Vol. 5 No.1
ISBN: 978-623-7144-28-1

MEMORY ANALYSIS FOR IPFS IMPLEMENTATION ON ETHEREUM SMART CONTRACT

Tiara Sabrina [1*], Adityas Widjajarto [2], Avon Budiyono [3]

[1] School of Industrial and System Engineering, Telkom University
[2] School of Industrial and System Engineering, Telkom University
[3] School of Industrial and System Engineering, Telkom University

Abstract. Smart contract is an agreement between two entities established in the program code. All Smart contract transactions are stored on the Blockchain. But storing large data on the Blockchain is expensive, so many developers are currently creating a DApp (Decentralized Application) that integrates IPFS on smart contract Ethereum. Files will be stored on IPFS while the Blockchain only stores hash files from IPFS to access them again. Blockchain & IPFS are distributed peer-to-peer technologies for storing and distributing digital data supported by the confidentiality, integrity and authenticity of the data. The study was conducted to measure memory usage to run the DApp web that integrates IPFS on smart contract Ethereum and find out the effect of the file size uploaded via the web DApp and the number of nodes connected in the network. The memory usage test results will be used as a standard for the memory capacity planning to implement a DApp web system that integrates IPFS on smart contract Ethereum in an organization. Based on the research result, to run a web DApp that integrates IPFS on a smart contract requires 774MB of memory. The result proves that IPFS is suitable for handling large files. The efficiency of DApp's web performance that integrates IPFS on the smart contract Ethereum are obtained by a small file size and a large number of nodes connected in a network. The smaller the file size, the less memory usage. The more nodes that are connected in the network, the less memory usage.

Keywords: Decentralized Application, IPFS, Smart Contract Ethereum, Blockchain, Memory Usage.

1. INTRODUCTION

At this time almost all business activities are carried out through written contracts. A contract is a binding agreement/written agreement that defines the rights and responsibilities of each party. A written contract has several weaknesses where the contract can be lost and damaged, one party broke the agreement and a written contract consumes more costs and time [8]. The Blockchain produce an invention, Ethereum with its feature is smart contract. By this smart contract, developers can write digital contracts with program code. Smart contract transactions has high credibility because the contract was made cannot be tracked and changed. Smart Contract is the solution to the problem of a written contract (Atzei dkk, 2017).

The smart contract and it transactions are stored on the Blockchain. The Blockchain is a distributed peer-to-peer technology for storing and distributing digital data such as

[*] Corresponding Author, Email: tiarasbrina@gmail.com

cryptocurrency. Smart contracts, property, stocks, files, or anything else valuable goods must have the confideality, integrity and authenticity of data (Rajalakshmi, 2018:1437).1

However, Blockchain is not suitable for storing large amounts of data, so many developers are currently creating a DApp Web that integrates IPFS on Smart contract Ethereum. Files will be stored on IPFS while the Blockchain only stores hash files from IPFS to has chance to access it again[11]. As the integration of IPFS on Smart contract, the gas costs provide for making contracts and transactions cheaper than to without the integration of IPFS on Smart contracts [12]. Gas is a small part of Ether that is used to pay miners for mining. Mining is the process of storing transactions in a block and adding blocks to the blockchain (Sinha & Kaul, 2018).

In this study a web-based DApp will be developed to integrates IPFS on the Smart contract Ethereum. To run Web DApp on operating system, requires memory resources to DApp web can operate properly. Research is conducted to measure the consumption of RAM and CPU by the web DApp. Some parameters that affect memory usage are the file size to be uploaded and the number of nodes interact with the system. The amount of memory usage by the DApp web will be a criteria in allocating memory according to the needs of user activities and available budgets. Moreover, market presents various specification hardware to meet user needs. The better the quality of the hardware the more expensive the device is.

2. RELATED WORK

In this section, we introduce some technologies related to our research.

2.1. Smart Contract Ethereum

Ethereum is a Blockchain based platform built specifically to create and run smart contracts. Smart contract is a feature of Ethereum to write a digital contract through program code written in EVM bytecode language. Developers generally write smart contract using Solidity than EVM bytecode language. Solidity is a Javascript-like programming language that is used to create Smart contracts. The code was created will be compiled into byte-code EVM so that it can be executed in the Ethereum network(Atzei dkk, 2017).

