An Automated Customization and Performance Profiling Framework for Permissioned Blockchains in a Virtualized Environment

Zeshun Shi*, Huan Zhou*, Jayachander Surbiryala†, Yang Hu*, Cees de Laat*, and Zhiming Zhao*
* Informatics Institute, Faculty of Science, University of Amsterdam, Amsterdam, Netherlands
{z.shi2, h.zhou, y.hu, delaat, z.zhao}@uva.nl
† Department of Electrical Engineering and Computer Science, University of Stavanger, Stavanger, Norway
{jayachander.surbiryala}@uis.no

***Abstract*—The permissioned blockchains have demonstrated their potential to provide trustworthy and security services in various industrial scenarios, especially in the Cloud-based virtualized environments. To customize the configuration of a blockchain application, an operator needs the performance characteristics of a blockchain network in different Cloud environments. However, manually profiling the performance characteristics of a blockchain network is very time-consuming. Therefore, in this paper, we propose a BlockchaIn-infRAstructure CustomIzation and Auto-profiLing (BIRACIAL) framework to automate the whole process of blockchain deployment and performance profiling. Based on the profile and performance requirements of a blockchain application, the framework aims to plan the virtual infrastructure for permissioned blockchain, to automate the provision of the required infrastructure, to deploy the customized permissioned blockchain, and to enable continuous monitoring of blockchain performance. Our evaluation results show that the proposed framework can achieve automated deployment of different permissioned blockchain networks under certain overheads. The performance profiling results can be used to compare and select the appropriate blockchain platforms and consensus algorithms.**

***Index Terms*—Permissioned Blockchain, Virtual Infrastructure, Performance Profiling**

I. Introduction

The blockchain technologies have demonstrated their great potential to provide trustworthy and security services in various industrial scenarios such as Internet of Things (IoT) [1], intellectual property management [2], Service Level Agreement(SLA) enforcement [3] [4], and stock exchange trading [5]. It records transactions among participants as identical copies through a decentralized ledger and generates new blocks based on a specific consensus algorithm. In general, blockchains can be permissionless or permissioned. Platforms like Bitcoin[1] and Ethereum[2] are permissionless blockchains, which means anyone can choose to join the network. However, for big enterprises that need their own blockchain infrastructure, this is highly undesirable. The permissioned blockchains, which are authorized to a small set of pre-selected trusted peers, have exhibited better performance compared to permissionless blockchains. Thus, they are more suitable for enterprise applications.

There have been several permissioned blockchain platforms developed, most notable examples include Hyperledger Fabric[3], Hyperledger Sawtooth[4], Hyperledger Iroha[5], R3 Corda[6], Quorum[7], and MultiChain[8]. Moreover, permissioned blockchain platforms often support pluggable consensus algorithms like Practical Byzantine Fault Tolerance (PBFT), Proof of Elapsed Time (PoET), and Raft. There are also different new consensus algorithms and their variants are being developed. It thus becomes an urgent need for developers to effectively select a suitable blockchain platform, and customize its consensus mechanisms.

The remarkable performance that permissioned blockchain platforms can achieve motivates lots of industrial application developers to choose permissioned blockchain as the basis. However, it is very time-consuming to manually profile the performance characteristics of permissioned blockchains. The large collection of platforms and consensus mechanisms are very difficult to exhaust different combinations. Moreover, there are lack of baselines for comparisons, due to those technologies only emerged in recent years. For example, some studies have tested the performance characteristics of blockchain platforms such as Hyperledger Fabric [6] and Hyperledger Sawtooth [7]. But as blockchain platforms and algorithms continue to be updated, customers and companies often need to re-test the performance when they need to choose their blockchain platforms, consensus algorithms, as well as the configuration parameters to meet the requirements of their applications. In addition, due to the unchangeable and immutable nature of the blockchain, once it is deployed, it cannot be changed and tuned anymore, which also makes it difficult to perform dynamic performance analysis and evaluation on the blockchain.

