Complex & Intelligent Systems (2022) 8:3707–3718
https://doi.org/10.1007/s40747-021-00466-y

ORIGINAL ARTICLE

Blockchain-based green big data visualization: BGbV

Iqra Shahzad[1] · Ayesha Maqbool[2] · Tauseef Rana[1] · Alina Mirza[3] · Wazir Zada Khan[4] · Sung Won Kim[5] · Yousaf Bin Zikria[5] · Sadia Din[5]

Received: 29 March 2021 / Accepted: 3 July 2021 / Published online: 10 August 2021

Abstract

The progression of Internet of Things (IoT) has resulted in generation of huge amount of data. Effective handling and analysis of such big volumes of data proposes a crucial challenge. Existing cloud-based frameworks of Big Data visualization are rising costs for servers, equipment, and energy consumption. There is a need for a green solution targeting lesser cost and energy consumption with tamper-proof record-keeping, storage, and interactive visualization with only demanded data. We have proposed a Blockchain-based Green big data Visualization (BGbV) solution using Hyperledger Sawtooth for optimum utilization of organization resources. BGbV will support current distributed data visualization platforms and guarantee benefits like security and data availability with lesser storage costs. It helps reduce costs by utilizing small resources that are already available and consume less energy, making it an environmentally friendly solution.

Keywords IoT · Decentralized visualization · Blockchain · Big data visualization · Hyperledger Sawtooth · Secure distributed storage · Green solution

Introduction

With the growth of technology, IoT generated Big data and its associated analytic have become common in the computing industry. The cloud-based Big data solutions have resulted in a significant increase in energy consumption at host data centers. As most of the time, organization's small yet distributed storage resources remain underutilized. A considerable amount of energy can be conserved by optimizing the utilization of local storage efficiently.

Nowadays, everything is digitally recorded; IoT systems generate and process massive amounts of data every day. However, to process and analyze these data are not enough. Our brain tends to find patterns more efficiently when the data are visually represented. Data visualization and analytics have an essential role in decision-making for Big data [3]. Data visualization leads to new opportunities by representing the data in innovative and visual means [26]. However, with the huge volume of data generated, interactive, scalable visualization and data processing are considered a constant

✉ Yousaf Bin Zikria
yousafbinzikria@ynu.ac.kr

✉ Sadia Din
saadia.deen@gmail.com

Iqra Shahzad
ishahzad.mscs24@students.mcs.edu.pk

Ayesha Maqbool
ayesha.maqbool@mcs.edu.pk

Tauseef Rana
tauseefrana@mcs.edu.pk

Alina Mirza
alinamirza@mcs.edu.pk

Wazir Zada Khan
wazir.zada@cust.edu.pk

Sung Won Kim
swon@yu.ac.kr

[1] Department of Computer Software Engineering, MCS, National University of Sciences and Technology (NUST), Islamabad, Pakistan

[2] Department of Computer Science, NBC, National University of Sciences and Technology (NUST), Islamabad, Pakistan

[3] Department of Electrical Engineering, MCS, National University of Sciences and Technology (NUST), Islamabad, Pakistan

[4] Department of Computer Science, Capital University of Science and Technology, Islamabad, Pakistan

[5] Department of Information and Communication Engineering, Yeungnam University, Gyeongsan 38541, South Korea

challenge. The size of this information, while visualization is so excessive that an ordinary processing technique is insufficient. Construction of each perception may requires in one hand traversing the whole data or on other reviewing a small subset of the entire data [11]. For this purpose, expensive servers are used. Therefore resources and their efficient utilization is a major concern.

Modern IoT systems involve complex interaction among different entities, demanding a secure, efficient, and distributed infrastructure. Therefore, distributed data visualization is considered an excellent solution to achieve perception and overview of this multi-dimensional massive data [19]. Big data visualization is different from the conventional visualization of data. The visualization of Big data involves a large variety of data. Consequently, high-level data processing techniques and speed are needed to visualize this vast volume of data [22]. There exist multiple web and user-based techniques for the visualization of Big data. These techniques are useful and provide decentralization but with an increase in cost for buying servers and equipment, which is a significant concern.

Blockchain techniques are attractive and have gained researchers' interest due to their decentralized architecture, immutability, and anonymous record-keeping without centralized authority [6]. Blockchains initially gathered interest to support cryptocurrency. However, due to its decentralized nature, blockchain has emerged as a distributive computing technology that provides a secure environment to support various interactions [21].

Blockchain is considered a highly immutable, decentralized, and modular platform to transfer assets during the last few years. It is considered a generalized approach for implementing decentralized computing applications. Blockchain technology is a decentralized and unchangeable database that is highly transparent and secure. It uses a peer-to-peer network with no central authority. It contains transaction records, and these records are shared among all participating nodes. These properties of blockchain make it a suitable candidate for decentralized Big data visualization with advantage of achieve cost-effectiveness, high security, and readily available results.

We have proposed a green solution BGbV as a support to existing distributed visualization systems. The BGbV framework is based on the blockchain platform, i.e., "Hyperledger Sawtooth" and is targeting to achieve the following features:

- Tamper proof record-keeping of data for distributed data visualization.
- Utilization of relativity smaller available distributed resources.
- Green solution with optimal and economical resource utilization with least energy consumption.
- No central point of failure as multiple copies of data are kept.
- Quality enforcement mechanism by ensuring high availability from the most rated node.

For our research, a big temporal data set for the crime rate has been taken as a case study to perform the proposed research. Data are gathered from decentralized nodes and visualization is created according to user's requirements. We have proposed a solution BGbV that will benefit in many ways, including less cost of buying resources for decentralized computing and lower energy consumption. It will allow users/people and organizations to gain secure and readily available data visualization that is decentralized and provides productive outputs for analysis.

In the rest of the paper, "Literature survey" presents the literature review. "BGbV: blockchain-based green big data visualization" section presents the BGbV paradigm. "Analysis of proposed system" discusses features and analysis of the BGbV framework. "Implementation and testing results" presents the implementation and testing details of our solution. We conclude this paper in "Conclusions and future work" and suggest future work.

