A Heuristic Approach to Improve Task Scheduling in Cloud Computing using Blockchain technology

MSc Research Project
Cloud Computing

Fatema Akbar Lokhandwala
x17101531

School of Computing
National College of Ireland

Supervisor: Mr. Vikas Sahni

National College of Ireland
Project Submission Sheet – 2017/2018
School of Computing

Student Name:	Fatema Akbar Lokhandwala
Student ID:	x17101531
Programme:	Cloud Computing
Year:	2017
Module:	MSc Research Project
Lecturer:	Mr. Vikas Sahni
Submission Due Date:	13/08/2018
Project Title:	A Heuristic Approach to Improve Task Scheduling in Cloud Computing using Blockchain technology
Word Count:	XXX

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature:	
Date:	14th September 2018

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
3. Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

A Heuristic Approach to Improve Task Scheduling in Cloud Computing using Blockchain technology

Fatema Akbar Lokhandwala
x17101531
MSc Research Project in Cloud Computing

14th September 2018

Abstract

In Cloud Data Centers (CDCs), energy cost is the major expense and many researchers aim to reduce it at different levels. Reducing the energy cost and improving its efficiency can be done at the DC level and at the server level where task scheduling takes place. Task scheduling not only focuses on improving the energy efficiency but also to allocate the resources efficiently in a timely manner by optimizing energy consumption. With the rise of new technologies in this era, one of the technologies is Blockchain. Blockchain technology has been used with the cloud in terms of security and storage. This paper proposes a heuristic approach to task scheduling in the cloud using blockchain. The work is carried out to improve the parameters such as wait time, execution time, and Service Level Agreement (SLA) that will eventually help to reduce the energy consumption and its cost. The aim of this research is to identify the benefits of blockchain in the cloud to schedule the tasks and explore its feasibility.

Keywords: Cloud Computing. Energy Consumption. Cost. Task scheduling .CloudSim. Blockchain.

1 Introduction

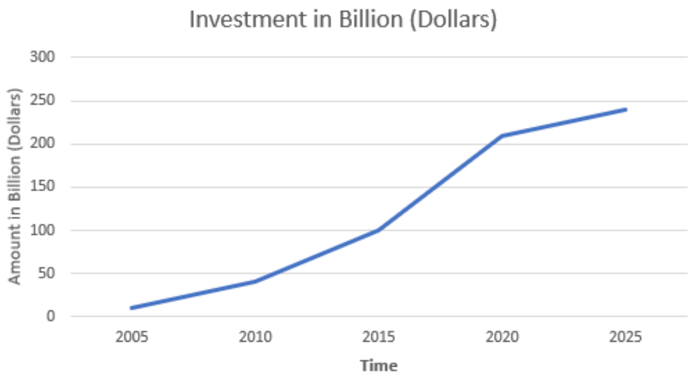
Cloud is widely used to provide online services such as Infrastructure as a service (IaaS), Platform as a service (PaaS), and Software as a service (SaaS) across the world. To perform these high complex computational task at a faster speed, huge DCs are built by top organizations to provide services to the customers 24x7. The aim of the Cloud Service Providers (CSPs) is to achieve reliability, availability, and scalability of the data. To process these data persistently, huge DCs consume a great amount of power. CSP invest millions of dollars to power these DCs as shown in the figure 1. Therefore, to optimize the energy is essential. The energy consumption comes from servers, cooling devices, virtual machines, and electrical equipment.(Lokhandwala; 2018) Also, many DCs aim to do green cloud computing using renewable sources like solar energy, hydroelectric energy, and fuel cells. This helps in reducing the carbon footprints as well as the energy cost.
Task scheduling plays an important role in allocating the resources and researchers aim

to improve scheduling algorithms to decrease the requests completion time which reduces energy consumption and avoids SLA violations. Blockchain technology has flexibility and security for transactions which would help to schedule the resources efficiently. (Buyya et al.; 2017)

1.1 Motivation and Background

International Data Corporation (IDC) has stated that 8.5 million DCs and more are operating globally and consumes a huge amount of energy to power these DC. (Mastelic et al.; 2015). According to the surveys conducted by Columbus (2017) and Stamford (2017), nowadays cloud organizations invest nearly $200 billion yearly and would eventually increase up to $240 billion by 2025. In the cloud, task scheduling is one of the oldest issues as it is related to various factors. Many algorithms were built to improve scheduling that would help in improving the Quality of Service (QoS) and other parameters. In the existing research papers, researches carry out their work using the CloudSim Simulator tool to build a virtual environment to schedule the tasks and evaluate their work.(Calheiros et al.; 2011) The scheduling is usually performed centrally and then later the task is distributed to various DCs considering the network-bandwidth cost. The proposed paper aims to schedule the task using the decentralized approach, the blockchain technology that would allocate the resources efficiently by consuming less energy that would eventually help in reducing the energy cost.

Figure 1: Investment in cloud computing
Columbus (2017) and Stamford (2017)



1.2 Project Specification

This research paper tries to resolve the issue of energy cost to task scheduling in cloud computing. The implementation relies on the CloudSim toolkit and Blockchain using Visual Studio. CloudSim is used to build virtual DCs and Virtual Machines (VMs) whereas Visual Studio is used to build a blockchain network, nodes, and smart contracts. The research question for the proposed paper is mentioned below.

1.2.1 Research Question

The current issues in relation to task scheduling are energy cost in task scheduling, Overall completion time (makespan), execution time, SLA violations, request rejection rate, DC over-load and under-load issues, and energy efficiency.
RQ: *Can energy cost of task scheduling be reduced in cloud computing using blockchain?*

1.2.2 Research Objective

The research question addresses the following objectives: Objective 1: Implementation of virtual DC environment using CloudSim and designing a blockchain network using Visual Studio, Objective 2: Queuing tasks on the basis of shortest length to reduce the waiting time, Objective 3: Scheduling task on a decentralized network to the least loaded DC, Objective 4: Calculating the energy of the DC and its total cost, Objective 5: Analyzing the research question by comparing it with existing work, and Objective 6: Documentation for NCI research report along with a configuration manual.

