Provenance Metadata Management in Distributed Storages Using the Hyperledger Blockchain Platform*

Andrey Demichev[1], Julia Dubenskaya[1], Elena Fedotova[1], Alexander Kryukov[1], Stanislav Polyakov[1], and Nikolay Prikhod'ko[2]

[1] Skobeltsyn Institute of Nuclear Physics
Lomonosov Moscow State University, Moscow, Russia
demichev@theory.sinp.msu.ru
[2] Yaroslav-the-Wise Novgorod State University, Velikiy Novgorod, Russia
niko2004x@mail.ru

Abstract. The paper is devoted to the development of principles and operation algorithms of provenance metadata management system, entitled ProvHL (Provenance HyperLedger), based on the integration of blockchain technology, smart contracts and provenance metadata driven data management. It is fault-tolerant and reliable in terms of the safety and security of provenance metadata records from accidental or intentional distortion.

Keywords: distributed storage · provenance metadata · blockchain · access rights · Hyperledger Fabric.

1 Introduction

This work is aimed at developing design principles of a system for storage and management of provenance metadata for data generated by large-scale scientific experiments. A vivid example of such installations is the Large Hadron Collider (CERN, Geneva; https://home.cern), the time of active work of which and, accordingly, the generation of large scientific data, is several tens of years, and the processing time of the data will be at least twice as much. Without detailed and correct provenance metadata, comparing the results obtained with an interval, for example, of a few years, will be simply impossible. Another important example is astroparticle physics which has become a data intensive science with many terabytes of data and often with tens of measured parameters associated to each observation (see e.g., [1]). While 10–15 years ago there were 1–10 Tb of data per year in astrophysics, new experimental facilities generate data sets ranging in size from 100's to 1000's of terabytes per year. Moreover new highly complex and massively large datasets are expected to be produced in the next decades by novel and more complex scientific instruments as well as results of data simulations needed for physical interpretation.

* This work was funded by the Russian Science Foundation (grant No. 18-11-00075).

Although a number of projects have been implemented in recent years to create systems for the support and management of metadata, including the data provenance, but the vast majority of implemented solutions are centralized [2, 3], which is inadequate for the use in distributed environments and the possibility of using metadata by organizationally unrelated or loosely related research communities. The use of centralized solutions in a distributed environment that includes different administrative domains is always associated with the problems of organizing the management of the central service, as well as with ensuring the trust in such a service from the side of participants of the distributed system. In addition, any central service is a bottleneck and a point of failure of the system. On the other hand, in recent years, distributed registries based on blockchain technology [4] have become very popular in various applied areas thanks to a number of important advantages. Just recently, blockchain-based provenance metadata (PMD) management systems have also been developed [5, 6]. Analysis of the proposed solutions shows that they are designed for cloud storage environments, are quite heavyweight and resource-consuming. The latter is related to the specific peculiarities of realization of the distributed registries as blockchains (namely, necessity for block mining). This makes doubtful the prospects for successful use of these solutions for the storage and management of provenance metadata generated by large scientific experiments in distributed environments.

In this paper, we propose an approach to solving this problem based on the use of blockchain technology and smart contracts within the Hyperledger platform (https://www.hyperledger.org) [7]. The basic principles of operation and algorithms of the ProvHL system (Provenance HyperLedger) for managing provenance metadata and data access rights in distributed storages are presented.

2 Basic principles of operation of the system for managing provenance metadata

2.1 Smart contracts

The basic scenario of using the proposed system assumes that a virtual organization (VO) is formed for the joint implementation of a certain project. VO includes several real organizations, in turn including data providers, users and data handlers affiliated with them. It is assumed that the implementation of such a project requires the use of a distributed data storage. This distributed storage can be formed by renting multiple cloud storage, as well as integrating the own storage resources of the organizations that form the VO. Thus, the hardware and software basis of the business environment in this case is formed by a set of storages (possibly of different types, e.g., cloud storages, file servers, tape storages, etc.), each of which can be managed by its own data management system (DMS). For certainty, it is further assumed that the data is stored as files, i.e. the file is a unit of data. Generally speaking, several VOs can coexist; the storages with which they interact can form partially overlapping sets.

