P2P Energy Trading on the Blockchain

Kareem Osama Mahmoud Elsayed Ahmed
School of Engineering
Monash University Malaysia
Selangor, Malaysia

Dr. Tan Wen Shan
School of Engineering
Monash University Malaysia
Selangor, Malaysia

***Abstract*—The energy sector is facing multiple problems when it comes to facing growing demand for users who require energy or just the growing population in general and how it affects the old infrastructure of transmission lines and the grid in general. A peer-to-peer Energy trading platform that is built utilizing Blockchain technology has been implemented, the platform's name is "NRG". The back end of the platform is built using the Golang programming language, while using the Angular framework for the front end of the platform and the user interface. Moreover, the platform also utilizes a NoSQL cloud database called MongoDB to securely and reliably store its data. The platform utilizes smart contracts that are able to govern and control all the transactions made without any human's interference, which makes this platform completely trust-less. Several test scenarios have been carried out utilizing Locust's open-source load testing tool coded in the Python programming language to measure and quantify the Blockchain's performance. The testing scenarios involved a number of users (150, 500, 1000) continuously transacting with each other repeatedly. The Blockchain that makes up the backbone of the platform was able to handle up to an average of 3.83 tps (transactions per second) through all test runs, while the average rps achieved for all test runs at maximum concurrent users amounted to 4.8 tps with a peak of 5.4 tps.**

Keywords—Blockchain; Peer-to-Peer; Energy; Renewable Energy; Smart Contracts; Energy Trading; Smart Grids; Micro-grids; Web App; Golang; Angular; Locust; Load testing; Market Place.

I. Introduction

As the world grows more and more people get access to electricity and have more money to buy bigger houses to live in filled with more appliances and electrical devices, it is only fair that the energy consumption for the average person throughout history has been increasing steadily [1], [2]. Since Grids in Malaysia are supplied with energy through main transmission lines that go from the generators all the way to distribution lines which then forward that energy to end users such as industrial, commercial, or residential users, it is in no doubt that as this demand and consumption for energy keeps increasing, at one point these transmission lines would have to be upgraded and renewed [3]. This would take a huge amount of time, resources and money, and if nothing is done the transmission lines will be extremely congested which is already starting to happen now during peak hours.

Another downfall of the energy sector according to consumers would be the houses with solar panels who do not own batteries that store their generated surplus of solar energy. They are not benefiting in any way from the surplus energy they generated as all of that energy is dissipated to the grid for free [4]. Not only does that make people feel like they have been cheated on the energy they produced, but also makes them reluctant to use any kind of renewable energy generation in their house.

One way to combat and put an end to the aforementioned problems would be to implement a P2P Energy trading Blockchain platform. Since prosumers would be able to provide surplus energy they generated for sale on the market, consumers would then be able to buy it from them without needing to get electricity from the main transmission lines and stress them [5]. This would eventually reduce the load on transmission lines as prosumers and consumers are using this P2P environment to facilitate their energy needs and consumption [6]. Consequently, Infrastructure upgrade costs and resources can be invested in another project as this upgrade will not be as necessary as before. Furthermore, an incentive will be created for people to buy their own solar panels at their house so they can start selling on the P2P platform. This will increase the number of people who are using renewable sources of energy, which will eventually lead to a cleaner and greener environment.

Integrating Blockchain in the energy trading environment would be a monumental improvement. It allows the whole market to be decentralized which makes people have more trust in the system as they know that it is not controlled by one single entity [7]. Not to mention that Blockchain enables the use of smart contracts which allows for a more transparent transaction process as the delivery of money and energy is monitored by the computer which minimizes the number of errors made. The use of smart contracts also allows for a more automated trading process among peers [8]. Therefore, it is no surprise that Blockchain could make some huge changes in the energy sector.

There has already been multiple research and trials of implementing a P2P energy trading platform/environment in the past with a few who made a huge success in the areas they were deployed in. The International Renewable Energy Agency (IRENA) issued a recent publication in July of 2020 detailing the benefits and possible gains from the implementation of a P2P trading platform [9]. There are many countries now who are currently using a P2P trading platform or at least conducted a pilot test to gauge feasibility. These countries include Germany, Japan, Malaysia, Netherlands, and many more [10].

Existing P2P Energy trading projects:

1. Piclo [11]:

◦ Established in the UK.

The meter data, consumer preference and generator pricing are used to match electricity demand and supply every half an hour.

◦ They provide contracts, meter data, billing, good customer service.