1.2.3 Research Contribution

The proposed research contributes: Reviewed results of the existing work, working on CloudSim to build DCs and VMs, the building of blockchain network, calculating the cost and the energy, and evaluating the results based on analysis of the proposed work.
The paper is structured in following sections: Section 2 describes a critical review of task scheduling to reduce the cost at DC level and server level It also discusses the implementation of various blockchain technologies in cloud computing. Section 3 presents the methodology of the proposed paper, Section 4 presents the implementation carried out to perform the research. Section 5 justifies the evaluation of the experiments performed. Section 6 defines the conclusion of the proposed paper and provides future work.

2 Related Work

Task scheduling in CDCs aims to decrease the energy consumption by improving its efficiency and reducing the overall cost incurred. Various task scheduling algorithms help to improve the energy efficiency by improving various parameters such as total completion time, total wait time, execution time, makespan and QoS.(Lokhandwala; 2018). The task scheduling can be improved at DC level and server level.

2.1 Task scheduling at DC level

In Hussin et al. (2011), proposed a deadline based priority algorithm that helped in improving the energy consumption and reduce the SLA violations. The priority scheduling worked beneficial in managing the workloads in Large Distributed System (LDS). Also, Amalarethinam and Kavitha (2017) proposed a meta-task scheduling algorithm. The task was scheduled on the user's priority. The high priority task was scheduled on the basis of the Min-Min algorithm whereas the normal prioritize tasks were scheduled on the basis of the Max-Min algorithm. This approach reduced the overall completion time and improved the resource utilization. Agarwal et al. (2014) proposed a generalized priority algorithm that assigned tasks to VMs on the basis of its task size. The tasks with high size had the highest priority. Er-raji et al. (2017) proposed a QuickSort algorithm. The tasks were sorted on the basis of their length in a descending order and VMs were sorted on the basis of their Millions of Instructions Per Second (MIPS).

Alworafi et al. (2018) proposed a budget constraint scheduler. This task scheduling algorithm significantly reduced the network, bandwidth and the storage cost. In Sanjeevi and Viswanathan (2017), the task was prioritized on the urgency and non-urgency basis. The proposed algorithm reduced the total completion time. Alla et al. (2016) proposed a Dynamic Priority Queue (DPQ) algorithm that improved the throughput, user's performance, and the load balancing. In Rani and Suri (n.d.), a unique search approach for task scheduling is proposed which is based on the Ant Colony Optimization (ACO) algorithm. The proposed algorithm reduced the makespan and improved the overall performance.

2.2 Task Scheduling at Server Level

This section describes the review of the existing work carried out to improve task scheduling at the server level.

Zhang et al. (2018) proposed a heuristic algorithm that prioritized the tasks on the basis of its deadline. The tasks with the loose deadlines were postponed without running more Physical Machines (PMs). This helped in reducing the VM overhead occurring during migration. The approach of placing VMs efficiently significantly reduced the energy consumption. Khosravi et al. (2017) proposed a dynamic approach to task scheduling. The resources were allocated to DCs and VMs running on green energy. This approach reduced VM migration issues and the energy cost consumed. In Manzoor et al. (2017), concluded a qualitative analysis that, energy consumption improvement relies on two factors namely VM allocation and VM scheduling. Deadline and cost constraints provide optimal results for the above two factors and help to improve energy efficiency.

2.3 Blockchain in cloud computing

This section discusses the existing work in cloud computing using blockchain.

Yang et al. (2017) proposed the benefits of blockchain in cloud computing for reducing the energy internet. The author briefly explains the working of the blockchain, consensus, and smart contracts. Also, a new approach to task scheduling was introduced. Xu et al. (2018) used this decentralized technology to minimize the energy consumption cost in task scheduling. The author used the concept of mining to calculate the energy cost. In this approach, the authors did not apply the Proof of Work (PoW), as the tasks were allocated to the DCs on the basis of First Come Fist Serve Scheduling.

2.4 Conclusion

Many task scheduling algorithms are developed to improve the energy consumption in a CDC. These algorithms help to improve the scheduling by minimizing the overall completion time, the wait time, the execution time etc. Also, Blockchain is feasible for transactions in a decentralized environment. In this paper, the aim is to utilize the blockchain to task scheduling and reduce energy consumption.

3 Methodology

3.1 CloudSim

CloudSim, a simulator tool used to develop a virtual environment for the cloud computing. Calheiros et al. (2011) In Figure 2, the layered architecture of the CloudSim is described. It provides support to functions such as task scheduling, Task allocation, VM migration, Load balancing, and communication between different entities in the Data Center such as Hosts, VMs, data center, broker, and cloudlets.
The main benefit of the CloudSim tool is its flexibility and time effectiveness. Many researchers utilize the CloudSim for conducting their experiments and evaluate the results. The key components of the CloudSim related to the proposed work are Cloudlets, DataCenterCharacteristics, DataCenterBroker, Network Topology and Virtual Machines.

1. Cloudlets: The cloudlets define the tasks performed in a Data center. The following parameters of the cloudlets are input, output size, and length of the tasks.

2. DataCenter Characteristics: This class stores information about the resources in the DC.

3. DataCenterBroker: This class acts as an intermediary between the cloud user and the cloud provider. This class maintains the QoS requirements.

