Blockchain Enabled IoT Edge Computing

Pankaj Mendki
Talentica Software Pvt Ltd
Pune, India 411045
pankajm@talentica.com

ABSTRACT
Number of internet connected devices is increasing as IoT is getting more prevalent. Volume of data collected by IoT sensors is very high and requires considerable resources for data processing like analytics. Edge processing enables getting the sensor data processed closer to the source. It's not always possible and/or economical to set up a resource intense infrastructure at the edge. This paper proposes use of blockchain based decentralized application to enable IoT edge processing. With this setup, it is possible for a resource owner to join the ecosystem and to lend the compute resources as needed. IoT devices can offload some of the edge computation to the resource owner nodes as and when required. This paper explains architectural components and overall solution for implementation. Based on the usecase of video analytics at edge, experimental setup is implemented for the IoT edge solution using hyperledger sawtooth blockchain.

CCS Concepts
• **Computing methodologies → Distributed computing methodologies**

Keywords
IoT; Edge computing; Blockchain; Hyperledger sawtooth.

1. INTRODUCTION
Internet of things (IoT) technology is getting adopted rapidly. Number of IoT enabled devices is increasing and it is predicted to reach about 25 billion by 2021 [1]. These devices produce immense volume of data. Generally, device data is sent over cloud for further processing, this demands high bandwidth and high computational resources at the cloud end. Edge and fog computing are evolving as complementary processing architectures to eliminate these shortcomings. [2] [3].

Blockchain technology initiated as decentralized electronic cash system. It gradually evolved to support smart digital contract and platform for developing decentralized application [4]. This paper proposes using blockchain based architecture for developing decentralized IoT edge processing. The paper is structured as follows. Section 2 explains the basic concepts of edge computing and blockchain. Section 3 explains the proposed architecture of blockchain based IoT edge processing. It explains the problem statement, different architectural components, processing flow and set of challenges. The next section – section 4, explains the usecase scenario and setup that is used for experimental implementation. This is followed by conclusions in section 5.

ICBCT 2019, March 15–18, 2019, Honolulu, HI, USA

ACM ISBN 978-1-4503-6268-9/19/03. . . $15.00
DOI: https://doi:org/10:1145/3320154:3320166

2. BACKGROUND
2.1 IoT Edge Computing
Most of the IoT deployments are based on cloud hosted central data processing and storage architecture, i.e. sensors are sending data to cloud either directly or via gateway. This type of processing encounters data transmission cost and latency. With the emergence of technologies like the fifth generation cellular network (5G) and advances in hardware capabilities, IoT devices would be able to send data at much higher rate imposing considerable scaling challenges for the cloud based central system [5]. Edge and fog computing architectures enable using resources available close to the edge of IoT network [2]. The idea of edge level analytics is proposed in this work by Satyanarayanan [3]. Work by Mendki [6] proposed using container based applications at the edge that enables using microservice architecture. Edge computing is not the replacement for the cloud; the latter is still required for the central point of references. Edge computing helps to offload some of the resource intensive work from the cloud.

2.2 Blockchain
Blockchain technology got introduced as Bitcoin cryptocurrency [7] - a peer to peer electronic cash system. Blockchain uses peer to peer distributed transaction ledger, there is no central authority to approve the transactions, it also ensures that transaction ledger is immutable. Some of the blockchain implementation frameworks provide mechanism for deploying smart contracts [8]. Smart contract is a programming script that is executed by the blockchain network to validate or enforce certain checks on a transaction or parties involved in the transaction. Smart contract enables deploying decentralized applications (DApps) over blockchain. There has been ongoing research of utilizing the features of blockchain for IoT and edge computing. The work by Sharma et al. [9] presents providing secure distributed fog node architecture using SDN and blockchain. Another research by Stanciu [10] proposes using blockchain smart contract for developing distributed hierarchical control system. Idea of using blockchain smart contract for automation of IoT ecosystem workflows is proposed in the work by Christidis and Devetsikiotis [11].

This paper is proposing idea of creating a permissioned blockchain network where IoT edge devices can get edge processing computation executed from any public infrastructure willing to share the resources.

