 400 transactions per second. However, when the transaction sending rate reaches 800 TPS, the RBaaS throughput is influenced much more significantly than the reference group. The throughput of RBaaS under 10 ms and 30 ms delays gets 645 and 688 TPS, respectively. As we can see, all three groups show a performance deterioration when the transaction sending rate reaches 800 TPS, which is partly caused by rising CPU saturation (node-load1-per-cpu) of database nodes. During experiments, the average CPU saturation of the reference group reaches about 120%, whereas that of RBaaS gets to about 150%. In addition, we find that the local receiving queue and sending queue of the write-set replication of MariaDB Galera database is greater than zero, which may result in replication throttling.

By using a local disk, the performance of RBaaS can get some promotion when the transaction sending rate is high, but not very significantly. Fig. 4(b) shows that the throughput with a transaction sending rate of 800 TPS is a little better than using specific GlusterFS volume.

VI. CONCLUSION AND FUTURE WORK

This paper introduces a cloud-edge collaborative BaaS paradigm (RBaaS) for a cloud-edge collaborative environment, extending blockchain deployment to the on-premises edge or private cloud with management capability from the public cloud. By integrating Kubernetes with Openyurt, redundant blockchain nodes, leader election, and edge network self-healing implementation, the RBaaS platform can function in an edge autonomy scenario with cross-data-center high availability and simplification of management. By localizing blockchain storage, RBaaS is capable of promoting privacy and trustability. The performance results indicate that the proposed RBaaS has an acceptable throughput under moderate transaction sending rates.

In the future, we plan to explore the possibility of extending the capability of the cloud control plane to the edge so that an organization can join an established consortium blockchain when the cloud-edge connection is interrupted, and the BaaS system can get better decentralization.