4. Hosts: This class is responsible for assigning the physical resources like CPU, VM, and Memory.

5. Virtual Machine: This class is responsible for allocating the resources to a VM based on its capabilities.

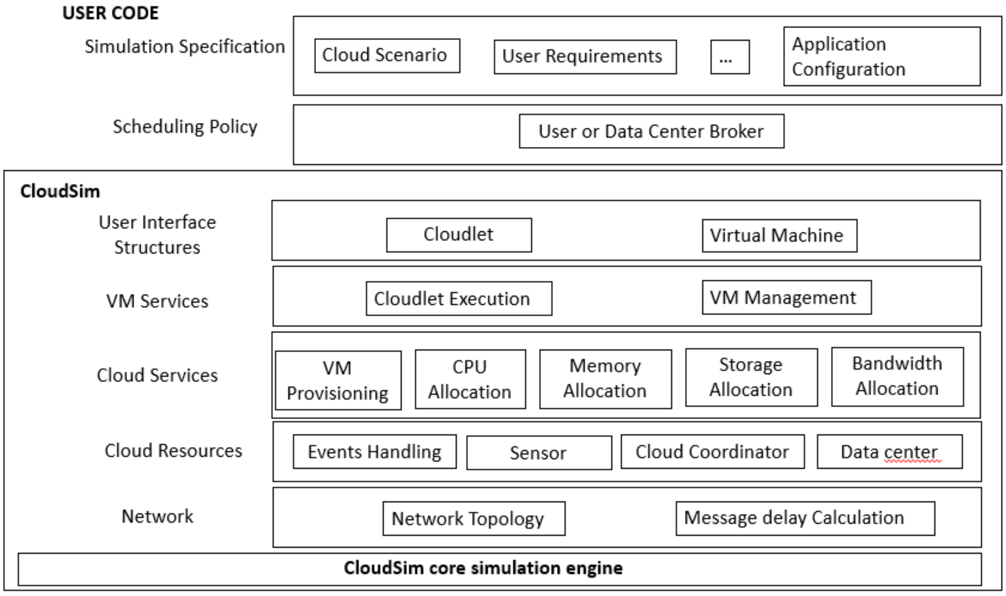
6. Network Topology: This class stores information about network topology in the cloudsim simulation.

Algorithm for task scheduling
Input: Task List
Sort: Sort the task list in ascending according to their shortest length
For all tasks in the Sorted List do
    Schedule the task to the VM
    If the current task length = next task length then
    Sort the task according to their arrival time
    End If
End for

3.2 Blockchain Network

The proposed paper focuses on implementing blockchain to schedule tasks in the cloud to reduce the energy consumption. Xu et al. (2018) used the FCFS scheduling technique to allocate the tasks. This eliminated the need for the scheduler. Later, a DC blockchain network was implemented and the tasks were assigned to the least loaded DC. By using

Figure 2: CloudSim Architecture
Calheiros et al. (2011)



RL technique, energy cost was calculated and the approach was compared with Round Robin (RR) and Max-Min algorithm.

The proposed research paper tries to replicate the work proposed by Xu et al. (2018). However, to minimize the total wait time of the requests, tasks are scheduled using Shortest Job First (SJF) Algorithm. Later, a blockchain network is created. The block stores the following information of the Data centers created such as ID, Timestamp, Previous Hash, and Data.

1. ID: This defines a unique ID for each DC.

2. Timestamp: This defines the current time and date of the block during its creation.

3. Previous Hash: This stores the hash of the previous block linked to it.

4. Data: The 'Data' represents the load of the data center to be stored.

In Figure 3, Multiple DCs form a network and each node(block) is created by defining the above parameters. Lokhandwala (2018)

3.3 Smart Contracts

Smart contracts are generally the code of the blockchain network that allows transactions when a requirement is fulfilled. In Blockchain, we write smart contracts using solidity (Python or JavaScript). The proposed paper creates a smart contract that helps to check the load of the data center stored inside the block. The smart contracts assign tasks to those Data center that has the least load. In figure 4, the working of the smart contract is explained. Lokhandwala (2018)

Figure 3: The Blockchain network

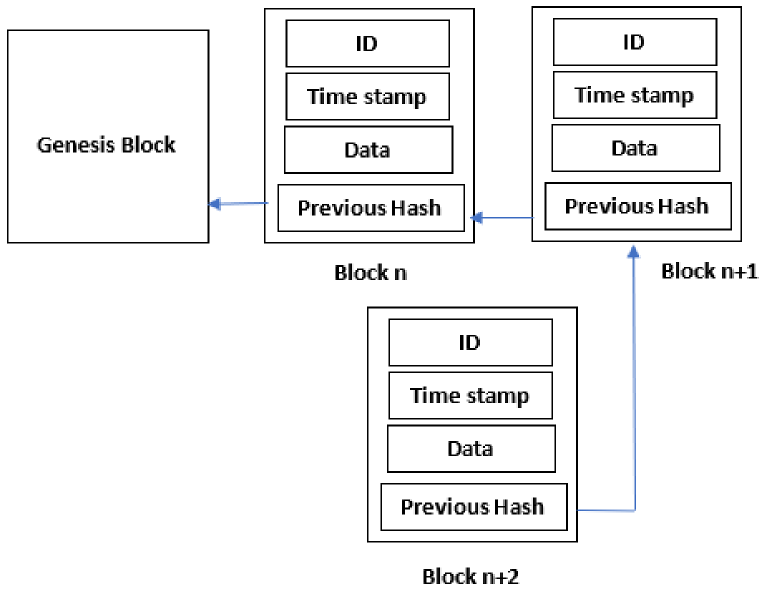


Figure 4: Working of Smart Contract
Lokhandwala (2018)