Literature survey

Many researchers have addressed Big data and its issues [24,31]. Exploring Big data offers numerous appealing features, but specialists and experts are also confronted with many difficulties while investigating such mines of data [23]. Many data sets are too large and complicated to manage on available memory units and are distributed across the clusters of computers [17]. Due to the growing nature of Big data, it is hard to avoid its challenges. The most immediate issues that need to be addressed are storage issues, management issues, and processing issues [16]. It is challenging to handle data connectivity issues, storage limitations, and data processing capabilities in real-time Big data [23]. The exponential growth in structured and unstructured data has compelled the need for an efficient and reliable storage approach. Therefore, the reliability of devices matters a lot concerning storage approaches chosen for handling Big data. While processing, backup, and archiving Big data, there could be many challenges like storage medium, data replication, and duplication [1]. Big data applications use high-speed servers and equipment, which results in increased cost.

A distributed paradigm is considered a suitable replacement for costly supercomputers. Distributed approaches are decentralized and aim to disrupt the existing central and conventional ways to deal with huge volumes of data. It also ensures to handle and deal with new expanding client's needs and application demands. Data storage, data access, data

transfer, and data visualization activities use distributed computing with low-cost machines to make Big data analysis and processing possible within a reasonable cost and time [20]. In distributed computing systems, the focus is on data representation. It attempts to addresses challenges of processing, interaction, and representation of data in proficient manners.

Instead of textual/numerical analysis data, visualizations are producing better perceptions and understanding of data. However, power and speed limitations are there when visualizing Big data. This leads to the scalability issue [9]. Current processing technologies and systems cannot satisfy the needs of big growing data, processing of data, and visualization. The increase in speed and storage capacity is much lesser than the amount of Big data. Speed and complex processors are needed [24]. Data visualization is a significant way to deal with Big data. It helps to get an overall perspective of Big data and find information esteems. It also helps to find the hidden patterns in data. The challenge is to manage the parallelism that blends between distributed and shared techniques with such massive data. For the issue mentioned above, the requirements of machines are a fundamental concern. Many researchers have shown that large-scale distributed data visualization is an input\output-bound problem. When interactivity is required, security and data access become significant problems, mainly when the data are distributed over wide-area networks. Many user interfaces and web-based data visualization techniques are available, which provide efficient decentralization with an additional cost for buying servers and equipment. This is the challenge for these techniques. Tables 1 and 2 provide Big data visualization techniques. Table 2 also provides websites to get these tools for the visualization of data. These tools offer visualizations of data based on its features. One of the key challenges in modern distributed data visualization today is how to provide users with an interactive experience. Techniques presented in Tables 1 and 2 are being used in many distributed applications to create and achieve the intended visualization of data [25]. These distributed data visualization frameworks are presented to attain interactive results and to address scalability. Distribution in these techniques is achieved with the help of substantial remote servers. Similarly, high-performance computing capability is needed to get interactive results [14].

Blockchain is emerging as new distributed technology. The blockchain concept is based on distributed ledger maintained by multiple parties [6]. Using blockchain, we can build decentralized systems [8], which can effectively mitigate problems associated with high access and communication costs. Also, a decentralized blockchain architecture is more resistant to a single point of failure [6]. Experiments show that the scalability issue can be effectively handled in blockchain systems.

Table 1 UI-based visualization techniques [25])

	User interface-based	Official websites
1	Tableau Desktop	https://www.tableau.com/
2	Gephi	https://gephi.org/
3	Jupyter	https://jupyter.org/
4	Sisense	https://www.sisense.com/
5	Qlik	https://www.qlik.com/
6	Team Mate	https://www.wolterskluwer.com
7	Infogram	https://infogram.com/
8	Datawrapper	https://www.datawrapper.de/

Table 2 Web-based visualization techniques [25])

	JavaScript based	Official website
1	FusionCharts	https://www.fusioncharts.com
2	HighCharts	https://www.highcharts.com/
3	Dygraphs	http://dygraphs.com/
4	Timeline JS	http://dygraphs.com/
5	Chart JS	http://www.chartjs.org/
6	D3 JS	https://d3js.org/
7	Leaflet	http://leafletjs.com
8	Google Charts	www.developers.google.com/chart/
9	RawGraph	http://rawgraphs.io/

Decentralized computing frameworks are aimed in several ways at disrupting the current cloud environment and scalability problem.

For achieving interactive outputs from huge streams and sets of big data, blockchain-based decentralized approaches have received great attention. Traditional distributed peer-to-peer (p2p) networks have inevitable disadvantages, including insecurity and lack of auditing and incentives [12]. Filecoin [7] and BigchainDB [4] are scalable blockchain databases that combine the characteristics of both blockchain and existing distributed databases. In all respects, blockchain is opening doors for solving numerous problems for many applications in a distributed manner [2]. Blockchain-based frameworks offer many benefits for distributed storage with features like availability, no single point of failure, confidentiality, privacy, and integrity [5]. Currently, Filecoin [7], Sia, Swarm, and Storj, etc. are mainstream distributed storage solutions built on blockchain [33].

Sia uses blockchain technology to provide an open market to buy and reserve unused computing space for users. Conditions of storage, such as availability and active duration that are agreed by the participants under file contracts, are encrypted service-level arrangements. These are immediately stored on the Sia blockchain and done [29].

Filecoin is another decentralized blockchain and native cloud service. It produces an incentive-based scheme that

provides additional storage. Filecoin uses IPFS, a P2P distributed protocol, where each file is encrypted with a hash key and containing indexing details. It enables large amounts of data to be searched and saved and distributed with high productivity [18].

Another protocol for decentralized storage network is SWARM. It is permission less and communication infrastructure [28]. The principal goal of swarm is to provide the decentralized app's creators with infrastructure resources. To implement the utilities, SWARM uses smart contract platforms (Ethereum). The strategy for decentralized storage is a peer-to-peer approach.

It is not deniable that blockchain has opened doors for solving many problems for many applications in a distributed manner with provision of various kinds of distributed storage approaches mentioned above [2].

Considering these blockchains based on distributed storage such as Filecoin, Sia, Swarm, etc, the set of transactions are stored within the of blocks of immutable blockchains and generate a type of decentralized database or storage of structured data. However, due to scalability concerns, the size of blocks cannot expand very large, and thus, it can be clearly seen that above-mentioned distributed storage blockchains are not meant to store and handle large amounts of data and will take too long to process and resources to accomplish these executions. These problems lead to increase the overall energy of the network. Due to consumed energies in nodes and long communication distances, the network lifespan, and energy consumption and maximization; it has been a critical issue and it increases the overall energy of the network [27]. Hence, these generic schemes are not much energy efficient also do not address optimizing network and storage cost.

