International Journal of Networked and Distributed Computing
Vol. **6**(**4**); December (2018), *pp.* 210–215
DOI: 10.2991/ijndc.2018.6.4.3; ISSN 2211-7946
https://www.atlantis-press.com/journals/ijndc

The Dependability of Crypto Linked Off-chain File Systems in Backend Blockchain Analytics Engine

Jongho Seol[1,*], Abhilash Kancharla[1], Nicole Park[2], Nohpill Park[1], Indy Nohjin Park[3]

[1]*Computer Science Department, Oklahoma State University, Stillwater, OK 74078, USA*
[2]*Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213, USA*
[3]*Computer Science Department, Oklahoma City University, Oklahoma, OK 73106, USA*

ARTICLE INFO

Article History

Received 15 August 2018
Accepted 8 October 2018

Keywords

Blockchain analytics
Ethereum
crypto link
crypto speed
IPFS
Hadoop

ABSTRACT

This paper presents ways to assure the efficacy of managing crypto links and crypto speed with respect to the dependability of blockchain. The dependability in this work is defined with reference to crypto links in a proportional manner and crypto speed in an inversely proportional manner, respectively. Also, note that dependability is proportional to the cost such that the more the number of crypto links come in place, the higher its cost is. It is a standard and intuitive practice to partition a large output file into smaller pieces at a certain threshold size provided in the blockchain analytics engine (e.g., 250 KB/partition in Interplanetary File Systems (IPFS)/Hadoop). Crypto link is defined by the hash address (i.e., a pointer link) stored in the blockchain for a partition of an output file off from a backend blockchain analytics engine. IPFS manages a crypto link to every partition of a file of interest whereas the proposed crypto link scheme is at a reduced number in order to address the cost and performance issue such that one crypto link to the entire file and the links between partitions within a file are maintained by local chains of links. Crypto speed is defined by the rate of the number of crypto links over the total turnaround time to encrypt all the crypto links (hash addressing) per file. Thus, notice that without loss of generality, the crypto speed increases inverse-linearly along with the number of crypto links. A binomial-based dependability model can be expressed as a rather straightforward function of the total number of partitions and the number of crypto links as proposed in this work. It is demonstrated in this work how the number of crypto links and the crypto speed influence the dependability in a numerical simulation. Also an implementation of an integrated procedure of a blockchain analytics engine is demonstrated along with Hadoop and IPFS modules.

1. INTRODUCTION

Blockchain systems on peer-to-peer (P2P) network have been ever expanding exponentially since its inception and is running against its capacity limit forcing itself to consider off-chain options for storage and computation in a selective manner [1].

A P2P file sharing system, namely Interplanetary File Systems (IPFS) [2,3], provides a service to store huge files in off-chain local storage and partition them into pieces, for each of which with a hash code created and assigned to be posted back in blockchain for reference, referred to as crypto link. IPFS manages hash codes on blockchain in a versioned manner based on the content-addressed distributed P2P model [4]. It is investigated in this research how to assure the efficacy of managing crypto links along with the performance of the processes with respect to the dependability of blockchain. The dependability [5] in this research is defined with reference to crypto links in a proportional manner and its performance, as referred to as the crypto speed, in an inversely proportional manner, respectively. Also note that the dependability is proportional to the cost such that the more the number of crypto links come in place, the higher its cost is. It is a standard and intuitive practice to partition a large output file into smaller pieces at a certain threshold size provided in the blockchain analytics engine (e.g., 250 KB/partition in IPFS/Hadoop). Crypto link (CL) is defined by the hash address (i.e., a pointer link) stored in the blockchain for a partition of an output file off from a backend blockchain analytics engine. Note that IPFS manages a crypto link to every partition of a file of interest whereas the proposed crypto link scheme is at a reduced number in order to address the cost and performance issue such that one crypto link to the entire file and the links between partitions within a file are maintained by local chains of links. Crypto speed (CS) is defined by the rate of the number of crypto links over the total turnaround time to encrypt all the crypto links (hash addressing) per file. Thus, notice that without loss of generality, the crypto speed increases inverse-linearly along with the number of crypto links. A binomial-based [6] dependability model can be expressed as a rather straightforward function of the total number of partitions and the number of crypto links as proposed in this research. At the same time a cost model can be expressed as a function of the ratio of the number of

* *Corresponding author. Email: jongho@okstate.edu*

crypto links over the crypto speed. Putting the dependability and the cost models together, the dependability can be expressed as a function of crypto speed with a certain coefficient as well.