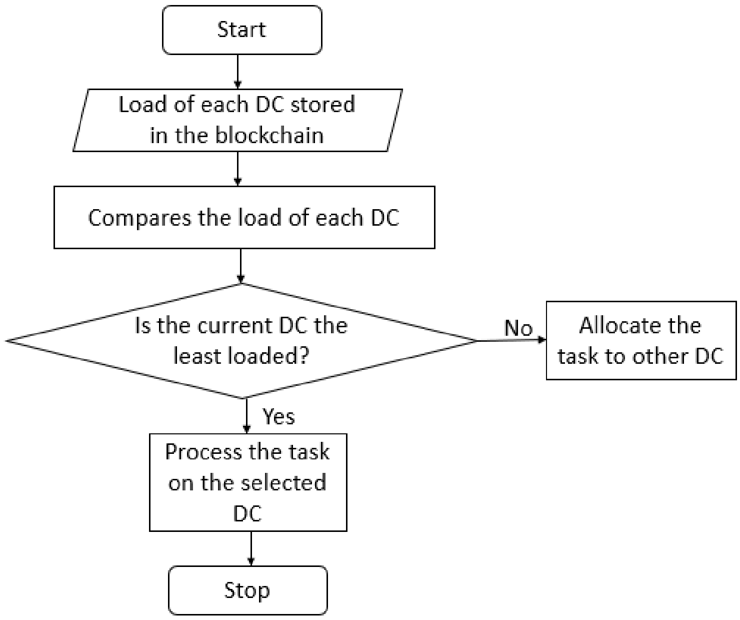


Table 1: Data Center Configuration

Data Center	
Parameters	Specification
System Architecture	x86
Operating System	Linux
Virtual Machine Monitor (VMM)	Xen

Table 2: Host Configuration

Host	
Parameters	Specification
Host ID	0
RAM	2048MB
Storage	1000000
Bandwidth	10000

3.4 Conclusion

In this section, a brief description of the methodologies is discussed. A detail information about the implementation of the research project is discussed below.

4 Implementation

4.1 CloudSim Setup

The proposed research paper used CloudSim 3.0.3 to analyze the algorithm. This simulator helps to test various algorithms for task scheduling, load balancing, virtual machine allocation algorithms etc and other issues of the cloud computing. The benefit of this simulator is that the faults can be known prior by testing the work in a simulating environment rather than real-time implementation.

The proposed experiment was performed on Intel i7 core processor, Windows 10 platform. The parameters used in this experiments were Data Center, Virtual Machine, Host, and Cloudlets. The configuration of each parameter is defined in the tables 1 , 2, 3, and 4.

Table 3: Virtual Machine Configuration

Virtual Machine	
Parameters	Specification
VM size	10000
RAM	512MB
MIPS	1000
Bandwidth	1000
No. of CPU	1

Table 4: Cloudlet Configuration

Cloudlet	
Parameters	Specification
Length	1000
File Size	300
Output Size	300

4.2 Experiment 1: Scheduling Tasks using SJF algorithm

This paper evaluates the technique of scheduling the tasks using the "Shortest Job First (SJF)" algorithm. The aim of using this algorithm is to minimize the wait time.
We performed the experiment using CloudSim 3.0.3. In this experiment, we created 1 DC, 1 VM and 40 cloudlets. The algorithm sorts the tasks on the basis of their shortest length in an ascending order. The length of the tasks are randomly allocated between 1000-3000. The following results were generated as shown in figures 8 9 10 and 11.

4.3 The Blockchain Setup

In this section, we proposed a simple blockchain network with 3 blocks including the genesis block. It takes parameters such as index, time stamp, previous hash, and data. The data parameter can store any value. A blockchain network is created which stores the load of each data center. The index parameter defines the block position, TimeStamp defines the time and date of the block creation and the PreviousHash stores the previous hash of the previous block in the current block.
Note: The load of the Data center is not constant.

To carry out the implementation, we use Visual studio code to create our simple blockchain network. We also validated the block by tampering with the data stored in one of the blocks. By applying some validation, the blockchain alerts about the changes by concluding whether the blockchain is valid or invalid. Therefore, this experiment helps in identifying the security level of the blockchain.

Another test is performed which uses the Proof of Work (PoW) mechanism. To tamper with a block and to re-calculate the hash of each block could be possible. To avoid such spamming, PoW is included. This helps in maintaining the difficulty level of the block by defining the number of zeroes at the beginning of the hash value. The higher the difficulty level, the longer it takes time to create the blockchain.

4.4 Experiment 2: Creating the Blockchain network

In this experiment, we use Visual Studio code to create the Blockchain network. Let's assume that we have created a blockchain network that stores information of 3 geographically distributed DCs. We define parameters such as Index, TimeStamp, PreviousHash and Data that stores the total load of the DC. We create a genesis block and add two other block that specifies a random number that defines the load of the DC. The following output was obtained as shown in figure 5.

The first block with index 0 is the Genesis Block and has no previous hash. The other two blocks have the previous hash parameter and some data stored in it that defines the load of the DC.

Figure 5: Creating the Blockchain network


```
PS C:\Users\[illegible]\node_modules> node main.js
{
    "chain": [
        {
            "index": 0,
            "previousHash": "0",
            "timestamp": "08/08/2018",
            "data": "Genesis block",
            "hash": "5555432c1ee1c5ce6ce22293824f8e041c99806c25802442c07b7d031fd57e17"
        },
        {
            "index": 1,
            "previousHash": "5555432c1ee1c5ce6ce22293824f8e041c99806c25802442c07b7d031fd57e17",
            "timestamp": "11/01/2019",
            "data": {
                "data": 200
            },
            "hash": "a1a91a8ab711c1f776d4eb3d2d6184c5d4c70a2bfea075daa5be001483c4bceb"
        },
        {
            "index": 2,
            "previousHash": "a1a91a8ab711c1f776d4eb3d2d6184c5d4c70a2bfea075daa5be001483c4bceb",
            "timestamp": "12/01/2019",
            "data": {
                "data": 250
            },
            "hash": "b5d2248f538b4496288a26a021dfddd80d04d05e48fcfb28f48ee95d05b205d2"
        }
    ]
}
```

4.5 Experiment 3: Validating the blockchain network

In the experiment, we tamper the data of block 1 and try to validate the blockchain network. If the blockchain network has not tampered, it returns true else false. Even if the hash of the block is re-calculated, it stills indicates that the blockchain network is not valid. In figure 6, the output indicates that the block was valid at first. Later, it identified the alterations and indicated that the blockchain is not valid. This shows that, when N number of transactions are stored in a blockchain, it could still tamper some data of the blockchain or re-calculate the hashes of all the blocks and make the blockchain valid. However, for task scheduling the load of the DC would not be constant. Therefore, the block should be able to automatically update the load of the DC after every task executed on it and calculate its load.