All projects are similar in the way as they are trying to provide decentralized storage, but existing solutions are controlled by centralized third party and are mainly commercial applications. The data stored are available to intermediaries who are providing services. The involvement of intermediaries and third parties leads to an increase in operational and transnational costs and lack of security which makes the system vulnerable.

Hyperledger Sawtooth is also a blockchain framework used in developing and running applications in a distributed manner smoothly [13]. Sawtooth also addresses the challenges of permissioned (private) networks. Sawtooth clusters with different approval can be implemented quickly. No central service may leak transaction trends or other classified details. Sawtooth Blockchain can help to provide efficient data storage and access framework for many private and small organizations and applications [2].

Despite, all good blockchain systems are facing scalability issues. Due to a number of scalability issues and limited on-chain storage capacity, the feasibility of such a blockchain-based data network is restricted [27]. A variety of methods have been suggested to increase scalability while maintaining decentralization and security in this regard. Sharding technique is one of it.

Sharding is one of the most practical approaches for achieving scalability by partitioning network into several shards, to reduce the overhead of scalability [32]. However, the issue is to maintain atomicity during cross-shard operations. Each shard is stored in a separate server instance [30].

Although this spreads the load, but increases the overall cost and increases the energy consumption. Sharding can be a good choice when it comes to public and permission-less blockchains, but for private small-level projects, sharding technique may be costly, complex, and consume more energy. Table 3 presents a comparison between some blockchain-based storage services. Provided in Table 3 is relative estimation four parameters, i.e., closed-source decentralization, exposure of data to third party, network cost and energy consumed, scalability, energy-efficient objectives, and free of cost solution.

Many blockchain-based frameworks offer benefits like availability, no single point of failure, confidentiality, privacy, and integrity [5]. Hyperledger Sawtooth is a blockchain framework used in developing and running applications in a distributed manner smoothly [13]. Blockchain can provide efficient data storage and access framework for many organizations and applications [2]. As the visual representation of Big data communicates ideas more clearly, the need for data visualization is increasing in different domains. However, it is a complex and demanding task to visualize a massive volume of data. Therefore, a distributed and green solution is needed to achieve interactive data visualization with less cost and energy consumption. Using blockchain-based mechanism for distribution and decentralization of big data and then visualize could be a good solution. However, aforementioned blockchain-based techniques also increase the chances of escalate overall network cost with high energy and resources consumption. Also users may somehow be required to be controlled by some third-party service making it centralized somehow.

Therefore, there is need of green solution with minimum energy consumption having low network cost providing optimum storage solution, for small and big levels of projects ensuring security, available and reliable data, and its visualization at same time.

BGbV aims to provide support to distributed data visualization. It ensures security and availability, and, most importantly, provides a framework for better utilization of existing distributed resources. By utilizing small units of storage effectively, it facilitates scalability challenges. We present a green solution that will reduce cost by utilizing small resources that are already available and lessen energy consumption and make it an environmentally friendly solution.

Table 3 Comparison between existing blockchain-based storage services

Parameters	Sia	Ipfs	Swarm	StorJ	BGbV
Closed-source decentralization	Yes	Yes	Yes	Yes	No
Exposure of data to third party	Yes	Yes	Yes	Yes	No
Network cost and energy consumed	High	High	High	High	Low
Scalability	Medium	Low–medium	Low	Low	Depends upon organization's free storage resources
Energy efficiency objective	Not defined	Not defined	Not Defined	Not defined	Yes
Free of cost adaptation	No (SiaCoin)	No (FileCoin)	Resource trading (No)	No (StorJCoin)	No

BGbV: blockchain-based green big data visualization

BGbV proposes a distributed architecture of blockchain-based green solution for Big data visualizations. Many organizations have loads of distributed storage resources available in unused disk space of workstations and terminals. Instead of getting new equipment that requires more hardware and power consumption, this solution provides optimal utilization of the existing smaller distributed storage units with the least energy consumption.

In BGbV, we have incorporated the hyperledger sawtooth framework with our front-end application. Hyperledger sawtooth is highly modular framework and isolates the client part of the core system's application. In our prototype, we have considered small-to-medium-level organization with mid-range resources to store the temporal big data set. According to our scope, the permissioned (private) hyperledger sawtooth blockchain is beneficial as it maintain the privacy and security. We have developed the client part of our application and core system in javascript. Our proposed system presents the design of a blockchain-based data storage and access framework for interactive visualization of big temporal data.

Proposed BGbV is different from existing techniques mentioned earlier, i.e., due to its integration with blockchain technology with optimum resource utilization and least energy consumption making it more green solution. It ensures many significant benefits like effective and optimal usage of distributed storage, temper proof record-keeping mechanism, no single point of failure, and availability.

In BGbV, metadata are stored on the blockchain, whereas the actual files are stored off-chain on multiple locations after distribution. This distribution is according to a pre-defined agreement using a peer-to-peer network. This scheme will provide decentralized storage on an economical and secure basis.

Architecture overview

BGbV framework achieves distributed data visualization with an interactive experience. Distributed processing and efficient lookup capabilities are also incorporated in our solution. The major contributions of this research are as follows:

- Design a sawtooth blockchain-based data storage and access framework that stores the metadata on the hyperledger blockchain, whereas the actual data are stored off-chain on multiple nodes.
- Design an off-chain distributed storage data block for cost-effective use of blockchain.
- Distributed and efficient storage of divided blocks of data on multiple nodes in which contents are accessed through permissioned hashes with high throughput.
- Instead of buying large servers for a remote decentralized solution, employing existing unoccupied storage of smaller machines (nodes).
- Temper proof record-keeping mechanism.
- Enforcing quality with no single point of failure and continuous availability of data.
- An optimized solution toward big data without third-party control for decentralization, with minimum network and storage cost.

We have introduced the basic distributed data visualization of temporal data and their interaction with the blockchain. In our solution, we have taken a data set for the US's crime rate as Big data in a temporal context. BGbV has two primary stages according to the user's request and demand for data. It entertains the client's request based on present nodes and node ratings. In BGbV, we have introduced two blockchains.**BGbV** for storing nodes available and their response-based rating and **BGbVfi** for storing metadata for the transaction on data.