It is further demonstrated in this research that how the number of crypto links and the crypto speed at a given cost constraint and a few coefficient values influence the dependability in a numerical simulation. Also an implementation of an integrated procedure of a blockchain analytics engine is demonstrated along with Hadoop and IPFS [7]. The procedure follows the steps such that first, it downloads transactions and blocks off the blockchain; second, it submits to and runs the downloaded data through Hadoop and collects the output report into a file; third, it triggers IPFS to partition the expectedly huge file of the output into pieces and creates and assigns a hash code to each partition; then finally, it posts those has codes (i.e., crypto links) back into blockchain managed by Truffle and Ganache.

The paper is organized as follows. The proposed crypto link and crypto speed are defined and characterized with respect to the dependability in the following section. In the third section, the efficacy of the proposed models is demonstrated in a numerical manner and followed by a section with the backend analytics engine integrated with Hadoop [8,9] and IPFS. Then, the conclusion is given in the last section.

2. PRELIMINARIES AND THE PROPOSED DEPENDABILITY

In general, the dependability of crypto links in this work can be viewed as the probability for a crypto link to be operational at an instance of time under the risk of security, authenticity, network connection and operational failures, to mention a few main factors, and in other words and ultimately, it can be viewed as the rate of a file to be operational (i.e., error-free) at an instance of time under the risk as mentioned above.

Specifically, in this work, the dependability [10] is defined with reference to the crypto links in proportional manner but to the crypto speeds in inversely proportional manner. In order to model the dependability of the crypto links, a single node failure is assumed in P2P file sharing network as each single node is the primary and significant point of communication with other nodes in the network. In this context, the dependability is defined by the probability for a single node to be successfully operational (i.e., P_{nf}) as expressed in Equation (1) with respect to four known main variables, namely P_{ND}, P_{NU}, P_{NCF}, and P_{NOF}, as a series product. Those variables are responsible for and influence the security (i.e., P_{ND}), authenticity (i.e., P_{NU}), network connection (i.e., P_{NCF}), and reliability (i.e., P_{NOF}) in a composite manner.

$$P_{nf} = (1 - P_{ND})\,(1 - P_{NU})\,(1 - P_{NCF})\,(1 - P_{NOF}) \tag{1}$$

where,

P_{nf}: Probability of a single node success, $0 \le P_{nf} \le 1$.

P_{ND}: Node defect rate by attack, $0 \le P_{ND} \le 1$.

P_{NU}: Node undefined rate by peers, $0 \le P_{NU} \le 1$.

P_{NCF}: Node connection failure rate, $0 \le P_{NCF} \le 1$.

P_{NOF}: Node operation failure rate, $0 \le P_{NOF} \le 1$.

Based on Equation (1), the probability for a crypto link failure can be expressed as follows (Equation 2).

$$P_{cf} = (1 - P_{nf}) \times f \tag{2}$$

where,

P_{cf}: Probability of a crypto-linked partition failure, $0 \le P_{cf} \le 1$.

P_{nf}: Probability of a single node success.

f: Failure rate of a single partition in a file, $0 \le f \le 1$.

Based on the definition of the dependability of crypto links, a dependability model for a file is proposed with respect to the total number of partitions and the number of crypto links as follows.

$$P_d = \binom{N}{\mathrm{CL}} \sum_{\mathrm{CL}}^{N} (P_{cf})^{\mathrm{CL}} (1 - P_{cf})^{N-\mathrm{CL}} \tag{3}$$

where,

P_{cf}: Single crypto-linked partition failure rate.

N: Total number of partitions in a file.

CL: Total number of partitions crypto-linked in a file.

P_d: Dependability of a file.

In Equation (3), it is assumed that the file is partitioned from CL = 1 up to 100 uniform-sized pieces for N = 100, and P_{cf} is assumed to be 0.25, 0.5 and 0.75 in Figs. 1–3, respectively.