Figure 6: Blockchain validation

```
PS C:\Users\fatem\node_modules> node main.js
 Is Blockchain valid? true
 Is Blockchain valid? false
```

4.6 Experiment 4: Proof of Work

In this experiment, we use mining to increase the difficulty level of security in the blockchain network. We set the difficulty level to 3 that adds 3 zero's at the start of the previous hash.

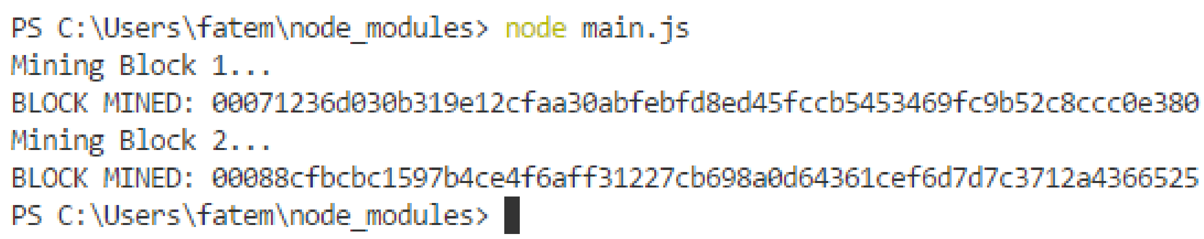
4.7 Conclusion

The description of the CloudSim architecture and the Blockchain network helps to achieve the research objective specified in the section 1.2.2. A detailed conclusion of the experiments performed is discussed in the next section.

5 Evaluation

The aim of the research is to analyze the feasibility of task scheduling using blockchain in cloud computing. The results are analyzed by conducting the following experiments. Also, a detailed conclusion and a future work are provided.

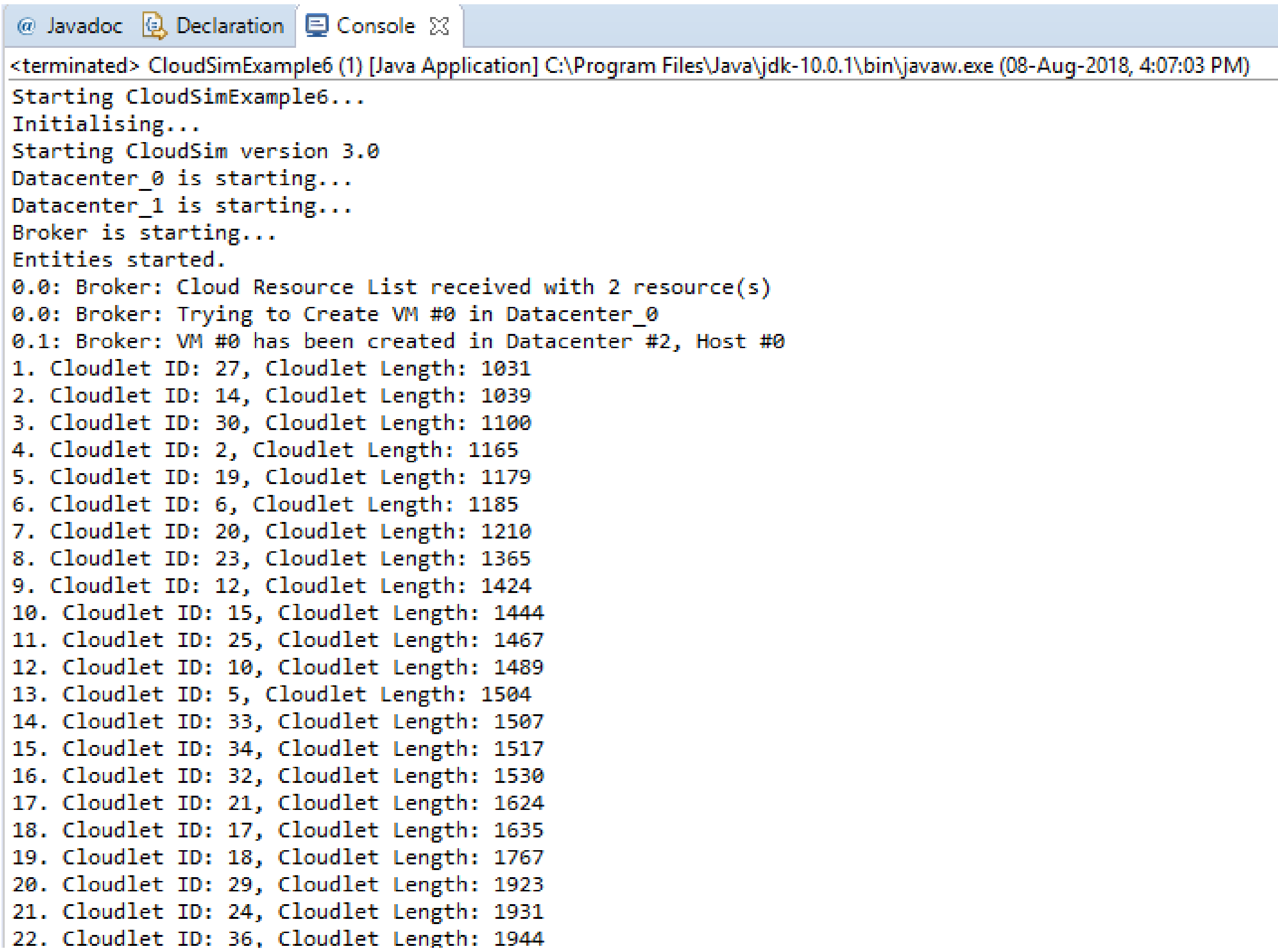
Figure 7: Blockchain validation



5.1 Experiment 1

The CloudSim Simulator schedules the tasks using SJF algorithm to minimize the wait time by sorting the shortest tasks first. After task scheduling, the tasks are allocated to the DCs that are linked on a blockchain network.

