Journal of Informatics Education and Research
ISSN: 1526-4726
https://doi.org/10.52783/jier.v3i2.104
Vol 3 Issue 2 (2023)

File allocation among the on trustworthiness areas attached to the distributed environment, using server storage testing method

D.Saravanan Dr.Dennis Joseph
Faculty of Operations & IT ICFAI Business School (IBS), Hyderabad,
The ICFAI Foundation for Higher Education (IFHE)
(Deemed to be university u/s 3 of the UGC Act 1956) Hyderabad-India.

Abstract
This paper focuses on the goal of consistency in the distributed environment, specifically the organization's inability to handle the data storage issues properly. The main drawback of this type of server is that it never keeps track of the connected clients, their capacity, storage size, and more. On the other hand, data redundancy helps the server if any of the connected clients don't work with the help of available copy system functions. It will improve the system's performance and never affect the system's flow. In the proposed work, each client-side system redundancy component is calculated, which will help the user identify exactly how many units of data transfer take place and the exact number of data losses. So any data loss is immediately notified to the user, and particle transactions are re-initialized to avoid data loss.

Key terms: Data storage, redundancy, Data isolation, Data failure, System reliability, Node distribution.

1. Introduction

The main challenge for many distributed environments is system storage, which will affect the major functions of the organizations. During data transfer, clients are not ready to accept the server input, or server communication is poor. If information among the nodes is not communicated properly, it will cause communication issues in the organizations. To avoid this, we need to maintain a highly configured system environment. This paper focuses on trustworthiness areas attached to the distributed environment, especially from the client storage point of view. The main drawback of the distributed system is data redundancy; it will occupy the maximum storage and reduce the system's performance, especially when users need to find any data on a particular client or server. It will increase the system searching time, and the same unwanted data will be stored in multiple systems. Storage utilization will also increase, which will create issues in other aspects. On other aspects, redundancy helps the users, especially on the server side. If a system crash takes place, it will affect other system functions. At this time, a duplicate copy helps to run the operations without any interruptions. Many existing systems reliability is achieved through system functions only; none of the systems focus on the hardware side, such as the network adapter, disc storage capacity, client configuration mechanisms, or other system-related hardware. Here, data may be duplicated, but these hardware components are unable to do so. Most of the hardware components are stand-alone, and due to this, if any of these components are not working or the hardware crashes, system operations get affected. Another drawback in the existing system is its inability to predict error functions. Even after system crashes take place, it will be very difficult to identify which client node is not working, how this error is identified, and which methods are used to solve this issue. To avoid this system duplication or create a duplicate copy and store any of the client nodes, it will help to run the process smoothly, even if an error occurs, and it will improve the system's performance.

The main advance in the proposed system is to create extra storage space on both the client and server; this space helps to store the redundancy copy it may use should any system errors occur. Another important function that is discussed in this paper is any data loss. No one has any clue how much data loss takes place, and due to this, which nodes get affected, all those information's are not yet recorded. In the proposed model, each client-side system redundancy component is calculated, which will help the user identify exactly how many units of data transfer take place and the exact number of data losses. So any data loss is immediately notified to the user, and particle transactions are re-initialized to avoid data loss. This information is stored so that the user can create prevention mechanisms to avoid this type of error in the future.

2. Existing System

Many distributed existing systems are highly dedicated to system reliability and system functions; none of these systems are bothered by the particular node failure effect. If any individual node in the network doesn't work, what will happen? How will communication be resumed without any interference? Another important issue in this system is how unsolvable errors are addressed, how these types of systems are identified, and what mechanisms are used to solve them. Many of these systems

Journal of Informatics Education and Research
ISSN: 1526-4726
https://doi.org/10.52783/jier.v3i2.104
Vol 3 Issue 2 (2023)

only address system reliability; they do not bother with system errors or solutions for this type of error. Many of these systems also provide solutions for the accrued errors but do not provide any mechanism for error prediction in advance. If the errors were predicted well in advance, many system crashes or system failures could be avoided.

2.1 Disadvantages of existing system

1. Many existing systems focus on system reliability functions only.
2. There is no error prediction mechanism.
3. Any unsolvable error is not identified; there is no proper mechanism or guidance for this type of system failure.
4. Many of these systems focus on frequent error patterns.
5. There is no proper solution for individual node failures.
6. There is no proper manual or guide for error prediction or error solving mechanisms.
7. In a distributed network, the data load for each system has not yet been calculated.
8. The capacity of individual systems is not addressed.

3. Proposed Model

The main objective of this proposed work is to show how data redundancy could be avoided if systems are connected through a network. Especially in this work, this type of error increases the storage size, which reduces system performance and reliability. So data duplication in multiple places needs to be avoided. This type of error affects the system storage structure but never creates any issues or problems in the network or network connecting devices. It will affect the system's reliability. Another important function of any connected node is how the system continues its functions without affecting other system functions. As soon as the mechanism identifies the affected device, it must be repaired immediately. To continue the operations without affecting the functions of the existing system, one or more buffer nodes must be added to any failure node; this node will balance the system load.