In Fig. 1, it is shown how dependability is affected by increasing the number of partitions (N) when P_{cf} = 0.25 and CL = 0, 20,

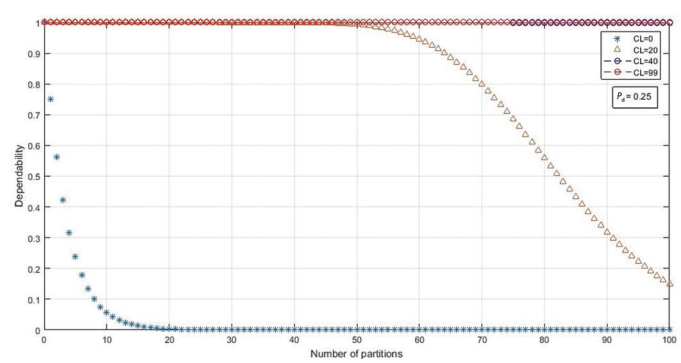

Figure 1 | Dependability versus number of partitions in P_{cf} = 0.25

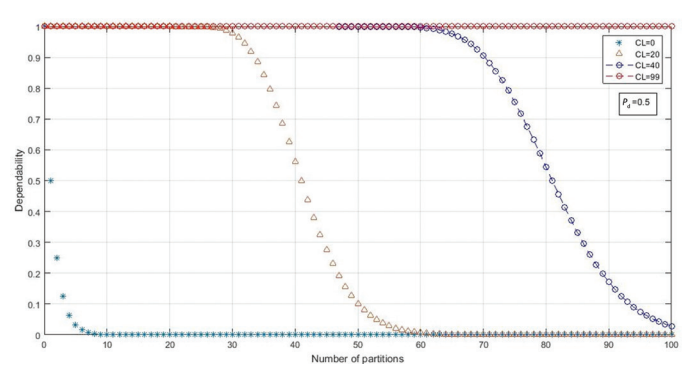

Figure 2 | Dependability versus number of partitions in P_{cf} = 0.5

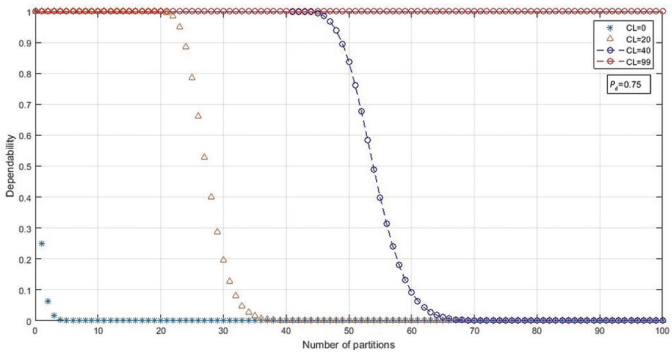

Figure 3 | Dependability versus number of partitions in $P_{cf} = 0.75$

40 and 100, respectively. It is observed that when CL = 40 and 100, their dependabilities are maintained at 1.0 throughout regardless of the N-values. If CL drops down to 20, its dependability is still maintained yet starts to slowly drop past $N = 50$. When CL = 0, it is observed that its dependability is dropping sharply in an exponential manner up to CL = 20 then maintained flat at 0 throughout for the rest of the range through $N = 100$.

In Fig. 2, $P_{cf} = 0.5$ raised from $P_{cf} = 0.25$ and it is observed that the dependabilities of CL = 40 and lower exhibit earlier dropping points by N-values than when $P_{cf} = 0.25$ such that when CL = 40, it starts to drop past $N = 60$ than maintaining flat at 1.0; and when CL = 20, it starts to drop past $N = 30$ earlier than 50. It is demonstrated that the same trends continue through as P_{cf} continues to rise as shown in Fig. 3.

As expected, the results as demonstrated in Fig. 1 through Fig. 3, it is evident that incorporation of crypto links (CL) definitely ensures the dependability of the off-chain file system in the backend blockchain analytics engine, and further it is demonstrated the dependability is quite sensitive to the failure rate of the CL's.

The dependability P_d from Equation (3) can be rewritten as follows:

$$P_d = \binom{N}{\mathrm{CL}} \sum_{\mathrm{CL}}^{N} (P_{cf})^{\mathrm{CL}} (1 - P_{cf})^{N-\mathrm{CL}} \tag{4}$$

$$= \binom{N}{\mathrm{CL}} \sum_{\mathrm{CL}}^{N} \left(\frac{P_{cf}}{1 - P_{cf}}\right)^{\mathrm{CL}} (1 - P_{cf})^{N} \tag{5}$$