[1] https://bitcoin.org/en/
[2] https://www.ethereum.org/
[3] https://www.hyperledger.org/projects/fabric
[4] https://www.hyperledger.org/projects/sawtooth
[5] https://www.hyperledger.org/projects/iroha
[6] https://www.r3.com/platform/
[7] https://www.goquorum.com/
[8] https://www.multichain.com

Currently, major Cloud providers such as Amazon AWS[9] and Microsoft Azure[10] provide Blockchain-as-a-Service (BaaS) to meet the urgent need of permissioned blockchains. However, most of these services are vendor lock-in [8], which means that a blockchain network is by default only deployed within the data centers of that provider. In the real industrial use cases, companies often need different kinds of Cloud resources to deploy decentralized and federated blockchain networks.

Therefore, there is an urgent industrial need for an automated customization and performance profiling framework, which could not only dynamically manage the planning, provisioning, deployment of the underlying Cloud infrastructure from the perspective of programmability and controllability, but also achieve continuously performance evaluation of various permissioned blockchain platforms and consensus algorithms. This paper aims at improving the existing development and the execution of permissioned blockchains by introducing a novel framework called BlockchaIn-infRAstructure CustomIzation and Auto-profiLing (BIRACIAL). The rest of the paper is organized as follows: Section II introduces the background knowledge of the blockchain and Cloud. Section III presents the framework architecture, characteristics, and implementation plan of our BIRACIAL framework. Section IV demonstrates the validation of the BIRACIAL's functional characteristics. Section V describes the related work about the blockchain performance analysis and the selection of blockchains. Finally, in Section VI we conclude this paper with the discussion of our contributions.

II. Background: Blockchain & Cloud

The popularity of blockchain can be traced back to the first generation of blockchain named bitcoin. Bitcoin is a digital cryptocurrency that has been developed with the adoption of existing technologies such as decentralized storage, peer-to-peer transmission, consensus mechanism, and encryption algorithms. Transactions are verified by network nodes through the proof of work (PoW) consensus algorithm and recorded in a permissionless distributed ledger. New generation blockchain platforms like Ethereum support smart contract, which is the self-executing code for business logic that runs on a blockchain.

Permissionless blockchains allow any user to participate in the blockchain network. However, for big enterprises that need their own blockchain infrastructure, this is highly undesirable. Thus, a more efficient permissioned blockchain that is authorized to a small set of trusted peers may be more suitable for enterprise applications. While there is growing consensus on the potential of linking permissioned blockchains to industries, a key question remains open: Where should the permissioned blockchain be hosted? Today, more and more companies are deploying their applications on virtual infrastructure such as Cloud in order to reduce costs and make it easy to use applications from any time and anywhere. In this context, Cloud-based virtualized environments provide a good opportunity to deploy the blockchain applications, especially for further promotion and development.

However, it is not easy to deploy a permissioned blockchain in a virtualized environment. Although some Cloud providers have provided basic deployment solutions to promote and support the use of BaaS, most of these solutions are vendor lock-in, which means that permissioned blockchain cannot be deployed across Cloud platforms and different data centers. In addition, the permissioned blockchain is totally unchangeable and immutable. This means once the deployment is completed, all the configuration settings cannot be modified anymore, which brings many difficulties to do the performance test of the blockchain. Industries and blockchain users need to test different blockchain platforms, blockchain parameters, and consensus algorithms, and choose the strategy that best suits their needs.

III. BIRACIAL: BlockchaIn-infRAstructure CustomIzation and Auto-profiLing framework

In this section, we will present the architecture, characteristics, and implementation of BIRACIAL framework.

A. BIRACIAL Framework Architecture

In this subsection, we present the BIRACIAL framework to enable collaborative customization, automated deployment, and performance profiling of permissioned blockchains in a virtualized environment. The architecture of the BIRACIAL framework is shown in Fig. 1. In general, BIRACIAL consists of three phases: planning phase, co-programming phase, and auto-profiling phase.

*1) **planning phase**:* This phase is to dynamically plan the blockchain-infrastructure combined strategy to meet the needs of the blockchain application. Firstly, the blockchain manager and Cloud manager in an organization need to negotiate a decision table for all the requirements when deploying a blockchain application. More specifically, the requirements to select blockchain platforms and consensus algorithms include three aspects of indicators: 1) blockchain architecture characteristics; 2) blockchain application characteristics; 3) blockchain performance characteristics, which can be further seen in Table I.