Figure 8: Initialization of the CloudSim



5.2 Experiment 2

This paper performs a replication of the work proposed by Xu et al. (2018). Although the work was carried out on a real-world platform, this paper measures the feasibility of blockchain to task scheduling in a simulating environment. A blockchain network is easily created by defining all the parameters and storing the data as well.

5.3 Experiment 3

In this experiment, blockchain was validated by tampering the data. This concludes that tampering the data within a blockchain could be difficult and therefore blockchain is widely used for security purposes.

5.4 Experiment 4

The security level is improved by using Proof of Work (PoW), also known as mining. We re-calculated the hash value by using the nonce parameter. This test shows that once a

Figure 9: Working of SJF algorithm

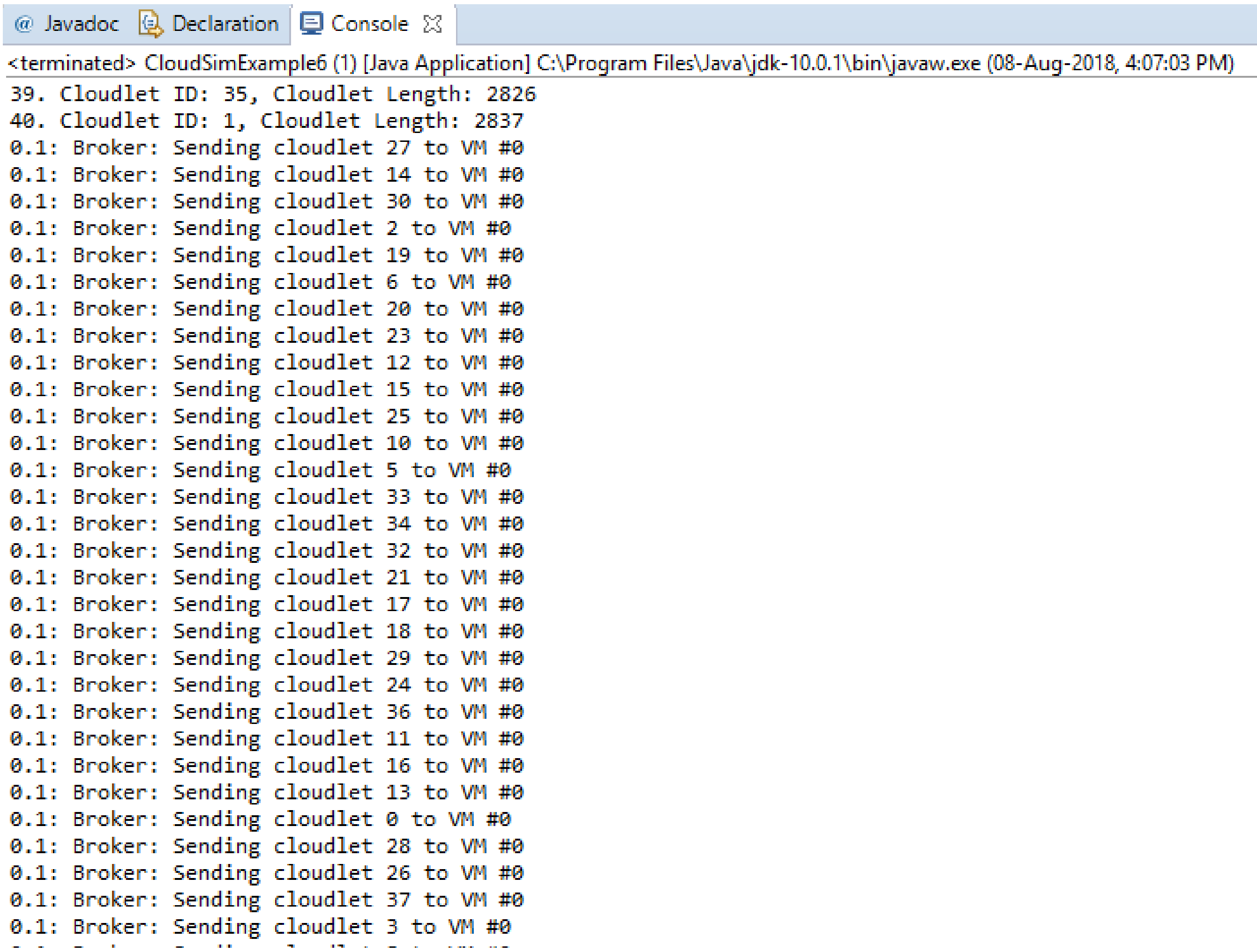


Figure 10: Allocating task to the VM

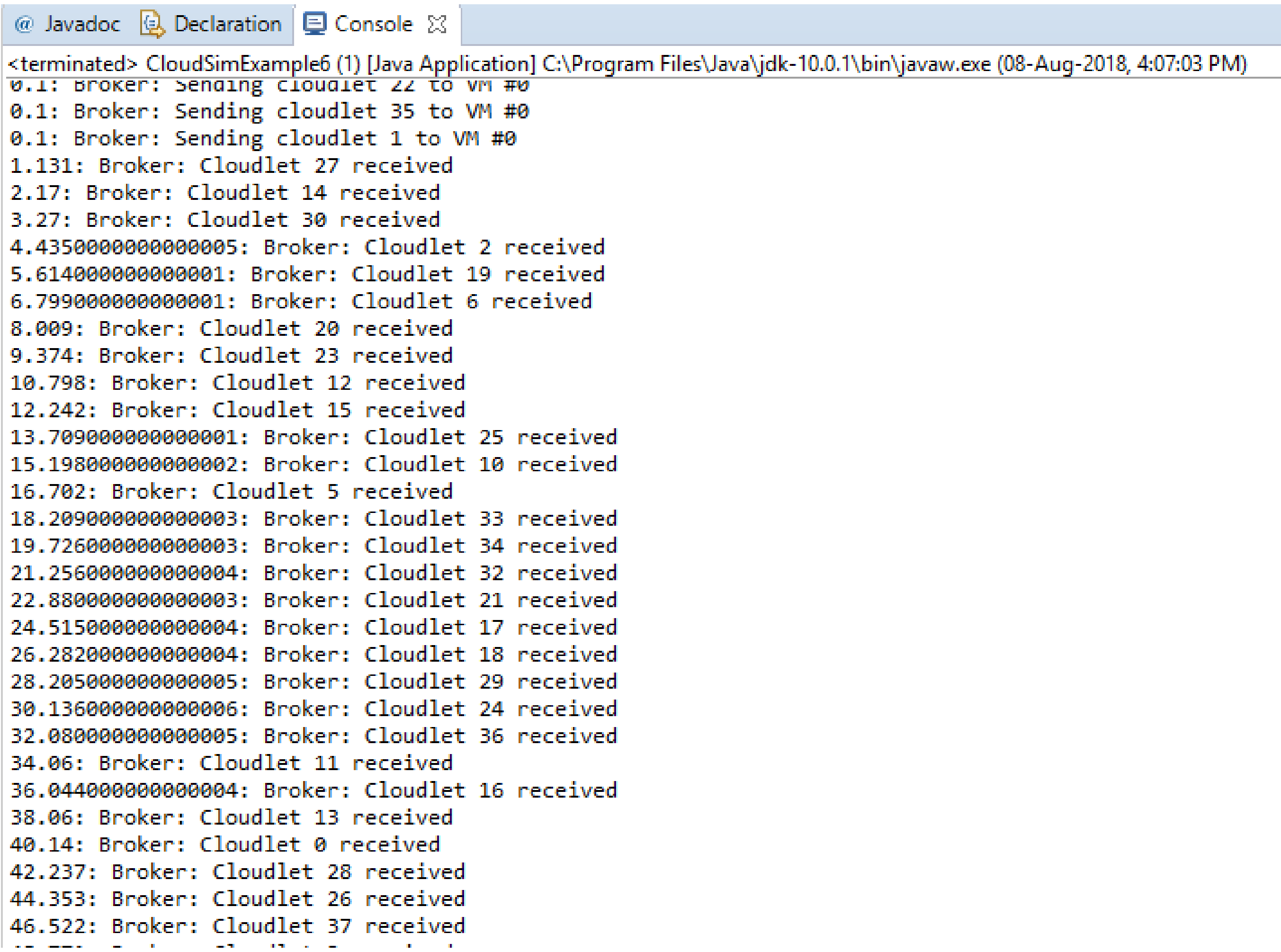


Figure 11: Allocating tasks successfully using SJF algorithm

```
@ Javadoc  Declaration  Console
<terminated> CloudSimExample6 (1) [Java Application] C:\Program Files\Java\jdk-10.0.1\bin\javaw.exe (08-Aug-2018, 4:07:03 PM)
74.22299999999998: Broker: Cloudlet 1 received
74.22299999999998: Broker: All Cloudlets executed. Finishing...
74.22299999999998: Broker: Destroying VM #0
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Datacenter_1 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.

========== OUTPUT ==========
Cloudlet ID    STATUS    Data center ID    VM ID    Time    Start Time    Finish Time
    27         SUCCESS        2              0      1.03      0.1           1.13
    14         SUCCESS        2              0      1.04      1.13          2.17
    30         SUCCESS        2              0      1.1       2.17          3.27
    2          SUCCESS        2              0      1.17      3.27          4.44
    19         SUCCESS        2              0      1.18      4.44          5.61