$$= \binom{N}{\mathrm{CL}} (1 - P_{cf})^{N} \sum_{\mathrm{CL}}^{N} \left(\frac{P_{cf}}{1 - P_{cf}}\right)^{\mathrm{CL}} \tag{6}$$

$$= \binom{N}{\mathrm{CL}} (1 - P_{cf})^{N} \left(\begin{array}{l} \left(\frac{P_{cf}}{1 - P_{cf}}\right)^{0} + \left(\frac{P_{cf}}{1 - P_{cf}}\right)^{1} + \ldots \\ + \left(\frac{P_{cf}}{1 - P_{cf}}\right)^{N-1} + \left(\frac{P_{cf}}{1 - P_{cf}}\right)^{N} \end{array} \right) \tag{7}$$

$$= \binom{N}{\mathrm{CL}} \frac{(1 - P_{cf})^{N} \left(1 - \left(\frac{P_{cf}}{1 - P_{cf}}\right)^{N}\right)}{1 - \left(\frac{P_{cf}}{1 - P_{cf}}\right)} \tag{8}$$

3. NUMERICAL STUDY ON THE PROPOSED DEPENDABILITY

The derivations from the previous section help reveal the more detailed functional relationships of the dependability (P_d) versus a series of variables such as N, CL and P_{cf}.

Figures 4 and 5 show the trend of the dependability (P_d) versus total number of partitions (N) and the number of crypto links (CL) at a failure rate of a single crypto-linked partition (P_{cf}).

In Fig. 4, the joint effect of N and CL is plotted varying CL at a fixed $P_{cf} = 0.5$ (note that this is an arbitrary value just for a simulation purpose) and it is observed that the more CL incorporated the higher dependability achieved as expected. Without loss of intuition, it is further expected that the lower P_{cf} turns, the higher the dependability goes.

In Fig. 5, CL is fixed at 5 instead (note that this also is an arbitrary value for a simulation purpose and also note that when CL = 5 is set implies N is at least 5 as well so as the plot starts from $N = 5$) and then the trend of the dependability (P_d) versus total number of partitions (N) and the failure rate of a single crypto-linked partition (P_{cf}). It is observed that the lower P_{cf} turns, the higher P_d goes. An interesting observation to note is that the impact of P_{cf} on P_d is less significant as P_{cf} approaches either to 1.0 or 0.0 as shown in Fig. 5, which leaves a question for a physical validation.

In Figs. 6 and 7, the trend of the dependability (P_d) versus the number of crypto links (CL) and the varying total number of

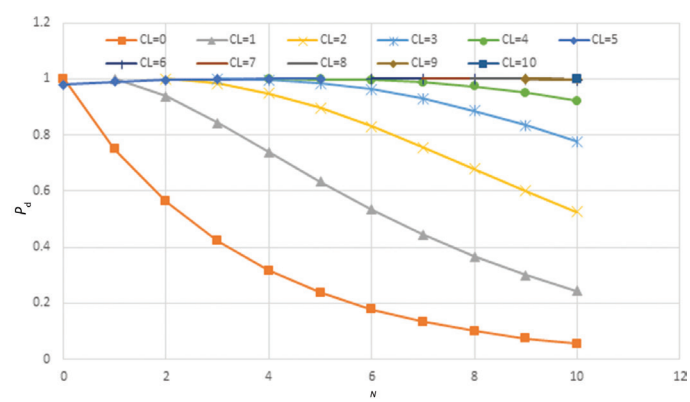

Figure 4 | A graph for P_d versus N with varying CL and fixed P_{cf}. $P_{cf} = 0.5$