In such an environment, an immutable and distributed (as the environment itself) registry and a consensus on the order of data operations are needed to resolve possible conflicts between the VO/project participants related to the use of the data. Conflicts may be caused by priority issues upon obtaining results of data processing, use of results (including funding issues), interrelations with storage providers, etc. In order to prevent possible conflicts, accurate implementation of mutual data access policies is required. In other words, support of business processes for data sharing and storage is needed.

A smart contract along with the registry form the basis of a blockchain system. While the registry contains information about the current and historical state of a set of business objects, a smart contract determines the executable logic that generates new states to be added to the registry. Before parties of a business process can enter into interactions with each other, they must define a common set of contracts covering common terms, data, rules, concept definitions and processes. Taken together, these contracts define a business model that governs all interactions between transactional parties. A smart contract defines these rules between the parties in the form of executable code.

2.2 Permissioned blockchains

A natural solution for the establishment of a distributed immutable registry for the provenance metadata (PMD) records is the use of the blockchain technology. The latter guarantees that no records were inserted into the registry in hindsight, no entries were changed in the registry and the registry has never been branched or bifurcated. An important question is how to provide validation of the chain of blocks with transaction records in the case of PMD registry. The use of the most popular proof-of-work (PoW) method [4] on the basis of mining is very resource-intensive, and is poorly suited for management systems for provenance metadata in the case of processing scientific data. Indeed, the calculations that are performed within the framework of PoW themselves do not serve any useful purpose, and this is a principle feature. It is very difficult to come up with a proof of work that would serve a socially useful role. Therefore, if possible, it is better to abandon it. Trying to solve these problems, a community of researchers in this field offers a variety of consensus algorithms that do not require “work”. The choice of the algorithm heavily depends on the way of access to transaction processing. From this point of view, blockchains are classified as follows:

- permissionless (public) blockchains, in which there are no restrictions on the transaction handlers;
- permissioned blockchains, in which transaction processing is performed by specified entities.

Public blockchains are more known because cryptocurrency networks are based on them. In contrast to the permissionless blockchains, in the systems based on permissioned blockchains, the built-in coins are usually not used. Built-in coins are required in permissionless blockchains to provide a reward for processing transactions. Permissioned blockchains can form a more controlled and predictable environment than public blockchains and does not require calculations related to the PoW algorithms. In the distributed storage environment, the local data management systems, data owners, representatives of real organizations participating in the project, etc., can act as the authorized parties that create and sign the blocks. In order to maliciously change a transaction confirmed by all the authorized parties in the distributed storage environment, the attacker must gain access to all the secret keys of the block handlers. This is very unlikely, and thus this approach provides a high degree of protection for the distributed registry. It is this approach to the construction of the metadata registry that was implemented in our PMD management system.

2.3 Hyperledger blockchain platform

To put this solution into practice, it is convenient to use existing blockchain platforms. Analysis of existing platforms shows that the required solution for the PMD management system most naturally can be implemented on the basis of the Hyperledger Fabric permissioned blockchain platform (HLF; hyperledger.org) [7] together with Hyperledger Composer (hyperledger.github.io/composer). The latter is a set of tools to simplify the use of blockchain. Hereafter we shall refer to these two components as HLF&C-platform. To describe the business process within the framework of HLF&C-platform, a number of concepts are used, the main ones are assets, participants, transactions and events. Assets are tangible or intellectual resources, services or property, records of which are kept in the blockchain. Assets must have a unique identifier, but they can also contain any properties defined for them. Participants are members of the business network who can own assets and make transaction requests. They also can have any properties if necessary. Transaction is the mechanism of interaction of participants with assets. Messages about the events can be sent by transaction processors to inform external components of changes in the blockchain. Very important that HLF&C-platform provides the operation of smart contracts (called chaincode), which allows us to organize the business process of sharing storage resources by project participants located in different administrative domains. The suggested ProvHL system for managing provenance metadata is a sophisticated adaptation of the HLF&C-platform for the business process of sharing storage resources.

Unlike public blockchain networks, which allow non-authenticated users to participate in their work, members of the HLF&C-network must be registered with Membership Service Provider (MSP), which, among other things, performs the functions of Certification Authority (CA).