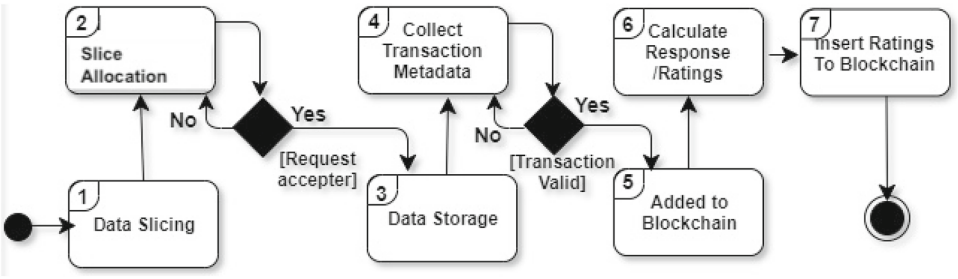


Fig. 1 Flow of activities in **BGbV** blockchain

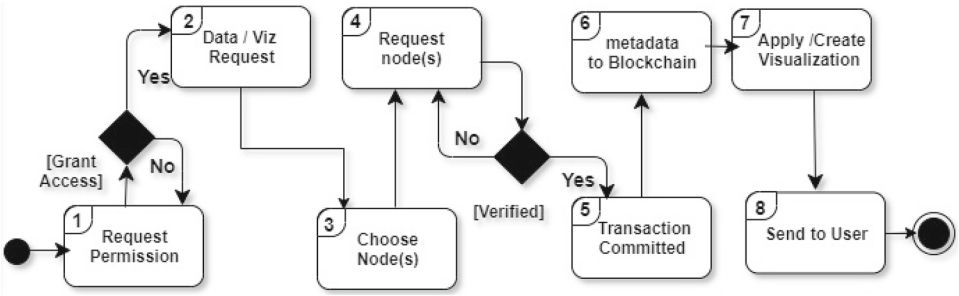


Fig. 2 Flow of activities in **BGbV** for information visualization

In the first stage, as an initial step, the data set is retrieved from bulk storage on an administrative view of a web-based application.

The flow diagram in Fig. 1 shows the activities perform on the administrator's machine to achieve distribution with the help of blockchain. The administrator has the right to slice the data on some pre-selected criteria. It can then be assigned to an appropriate node that desires to participate in storing Big data. Before this assignment of data, a dialogue mechanism is implemented between the administrator machine and nodes. This mechanism ensures that the node has agreed to share its resources and disk space. To store data slice, the administrator selects the available node according to its storage capacity and rating. It then prompts the storing note by sending a request for sharing space to save data on it. At present, administrator role is defined as an intermediatory for transfer of data from centralized bulk storage to sliced distributed storage. Administrator's task is to divide the data on agreed/pre-defined criteria and send data to storage nodes. When the data are once stored on agreed nodes, to ensure the integrity, the role of administrator has no role to play in access of data. The role of administrator will be automated in future by defining performance metrics and status updates of free storage nodes.

If this device agrees on saving data on it, the administrator sends the slice of distributed data to that node, and its response time is captured. Both the rating and available devices are stored in blockchain **BGbV**. After this agreement, sliced data are sent to a candidate node. Information about all those nodes that are ready to share space is stored in one blockchain **BGbV**. This blockchain is implemented to store two parameters: first is the candidate node identity; second is its ratings and response time.

After this process, control is transferred to the second blockchain **BGbVfi**. Multiple copies of same slice are to be distributed among multiple nodes thus providing safeguards against data loss; when data slice is requested the node with better rating and availability is chosen (step 3 of Fig. 2). The **BGbVfi** will be used to store the metadata for transactions in our solution. The flow of activities in both stages is described with flowcharts in Figs. 1 and 2. We have use used two blockchains for our design prototype instead of storing the metadata and ratings in one blockchain. The reason for using two blockchains is to maintain the ideal scale of blockchain and not to make solution and blockchain bulky due to increased number of requests. We have used two blockchains **BGbV** and **BGbVfi** for keeping our transaction metadata log and distribution and slicing information separately. Using two blockchains leads to the benefit of separation of read and write operations from logs and ratings. Read/write operation is one operation and metadata of slice is other operation. Keeping them separate is a major reason for two blockchains. It disrupts the need for again and again replication of data in block.

Flowchart in Fig. 1 represents activities initiated in response to the user's request for information.

Our first blockchain **BGbV**is used to store information about nodes that holds data after agreement mechanism and their response-based ratings. Blockchain **BGbVfi** shown in Fig. 2 stores the metadata about the transaction. This blockchain will store parameters:

- How administrator sliced data.
- Amount of data stored on each node with its address.
- Sending time from administrator's machine, hash where our data have been stored.

Above-mentioned resources are on-chain resources for both blockchains. The second stage defines retrieving mechanism. For retrieving mechanism, when the user wanted to retrieve data, the hash and first blockchain are consulted, which helps users check the node with the most ratings and response time. That node is requested to send data to the user. The quality of the node is maintained by this blockchain **BGbV**. The information that how much data are stored on any node is also available on blockchain **BGbVfi**, which stored the distributed mechanism of data.

This lets the system choose the amount of data it wants to visualize and let them know which node they can get data to visualize. When the user sends a request, the node signs the batch and then connects and sends requests to the validator node on the Sawtooth network by using Rest API. Hyperledger Sawtooth provides a REST API for clients to interact with a validator using common JSON/HTTP standards. REST API uses a JSON envelope to send metadata back to clients in an easy and customized manner. The sawtooth's validator sends this transaction request to the transaction processor, which is a blockchain featuring smart contracts. It can apply any rules for the validation of transactions. The transaction processor thus attaches the payload to it. The transaction processor makes sure that it is valid, all dependencies are met, and then commits it. At present, data are sent to the node using FTP which entertains the user's request.

Off-chain resources

Some off-chain resources will help in the proposed solution. Blockchains are not general-purpose databases. It is not possible to store distributed blocks of data as this leads to an unmanageable size and causes the problem in the blockchain. Thus, blockchain only stores metadata, and actual data are divided into blocks and are stored as off-chain resources at multiple storage locations on the network. Therefore, off-chain resources contain distributed data on multiple nodes. This adds resilience toward failure and ensures smooth working and high availability. If one of the nodes failed due to any reason, availability will be ensured from some other nodes. Ratings from **BGbV** are also linked to this decentralized system that gives a score of the node's reputation. This in term helps the user to better judge if node and their contents should be trusted or not. These off-chain resources can be open to the public or restricted to authorized members. A visualization pipeline is applied to that data after transfer. The visualization of that data is created for that corresponding user that requests for data visualization. Their computation is thus distributed. Update the state and outcome as a response is sent back. Then, the network's chosen consensus mechanism publishes it in a block to the rest of the network. Figure 3 shows the layered architecture of our given solution.