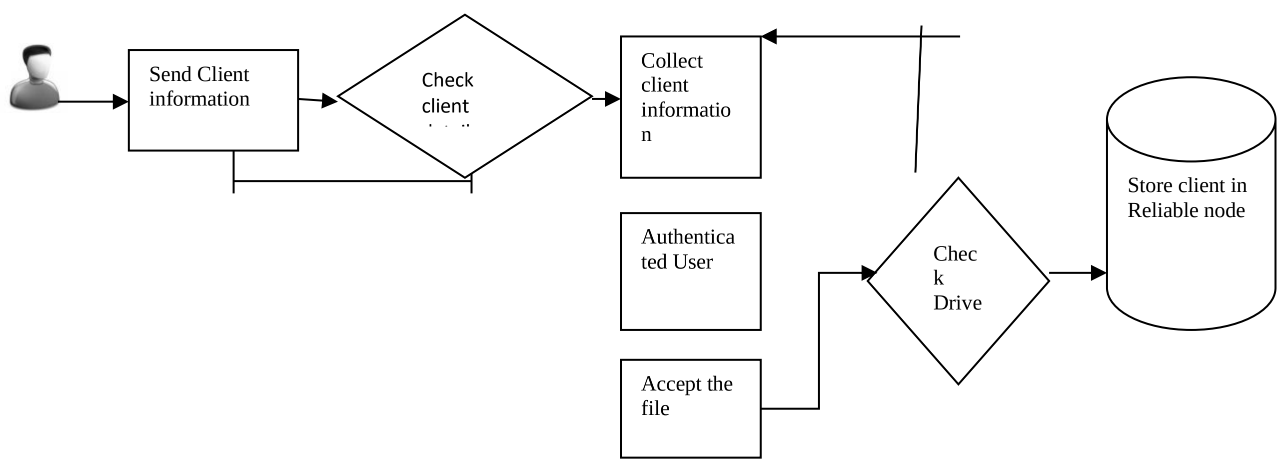

Fig 1. Proposed Architecture

3.1 Advantages

1. A Buffer node is introduced; in the event of a system failure, this node will take care of system load and other functions.
2. System storage is increased in the proposed model, so any overload won't affect the performance of the system.
3. The proposed model easily identified the system redundancy, and due to this, many storage problems were rectified.
4. System duplication was identified and removed.
5. System performance won’t be affected even if any individual node gets disturbed or doesn't work.
6. Alternate systems or nodes are introduced so that system performance is never distributed.

.

4. Experimental design

In the proposed scheme, the data replication problem is confounded because this system occupies more storage for storing the same content in different locations. It will reduce system performance and increase content retrieval time. This model works well in a system-sharing environment, especially in larger enterprises. The proposed model focuses on the system storage point of view to avoid data redundancy. The proposed model easily identified the system redundancy, and due to this, many storage problems were rectified. In the proposed system is to create extra storage space on both the client and server; this space helps to store the redundancy copy it may use should any system errors occur. Another important function that is discussed in this paper is any data loss. No one has any clue how much data loss takes place, and due to this, which nodes get affected, all those information's are not yet recordedMany existing systems focus on this issue in different aspects, but the proposed model focuses on system storage structure. It will improve the system's reliability and also improve its functions

4.1 System experimental design phase:

- Accumulate Client arrangement and File Sending.
- Ensure the Server constrain.
- Create space for client in reliable storage locations.

4.1.1 Accumulate Client arrangement and File Sending

Data sharing in distributed system one of the challenging functions to the users. Where never data get transferred in the distributed system connectivity among the clients should more secure and more reliable. Before any transition take palace system ensure the security and availability of the system before the operations gets executed. Both the sender and receiver get ready for data transfer and accept the input. Performing this operations first check the system availability, in the network user need to ensure all the clients are ready to send/receive the information's. For achieving this first users need to check reliability analysis among the trusted clients it will helps smooth data transfer in the distributed system.