2. PeerEnergyCloud [12]:

◦ Established in Germany.

◦ Provide virtual marketplace for power trading and value-added services within a Micro-grid.

3. TransActive Grid [11]

◦ Located in Brooklyn, US.

◦ Current Prototype uses the Ethereum Blockchain.

A. *Contributions*

This paper and NRG as a platform in general aim to transform the future of energy to a much more decentralized, more reliable, and cleaner version of what it at these current times. The contributions are classified below:

- A Blockchain integrated P2P trading platform that is free for everyone and completely transparent.
- High standard security of data and immutability.
- Reliable and trustless platform that strays away from monopolistic behaviors that possess the current state of the energy industry.
- Clean and intuitive interface that makes it easy for everyone, no mater who they are, to be able to make energy transactions.
- Evaluation of how the Blockchain's performance varies with different test scenarios.

B. *Organization of the paper*

The following sections aim to document the process of building such a platform and they are organized as follows. Section II outlines the development of the platform and what kind of database it uses in addition to the programming languages used in terms of front end and back end. Different test case scenarios are also explored and evaluated in section III. Finally, being such a big project that has endless potential of scalability, some future improvements will be explored and discussed and the paper will be concluded in section IV.

II. DEVELOPMENT

At the beginning of this project, it was decided that an open-source consortium Blockchain platform (HyperLedger Fabric) will be used, which was at the level of knowledge acquired at that point an actually good choice. Nonetheless, after further research into HyperLedger fabric and diving into its complexity, capabilities and scaling, it was realised that the platform was too complex for this type of project and its context. Furthermore, after encountering a plethora of conflicts and intricacies when working with fabric's own code, it was recognized that fabric needed a whole team/organization in order to accomplish such a project in a short time, not to mention that fabric was coded by almost 300 contributors and examining the code written in order to use fabric properly was extremely difficult. With all that being said, immersing in fabric for the better part of half a year provided plenty of knowledge and context when it comes to Blockchains and coding them, which is why the decision to code a Blockchain from scratch was made. Interestingly enough, the coding language used for the back end was Golang, which is the same language use by HyperLedger Fabric.

This section of the report will outline and document the development of the platform technically while showcasing and explaining all the code used. The first section will be going over the database, showing the clusters and servers used and their regions. Secondly the back end development/code will be showcased and explained in great detail, while also highlighting reasons for certain actions or choices and also showcasing the Blockchain's architecture, crucial components and validation. Finally, the last part of this section will be exploring the development of the front end, the technical features and the code used.

A. *Cloud Database*

The database is the final layer of the project where it interacts with the middle layer (back end) to store pieces of information safely so they can be retrieved, updated, or even removed whenever the need arises. The database of choice for the type of data and applications used in this project was a NoSQL database, namely MongoDB. This decision was made based on the advantages of the "NoSQL" databases such as supporting storage of unstructured data unlike SQL databases [13]. In addition, unlike SQL databases, MongoDB enables horizontal scaling, also known as 'sharding', which means adding additional low- cost commodity hardware servers or clusters for the purpose of scalability, in other words, instead of one server serving the queries, two or more are [14]. This, of course, is very cost-effective compared to SQL databases where generally only vertical scaling is possible which involves upgrading the memory size/disk space/computing power of the single server storing the database [15],[16].

The flowchart in Fig. 1 shows the structure of the data to be stored in the database.

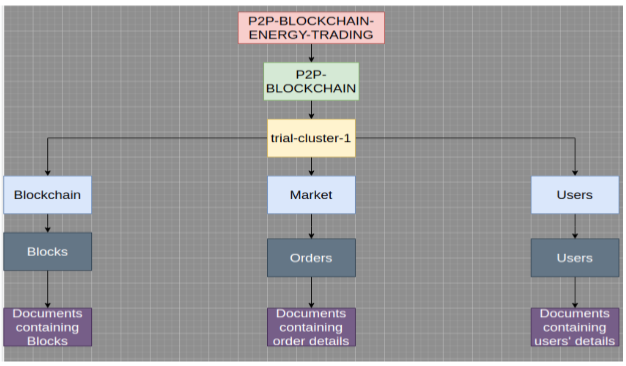

Fig. 1. MongoDB structure

B. *Back end*

The back end is the middle layer of platform, it interacts with not only the database (the final layer), but also interacts with the front end of the platform (the first layer). The back end of the project was developed using the Go (Golang) Programming language for a plethora of reasons, but mainly due its speed, concurrency, and efficiency. The back end consists of nine go files, each made to serve a specific purpose for the platform, most importantly, the “smartcontracts.go” file which contains all the code that governs the entire platform and it will be explored below in a series of flowcharts.