In order to obtain an exhaustive blockchain-infrastructure requirement table, a screening and survey method similar to [9] can be used. In this process, a detailed blockchain screening questionnaire for different use cases was produced and distributed to blockchain industry practitioners. This will investigate the blockchain platforms in the current market and analyze the usage of blockchain platforms and consensus algorithms for different use cases. Next, a BIRACIAL *Planning Engine* with two components is activated. More specifically, the role of the *Blockchain Planner* is to generate the decision tree of the blockchain planning so as to support two different perspective decisions about 1) the blockchain platform and consensus algorithm for implementing application scenarios

[9]https://aws.amazon.com/blockchain/

[10]https://azure.microsoft.com/en-us/solutions/blockchain/

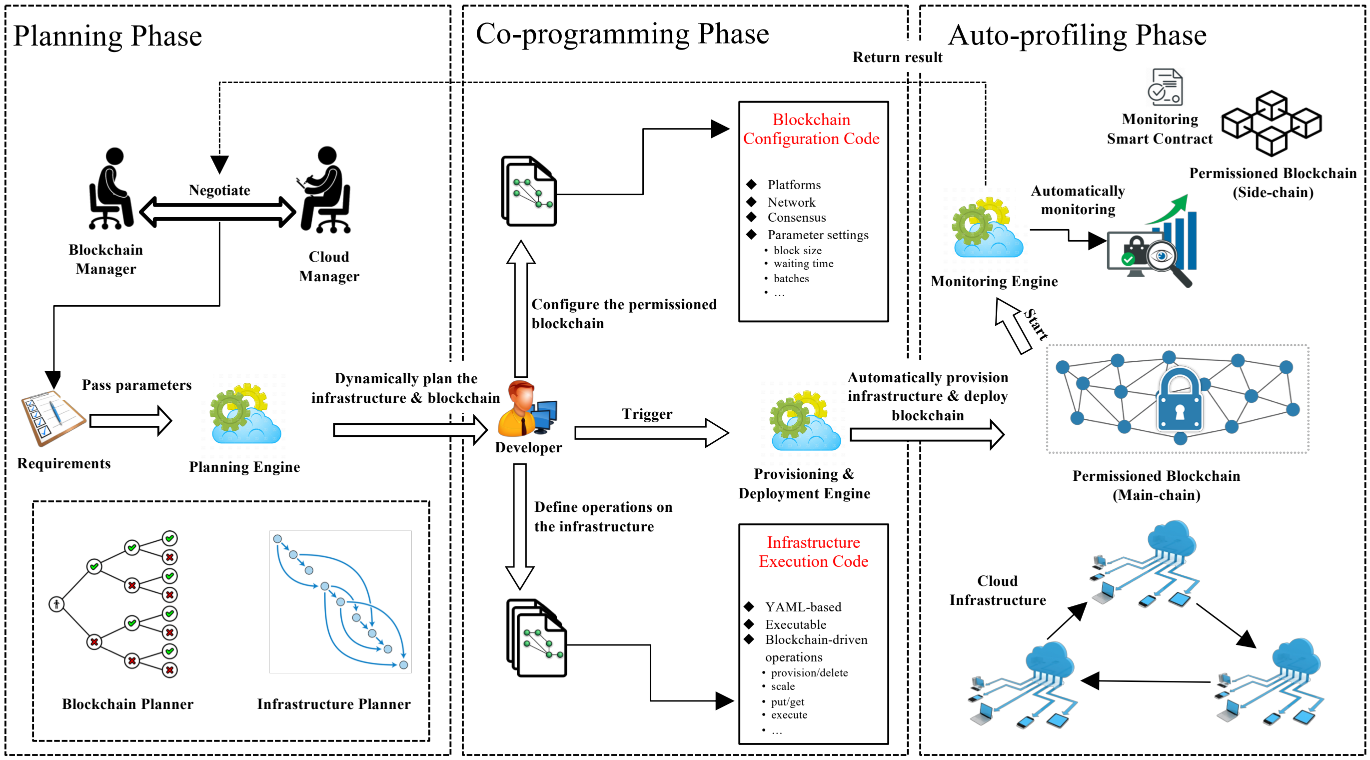

Fig. 1. The architecture of BIRACIAL framework

(main-chain); and 2) the blockchain platform and consensus algorithm for implementing monitoring service (side-chain). In contrast, the role of *Infrastructure Planner* is to meet the needs of multiple deadlines for fast infrastructure provisioning. BIRACIAL's *Planning Engine* will then pass the well-planned blockchain and infrastructure parameters to the developer. Then, the BIRACIAL framework enters the second phase.