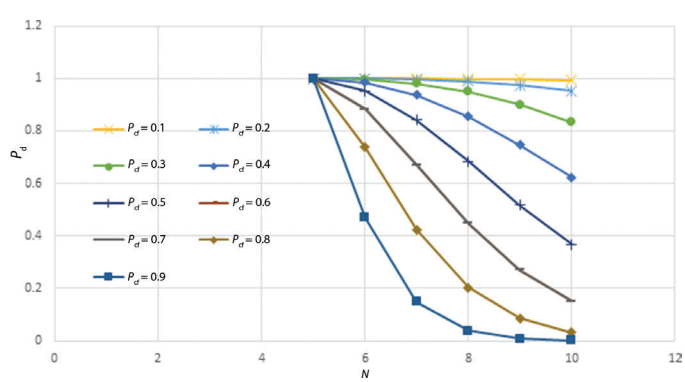

Figure 5 | A graph for P_d versus N with varying P_{cf} and fixed CL. CL = 5

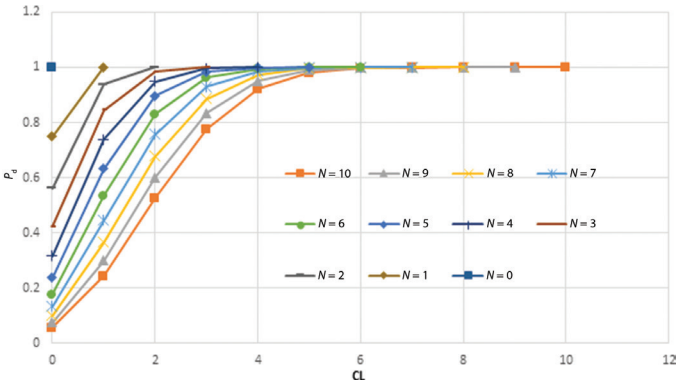

Figure 6 | A graph for P_d versus CL with varying N and fixed P_{cf}. $P_{cf} = 0.25$

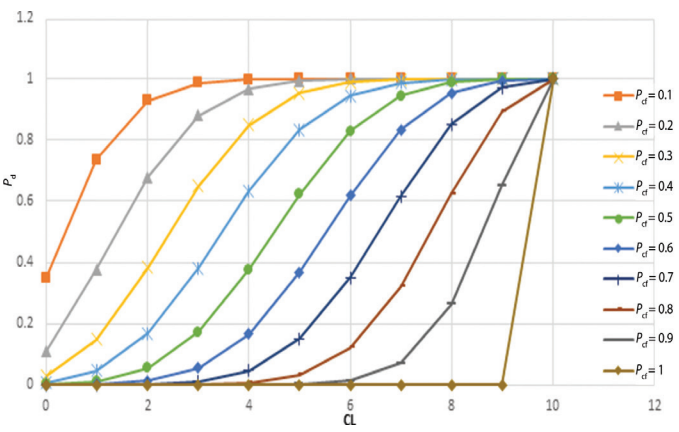

Figure 7 | A graph for P_d versus CL with varying P_{cf} and fixed N. $N = 10$

partitions (N) and at a failure rate of a single crypto-linked partition (P_{cf}), and versus the number of crypto links (CL) and the varying failure rate of a single crypto-linked partition (P_{cf}) at a total number of partitions (N), respectively.

In Fig. 6, notice that N is varying instead of CL in order to observe the impact of N on P_d at a fixed P_{cf} (0.25). It is observed that at a CL set, the impact of P_{cf} turns more significant as N goes down while P_d picks up. Also it is observed that beyond a CL value (e.g., 5–6) P_d gets saturated provided the set of values of other variables. Without loss of intuition, it can be further expected that as P_{cf} goes lower the saturation point will be formed earlier than when P_{cf} is higher.

It is noticed in Fig. 7 that the trends of picking up P_d as both CL and N pick up is quite steady and gradual indicating that the impact of CL on P_d provided other variables is quite consistent no matter how. Yet, it is again observed that the impact of P_{cf} on P_d is less significant as P_{cf} approaches either close to 1.0 or 0.0, which also leaves a question for a physical validation.

In Figs. 8 and 9, the trend of P_d versus P_{cf} and the varying CL and at an N, and versus P_{cf} and the varying N and at a CL, respectively.

4. BACKEND BLOCKCHAIN ANALYTICS ENGINE

In order to demonstrate a CL implementation, a blockchain analytics engine has been developed and is presented in this section.