REFERENCES

- [1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Bus. Rev.*, Oct. 2008. [Online]. Available: <https://www.debr.io/article/21260-bitcoin-a-peer-to-peer-electronic-cash-system>
- [2] M. M. H. Onik and M. H. Miraz, "Performance analytical comparison of blockchain-as-a-service (BaaS) platforms," in *Proc. Int. Conf. Emerg. Technol. Comput.* Cham, Switzerland: Springer, 2019, pp. 3–18.
- [3] Q. Lu, X. Xu, Y. Liu, I. Weber, L. Zhu, and W. Zhang, "uBaaS: A unified blockchain as a service platform," *Future Gener. Comput. Syst.*, vol. 101, pp. 564–575, Dec. 2019.
- [4] J. Song, P. Zhang, M. Alkubati, Y. Bao, and G. Yu, "Research advances on blockchain-as-a-service: Architectures, applications and challenges," *Digit. Commun. Netw.*, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2352864821000092>
- [5] M. Castro and B. Liskov, "Practical Byzantine fault tolerance and proactive recovery," *ACM Trans. Comput. Syst.*, vol. 20, no. 4, pp. 398–461, 2002.
- [6] M. Hosseini Shirvani, A. M. Rahmani, and A. Sahafi, "An iterative mathematical decision model for cloud migration: A cost and security risk approach," *Softw., Pract. Exp.*, vol. 48, no. 3, pp. 449–485, Mar. 2018.
- [7] M. Shuaib, A. Samad, S. Alam, and S. T. Siddiqui, "Why adopting cloud is still a challenge?—A review on issues and challenges for cloud migration in organizations," in *Ambient Communications and Computer Systems*. Singapore: Springer, 2019, pp. 387–399.
- [8] D. S. Linthicum, "Cloud-native applications and cloud migration: The good, the bad, and the points between," *IEEE Cloud Comput.*, vol. 4, no. 5, pp. 12–14, Sep. 2017.
- [9] R. Yang, F. R. Yu, P. Si, Z. Yang, and Y. Zhang, "Integrated blockchain and edge computing systems: A survey, some research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1508–1532, 2nd Quart., 2019.
- [10] P. K. Sharma, M.-Y. Chen, and J. H. Park, "A software defined fog node based distributed blockchain cloud architecture for IoT," *IEEE Access*, vol. 6, pp. 115–124, 2017.
- [11] S. Guo, X. Hu, S. Guo, X. Qiu, and F. Qi, "Blockchain meets edge computing: A distributed and trusted authentication system," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 1972–1983, Mar. 2020.
- [12] K. Gai, Y. Wu, L. Zhu, L. Xu, and Y. Zhang, "Permissioned blockchain and edge computing empowered privacy-preserving smart grid networks," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7992–8004, Oct. 2019.
- [13] A. Stanciu, "Blockchain based distributed control system for edge computing," in *Proc. 21st Int. Conf. Control Syst. Comput. Sci. (CSCS)*, May 2017, pp. 667–671.
- [14] L. Yuan, Q. He, S. Tan, B. Li, J. Yu, F. Chen, H. Jin, and Y. Yang, "CoopEdge: A decentralized blockchain-based platform for cooperative edge computing," in *Proc. Web Conf.*, Apr. 2021, pp. 2245–2257.
- [15] J. Kang, R. Yu, X. Huang, M. Wu, S. Maharjan, S. Xie, and Y. Zhang, "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4660–4670, Jun. 2019.
- [16] M. Samaniego, U. Jamsrandorj, and R. Deters, "Blockchain as a service for IoT," in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber, Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData)*, Dec. 2016, pp. 433–436.
- [17] S. Pešić, M. Radovanović, M. Ivanović, M. Tošić, O. Iković, and D. Bošković, "Hyperledger fabric blockchain as a service for the IoT: Proof of concept," in *Proc. Int. Conf. Model Data Eng.* Cham, Switzerland: Springer, 2019, pp. 172–183.
- [18] A. Bose, G. S. Aujla, M. Singh, N. Kumar, and H. Cao, "Blockchain as a service for software defined networks: A denial of service attack perspective," in *Proc. IEEE Int. Conf. Dependable, Auton. Secure Comput., Int. Conf. Pervasive Intell. Comput., Int. Conf. Cloud Big Data Comput., Int. Conf. Cyber Sci. Technol. Congr. (DASC/PiCom/CBDCom/CyberSciTech)*, Aug. 2019, pp. 901–906.
- [19] G. S. Aujla, M. Singh, A. Bose, N. Kumar, G. Han, and R. Buyya, "BlockSDN: Blockchain-as-a-service for software defined networking in smart city applications," *IEEE Netw.*, vol. 34, no. 2, pp. 83–91, Mar. 2020.
- [20] Z. Wan, M. Cai, J. Yang, and X. Lin, "A novel blockchain as a service paradigm," in *Proc. Int. Conf. Blockchain*. Cham, Switzerland: Springer, 2018, pp. 267–273.
- [21] Y. Chen, J. Gu, S. Chen, S. Huang, and X. S. Wang, "A full-spectrum blockchain-as-a-service for business collaboration," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Jul. 2019, pp. 219–223.
- [22] W. Zheng, Z. Zheng, X. Chen, K. Dai, P. Li, and R. Chen, "Nut-BaaS: A blockchain-as-a-service platform," *IEEE Access*, vol. 7, pp. 134422–134433, 2019.
- [23] H. Chen and L.-J. Zhang, "FBaaS: Functional blockchain as a service," in *Proc. Int. Conf. Blockchain*. Cham, Switzerland: Springer, 2018, pp. 243–250.
- [24] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in *Proc. USENIX Annu. Tech. Conf. (USENIX ATC)*, 2014, pp. 305–319.
- [25] H. Linbo. (2021). *Detailed Explanation of Yurt-Tunnel| Resolving the O&M Monitoring Challenges of Kubernetes in Cloud-Edge Collaboration*. [Online]. Available: https://www.alibabacloud.com/blog/detailed-explanation-of-yurt-tunnel-%7C-resolving-the-o%26m-monitoring-challenges-of-kubernetes-in-cloud-edge-collaboration_598160