1. Making a new order

When a user decides to make a new order, be it a buy order, or a sell order, the asset that they are trading will need to be locked so they cannot use it in any type of transaction. The following flowchart in Fig. 2 illustrates how this is done via the “new order smart contract”.

2. Transaction between two users

Carrying over from the previous flowchart in figure 2, and assuming that Kareem (User A) has decided to go with a sell order, the next flowchart in Fig. 3 introduces Mark (User B) who decides to buy some of the energy that Kareem has put for sale in his order (200 Kilo Watt hour). After Kareem and Mark have their new balances updated, a new block containing the transaction details will be generated and added to the Blockchain.

3. Deleting an order

In the case of a user who no longer wishes to keep their order on the market, they can choose to delete it instantly. The following flowchart in Fig. 4 introduces Jason (User C) who would like to delete his order and how smart contracts could handle that.

C. *Front end*

The front end is the first layer of platform, it allows the user to use the platform’s functions and features through a clean interface that is easy to understand. It interacts with the back end through HTTP requests and receives relevant data from the back end based on the requests made. The front end of the platform was developed and implemented using the Angular framework which is an open-source front end framework, the decision to use Angular was made for several reason mainly because its modularity and reusability. The user interface created is a responsive web application which means that the interface will adapt (respond) to the size of the device that is currently displaying it (i.e., Smartphone, Tablet, or Computer). Moreover, forms are validated and required, which means that the user cannot submit forms until the data is entered correctly and is notified through visual cues by the interface that the data is invalid. Some important interfaces are shown in Fig. 5 to Fig. 15 while Fig. 16 shows the validation flow chart of making a new transaction.