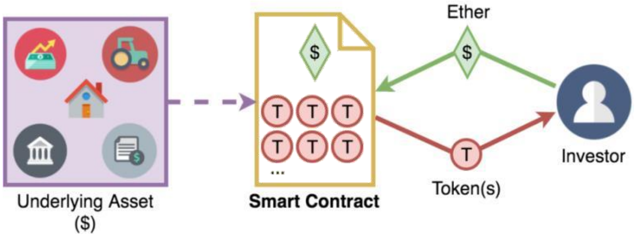

Fig. 1: Smart Contract.

Smart contracts and transactions are created, stored and executed on Blockchain in each network node to speed up the communication process and prevent data loss conflicts(Dannen, 2017). Smart contract execution is triggered by a transaction and each transaction consumes Ether. There are three reasons in transactions for users in the Ethereum network, just as : (i) making new contracts;

(ii) carry out the contract function; (iii) transfer ethers to contracts or to other users (Atzei dkk, 2017). When someone makes a transaction, a miner is required to do the mining process because a distributed system does not have a single owner (Wang, 2018).

2.2. Blockchain

Blockchain technology are useful for storing and distributing digital data with the confidentiality, integrity and authenticity of data(Rajalakshmi, 2018:1437). Blockchains are usually used to store various records namely cryptocurrency transactions, smart contracts, certificates and digital signatures(Grech & Camilleri, 2017).

Blockchain uses data block-chain structures to store and verify data, using concencus distribution node algorithms to update transaction data, using cryptography asymmetric to ensure security of access and data transmission (Chen dkk, 2017).

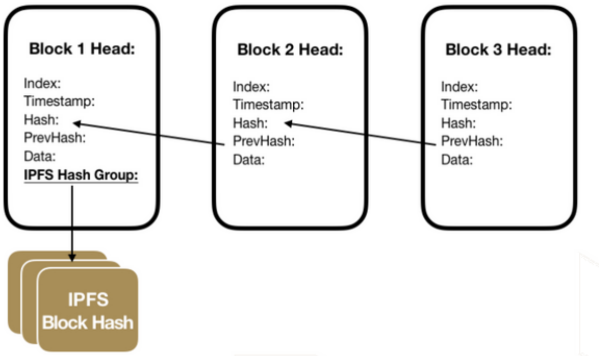

Fig. 2: Blockchain Structure.

All participants (nodes) on the Blockchain network have the same copy of the database. Each block on the blockchain consists of a list of transactions. Chains will grow when new blocks are added continuously. Each block has a timestamp, its own hash cryptography and a hash reference from the previous block listed in the block header as seen in the Fig.2 about Blockchain Structure (Zheng, 2017). Cryptographic hash algorithms can make it easier to verify that transaction data in each block cannot be changed, and blocks that are linked in the Blockchain cannot be broken (Wang dkk, 2016:4).

If IPFS is integrated with a smart contract, the whole file will be stored on IPFS while the Blockchain only stores the hash of the IPFS file. The hash files stored on the Blockchain are links to files stored on IPFS, to be accessed later (Rajalaksmi dkk, 2018).

Vol. 5 No.1
ISBN: 978-623-7144-28-1

2.3. IPFS

IPFS is an open source project, originally designed by Juan Benet and then developed since 2014 by Protocol Labs, a laboratory for research and development of network protocols. IPFS is a distributed file system on peer-to-peer (P2P) network, functioned to connected all computer set to the same file system. IPFS combines the former succesfeull idea from peer-to-peer such as DHT, BitTorrent, Git, dan SFS (Bennet, 2014). Each node will identify with hash cryptography from public key which was made by Crypto S/Kademlia (Wennergen, 2018). IPFS network using routing system based on DHT (Distributed Hash Table) that can be accomodated millions of nodes (Chen dkk, 2017). IPFS distributed file through a protocol names Bitswaps. The file distribution conducted by changing the block between peer. Bitswap will record transfer history on Bitswap Ledger. The more byte notes in the Biswap Ledger, the more the node will be trusted by Bitswap (Bennet, 2014).