    6          SUCCESS        2              0      1.19      5.61          6.8
    20         SUCCESS        2              0      1.21      6.8           8.01
    23         SUCCESS        2              0      1.37      8.01          9.37
    12         SUCCESS        2              0      1.42      9.37          10.8
    15         SUCCESS        2              0      1.44      10.8          12.24
    25         SUCCESS        2              0      1.47      12.24         13.71
    10         SUCCESS        2              0      1.49      13.71         15.2
    31         SUCCESS        2              0      2.53      60.5          63.03
    38         SUCCESS        2              0      2.75      63.03         65.78
    22         SUCCESS        2              0      2.78      65.78         68.56
    35         SUCCESS        2              0      2.83      68.56         71.39
    1          SUCCESS        2              0      2.84      71.39         74.22
SJF applied Successfully
```

block is mined, it is nearly impossible to tamper with the data.

5.5 Discussion

From the above experiments, we can conclude that blockchain has many security features and is more reliable for storing transactions. However, when a large number of transactions are stored in a blockchain network it would become difficult to create new blocks. It would consume time as well as energy. It is more feasible for imposing securities in a transaction by increasing its difficulty.

A task scheduling in cloud computing relies on various factors. Using blockchain in cloud computing for storage is a good option. However, Task scheduling would not be that feasible as it would just store a large amount of data. But for calculating of load of DCs after every certain interval would consume a huge amount of energy as well as time. Also, Blockchain requires few minutes for mining. Therefore, blockchain is more feasible for security purposes.