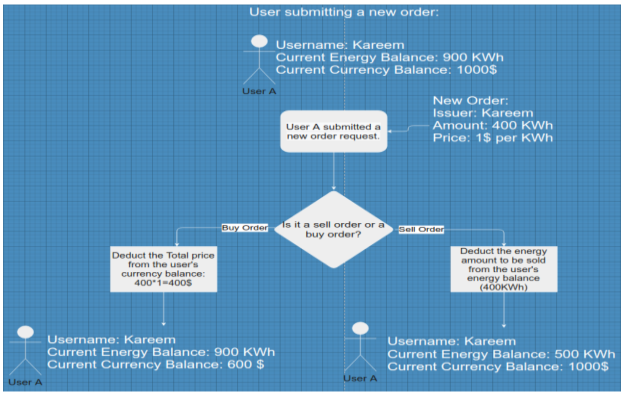

Fig. 2. New Order Flowchart

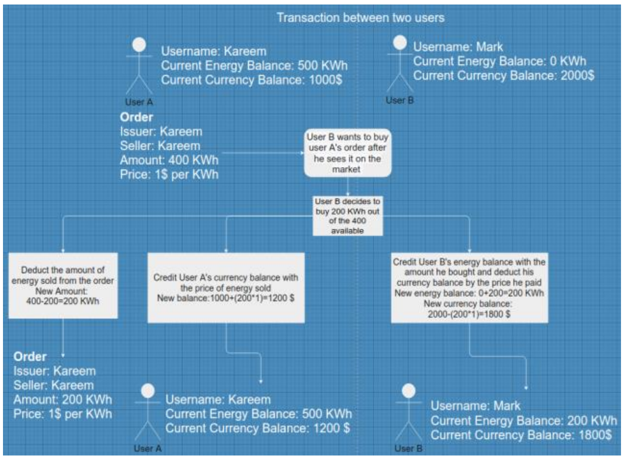

Fig. 3. Transaction Flowchart

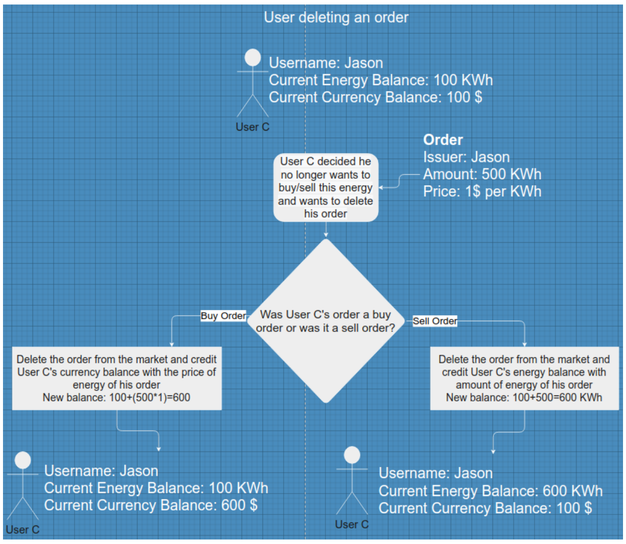

Fig. 4. Order Deletion Flowchart

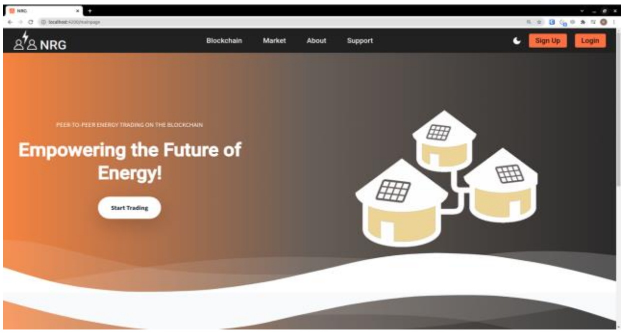

Fig. 5. Main Page user interface

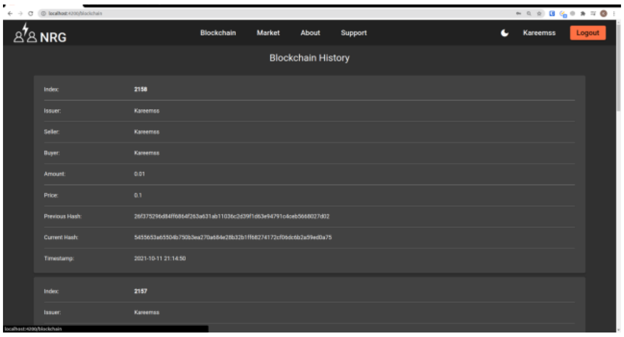

Fig. 6. Blockchain Page user interface

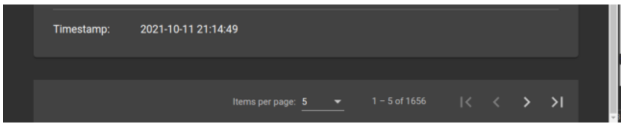

Fig. 7. Blockchain Page Paginator

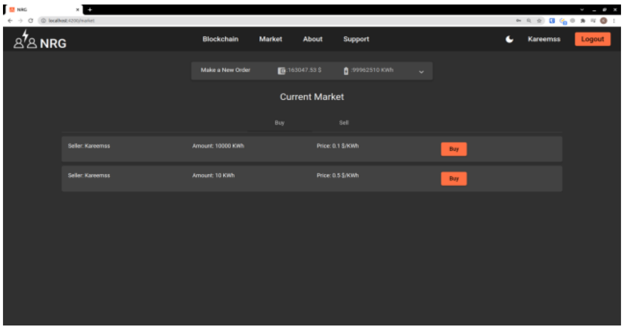

Fig. 8. Market Page user interface

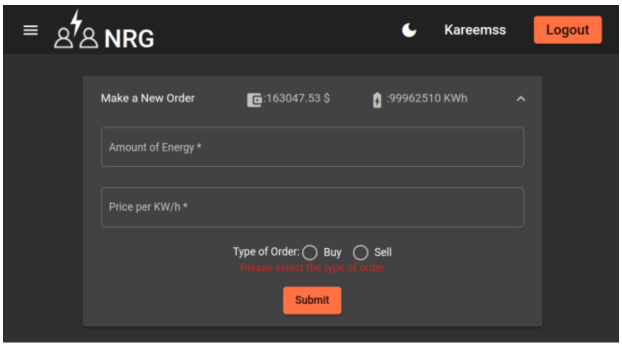

Fig. 9. New Order Expansion Panel

Fig. 10. Validation visual cues

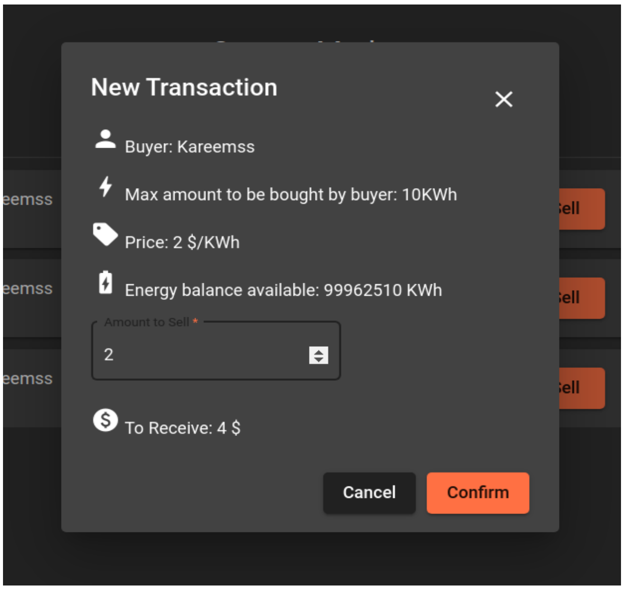

Fig. 11. Transaction dialog form

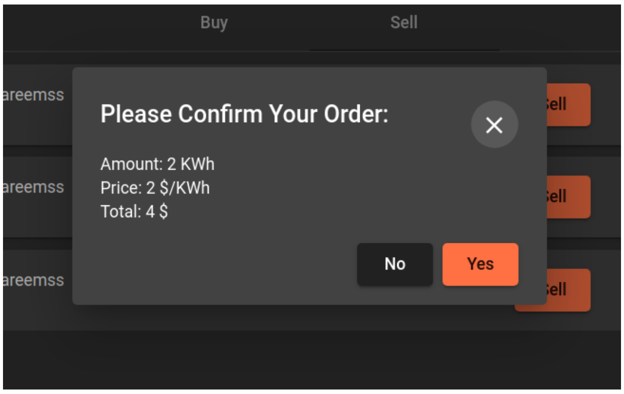

Fig. 12. Confirmation dialog

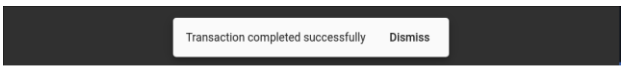

Fig. 13. Snackbar notification

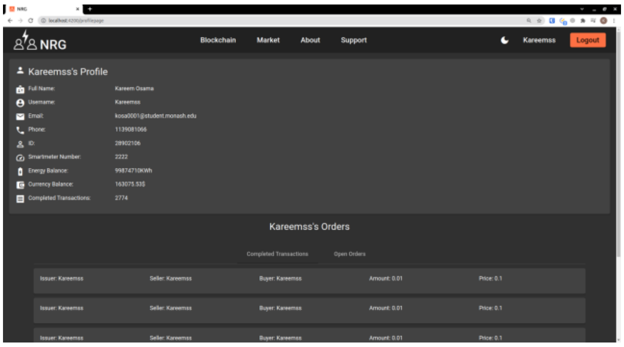

Fig. 14. Profile Page user interface I