2.4. Distributed Network

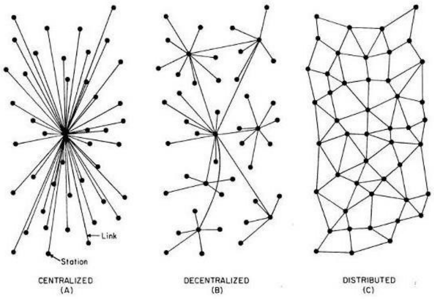

Fig. 3:Computer Network Types.

Distributed networks connect all nodes without involving a server. All nodes have the same position to provide information or communicate with other nodes. The more connections maintained by a node, the more reliable the communication and distribution of data with the entire network. In distributed networks, nodes are connected peer-to-peer. (Port, 2019)

2.5. RAM

RAM is divided into two types, namely Static RAM (SRAM) and Dynamic RAM (DRAM). SRAM is used to store cache that is inside or outside the CPU chip. DRAM is used for main memory (physical RAM) and frame buffer from the graphic system. SRAM usually only has a few megabytes, while DRAM can reach hundreds or thousands of megabytes [2]. Main memory (physical RAM) is a temporary storage device that holds data in the programs when the processor executes the program.

Vol. 5 No.1
ISBN: 978-623-7144-28-1

3. PROPOSED ARCHITECTURE

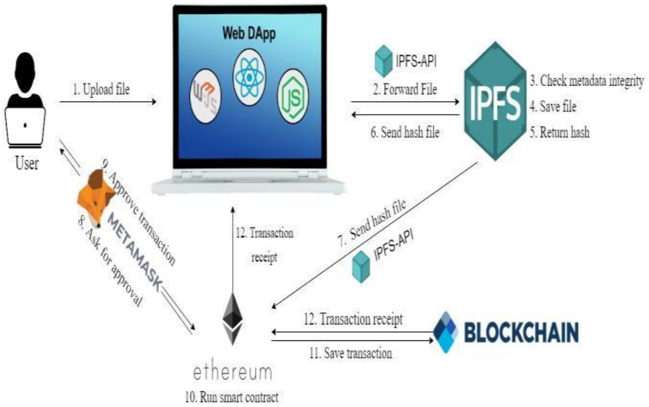

Fig. 4: Proposed Architecture.

The system built is a simple web Dapp that integrates IPFS (Interplanetary File System) on the smart contract Ethereum. DApp web system is built as a media to store data with a contract. Some applications are used in making and running web DApp will be explained in the table below.

Table. 1: Software Istrument.

Type	Software	Version
Operating System	Linux Ubuntu	16.04 LTS
Main Software	IPFS	go-ipfs v0.4.18
	Go-lang	go1.12
	NPM	6.4.1
	Node JS	V10.15.3
	Create-react-app	3.0.0
Dependency NPM	Bootstrap	3.3.7
	fs-extra	7.0.1
	Ipfs-api	26.1.2
	react-script	3.0.0
	*Web*3	1.0.0-beta.34
Third party Software	Htop	2.6.3
	Google Chrome	74.0.3729.131 (64-*bit*)
	Metamask	6.4.1
	Sublime Text	3.2.1, Build 3207

The DApp Web is built on the Ubuntu 16.04 LTS Linux operating system. The first step in building a system is to create a Metamask account to get a wallet and manage transactions from Ethereum Blockchain. Then create and deploy contracts using Remix from the Metamask account.

Remix is an open source web application that helps in writing smart contracts in the programming language Solidity. The next step is to build the DApp web. The applications that help in developing web DApp are Node.js, and NPM dependencies such as create-react-app, react-bootstrap, fs-extra, ipfs-api and web3. Then integrate DApp web with IPFS and Smart Contract via ipfs-api and web3.

The user opens the DApp web through the browser and uploads the file via the DApp web. The browser will convert the file into a buffer. Then the Web DApp will send files to IPFS via IPFS-api. IPFS will check the integrity of the file metadata by IPNS, save the file on the IPFS by Bitswap and return the hash from the file by IPNS. IPFS will send a hash from the file to the web DApp and Ethereum via IPFS-api. Before running the smart contract Ethereum will ask for transaction confirmation through Metamask. The user will provide confirmation and Metamask will forward it to Ethereum. With the user's agreement, the smart contract will be exploited by Ethereum with input data in the form of an IPFS hash. Smart contract transactions that have file hash data will be stored on the Blockchain. Blockchain will provide transaction recipt to the DApp web via web3. Transaction receipt is smart contract transaction information such as transaction hash, nonce, hash block, block number, gas used, gas price, input and others.