*2) **Co-programming Phase**:* This phase is to achieve collaborative customization between the virtual infrastructure and permissioned blockchains. First, developers need to configure the *Blockchain Configuration Code* and the *Infrastructure Execution Code* according to the blockchain-infrastructure customization strategy during the planning phase. The *Blockchain Configuration Code* will contain various configuration parameters (e.g. blockchain platforms, network topology, consensus algorithms, and various customized parameters) of the blockchain that needs to be deployed. At the same time, the *Infrastructure Execution Code* contains basic description and execution information for the underlying infrastructure. When two parts of the code are successfully configured, the developer needs to start the *Provisioning Engine* and the *Deployment Engine* respectively, which are responsible for the automatic provisioning of infrastructure and the deployment of the permissioned blockchain. The infrastructure *Provisioning Engine* uses the open Cloud computing interface as its default provisioning interface and currently supports Amazon AWS[11], EGI FedCloud[12], and ExoGeni[13]. Since BIRACIAL relies on multiple Cloud providers, it offers a besteffort approach for the provision, stability, and performance of the underlying virtual infrastructure. The deployment engine can deploy any type of permissioned blockchains based on the *Blockchain Configuration Code*. It also supports deploy overlay Docker clusters such as Docker Swarm or Kubernetes.

*3) **Auto-profiling Phase**:* This phase is to achieve dynamic performance monitoring and profiling of the permissioned blockchain, in which a two-layer blockchain structure is constructed. When the permissioned blockchain (main-chain) is deployed successfully on the Cloud infrastructure, a *Monitoring Engine* will be launched at the same time to manage automated performance monitoring and profiling. In order to ensure the security and non-tamperability of the monitoring data, another permissioned blockchain (side-chain) is then activated. On the side-chain, a monitoring smart contract is used to achieve performance monitoring and data storage. In order to meet the needs of continuously monitoring, it is necessary to select a blockchain platform and a consensus algorithm with high throughput. Finally, the results of the auto-

[11]https://aws.amazon.com/
[12]https://www.egi.eu/services/cloud-compute/
[13]http://www.exogeni.net/

TABLE I
DESCRIPTION AND CLASSIFICATION OF SOME INDICATORS OF A PERMISSIONED BLOCKCHAIN

Category	Indicators	Description	Examples
Architecture Characteristics	Consensus protocol	Consensus protocol used for testing	PBFT, PoET, Raft
	Network topology	The Network topology of the blockchain	Centralized, Decentralized
	Blockchain platform	The platform where the testing is performed	Hyperledger Fabric, Sawtooth
	Smart contract	Indictes whether to use smart contract	Required, N/A
Application Characteristics	Security	Describe the security requirements of the application	Low, Medium, High
	Scalability	Describe the scalability requirements of the application	Low, Medium, High
	Reliability	Describe the reliability requirements of the application	Low, Medium, High
	Key management	Describe whether the key management is required during the testing	Required, N/A
	Data model	Describe the data model used in testing, can be either fixed or customised	Fixed, Customized
	Programming language	Describe the programming language used in the testing	Python, Java
Performance Characteristics	Geographic distribution of nodes	Depict the location and dispersion of all nodes	USA, UK
	Infrastructure types of all peers	Describe the machinery characters such as CPU core number, processing speed and memory	EC2 small, medium, large
	Number of nodes involved in the test transactions	Transactions can be transmitted either to all nodes or within a subset of nodes	20, 50
	Software component dependencies	Describe the supplementary components for special requirements of a test or platform	Required, N/A
	Transaction patterns	The pattern that is used to generate and check the validity of the transaction	Dense, Sparse, N/A

profiling test are returned to the Cloud/blockchain manager in real-time for them to perform more testing. These performance profiling results will help them make better decisions on how to choose blockchain platforms, consensus algorithms, and more setting parameters.