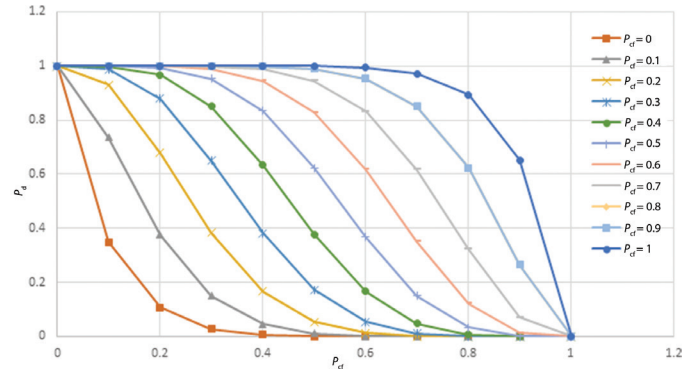

Figure 8 | A graph for P_d versus P_{cf} with varying CL and fixed N. $N = 10$

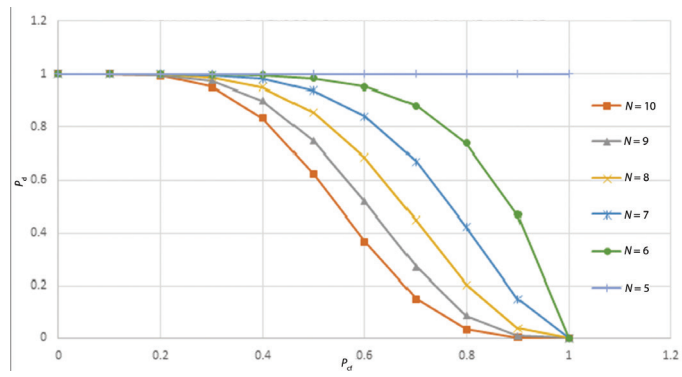

Figure 9 | A graph for P_d versus P_{cf} with varying N and fixed CL. CL = 5

A bash shell script has been written along in the truffle development mode on Ubuntu OS (16.04).

The procedure is such that it starts downloading the blockchain transactions and blocks from https://etherscan.io/blocks and obtains the latest height page first. Each height of block contains several transactions [4] which is downloadable to the local hard disk storage. When the downloading is completed, it starts performing analytics on the blocks and transactions through the Apache Hadoop map reduced scheme and then generates an output report on the analytics results. In this particular work, a WordCount java program was used as a benchmark.

Next, it processes the report file in order to firstly partition it and then to create and assign a hash code to each partition by IPFS which coordinates to store the hash codes both in the local storage and in the P2P IPFS network. Note that when an account is connected to IPFS network, it is notified that how many peers are connecting through IPFS. Then, those IPFS-generated hash codes are posted back on the blockchain through Node.js coordinated by truffle [11]. A hash code is posted back on the blockchain by executing transactions of storing and posting functions in web3.js protocol in Ethereum. In Fig. 10, the flow of the blockchain analytics engine is shown.

5. CONCLUSION

This paper has presented a study on the dependability of a blockchain analytics engine with respect to crypto links, crypto speed.