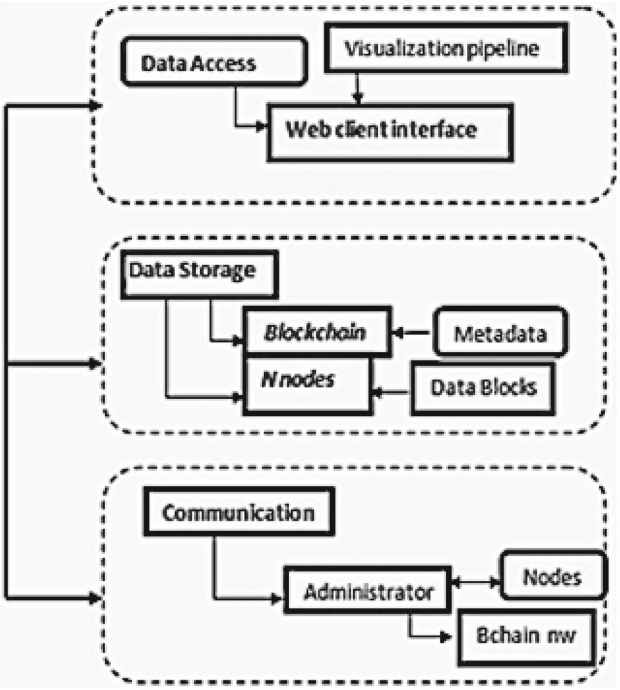


Fig. 3 Layered architecture of **BGbV**

Analysis of proposed system

In this section, we present the analysis of the energy and cost saved by BGbV and its adaptation. It is a fact that IT storage infrastructure is under utilized by many organizations. As per a survey conducted by CISCO [10], the utilization of direct access storage on average is 40% and is 60% for the storage access network. Organizations have underutilized distributed storage available in free disk space on several desktops and laptops. Such organizations can improve the utilization of already available resources by employing BGbV for their big data storage solutions. On average, the cloud-based solution for Big data visualization costs more than 2–4 USD per GB per month. Similarly, as Table 3 shows, all decentralized solutions are also commercial solution with associated costs. For most organizations, this cost is besides the cost of maintenance on site underused storage equipment.

In the following analysis, we present the details of energy consumed in storing data using BGbV step as compared to cloud-based storage. it can be seen that, in terms of energy consumption, the utilization of existing direct access storage solutions consumes less energy than cloud-based storage. In BGbV, we propose two blockchains **BGbV** and **BGbVfi**. As mentioned earlier, blockchain **BGbV** stores the rating mechanism, and blockchain **BGbVfi** saves metadata about data fragments of big data stored at multiple nodes. Each block of **BGbV** has two parts block header and body. Each block's body contains transaction data comprising approx. 264 bytes. The block header's information contains block

number, version, previous BlockHash, and the time-stamp at which this block is committed. The block's header is 46 bytes. Thus, the total size of each block is 310 bytes, which approximately equals 0.3 KB. In **BGbVfi**, the body of the block is 134 bytes. By adding the standard header size of 46 bytes, the net transaction size is approximately equal to 0.1 KB. By combining both blockchains of size 0.4 KB to 0.5 KB, the BGbV takes less than 1 KB overhead for storing each fragment.

Figure 4 shows the amount of energy saved using direct access storage devices compared to the cloud-based solution. The energy is calculated by

$$E_S = E_{cl} - E_{BGbV} \tag{1}$$

$$E_{cl} = F_{sz} * \lambda_{cl} \tag{2}$$

$$E_{BGbV} = \left(\sum_{i=1}^{K} F_i + K * N_{nodes} * Tr_{BGbV} \right) * \lambda_{HDD}. \tag{3}$$

Here, E_S is net energy saved, E_{cl} is net energy consumed by cloud, E_{BGbV} is net energy consumed by BGbV, Fsz is the total number of bits in the file, λ_{cl} is the energy consumed to write 1 bit on the cloud, λ_{HDD} is the energy consumed to write one-bit disk, Fi are the number of bits in each segment stored, N_{nodes} is the number of nodes maintaining BGbV blockchain, K is number to fragments, and Tr_{BGbV} is the number of bits in the transaction. Energy rate λ_{HDD} is kept as 100FJ and λ_{cl} is kept as 1nJ [15].

Figure 4 shows the energy saved against file size ranging from 1 to 60 TB is shown. For BGbV, we have sliced each file into fragments of size 1 GB. Also, we have kept the number of nodes that keep the copy of the blockchain to 100. The red line shows the upward trend in energy saved with respect to file size. Even though the number of fragments and transaction overload is also increased (shown as bars), but net energy saved in mJ improves with larger files. The above calculation is performed to establish the bottom line of the energy saved by considering just storage. Here, any additional costs of bit transfer to the cloud are not considered.

We have calculated only the energy spend in saving original files. It can be seen that backup and replication overhead will show the same trends.

Also, for IOT big data, the greatest obstacle for implementing blockchain in IOT is scalability, as the response to requests expands as the number of computing machine increases. This leads to high energy consumption. Our solution will also help with IOT big data. As mention the proposed system is decentralized and also use free local underutilized storage will be used to save data. The continuously growing IOT data can be efficiently stored and accessed with minimum network energy. BGbV is flexible in terms of source of data. Whether big data are transferred from the cloud or is generated by IOT, the BGbV framework can easily mange it.

Features

The proposed system is intended to ensure many features and add support to current distributed data visualization systems. Following are the prominent features of the proposed system.

Integrity

In the proposed system, the key feature is that all the information is unaltered and immutable, increasing the solution's integrity. The incorporation of blockchain ensures not only integrity but also provides control over data. Replicas of data on multiple locations and its metadata as evidence are stored on the blockchain, ensuring its persistence and integrity.

Availability

The ranking-based node selection in $BGbV\beta$ provides framework for data to be efficiently available for the users. Users can achieve and retrieve that corresponding response, i.e., visualization from any other node if one of the part nodes is offline. This ensures the data's availability for visualization and analysis from other locations(nodes) on the network.