B. BIRACIAL Characteristics

Based on this framework, we want to provide the following features and benefits to use BIRACIAL deploy a permissioned blockchain in a virtualized environment. Basically, BIRACIAL demonstrates its impact on the following perspectives:

- *Programmable:* The permissioned blockchain, as well as the underlying Cloud virtual infrastructure, can be customized at the BIRACIAL co-programming phase to fit the blockchain application requirements. Furthermore, the network topology, pluggable consensus protocols, and other configuration settings within a permissioned blockchain can also be defined in advance. In total, a programmable blockchain is achieved in our framework.
- *Controllable:* BIRACIAL affords high-level controllability of Cloud infrastructure, including real-time monitoring, failure recovery, and auto-scaling. Besides, the corresponding permissioned blockchain platform and application can be easily configured and controlled at the co-programmable phase.
- *Interoperable:* BIRACIAL is able to manage virtual infrastructure among federated Clouds. Hence, this is not a vendor lock-in solution, which means you can deploy a permissioned blockchain network across the federated Clouds. This is well suited to the decentralization characteristic of blockchain. In addition, BIRACIAL also enables collaboration between two permissioned blockchains. The use of non-tamperable and decentralized sidechains avoids the risk of single points of failure and data manipulation in the data center.
- *Reusable:* We propose *infrastructure code* and *blockchain code* to describe dynamic operations on the infrastructure and permissioned blockchain. They are able to define and customize the whole process of infrastructure provisioning, blockchain deployment, and performance profiling. These codes are totally reusable and the entire process is reproducible, which means you can test the blockchain multiple times as long as you have enough budget for Cloud services.

C. BIRACIAL Implementation

For BIRACIAL development and implementation, we followed the DevOps practices and modular development. BIRACIAL works in the following steps, which is shown in Fig. 2.