4. RESEARCH METHOD

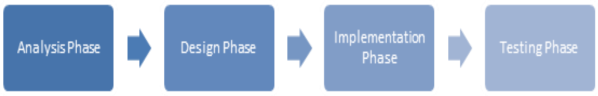

Fig. 5: Research Method.

There are 4 phases in this research : the analysis phase, the design phase, the implementation phase and the testing phase. The analysis phase is the hardware and software analysis phase that will be used to build the DApp web that integrates IPFS on the smart contract Ethereum. The design phase is to describe the relationships between applications and the DApp web workflow that integrates IPFS on Smart Contract Ethereum.

The implementation phase is the stage of building a web-based DApp that integrates IPFS on the Smart contract Ethereums in accordance with the system architecture that has been designed.

The testing phase is carried out on three laptops that are connected and run an integrated DApp web with IPFS and Smart Contract Ethereum through the process of uploading files. The purpose of testing is to analyze the effect of uploaded file size and number of node interactions on memory usage when running the DApp web. There are three test scenarios in this study, namely (1) one node uploads the file via the DApp web, (2) two nodes upload files via the DApp web at the same time and (3) three nodes upload files via the DApp web at the same time. Memory usage of the three scenarios is the result of the calculation of the memory usage of all applications, namely Metamask, Web DApp, IPFS, NPM, and Node.js. The size of memory usage is obtained by monitoring memory usage through the Htop application.

5. RESULT AND DISCUSSION

The first scenario is the process of uploading files carried out by one node that is not connected to the other nodes. The second scenario is the process of uploading files that are carried out by one node connected to another node. The third scenario is the process of uploading files carried out by one node connected to the other two nodes. All test results are measured in megabytes.

Table. 2: Memory Usage Scenario 1.

Process Size	Metamask	*Web*DApp	IPFS	NPM	*Node*.js	Total
idle	285	128	34	41	302	791
10MB	335	176	53	41	302	907
50MB	364	205	126	41	303	1041
100MB	392	237	241	41	303	1215
200MB	406	331	376	41	303	1458
300MB	420	429	446	41	303	1639
400MB	427	521	512	41	303	1805
500MB	473	611	545	41	303	1973
600MB	492	707	545	41	303	2089
700MB	527	807	545	41	303	2224
800MB	530	900	545	41	303	2320
900MB	557	997	545	41	303	2443
1GB	567	1084	536	41	303	2532
1.1GB	585	1187	569	41	304	2686
1.2GB	617	1209	588	41	303	2758
1.3GB	650	1381	606	41	304	2982
1.4GB	665	1473	617	41	304	3101
Average	488	728	437	41	303	1998

Table. 3: Memory Usage Scenario 2

Proses Size	Metamask	*Web*DApp	IPFS	NPM	*Node*.js	Total
idle	295	126	45	41	305	813
10MB	328	145	63	41	306	883
50MB	354	185	136	41	305	1021
100MB	369	238	234	41	306	1188
200MB	409	327	323	41	305	1406
300MB	411	428	360	41	306	1547
400MB	434	522	360	41	305	1663
500MB	465	619	408	41	306	1840
600MB	482	709	415	41	305	1953
700MB	492	807	412	41	306	2058
800MB	522	902	415	41	306	2186

Vol. 5 No.1
ISBN: 978-623-7144-28-1

900MB	545	996	415	41	306	2303
1GB	559	1090	405	41	306	2402
1.1GB	581	1187	406	41	306	2521
1.2GB	588	1281	406	41	306	2622
1.3GB	613	1377	462	41	306	2799
1.4GB	647	1484	470	41	306	2948
Average	476	731	337	41	306	1891

The increase of RAM usage in web DApp and IPFS is influenced by the size of the file size. The larger file size uploaded via web DApp , the more memory is needed to run the entire work process from web DApp. The most consuming memory is subprocess of converting files into buffers.