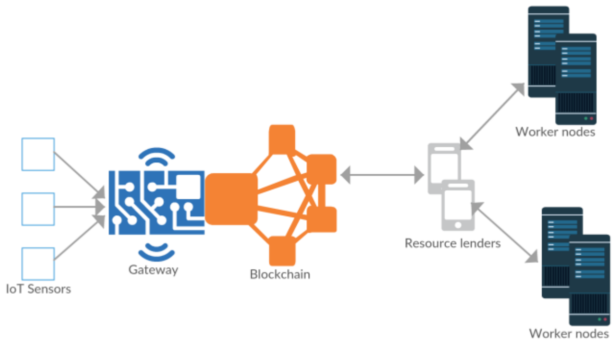

Figure 1: Architecture for Edge Computing over Blockchain

3. PROPOSED ARCHITECTURE

This section discusses about the problem statement, proposed architecture of blockchain based edge computing for IoT, overall workflow of the processing and possible challenges in the system.

3.1 Problem Statement

One of the issues with edge level processing is its horizontal scalability. Resource infrastructure deployed at the edge is not as easy to scale as compared to the cloud based. Chiang and Zhang [12] have proposed pooling idle resources that can contribute to edge or fog. The resource pooling mechanism for horizontal edge scalability can be implemented using blockchain DApps mechanism. Blockchain will ensure secured and contractual way of sharing resources between idle publicly available users' resources and edge devices. Users having idle resources can execute the edge processing work and get paid.

3.2 Architectural Components

Different components of the architecture are depicted in Figure 1. The main components of the architecture are:

3.2.1 *Blockchain network*

This is a permissioned blockchain network. Gateway and other edge devices are the members of this network. These devices act as validators and validate all the transactions submitted to the blockchain, as per the business rule. The network also keeps track of series of transactions as immutable ledger.

3.2.2 *Client or requester*

Client or requester is a device requesting for the compute resources to solve computation intensive task a.k.a job. Requester submits a request transaction to the blockchain network defining the details of the job to be executed. Typically gateway devices would run requester/client application for getting the edge level processing done. All the interaction with the blockchain is treated as transaction e.g. sending a request proposal for executing computation.

3.2.3 *Resource lender*

Any user can join as a resource lender. Resource lenders would lend out their compute resources for getting the job executed. Once the job is executed successfully on the worker node, output is submitted as transaction

3.2.4 *Worker nodes*

Resource lenders own one or more worker nodes that are responsible for job execution. Job should be executed in uniform sandboxed environment that would ensure support for different types of worker node operating systems.

3.3 Processing Flow

Figure 2 depicts the processing flow. Requester gathers data from IoT sensors and creates a batch of input data for processing a set of jobs. This job data can be stored over blockchain based decentralized storage built over edge devices. Requester submits transaction to blockchain network requesting for execution of a job or batch of jobs. This transaction contains the reference to the input data file and commands for executing the job. Once the above transaction is approved by the blockchain network, resource lender fetches the details of job and delegates the job execution to one of the worker nodes. Worker node fetches the run-time/executable files and input data required for the job execution and executes the job. Resource lender gets the processing output results from worker node and submits the results to the blockchain network as a transaction. One of the validators validates this transaction using the test data output submitted, if the transaction submitted is valid then transaction will be approved by blockchian network else it is discarded.

3.4 Challenges

3.4.1 *Validation of results*

Resource lender executes a job on worker node and submits the results to the blockchain. One of the major challenges for this solution is verification of the results without executing the job itself. Yu et al. [13] have surveyed different techniques for verifiable computation but still this is active area of research. Some of the pragmatic approaches can be implemented for result verification like every batch of jobs may contain a sample test job for which result is known. The submitted results can be verified against the known value of the test job implanted in batch. There could be multiple such test jobs and a batch may contain one or more randomly selected test jobs.

Another way is to use intermediate processing logs generated by worker node to validate as proof of job execution. Using redundant jobs would also help validating the results. None of these approaches is the perfect solution but a probabilistic measure of evaluating results correctness. The reputation and/or economic penalty-based system can enhance the authenticity of the resource supplier. Malicious resource supplier can be blacklisted and transaction request from that can be discarded.