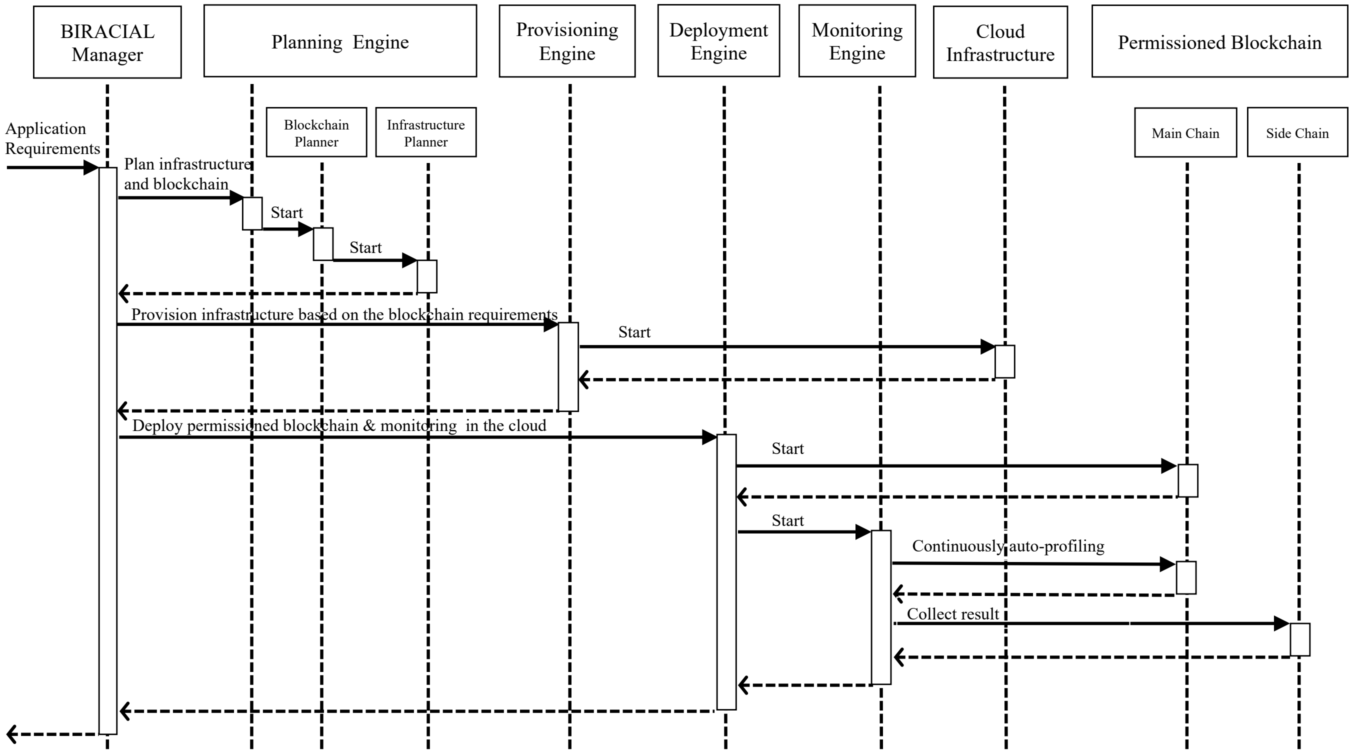

Fig. 2. Sequential diagram of the implementation steps for BIRACIAL framework

BIRACIAL *Manager* is responsible for the whole lifecycle of planning of the blockchain-infrastructure strategy, provisioning the underlying infrastructure, deploying the customized permissioned blockchains, and auto-profiling the performance characteristics. When the external application requirements are passed to BIRACIAL, BIRACIAL *Manager* starts the *Planning Engine*. Then the *Blockchain Planner* calculates the best main-chain and side-chain solutions based on a decision tree-based algorithm, and the *Infrastructure Planner* returns the best infrastructure strategy to support permissioned blockchain deployment. After that, BIRACIAL *Manager* calls the *Provisioning Engine* to provide the underlying infrastructure based on the infrastructure plan. When all the infrastructure is provided as planned, BIRACIAL *Manager* will call the *Deployment Engine* to deploy the main-chain, either customize each node to be deployed within a Virtual Machine (VM) or within a docker container. When the main-chain starts successfully, the *Deployment Engine* then calls the *Monitoring Engine* to initiate continuous performance monitoring of the main-chain. The *Monitoring Engine* also enables another permissioned blockchain (side-chain) and smart contract to prevent data from being altered or manipulated. The final monitoring results then returns to the BIRACIAL *Manager* and output the BIRACIAL framework.

IV. BIRACIAL Functional Validation

According to the design requirements, BIRACIAL should achieve at least two key features: 1) automate the deployment of the permissioned blockchains on Cloud infrastructures; and 2) automate the monitoring and analysis of the performance characteristics of the deployed blockchain. To evaluate these functionalities of BIRACIAL, we thus designed two experiments which will be discussed in this section. We leveraged Hyperledger Sawtooth as the permissioned blockchain platform and used ExoGENI and Amazon EC2 as our Cloud providers.

A. Automated Deployment Functional Validation

We tested the execution overhead required by BIRACIAL for infrastructure provisioning and blockchain deployment. BIRACIAL can automate the deployment of Hyperledger Sawtooth blockchain (less than 400 seconds) with different nodes in both ExoGENI and EC2, which can be seen in Fig. 3. More specifically, in the ExoGENI Cloud, although some fluctuations exist (possibly due to the dynamic nature of the Cloud), its overall execution overhead increases as the number of blockchain nodes increases. One of the reasons is that when the number of nodes increases, both the provisioning time of infrastructure and the deployment time of the blockchain will increase, in particular in the private Cloud environment like ExoGENI. In contrast, the execution overhead of Amazon EC2 did not increase obviously and remained at around 150 seconds. We believe that one possible reason is that as a mature commercial Cloud provider, Amazon EC2 offers better quality Infrastructure-as-a-Service than private Clouds. In this way, when the number of VM nodes increases, public Cloud can still maintain an efficient provisioning service.