The larger file size uploaded via web DApp , the more RAM is needed by IPFS to perform all processes such as checking integrity in the metadata of the file to be stored, storing files in IPFS blocks, generating hashes from files and giving hash files to web DApp. However, the upload process up to 500 MB, as the size of the file increases, memory usage by IPFS tends to be stable, not increasing significantly. The result proves that IPFS is suitable for handling large files.The increase of RAM usage in Metamask is not influenced by the size of the file uploaded via the DApp web. The increase in RAM usage in Metamask is caused by an increasing quantity of transactions. With the increase in transactions with Metamask, the smart contract transaction data managed and stored by Metamask also increased.

The size of the file does not affect NPM. NPM is a package manager for building interfaces from DApp. So when uploading files, there is no change in memory usage by NPM. The size of the file also has no effect on Node.js. Node.js is an runtime enviroment for NPM. So when uploading files, there is no change in memory usage by Node.js.

The picture below is the average total memory usage from three scenario for each file upload test of 10MB-1.4GB. It can be concluded that the larger file size uploaded via web DApp , the more memory needed to run the web DApp that integrates IPFS on the smart contract Ethereum.

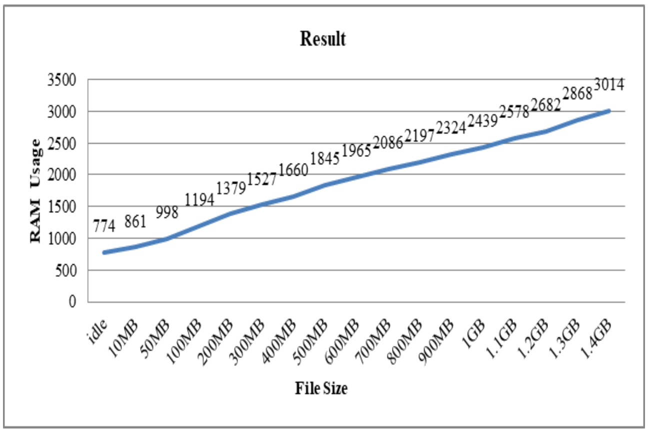

Fig. 6: The Average of Memory Usage From 3 Scenarios.

From the overall results of the memory usage above, forecasting can be done using the Trend Least Square method quoted from Iskandar in 2014.

Table. 5: Forecasting Process.

N	y	x	x.y	x^2
100 MB	1194	-13	-15520.96	169
200 MB	1379	-11	-15166.38	121
300 MB	1527	-9	-13740.97	81
400 MB	1660	-7	-11620.84	49
500 MB	1845	-5	-9225.41	25
600 MB	1965	-3	-5895.82	9
700 MB	2086	-1	-2085.68	1
800 MB	2197	1	2196.74	1
900 MB	2324	3	6971.13	9
1 GB	2439	5	12195.86	25
1.1 GB	2578	7	18044.85	49
1.2 GB	2682	9	24141.52	81
1.3 GB	2868	11	31552.97	121
1.4 GB	3014	13	39188.44	169
14	29758	0	61035.44	910

With the trend equation:

$$Y = a + bx,$$

To find a and b using the formula below:

$$a = \frac{\sum Y}{N}$$

$$b = \frac{\sum XY}{X2}$$

After the calculation is done, the following equation is obtained :

$$y = 2143.13 + 65.86x$$

Table. 6: Forecasting Result.

File Size (X)	**RAM Usage (Y)**	**File Size (X)**	**RAM Usage (Y)**
1500 MB	3132 MB	3300 MB	5503 MB
1600 MB	3266 MB	3400 MB	5635 MB
1700 MB	3386 MB	3500 MB	5766 MB
1800 MB	3518 MB	3600 MB	5898 MB
1900 MB	3643 MB	3700 MB	6029 MB