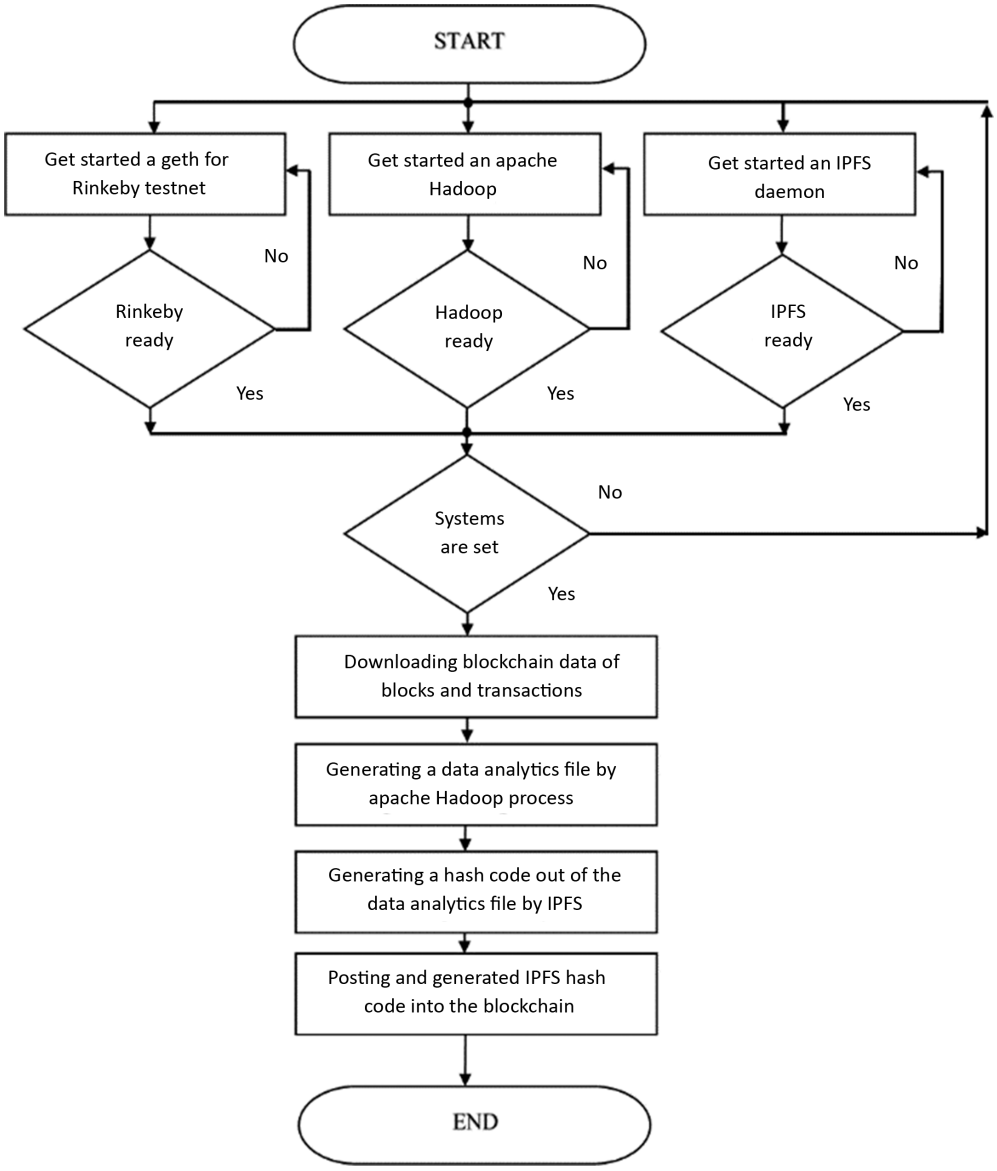

Figure 10 | The proposed flow of blockchain analytics engine by Apache Hadoop Map reduced scheme through IPFS distributed file systems on the Ethereum blockchain Rinkeby testnet

Also an implementation of the proposed blockchain analytics engine is presented. The engine downloads blockchain data, performs analytics and then post the resulting hash code(s) of the output file back into the blockchain by using the Hadoop/IPFS. A blockchain engine has been tested on Ethereum Test network Rinkeby. The engine has fulfilled an effective system to perform automated blockchain analytics by using bash shell script for downloading blockchain data and posting crypto links back into blockchain by using IPFS, Hadoop, Truffle, Ganache (private network based) or Rinkeby Testnet (Ethereum test network based) in an integrated manner.

REFERENCES

[1] V. Buterin, A Next Generation Smart Contract & Decentralized Application Platform, Ethereum White Paper, 2014.

[2] J. Benet, IPFS – Contents Addressed, Versioned, P2P File System (Draft 3), White Paper, 2015. Available from: https://github.com/ipfs/papers/blob/master/ipfs-cap2pfs/ipfs-p2p-file-system.pdf

[3] T. Renner, J. Muller, O. Kao, Endolith: A Blockchain-based Framework to Enhance Data Retention in Cloud Storages, 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), March 21–23, IEEE, Cambridge, UK, 2018.

[4] S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2009.

[5] B.W. Johnson, Design and Analysis of Fault Tolerant Digital Systems, Addison-Wesley Publishing Company, Reading, MA, USA, 1989.

[6] S.M. Ross, Introduction to Probability Models, 11th ed., Elsevier Academic Press, Taiwan, 2014.

[7] D. Vegros, J. Saenz, Peer-To-Peer Networks and Internet Policies, Nova Science Publishers, Inc., NY, USA, 2010.

[8] P. Pirzadeh, M. Carey, T. Westmann, A Performance Study of Big Data Analytics Platforms, 2017 IEEE International Conference on Big Data, December 11–14, IEEE, Boston, MA, USA, 2017.
[9] M. Sogodekar, S. Pandey, I. Tupkari, A. Manekar, Big Data Analytics: Hadoop and Tools, 2016 IEEE Bombay Section Symposium, December 21–22, IEEE, Baramati, India, 2016.
[10] U.M. Borghoff, Catalogue of Distributed File/Operating Systems, Springer-Verlag, Berlin Heidelberg, 1992.
[11] https://github.com/ethereum/web3.js/
[12] I. Koren, C. Mani Krishna, Fault-Tolerant Systems, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2007.