6 Conclusion and Future Work

The main goal of the research was to improve energy efficiency to task scheduling using blockchain. The CloudSim simulator helped for implementing the SJF algorithm, which aims to reduce the wait time for other tasks. Scheduling using shortest job first would eventually reduce the overall completion time but might fail to reduce the SLA violations. Therefore, In future work, scheduling using deadline-priority would be useful for reducing SLA violation.

Further, we proposed Blockchain architecture to task scheduling by creating a blockchain network that links DCs. The tasks are executed on DCs that is least loaded. The block defines the load of the DC and it needs to be incremented after every certain interval. The limitation of the proposed paper was to develop a smart contract that would help to choose the DC with the least load and execute the tasks on it. However, we failed to develop the contract.

The future work for the research paper is to evaluate the proposed work in a real-time environment to conclude better results to task scheduling using blockchain. Also, to develop smart contracts that would help to that allocate the task to the least loaded DCs.

Acknowledgement

I would sincerely like to thank all my college professors for supporting me throughout my course in MSc in Cloud Computing. I am sincerely grateful to Mr. Vikas Sahni for his constant support, motivation and, encouragement during the MSc Research Project.

References

Agarwal, D., Jain, S. et al. (2014). Efficient optimal algorithm of task scheduling in cloud computing environment, *arXiv preprint arXiv:1404.2076* .

Alla, H. B., Alla, S. B., Ezzati, A. and Touhafi, A. (2016). An efficient dynamic priority-queue algorithm based on ahp and pso for task scheduling in cloud computing, *International Conference on Hybrid Intelligent Systems*, Springer, pp. 134–143.

Alworafi, M. A., Al-Hashmi, A., Dhari, A., Darem, A. B. et al. (2018). Task-scheduling in cloud computing environment: Cost priority approach, *Proceedings of International Conference on Cognition and Recognition*, Springer, pp. 99–108.

Amalarethinam, D. G. and Kavitha, S. (2017). Priority based performance improved algorithm for meta-task scheduling in cloud environment, *Computing and Communications Technologies (ICCCT), 2017 2nd International Conference on*, IEEE, pp. 69–73. DOI: 10.1109/ICCCT2.2017.7972250, Conference Location: Chennai, India.

Buyya, R., Srirama, S. N., Casale, G., Calheiros, R., Simmhan, Y., Varghese, B., Gelenbe, E., Javadi, B., Vaquero, L. M., Netto, M. A. et al. (2017). A manifesto for future generation cloud computing: Research directions for the next decade, *arXiv preprint arXiv:1711.09123* .

Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A. and Buyya, R. (2011). Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, *Software: Practice and experience* **41**(1): 23–50.

Columbus, L. (2017). Roundup of cloud computing forecasts, 2017.
URL: *https://www.forbes.com/sites/louiscolumbus/2017/04/29/roundup-of-cloud-computing-forecasts-2017/31a0d0a531e8*

Er-raji, N., Benabbou, F. and Eddaoui, A. (2017). A new task scheduling algorithm for improving tasks execution time in cloud computing, *Proceedings of the Mediterranean Symposium on Smart City Applications*, Springer, pp. 298–304.

Hussin, M., Lee, Y. C. and Zomaya, A. Y. (2011). Priority-based scheduling for large-scale distribute systems with energy awareness, *Dependable, Autonomic and Secure Computing (DASC), 2011 IEEE Ninth International Conference on*, IEEE, pp. 503–509. DOI: 10.1109/DASC.2011.96 , Conference Location: Sydney, NSW, Australia.

Khosravi, A., Andrew, L. L. and Buyya, R. (2017). Dynamic vm placement method for minimizing energy and carbon cost in geographically distributed cloud data centers, *IEEE Transactions on Sustainable Computing* **2**(2): 183–196. DOI: 10.1109/TSUSC.2017.2709980.

Lokhandwala, F. A. (2018). A heuristic approach to improve scheduling in cloud computing using blockchain.

Manzoor, S., Manzoor, M. and Hussain, W. (2017). An analysis of energy-efficient approaches used for virtual machines and data centres, *e-Business Engineering (ICEBE), 2017 IEEE 14th International Conference on*, IEEE, pp. 91–96. DOI: 10.1109/ICEBE.2017.23.

Mastelic, T., Oleksiak, A., Claussen, H., Brandic, I., Pierson, J.-M. and Vasilakos, A. V. (2015). Cloud computing: Survey on energy efficiency, *Acm computing surveys (csur)* **47**(2): 33.

Rani, S. and Suri, P. (n.d.). An efficient and scalable hybrid task scheduling approach for cloud environment, *International Journal of Information Technology* pp. 1–7.

Sanjeevi, P. and Viswanathan, P. (2017). Nuts scheduling approach for cloud data centers to optimize energy consumption, *Computing* **99**(12): 1179–1205.

Stamford (2017). Gartner says worldwide public cloud services market to grow 18 percent in 2017.
URL: *https://www.gartner.com/newsroom/id/3616417*

Xu, C., Wang, K. and Guo, M. (2018). Intelligent resource management in blockchain-based cloud datacenters, *IEEE Cloud Computing* **4**(6): 50–59. DOI: 10.1109/MCC.2018.1081060.

Yang, T., Guo, Q., Tai, X., Sun, H., Zhang, B., Zhao, W. and Lin, C. (2017). Applying blockchain technology to decentralized operation in future energy internet, *Energy Internet and Energy System Integration (EI2), 2017 IEEE Conference on*, IEEE, pp. 1–5. DOI: 10.1109/EI2.2017.8244418 , Conference Location: Beijing, China.

Zhang, Y., Cheng, X., Chen, L. and Shen, H. (2018). Energy-efficient tasks scheduling heuristics with multi-constraints in virtualized clouds, *Journal of Grid Computing* pp. 1–17.