The proposed architecture is not using any centralized entity for results verification; using such central entity violates the true blockchain based nature of the solution.

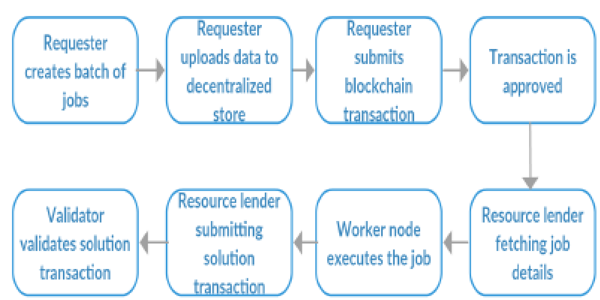

Figure 2: Processing Flow for Edge Computing over Blockchain.

3.4.2 Reward system

Authentic resource supplier should get paid. The reward amount could be in terms of stable cryptocurrency pegged to the USD. The extent of reward could be proportional to the amount of computation done by worker node. The system should ensure that when the result submitted is verified and acceptable; the resource supplier should be paid appropriately. One of the possible ways to achieve this is using hash time locked contract [14].

3.4.3 Packaging and distribution

Worker nodes should execute the job in sandboxed environment; this would ensure consistent reproducible output and system independent job processing. One of the ways to achieve this is using container based environment like docker. Container images can be stored over decentralized blockchain based storage and worker nodes can fetch those images on need basis before executing the job.

4. USECASE

Edge level video analytics usecase is selected for implementing the above solution. Surveillance cameras are getting used widely for the security reasons. The volume of video data produced by these cameras is huge. As per [15], global data generated by surveillance cameras is about 560 petabytes per day. This makes them a good candidate for deploying edge-based analytics solution. One of the possible usecases is to analyze the content of videos and storing that as searchable indexed repository for the video content [16]. A simplified version of this usecase is detecting different objects present in video frames using deep learning image classification techniques. This processing will be executed over worker nodes provided by resource lenders.

4.1 Experimental Setup

CCTV camera is connected to Raspberry Pi using camera module. Raspberry Pi acts as a client/requester that captures the video frames and constructs a batch for processing. Detecting objects in a set of captured images is a job. Hyperledger Sawtooth is used [17] for setting up the blockchain network in this setup. This framework doesn't support ARM architecture, but it provides REST based APIs [18] that can be consumed from Raspberry Pi. Because of this limitation, the current implementation does not include Raspberry Pi as a node on the blockchain network. There are four types of actors in the application; requester, validator, resource lender and worker node. Requester is a Raspberry Pi device. Ubuntu server 16.04 based virtual machines are used as validators, the current setup uses four validators. These validators constitute blockchain network. Hyperledger Sawtooth node constitutes these two primary components; REST based APIs and transaction processors. REST APIs are used to interact with blockchain whereas transaction processors are used to validate the submitted transactions as per the business rules. Resource lender and worker node are Ubuntu server 16.04 based virtual machines. IFPS is setup on Ubuntu server 16.04 virtual machine.

Requester captures the video frames and uploads a group of captured frames to IPFS decentralized storage as processing input to a job. Requester then creates a job request by submitting the job creation transaction to the blockchain. The job object that is stored on blockchain has fields like job type, job ID, job state and reference to input data on IPFS. Job state is enumerated data type, it can have values "REQUESTED", "LOCKED", "SOLVED". Initial value of job state is "REQUESTED". When the job creation transaction is added successfully to the blockchain network, resource lender gets notified about the transaction. The resource lender that is interested to solve the job should try to lock the job so that no other resource lender can pick the job. This is done by submitting a transaction that changes job state to "LOCKED". Ideally the lock state on job should be released after a finite time if resource lender fails to solve the job or intentionally trying to make the job unavailable for others. This security mechanism is not implemented in the current setup. Once the lock transaction request is successfully accepted by the blockchain network, resource lender can start processing the job by invoking job specific scripts on one of the worker nodes. This script fetches the required docker images, executes the job using the input files and uploads the results to IPFS. Resource lender submits a transaction that changes the job state to "SOLVED", this transaction also has reference to the results of the job. One of the validators will approve the result if the test job output is as expected else the transaction is rejected. On successful validation of the transaction, requester can fetch the results over IPFS.