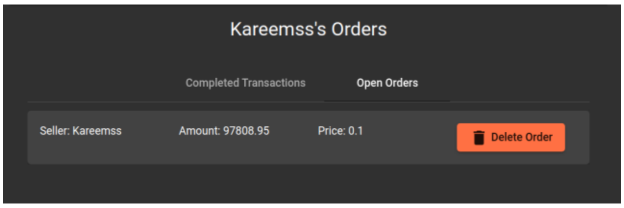

Fig. 15. Profile Page user interface II

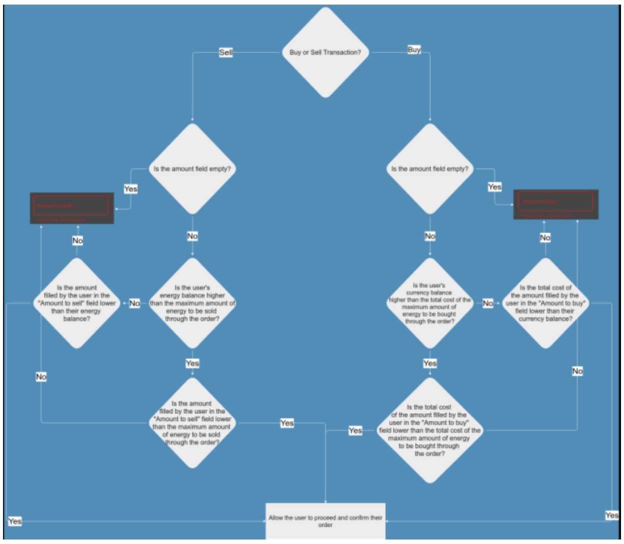

Fig. 16. Transaction form validation flowchart

III. TESTING RESULTS AND DISCUSSION

Several load tests have been made to quantify and substantiate the speed and throughput rate of the Blockchain, and they will be explored in the following sections. Before getting into the test details, there are four main conditions/variables that can affect the speed of the Blockchain that will have to be mentioned. Firstly, the design of the program and what type of algorithms are being used in the code. Secondly, the efficiency of the code and minimization of RAM usage. Thirdly, the technical specifications of the server/computer that is hosting the program (i.e., RAM, CPU, HDD/SSD etc…). Finally, the speed of the database being used (to note: this platform uses the MongoDB free tier as shown previously in the development section).