Fault tolerance

BGbV has no single point of failure, and the system is more tolerant of faults. The data are vast and divided into different chunks, stored at multiple nodes of the P2P network. This makes our solution more robust and good in terms of throughput.

Node rating mechanism

BGbV maintains nodes and their rating with respect to their response to users. Each node available for storage is analyzed, and data are collected locally based on the performance. It defines their rating. It also enforces a quality mechanism for our proposed system.

Effective resource utilization

When working with a huge amount of data resource utilization is a very critical concern. It should be logical and intuitive. In BGbV, we are not using remote big servers. To make our solution as best economical, we are occupying unused smaller units of storage on different machines that are available for sharing their space and store data. Instead of using high-cost remote servers, data storage is distributed and efficient using significantly less cost and smaller units.

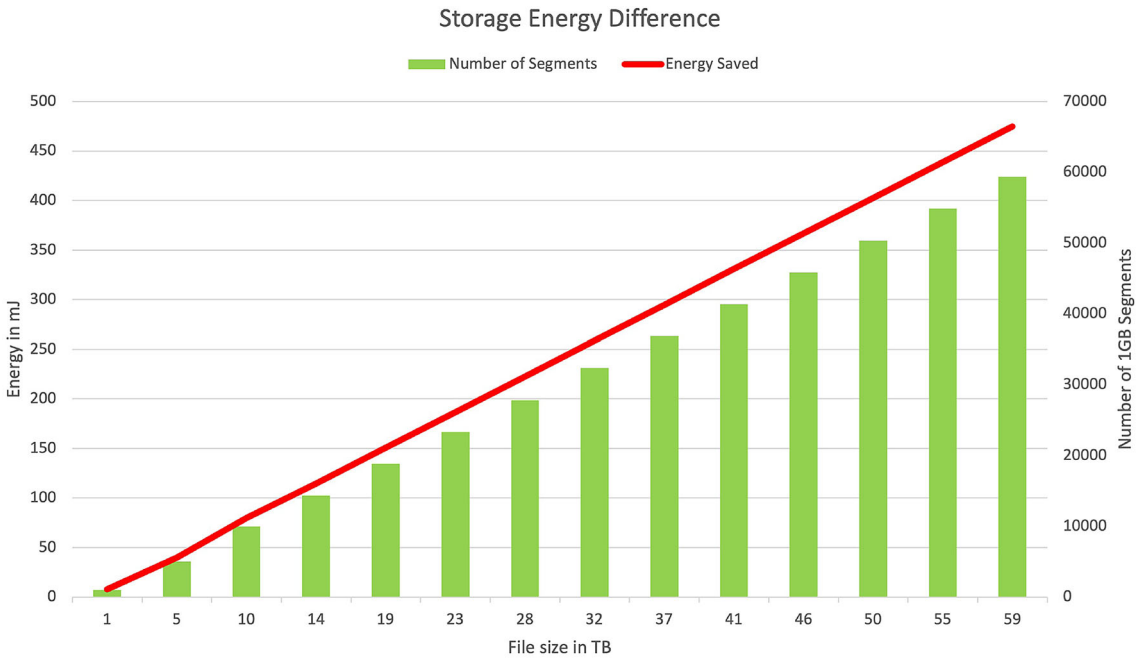


Fig. 4 Net energy saved by BGbV

Security

In BGbV, it is difficult for malicious participants to tamper with the data. Only authorized users have permission and access to data stored, and malicious activities are prevented. Blockchain networks also achieve strict security. After passing through a strictly ordered distributed blockchain, transactions are routed to the appropriate transaction processor, and hence, this processor then attaches the transactions' payload. By creating a domain-oriented transaction processor, actions are limited that can be performed on a blockchain network, which can improve security and performance.

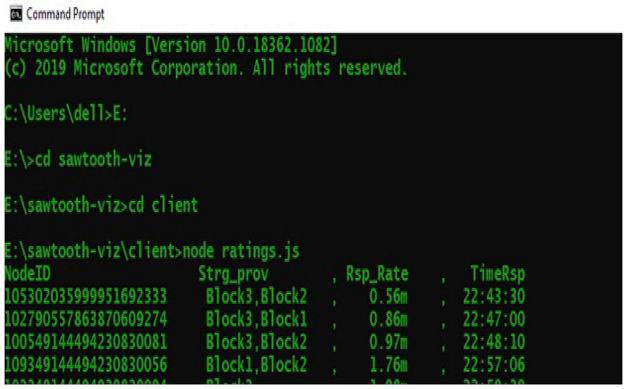


Fig. 5 Nodes with details of data stored and response rates

Implementation and testing results

This section discusses the implementation and the testing details. This application is developed in HTML 5.0 and Node JS as front-end languages.

Implementation details

This section will explain the prototype developed for the proof of our concept. Proposed architecture has the following components:

- A peer-to-peer network committing transactions between nodes.
- A distributed log, which contains a list of transactions.
- A transaction processor for processing those transactions.
- An off-chain storage mechanism for storing actual data files.
- A distributed storage mechanism for storing the resulting state after processing transaction.
- A consensus algorithm for ensuring consensus across the network.

For deploying, building, and running blockchain-based green Big data visualization (BGbV), we have integrated hyperledger sawtooth framework, which provides a modular and flexible platform for implementing transactions between nodes. For the proposed BGbV, we have taken m nodes to store the Big dataset of the US crime rate from 2001 to the present. Nodes with details of data stored and response rates

are shown in Figs. 4, 5, 6f. The application is developed in Html 5.0 and Node JS as front-end language.

The prototype we have implemented is highly modular and has two basic parts depending on user request and demand for data. The first part is a web-based interface for slicing and distributing the data over m nodes. This web interface is initial and mainly runs on the administrator's machine. The administrator slices the data according to pre-defined criteria and sends the data on agreed nodes after a consensus mechanism. The dataset we have taken for this implementation is the US crime rate from 2001 to the present. It is collected based on 20 different attributes of crimes in the US, e.g., to crime types and locations.

For our prototype, administrator has sliced the Big dataset into 5 blocks based on these attributes. Each block consists of 1000 rows of crime datasets along with its attributes. The administrator can send the sliced data to k nodes agree to save the data and share the space in different replications after consensus.