2000 MB	3774 MB	3800 MB	6161 MB
2100 MB	3913 MB	3900 MB	6294 MB
2200 MB	4046 MB	4000 MB	6426 MB
2300 MB	4183 MB	4100 MB	6559 MB
2400 MB	4316 MB	4200 MB	6692 MB
2500 MB	4448 MB	4300 MB	6823 MB
2600 MB	4581 MB	4400 MB	6956 MB
2700 MB	4705 MB	4500 MB	7087 MB
2800 MB	4836 MB	4600 MB	7219 MB
2900 MB	4971 MB	4700 MB	7351 MB
3000 MB	5103 MB	4800 MB	7484 MB
3100 MB	5238 MB	4900 MB	7616 MB
3200 MB	5370 MB	5000 MB	7748 MB

Table VI above is the forecasting result performed on memory usage when uploading files of a certain size using the Trend Least Square method. Forecasting results can be applied as a reference in allocating memory to run a web DApp system that implements IPFS on Smart Contract Ethereum.

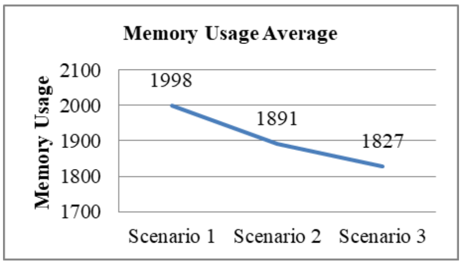

Fig. 7: The Average of Memory Usage Total From 3 Scenarios.

The first scenario is the process of uploading files carried out by one node that is not connected to the other nodes. The second scenario is the process of uploading files that are carried out by one node connected to another node. The third scenario is the process of uploading files carried out by one node connected to the other two nodes. The average total RAM usage in the first scenario is 1998 MB. The average of the total RAM usage in the second scenario is 1891MB. The average of the total RAM usage in the third scenario which is 1827MB. A decrease in memory usage occurs in each scenario. Web DApp that was created is a open distributed system. When there is data communication as in the file upload process, data will be distributed to all connected nodes. The more nodes are connected in network , the less data will be stored in a node so the memory resources used by each node are also fewer.

REFERENCE

Atzei N., Bartoletti M., & Cimoli, T. (2017) : A Survey of Attacks on Ethereum Smart contracts.

International Conference on Principles of Security and Trust hal. 164-186.

Bennet, J. (2014) : IPFS - Content Addressed, Versioned, P2P File System(DRAFT 3). ArXiv.

Bryant, R.E & O'Hallaron, D.R. (2013) : Computer Systems: A ProgRAMmer's Perspective, Second Edition. United States of America : Pearson.

Chen, Yongle, dkk. (2017) : An Improved P2P File System Scheme based on IPFS and Blockchain.

IEEE International Conference on Big Data.

Dannen, C. (2017) : Introducing Ethereum and Solidity:Foundations of Cryptocurrency and Blockchain ProgRAMming for Beginners. New York : Apress.

Grech, A. dan Camilleri, A. F. (2017) : Blockchain in Education. In Inamorato dos Santos, A. (ed.) EUR 28778 EN; doi:10.2760/60649.

Kenneth, H. (2018) : Ethereum Test Network , https://medium.com/coinmonks/ ethereum-test-

network-21baa86072fa, Accessed 17 June 2019.

Parizi, dkk. (2018) : Empirical Vulnerability Analysis of Automated Smart contracts Security Testing on Blockchains. Centre for Advanced Studies Conference.

Port, T., (2019) : Centralized vs Decentralized vs Distributed Networks + Blockchain, Accessed 24 June 2019.

Rajalakshmi, A., dkk. (2018) : A Blockchain and IPFS based Framework for Secure Research Record Keeping. International Journal of Pure and Applied Mathematics ,119,1437-1442.

Sinha, P. dan Kaul, A. (2018) : Decentralized KYC System. International Research Journal of Engineering and Technology (IRJET), 5(8), 1209-1210.

Wang, S., dkk. (2018): A Blockchain-Based Framework for Data Sharing with Fine-grained Access Control in Decentralized Storage Systems. IEEE Access, 4, 1-5.

Wennergen, O., dkk. (2018) : Tranparency Analysis Of Distributed File System With a Focus on Interplanetary File System. Swedia: University of Skovde.

Zheng, Z., dkk. (2017) : An Overview of Blockchain Technology Architecture, Consesus and Future Trends, IEEE 6th International Congress on Big Data.