To carry out the performance test, an open-source performance benchmarking and load testing tool that is named Locust has been used. Locust is extremely efficient at swarming HTTP services with users while minimizing the resources needed to do so, which is why it is being used by extremely big companies such as Google, Microsoft, and RIOT Games. Locust performs testing by running a locust test file that is written in the Python programming language by the user. Finally, to ensure comparable and valid results when testing, the locust test and the back end have been hosted on the same personal computer and its specifications are shown in the Fig. 17.

The test scenarios have been carried out by making a user class in the locust file that performs exactly one task (Making a transaction) and keeps repeating it depending on the wait time range assigned. Upon running the load testing file, Locust will keep spawning new instance of that user class based on two variables: The number of peak concurrent users and the user spawn rate per second. For each of the scenarios below, these variables will be declared, in addition to graphical data that aid in explanations and quantifying the speed of the Blockchain.

1. Frist Test Scenario:
   - Starting user number: 0
   - Peak concurrent users: 150
   - User spawn rate: 2 users per second
   - Task: Perform a transaction
   - Wait time before repeating the task: Between 1 and 2 seconds
   - Total number of requests that have been completed: 1165
2. Second Test Scenario:
   - Starting user number: 0
   - Peak concurrent users: 500
   - User spawn rate: 2 users per second
   - Task: Perform a transaction
   - Wait time before repeating the task: Between 1 and 5 seconds
   - Total number of requests that have been completed: 1320
3. Third Test Scenario:
   - Starting user number: 0
   - Peak concurrent users: 1000
   - User spawn rate: 5 users per second
   - Task: Perform a transaction
   - Wait time before repeating the task: Between 1 and 5 seconds

- Total number of requests that have been completed: 2015

A. *First Test Scenario*

As seen in Fig. 18, as the number of users increases until it reaches maximum peak concurrent users, the total successful Requests per Second (rps) is very unstable. However, after it reaches the maximum number of users for a couple of seconds, the rps starts to stabilize within a range of 4.7 to 5.2 which puts the calculated average of the entire test scenario at 4.4 rps, with the peak rps being 5.2. Moreover, it should be noted that there were zero failures for the entirety of the test run, which means that all 1165 transactions have been completed successfully.

B. *Second Test Scenario*

As seen in Fig. 19, as the number of users increases until it reaches maximum peak concurrent users, rps is very unstable, similar to Test Scenario 1. However, after it reaches the maximum number of users for a couple of seconds, the rps starts to stabilize within a range of 4.6 to 5.4 which puts the calculated average of the entire test scenario at 3.7 rps, with the peak rps being 5.4. Moreover, just like the previous scenario, there were zero failures for the entirety of the test run, which means that all 1320 transactions have been completed successfully.

C. *Third Test Scenario*

An As seen in Fig. 20, as the number of users increases until it reaches maximum peak concurrent users, the rps is very unstable, similar to scenario 1 & 2. However, after it reaches the maximum number of users for a couple of seconds, the rps starts to stabilize within a range of 4.5 to 5.4 which puts the calculated average of the entire test scenario at 3.4 rps, with the peak rps being 5.4. Moreover, just like the previous scenarios, there were zero failures for the entirety of the test run, which means that all 2015 transactions have been completed successfully.

Fig. 17. System Specifications

Fig. 18. First Test Scenario

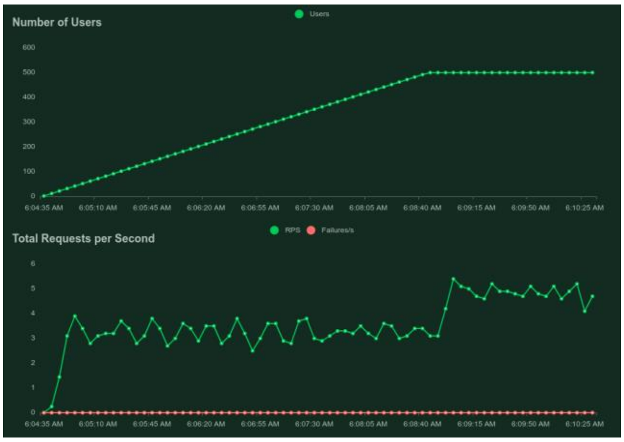

Fig. 19. Second Test Scenario

Fig. 20. Third Test Scenario

With the testing out of the way, it’s time to summarize the results and add some context to put things into perspective. Even though the rps during the linear increase to maximum number of concurrent users has not been the best, the performance at peak concurrency was above average.