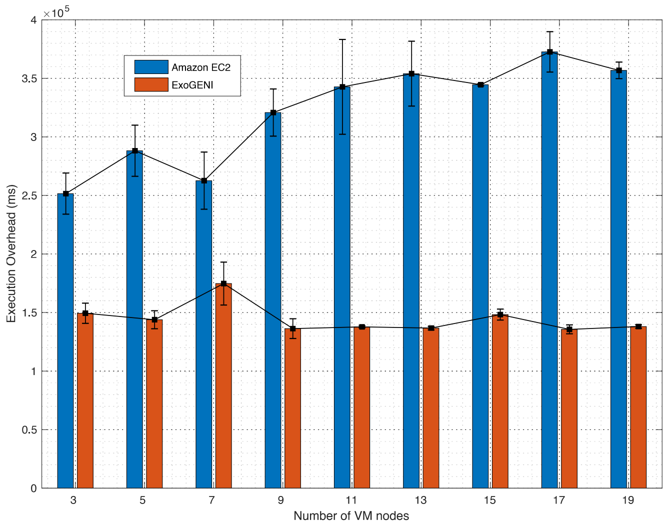

Fig. 3. Variation in the overhead of deploying permissioned blockchains and provisioning infrastructure by different Cloud providers

B. Performance Profiling Functional Validation

In this experiment, we compared the blockchain performance profiling results with different transaction patterns and consensus algorithms to evaluate the performance profiling functionality of BIRACIAL. As can be seen from the Fig. 4, when the *Transaction Input Rate* is increased, the throughput of the permissioned blockchain first increases and then reaches the performance bottleneck and gradually stabilize. In the comparison of the three consensus algorithms, PBFT is slightly higher than Raft, and both of them are significantly better than PoET when dealing with a huge workload. From this experiment, we can conclude that BIRACIAL can achieve monitoring and performance profiling functionality of the permissioned blockchain. The monitoring results can be used as a reference in choosing the appropriate blockchain platforms and consensus algorithms.

V. Related Work

Regarding the analysis of performance characteristics of the blockchain, there already some studies focus on the evaluation of popular blockchain platforms like Hyperledger Fabric [6] [10], Hyperledger Sawtooth [7], and Ethereum [11].

With respect to how to choose the blockchain platform and consensus algorithm, Pandey et al. [12] proposed a blockchain planning simulator called BlockSIM which can be used to compare and filter different blockchain parameters and consensus algorithms. However, their simulator is relatively simple, and it is doubtful whether their simulation results are credible in practice. Kondo et al. [9] presented a blockchain screening method based on user requirements. They use a screening form to select suitable blockchain software products that demonstrate good coverage of requirements for a use

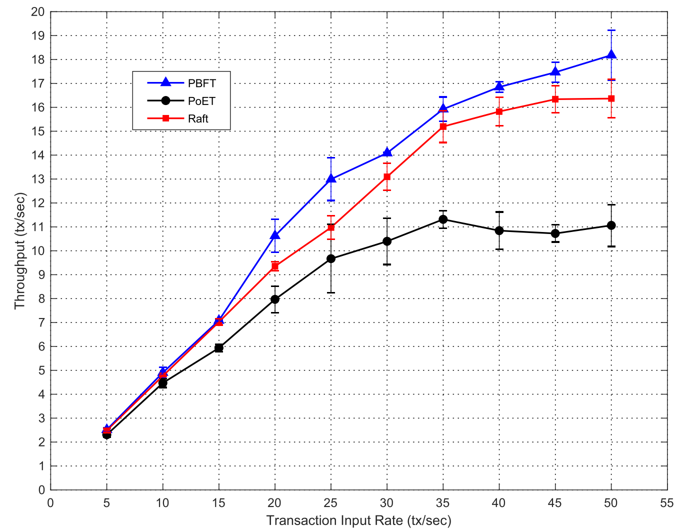

Fig. 4. Variation in performance profiling of different consensus algorithms and transaction patterns.

case in terms of network topology, settlement finality, governance, privacy/confidentiality, and application architecture. Diarra [13] also proposed a scheme for selecting a consensus algorithm based on different use cases. However, their idea is still in the prototype stage.