We have created a consensus mechanism in our prototype for running the tests and saving the data on available nodes. It ensures the agreement of nodes and the administrator's machine to share the space and save data. For this agreement, OAuth 2.0 is used as a protocol for authorization for the specific flow of authentication and authorization for data sending to client nodes. We generate a public/private key pair for every available node with their node id for authentication purposed to configure the network. For keeping our application highly modular and not bulky, integration of hyperledger sawtooth at this stage is kept separately. For incorporating Sawtooth blockchain, design-specific transaction processor TP is written in Node JS. When the node agrees to save data, storage metadata are collected and stored on **BGbVfi**. At the same time, the response rate is stored on BGbV. The second part of our front end application is for the user to retrieve data. While the user wants to retrieve the data, it will select nodes with the most higher response ratings from BGbV. A REST API is used, which helps users communicate with the sawtooth network implemented in our prototype.

Communication with the validator and submit any transactions with the help of HTTP requests. A data model is designed to tell the allowed operation and transaction types implemented in the transaction family. For our prototype evaluation, tests are run using the VS code and docker using version 1.49 and version 2.0.0.3, respectively. For our testing scenario, transaction processor runs on administrator machine using a unique machine address " **0f5a8cbbf216e93b6b3deed01c7b6866bedb4ed**". For evaluation and a test run, the administrator has sent the request to 10 nodes to share their space and hold data. The node receives updates about new blocks from the administrator machine. Out of offered, 4 nodes are agreed and ready for sharing space and data storage. As measurements for our project, we have collected the time and response rate of nodes with respect to their nodes IDs and saved them to BGbV.

Before the administrator sends the data, client authentications are established using Oauth2.0 . The administrator sends the data to these nodes. Sliced blocks are sent to each node with alternate replication. Node A is saving blocks 1 and 3, and Similarly, Node B has data blocks 2 and 3. Node 3 is holding blocks 1,2. Node D, the fourth agreed node, is holding blocks 3,2. Once the storage node has received the data, it will verify and send the administrator machine's acknowledgement. After the administrator receives confirmation from the storage node. TP running will commit the transaction are published to **BGbV**, as shown in Fig. 6f.

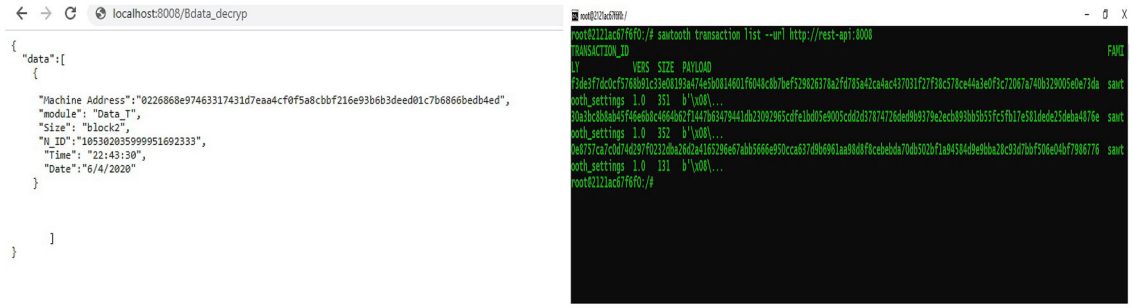
For testing and retrieving data, user requests node two for retrieval of data. The transaction processor also receives the request sent by the user. Transaction processor assesses the transaction on this node. The transaction list in the transaction processor is shown in Fig. 6d. Given that all verification pass, the storage node B sends the data back at a high response rate. A basic visualization pipeline is associated at this stage, and data visualization is created according to user-requested data to visualize data to understand better and get insights into this data. This helps to produce productive output from it. Distributed data visualization is presented as DApp using hyperledger sawtooth integrated with it. Implementation and testing scenarios are presented, so that the proposed solution can potentially express here.

Figure 6a, b shows the blocks created; the state list shows that data are saved in blockchain, i.e., metadata. After the agreement mechanism and after the verification, the transaction is submitted. Block is added to the blockchain and broadcasted on the network. After this transfer node, grant access to the user requested for data. Transactions committed are shown in Fig. 6b.

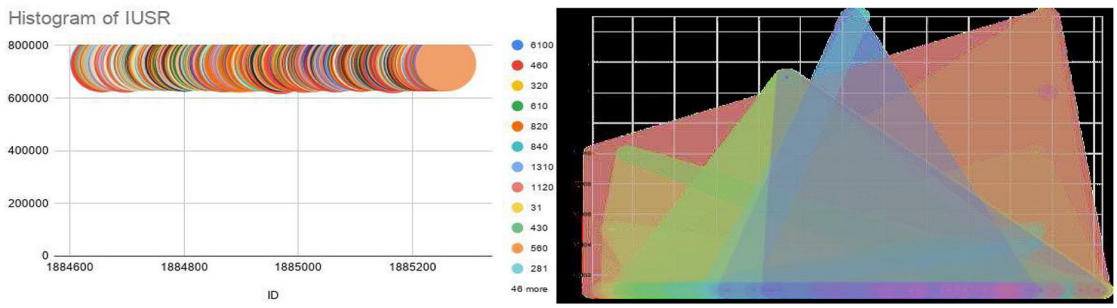
Figure 6c shows the successful execution data transfer from the administrator machine and node. A basic visualization pipeline is associated at this stage, and data visualization is created according to criteria the user selected to visualize data to understand better and get insights into these data. This helps to produce productive output from it.

A distributed data visualization is presented as DApp using hyperledger sawtooth integrated with it. Implementation and testing scenarios are presented to demonstrate the proposed solution. Outcome and some of the interactive visualizations based on the user's request are shown in Fig. 6e, f. It is established from our implementation that data can easily be stored in spare storage available in local network, thus reducing overall energy consumptions. The private blockchain framework using hyperledger Sawtooth provides support for temper proof record-keeping. Data from centralized storage are not only successfully stored on multiple nodes; it is also flawless retrieved for visualization.

(a) Blocks created in sawtooth blockchain. (b) metadata stored in blockchain



(c) Successful execution data transfer from administration machine and node (d) Transaction list in Transaction Processor TP



(e) Visualized IUSR parameter on the base of ID and Case no (f) Crime Rate spread by Year

Fig. 6 Results

Conclusions and future work

In emerging IoT systems, effective resource utilization, security, availability, and throughput of Big data are of main concern. We have proposed BGbV, a blockchain-based solution for effective storage and retrieval of Big data visualization. The integration of blockchain solutions provides cost-effective, secure, and available visualizations of Big data. BGbV presents a mechanism of the utilization of relatively small distributed resources optimally for storing Big data. As our system provides an alternative to the purchase of additional resources, thus reduces energy, costs, and electronic waste. Additional benefits of our scheme are the high availability and better extensibility.