The average rps through all three test scenarios amounted to 3.83 rps, while the average rps achieved for all test runs at maximum concurrent users amounted to 4.8 rps with a peak of 5.4 rps. This shows that the transaction validation process is much smoother when the the number of users becomes stable and stops increasing.

Overall, The Blockchain has performed extremely well considering the minimal resources that have been allocated, from the low computer specifications to all the free low tier services that have been utilized.

IV. Future Improvements & Conclusion

As outlined in the previous sections, this project has a huge amount of potential for scalability and a plethora of improvements to be made, these improvements will be explored in the following subsections.

A. Upgrading database

Currently, this project utilizes the MongoDB cloud database's free tier, which is a sever that shares its RAM (Random Access Memory) with other free tier users. Upgrading the MongoDB will not only allow the database to be dedicated only for the project and its ram not shared with anyone, it will also increase the maximum storage size from 512 Megabytes to a higher storage cap. Moreover, CRUD (Create, Read, Update, and Delete) Operations speeds are improved and the number of maximum connections to the database which is currently capped at 500, can also be further increased. In addition, a higher tier will also provide more virtual core processing units (vCPUs), and that will enhance the performance of the server marginally.

B. Upgrading hosting services

The back end is currently hosted via a starter free service that is called Heroku. This service is good for starting out and handling low number of users, however, in the future, when the number of users is bound to increase, a better cloud hosting service will need to be used like a paid subscription of google cloud or entirely renting a cloud web server with good specifications for hosting purposes. A better cloud hosting solution will allow the back end to process more requests (transaction) per second and provide a way smoother experience to the user at peak times.

C. Adding more nodes of the Blockchain

At the current moment, the NRG platform only has one node, which means there's only one copy of the Blockchain at any given moment, and this is purely because of budgeting and resource reasons. Therefore, in the future multiple instances/copies of the Blockchain can be initialized in various servers and databases to increase the robustness of the network and ensure the maximum potential security for the clients.

D. Introduce 'tokenomics'

The NRG can introduce a tokenized asset (let us call it NRG token) with its own tokenomics for various strategic reasons. First of all, NRG can profit from the amount of transactions being made daily by introducing a small transaction fee like a (0.1%) of the price of energy traded, which the users will only be able to pay with the NRG token. This will ensure that NRG has enough capital to facilitate development moves such as doing more research and regular maintenance to the platform or business moves such as marketing or public relations initiatives.

Another area where tokenomics can really aid NRG is with user set-up nodes. To elaborate further, NRG can incentivize its users by paying them in NRG tokens to set up their own validating nodes at their home computers. These validation nodes will secure a copy of the latest instance of the Blockchain. The users can be paid for the time they continue to keep their computer running and connected to the network for validating new transactions. This is different from mining since their computers are not doing any algorithmic work, they are simply continuously storing the latest copy of the Blockchain, so it can be invoked by the main program whenever it is validating a new transaction. This will make sure that all copies of the Blockchain in all places are completely identical with each other. Such a concept, will not only benefit NRG greatly, but will also benefit its users, since they're being paid for their work or more precisely, their computer's work. In addition, it will also build user loyalty to the platform and increase the users' trust in it.

E. Conclusion

Based on all the previous sections, it is clear that the Blockchain technology is here to stay and keep penetrating even more industries. NRG has a huge potential to make ripples through the energy industry. Moreover, the smart contracts have been able to govern the platform efficiently and the performance of the blockchain has been spectacular relative to the resources being used. To put things into perspective, Bitcoin's throughput rate is around 7 tps while NRG has achieved an average of 4.8 tps during peak user concurrency with a maximum tps of 5.4.