VI. Conclusion

In this paper, we proposed a BlockchaIn-infRAstructure CustomIzation and Auto-profiLing (BIRACIAL) framework to help application developers to choose the appropriate blockchain platforms and consensus algorithms, and to customize their deployment. By considering requirements from both the blockchain and the virtual infrastructure, our framework optimizes the customization of blockchain deployment and provides features for continuously monitoring and performance profiling. The experiment demonstrated the features of our framework. Currently, our BIRACIAL framework supports two permissioned blockchain platforms (Hyperledger Fabric and Hyperledger Sawtooth) and three Cloud providers (EC2, EGI, and ExoGENI). It is able to deploy blockchains across different providers. We aim to make it as an open framework and attract community effort for supporting more blockchain platforms and consensus protocols in the future.

Acknowledgment

This research is funded by the EU Horizon 2020 research and innovation program under grant agreements 825134 (ARTICONF project), 862409 (BlueCloud project) and 824068 (ENVRIfair project).

References

[1] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for iot security and privacy: The case study of a smart home," in *2017 IEEE international conference on pervasive computing and communications workshops (PerCom workshops)*, pp. 618–623, IEEE, 2017.

[2] W.-T. Tsai, L. Feng, H. Zhang, Y. You, L. Wang, and Y. Zhong, "Intellectual-property blockchain-based protection model for microfilms," in *2017 IEEE Symposium on Service-Oriented System Engineering (SOSE)*, pp. 174–178, IEEE, 2017.

[3] H. Zhou, X. Ouyang, Z. Ren, J. Su, C. de Laat, and Z. Zhao, "A blockchain based witness model for trustworthy cloud service level agreement enforcement," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pp. 1567–1575, April 2019.

[4] H. Zhou, X. Ouyang, J. Su, C. de Laat, and Z. Zhao, "Enforcing trustworthy cloud sla with witnesses: A game theorybased model using smart contracts," *Concurrency and Computation: Practice and Experience*, vol. 0, no. 0, p. e5511. e5511 cpe.5511.

[5] L. Lee, "New kids on the blockchain: How bitcoin's technology could reinvent the stock market," *Hastings Bus. LJ*, vol. 12, p. 81, 2015.

[6] P. Thakkar, S. Nathan, and B. Viswanathan, "Performance benchmarking and optimizing hyperledger fabric blockchain platform," in *2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pp. 264–276, IEEE, 2018.

[7] Z. Shi, H. Zhou, Y. Hu, J. Surbiryala, C. de Laat, and Z. Zhao, "Operating permissioned blockchain in clouds: A performance study of hyperledger sawtooth," in *2019 18th International Symposium on Parallel and Distributed Computing (ISPDC)*, pp. 50–57, IEEE, 2019.

[8] H. Zhou, Y. Hu, X. Ouyang, J. Su, S. Koulouzis, C. de Laat, and Z. Zhao, "Cloudsstorm: A framework for seamlessly programming and controlling virtual infrastructure functions during the devops lifecycle of cloud applications," *Software: Practice and Experience*, vol. 49, no. 10, pp. 1421–1447, 2019.

[9] Y. Kondo, T. Shimosawa, and S. Oshima, "A practical screening method for blockchain-based systems," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 1483–1490, IEEE, 2018.

[10] H. Sukhwani, J. M. Martínez, X. Chang, K. S. Trivedi, and A. Rindos, "Performance modeling of pbft consensus process for permissioned blockchain network (hyperledger fabric)," in *2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS)*, pp. 253–255, IEEE, 2017.

[11] S. Rouhani and R. Deters, "Performance analysis of ethereum transactions in private blockchain," in *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pp. 70–74, IEEE, 2017.

[12] S. Pandey, G. Ojha, and B. Shrestha, "Blocksim: A practical simulation tool for optimal network design, stability and planning.," in *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pp. 133–137, IEEE, 2019.

[13] N. Diarra, "Choosing a consensus protocol for uses cases in distributed ledger technologies," in *2019 Sixth International Conference on Software Defined Systems (SDS)*, pp. 306–309, IEEE, 2019.