5. CONCLUSION

Edge and fog computing have been evolving as complimentary architecture to the central cloud processing and can offload the ever-increasing scaling demands. Likewise blockchain based resource pooling solution can help edge and fog architecture to address their scaling needs. Using the proposed blockchain based solution for IoT edge can lead towards building the crowdsourced public infrastructure for IoT edge. Ongoing research in the areas like verifiable computation would strengthen the possibility of adoption of similar solutions.

6. REFERENCES

[1] Gartner identifies top 10 strategic iot technologies and trends. https://www.gartner.com/en/newsroom/press-releases/2018-11-07-gartner-identifies-top-10-strategic-iot-technologies-and-trends, 2019 (accessed February 3, 2019).

[2] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. Fog computing and its role in the internet of things. In Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, MCC '12, pages 13-16, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1519-7. URL http://doi.acm.org/10.1145/ 2342509.2342513.

[3] M. Satyanarayanan. The emergence of edge computing. Computer, 50(1):30-39, Jan 2017. ISSN 0018-9162.

[4] Vitalik Buterin. Ethereum white paper.pdf. https://cryptorating.eu/whitepapers/Ethereum/Ethereum_white_paper.pdf (Accessed on 02/03/2019).

[5] Nadeem Javaid, Arshad Sher, Hina Nasir, and Nadra Guizani. Intelligence in iot-based 5g networks: Opportunities and challenges. IEEE Communications Magazine, 56, 06 2018.

[6] P. Mendki. Docker container based analytics at iot edge video analytics usecase. In 2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU), pages 1-4, Feb 2018.

[7] Satoshi Nakamoto. bitcoin.pdf. https://bitcoin.org/ bitcoin.pdf. (Accessed on 02/03/2019).

[8] Ethereum project. https://www.ethereum.org/. (Accessed on 02/03/2019).

[9] Pradip Kumar Sharma, Mu Yen Chen, and Jong Hyuk Park. A Software Defined Fog Node Based Distributed Blockchain Cloud Architecture for IoT. IEEE Access, 6:115-124, 2018. ISSN 21693536.

[10] Alexandru Stanciu. Blockchain Based Distributed Control System for Edge Computing. Proceedings - 2017 21st International Conference on Control Systems and Computer, CSCS 2017, pages 667-671, 2017. ISSN 2379-0482.

[11] Konstantinos Christidis and Michael Devetsikiotis. Blockchains and Smart Contracts for the Internet of Things. IEEE Access, 4:2292-2303, 2016. ISSN 21693536.

[12] M. Chiang and T. Zhang. Fog and iot: An overview of research opportunities. IEEE Internet of Things Journal, 3(6):854-864, Dec 2016. ISSN 2327-4662.

[13] Xixun Yu, Zheng Yan, and Athanasios V. Vasilakos. A Survey of Verifiable Computation. Mobile Networks and Applications, 22(3):438-453, 2017. ISSN 15728153.

[14] Hash time locked contracts. https://en.bitcoin.it/wiki/ Hash Time Locked Contracts, 2019 (accessed February 1, 2019).

[15] Data generation from new video surveillance cameras was 566 petabytes per day in 2015 - storage servers. https://storageservers.wordpress.com/2016/01/22/data-generation-from-new-video-surveillance-cameras-was-566-petabytes-per-day-in-2015/. (Accessed on 02/04/2019).

[16] M. Satyanarayanan, P. Simoens, Y. Xiao, P. Pillai, Z. Chen, K. Ha, W. Hu, and B. Amos. Edge analytics in the internet of things. IEEE Pervasive Computing, 14(2):24-31, Apr 2015. ISSN 1536-1268.

[17] Hyperledger sawtooth - hyperledger. https://www.hyperledger.org/projects/sawtooth, (Accessed on 02/21/2019).

[18] Rest api - sawtooth v1.1.4 documentation. https://sawtooth.hyperledger.org/docs/core/releases/1.1/architecture/rest_api.html, (Accessed on 02/21/2019).