2.4 Management of data access rights

In addition to the task of recording the immutable history of working with data in a distributed storage environment, the task of providing distributed management of access rights to data is set. For example, the owner of a data file (the user who created the data, the organization to which it belongs through its authorized representative, etc.) must be able to manage access to it for other users. Another example is when a cloud storage service grants access to data stored on it only to users from organizations that have paid for this storage service.

Detailed management of rights to initiate transactions related to operations with data in the distributed storage is based on the use of special Hyperledger Composer tools. The rights have to be described in acl-files located in the nodes of the blockchain network with the help of the special Access Control Language (ACL). Modification of the contents of the acl-file is also carried out by initiating the corresponding transaction by duly authorized users.

2.5 Metadata driven data management

From the general point of view, two approaches are possible. In the first approach, data management systems (DMS) manage data and use a blockchain simply as a distributed log (data driven data management). In the second approach, the metadata is written to the blockchain beforehand, and DMSs refer to the blockchain and perform the transactions recorded there (metadata driven data management). In the first case, the functionality of the blockchain system is very limited, it only provides a distributed ledger which is resistant to occasional or malicious attempts to modify the history of data in distributed storage. HLF&C-platform enables one to implement the second approach, which, in addition to simply maintaining the ledger, allows us to solve the problem of distributed data access management.

In our case, participants (in the sense of the HLF&C-platform) include persons (users and administrators of different levels) and storage providers. The main assets are data files. Their properties (attributes) are provenance metadata, including local file name in a storage, storage ID, creator ID, file owner ID, type of the file (primary, secondary or replica), etc. Another important type of the assets consists of the (local) storages constituting the distributed environment. We also defined user groups as assets, because we found it useful for managing data access rights. Finally, operations with files are treated as assets too because each operation actually comprises of a several atomic transactions. The basic operations can be of the following types: new file upload; file download; file copy within a storage; file deletion; file copy to another storage; file transfer to another storage.

The algorithm which we propose for recording transactions with provenance metadata and managing data access rights in the framework of ProvHL in a very simplified form reads as follows:

- the owner accesses the chaincode function, which, according to the acl-file (acl stands for access control language), allows the owner of the data to grant access rights to these data to another user or group of users;
- a user who has been granted access rights by the owner accesses the chaincode with a request to make an operation (ClientRequest transaction) with data (for example, file download, upload or copy);
- chaincode verifies that such a transaction complies with the rules defined in the acl-file and, if it does, sends a request to the HLF environment to complete the transaction;

- HLF performs transaction processing (transaction workflow: simulation and endorsements → ordering → validation → state updating);
- HLF sends a message (event) to the user about the successful transaction and its recording in the blockchain; the message also contains the transaction identification number;
- the user accesses the data management system (DMS) with a request to perform a data operation that contains the number of the corresponding transaction;
- DMS checks for a record of this transaction in the blockchain;
- if there is a record of the valid transaction, the DMS performs the required operation and, in turn, initiates a transaction record confirming that a data operation was performed (ServerResponse transaction).

As it can be seen, for each data operation, at least two transaction records are made in the blockchain: the first corresponds to the client request (ClientRequest), and the second corresponds to the server response (ServerResponse). In general case, an operation comprises of even more transactions. In the simplified description of the algorithm, some details specific to certain types of transactions are omitted for brevity. In particular, when the new file upload operation is performed, the creation of the new asset, that is the data file, is performed only after the actual upload of the file in the storage when DMS makes a ServerResponse transaction and turns the uploaded file into a fully valid asset. Together with the above-mentioned splitting of transactions into the client and server parts, this makes the level of correspondence between the history recorded in the blockchain and the real history of the data in the distributed storage practically acceptable.