Acknowledgments

This piece of work would have not been possible without the support of my family and their firm belief in my potential, nor without the support of my dear friends who encouraged me and tried to lift me up at some of the times I have been down in. I would also like to acknowledge the efforts of my supervisor Dr. Tan Wen Shan, for believing in my capabilities since he chose me for this project until the very end of it, and for supporting and trusting my decision making in all the decisions and turning points throughout the project. Moreover, I would like to acknowledge the efforts of the Undergraduate Research Opportunities Program (UROP) students who have been assigned to this project for their efforts, namely, Chia Jun Shen, Mohammed Rafaquat Alam, Drayang Chua Kai Yang, and Murtaza Hatim Rangwala.

References

[1] G. "U.S. energy facts explained - consumption and production - U.S. Energy Information Administration (EIA)", Eia.gov, 2021. [Online]. Available: https://www.eia.gov/energyexplained/us-energy-facts/. [Accessed: 13- Apr- 2021].

[2] H. Ritchie and M. Roser, "Energy", Our World in Data, 2021. [Online]. Available: https://ourworldindata.org/energy/country/malaysia. [Accessed: 13- Apr- 2021].

[3] "Outline of Malaysia's electricity system - Power Engineering International", Power Engineering International, 2021. [Online]. Available: https://www.powerengineeringint.com/world-regions/asia/outline-of-malaysiaselectricity-system/. [Accessed: 13- Apr-2021].

[4] "What Happens With Unused Solar Power? Net Metering Explained", ReVision Energy, 2021. [Online]. Available: https://www.revisionenergy.com/blogs/net-metering-101happens-excess-solar- generation/#:~:text=If you produce more solar,power out to the grid.&text=When this happens%2C your unused,use when it's not sunny. [Accessed: 13- Apr- 2021].

[5] J. Yang, A. Paudel, H. B. Gooi, and H. D. Nguyen, “A proof-of-stake public blockchain based pricing scheme for peer-to-peer energy trading,” *Applied Energy*, vol. 298, p. 117154, 2021.

[6] S. Aggarwal and N. Kumar, “A consortium blockchain-based energy trading for demand response management in vehicle-to-grid,” *IEEE Transactions on Vehicular Technology*, vol. 70, no. 9, pp. 9480–9494, 2021.

[7] K. Heck, E. Mengelkamp and C. Weinhardt, "Blockchain-based local energy markets:

Decentralized trading on single-board computers", Energy Systems, 2020. Available: 10.1007/s12667-020-00399-4.

[8] A. Esmat, M. de Vos, Y. Ghiassi-Farrokhfal, P. Palensky and D. Epema, "A novel decentralized platform for peer-to-peer energy trading market with blockchain technology", Applied Energy, vol. 282, p. 116123, 2021. Available: 10.1016/j.apenergy.2020.116123.

[9] "P2P Energy Trading", Irena.org, 2021. [Online]. Available: https://www.irena.org/-/media/Files/IRENA/Agency/Publication/2020/Jul/IRENA_Peer-topeer_trading_2020.pdf?la=en&hash=D3E25A5BBA6FAC15B9C193F64CA3C8CBFE3 F6F41. [Accessed: 13- Apr- 2021].

[10] "What recent trials teach us about peer-to-peer power trading", Power-technology.com, 2021. [Online]. Available: https://www.power-technology.com/features/peer-to-peerenergy-trading- p2p-irena-malaysia-seda-power-ledger-blockchain-automation/. [Accessed: 13-Apr- 2021].

[11] V. Brilliantova and T. Thurner, "Blockchain and the future of energy", Technology in Society, vol. 57, pp. 38-45, 2019. Available: 10.1016/j.techsoc.2018.11.001.

[12] A. Ahl, M. Yarime, K. Tanaka and D. Sagawa, "Review of blockchain-based distributed energy: Implications for institutional development", Renewable and Sustainable Energy Reviews, vol. 107, pp. 200-211, 2019. Available: 10.1016/j.rser.2019.03.002.

[13] B. Jose and S. Abraham, “Performance analysis of NoSQL and relational databases with MongoDB and MySQL,” Materials Today: Proceedings, vol. 24, pp. 2036–2043, 2020.

[14] R. Čerešňák and M. Kvet, “Comparison of query performance in relational a non-relation databases,” Transportation Research Procedia, vol. 40, pp. 170–177, 2019.

[15] Á. Vathy-Fogarassy and T. Hugyák, “Uniform data access platform for SQL and NoSQL database systems,” Information Systems, vol. 69, pp. 93–105, 2017.

[16] B. Berry and R. Abueg, “Mongodb vs sql,” Knowi, 18-May-2020. [Online]. Available: https://www.knowi.com/blog/mongodb-vs-sql/. [Accessed: 14-Sep-2021].