In future work, the crowd-sourcing of storage nodes will be explored by adding an effective incentive mechanism. Nodes will be awarded incentives depending upon the rating and reviews they get from the users. Developing such an architecture that allows large data visualization with incorporating benefits like high availability, no single point of failure, quality, and interactive outputs will prove beneficial and will noticeably improve the business sector's performance.

Acknowledgements The authors are obliged to the National University of Sciences and Technology for funding this work through the Researchers Supporting Grant, National University of Sciences and Technology, Islamabad, Pakistan and supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2021R1A6A1A03039493).

Declarations

Conflicts of interest The authors declare no conflicts of interest.

References

1. Agrawal R, Nyamful C (2016) Challenges of big data storage and management. Global J Inf Technol Emerg Technol 6(1):1–10
2. Ali S, Wang G, White B, Cottrell RL (2018) A blockchain-based decentralized data storage and access framework for pinger. In: 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE), pp 1303–1308. IEEE
3. Ali SM, Gupta N, Nayak GK, Lenka RK (2016) Big data visualization: tools and challenges. In: 2016 2nd International conference on contemporary computing and informatics (IC3I), pp 656–660. IEEE
4. Angeline R, et al. (2018) An immortal database system for the decentralized internet. In: 2018 3rd International conference on communication and electronics systems (ICCES), pp 994–998. IEEE
5. Asharaf S, Adarsh S (2017) Decentralized computing using blockchain technologies and smart contracts: emerging research and opportunities. IGI Global
6. Ayoade G, Karande V, Khan L, Hamlen K (2018) Decentralized iot data management using blockchain and trusted execution environment. In: 2018 IEEE international conference on information reuse and integration (IRI), pp 15–22. IEEE
7. Benisi NZ, Aminian M, Javadi B (2020) Blockchain-based decentralized storage networks: a survey. J Netw Comput Appl pp 102656
8. Bistarelli S, Pannacci C, Santini F (2019) Capbac in hyperledger sawtooth. In: IFIP international conference on distributed applications and interoperable systems, pp 152–169. Springer
9. Childs H, Geveci B, Schroeder W, Meredith J, Moreland K, Sewell C, Kuhlen T, Bethel EW (2013) Research challenges for visualization software. Computer 46(5):34–42
10. cisco: (2009) https://www.cisco.com/c/dam/en_us/about/ciscoitatwork/downloadszciscoitatwork/pdf/Cisco_IT_Ops_Practices_Storage_Utilization.pdf
11. Drossis G, Birliraki C, Patsiouras N, Margetis G, Stephanidis C (2016) 3d vizualization of large scale data centres. Closer 1:388–395
12. Huang H, Lin J, Zheng B, Zheng Z, Bian J (2020) When blockchain meets distributed file systems: an overview, challenges, and open issues. IEEE Access pp 50574–50586
13. Hyperledger: Sawtooth: an introduction (2019). https://www.hyperledger.org/learn/white-papers
14. Javed MU, Rehman M, Javaid N, Aldegheishem A, Alrajeh N, Tahir M (2020) Blockchain-based secure data storage for distributed vehicular networks. Appl Sci 10(6):2011
15. Jiang W (2018) http://large.stanford.edu/courses/2018/ph240/jiang2/
16. Kaisler S, Armour F, Espinosa JA, Money W (2013) Big data: issues and challenges moving forward. In: 2013 46th Hawaii International conference on system sciences, pp 995–1004. IEEE
17. Kosar T (2012) Data intensive distributed computing: challenges and solutions for large-scale information management. Inf Sci Ref
18. Labs P, Benet J (2014) A decentralized storage network for humanity's most important information | filecoin . https://filecoin.io/
19. Leigh J, Johnson A, Renambot L, Vishwanath V, Peterka T, Schwarz N (2012) Visualization of large-scale distributed data. In: Data intensive distributed computing: challenges and solutions for large-scale information management, pp. 242–274. IGI Global
20. Mazumder R, Bhadoria RS, Deka GC (2017) Distributed computing in big data analytics. In: InCon-cepts, technologies and applications, Springer, New York
21. Nguyen BM, Dao TC, Do BL (2020) Towards a blockchain-based certificate authentication system in Vietnam. PeerJ Comput Sci 6:e266
22. Olshannikova E, Ometov A, Koucheryavy Y, Olsson T (2016) Visualizing big data. In: Big data technologies and applications, pp 101–131, Springer, New York
23. Oussous A, Benjelloun FZ, Lahcen AA, Belfkih S (2018) Big data technologies: a survey. J King Saud Univ Comput Inf Sci 30(4):431–448
24. Samal N, Mishra N (2015) Big data processing: big challenges and opportunities. J Comput Sci Appl 3(6):177–180
25. Segall RS, Cook JS (2018) Handbook of research on big data storage and visualization techniques. IGI Global
26. Seref, Sinanc D (2013) Big data: a review. In: 2013 International conference on collaboration technologies and systems (CTS), pp 42–47, IEEE
27. Swain A, Salkuti SR, Swain K (2021) An optimized and decentralized energy provision system for smart cities. Energies 14(1451)
28. Swarm E (2020) Sawrm documentation . https://swarm-guide.readthedocs.io/en/latest/introduction.html
29. Tech S (2020) About-sia . https://sia.tech/about
30. Terado T (2018) What is decentralized storage? (ipfs, filecoin, sia, storj & swarm) . https://medium.com/bitfwd/what-is-decentralised-storage-ipfs-filecoin-sia-storj-swarm-5509e476995f
31. Yang R, Xu J (2016) Computing at massive scale: Scalability and dependability challenges. In: 2016 IEEE symposium on service-oriented system engineering (SOSE), pp 386–397, IEEE
32. Yu G, Wang X, Yu K, Ni W, Zhang JA, Liu RP (2020) Survey: sharding in blockchains. IEEE Access 8:14155–14181
33. Zhu Y, Lv C, Zeng Z, Wang J, Pei B (2019) Blockchain-based decentralized storage scheme. J Phys: Conf Ser 1237:042008