2.6 Consensus

One of the first and most well known consensus algorithms is the Paxos algorithm [8]. This algorithm is not designed to work in distributed systems with possible Byzantine errors (malicious distortion of information by nodes). It is very difficult for understanding and implementing. In addition, Paxos uses an approach in which each node (consensus member) interacts with each, so the complexity of the decision is $O(n^2)$, where n is the number of the nodes. As a result, practical implementations have little to do with Paxos. Each implementation begins with Paxos, detects difficulties in its implementation, and then significantly changes the architecture. It takes a lot of time and leads to errors. Because of these problems, Paxos is not a good choice for building real systems. The Raft algorithm [9] implements the consensus by choosing a single leader, giving it full responsibility for managing the replicated log. The leader accepts requests from consensus members, copies them to other nodes, and tells the rest of the nodes when it is safe to use log entries in their replicated state machines. The idea of having a special leader simplifies the management of the replicated log. If the leader for some reason stops working, the procedure for selecting a new leader begins. However, Raft is also not designed to work in distributed systems in which the Byzantine type of error is possible.

The Practical Byzantine Fault Tolerance (PBFT) algorithm [10] was the first practical solution to achieve consensus in the face of Byzantine failures. It uses the concept of replicated state machine, and nodes in a PBFT system are sequentially ordered with one node being the leader and others referred to as backup nodes. All nodes in the system communicate with one another with the goal being that all honest nodes will come to an agreement of the state of the system using the majority rule. This algorithm requires $3n + 1$ replicas to be able to tolerate n failing nodes. Communication between nodes has two functions: nodes must prove that messages came from a specific peer node, and they must verify that the message was not modified during transmission. This approach imposes a low overhead on the performance of the HLF&C-platform ordering services which are consensus nodes in our case. However messaging overhead in the case of PBFT increases significantly as the number of the ordering nodes increase. However it remains acceptable for a couple of dozens of ordering service nodes (parties in a project using a distributed storage under the ProvHL management). Currently we consider PBFT as a most suitable distributed consensus algorithm.

3 Conclusion

In this paper, using the novel approach based on the integration of blockchain technology, smart contracts and metadata driven data management, the principles and algorithms of the new system, entitled ProvHL (Provenance HyperLedger), have been developed. This system is intended for fault-tolerant, safe and secure management of provenance metadata, as well as of access rights to data in distributed storages. The problems of optimal choice of the blockchain type for such a system, as well as the choice of the blockchain platform are studied. Namely, it is proposed to use the permissioned type of blockchain and the Hyperledger blockchain platform, on the basis of which the ProvHL system is implemented.

At present, a testbed has been created on the basis of SINP MSU, where a preliminary version of the ProvHL prototype is deployed to implement the developed principles and refine the algorithms of the system. The creation of ProvHL production level system will significantly improve the quality and reliability of the results obtained on the basis of processing and analysis of data in a distributed computer environment.

References

1. Berghöfer, T., et al.: Towards a model for computing in european astroparticle physics. arXiv preprint, arXiv:1512.00988 (2015)
2. Zafar F., et al.: Trustworthy Data: A Survey, Taxonomy and Future Trends of Secure Provenance Schemes. Journal of Network and Computer Applications **94**, 50-68 (2017)
3. da Cruz S. M. S., Campos M. L. M. and Mattoso M.: Towards a Taxonomy of Provenance in Scientific Workflow Management Systems. In: World Conference on Services-I, pp. 259-266. IEEE (2009)

4. Baliga A.: Understanding Blockchain Consensus Models. Tech. rep., Persistent Systems Ltd (2017)
5. Ramachandran A. and Kantarcioglu M.: SmartProvenance: A Distributed, Blockchain Based Data Provenance System. In: CODASPY'18: The 8th ACM Conference on Data and Application Security and Privacy. Tempe, AZ, USA (2018)
6. Liang X. et al.: Provchain: A Blockchain-based Data Provenance Architecture in Cloud Environment with Enhanced Privacy and Availability. In: Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, pp. 468-477. IEEE Press (2017)
7. Androulaki E.,et al.: Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. In: Proceedings of the Thirteenth EuroSys Conference, article No. 30. ACM, Porto, Portugal (2018)
8. Lamport, L.: The Part-Time Parliament. ACM Transactions on Computer Systems **16**(2) 133–169 (1998)
9. Ongaro, D. and Ousterhout J.K.: In search of an understandable consensus algorithm. In: USENIX Annual Technical Conference, pp. 305–319. USENIX Association (2014)
10. Castro M. and Liskov B.: Practical Byzantine Fault Tolerance. In: Proceedings of the 3rd Symposium on Operating Systems Design and Implementation, pp. 173-186 (1999)