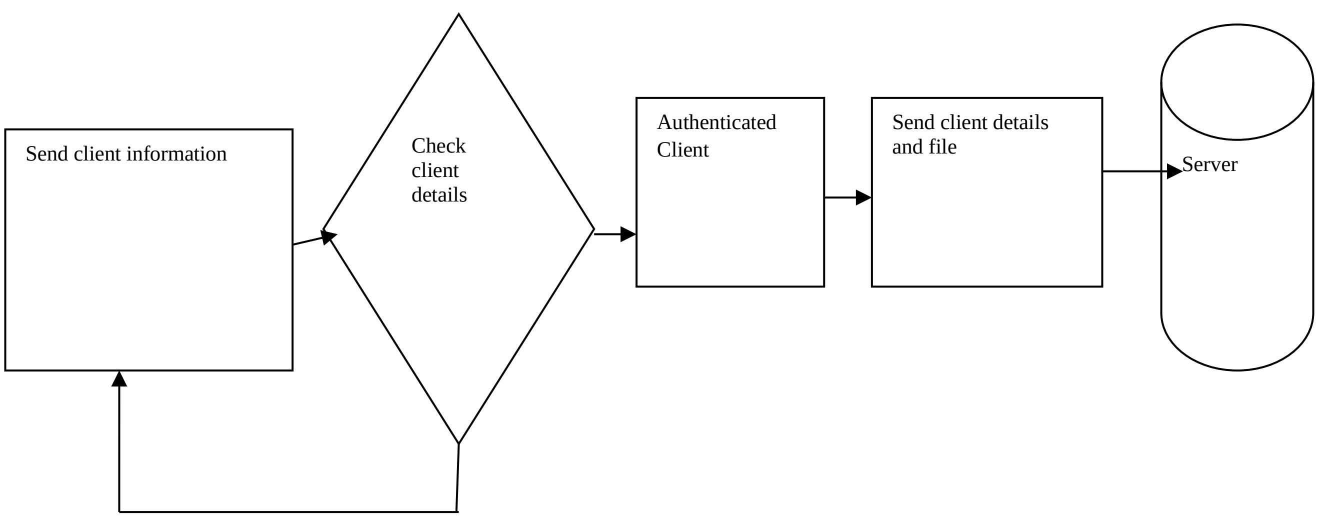

Fig.2 Client file sending process

4.1.1.1 Pseudo code for file sending process:

```
File.Copy(filename, "\\\\" + textBox2.Text + "\\ITDNS02-Server\\" + "\\" + Path.GetFileName(filename).ToString(),
MessageBox.Show("The File is received by the server","Alert", Input,
Input.functions..Data);      }
        else          {
Input.functions..Data ("Please give the server name and browse the file before the sending", "Alert", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation);
```

4.1.2 Ensure the Server constrain

During the data communication, both clients and servers are getting ready to accept the inputs. Any communications between the client and server ensure safety and smooth transactions among the nodes. This module works on any error among the nodes that makes the system unable to transfer any data, if any of these errors need to be identified and retrieved immediately. Otherwise, it will affect the system's performance. This module helps identify the type of error and what mechanisms need to be adopted for this error. It will help the user have smooth operations.

4.1.2.1 Pseudo code for server drive check process:

```
cmd = new SqlCommand("Insert into
Server_Setting(Server_Name,Total_size,Actual_Free_Space,Storage_Space,Restricted_Space)values('" + server_name + "','"
+ Convert.ToInt32(textBox1.Text) + "','" + Available_Hard_disk + "','" + Change. Input(Script. Information)+
Change.Input(Script.Information)+);
                affected = cmd.ExecuteNonQuery();
                if (affected > 0)
                {
                  MessageBox.Show("You have established the setting on the server successfully!", "Alert",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
```

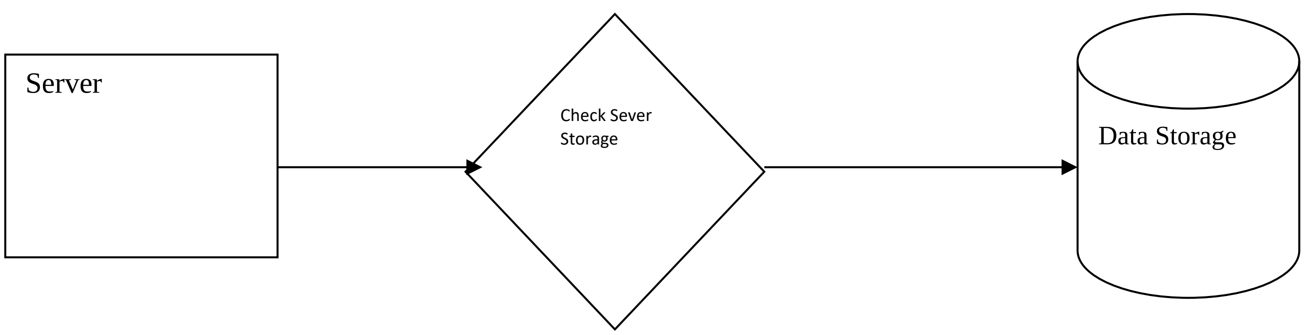

Fig3. Check server status operation.

4.13Create space for client in reliable storage locations.

In a distributed network, whenever clients send any data request, the server first checks the client's authentication, credentials, and more. Based on the data, the server initiates the transaction. The server first selects the client system and transfers data or files to the client. The server also checks the file size, which is communicated to the client. If the size is beyond the communication width, the files are broken into small packets and sent to the client. On the client side, clients check if they have received all the packets. It also checks for any data loss during data transfer; if any loss occurs, it will replicate the transaction again. This module also confirms send/receive data information that is sent by the server or received by the client.

Journal of Informatics Education and Research
ISSN: 1526-4726
https://doi.org/10.52783/jier.v3i2.104
Vol 3 Issue 2 (2023)

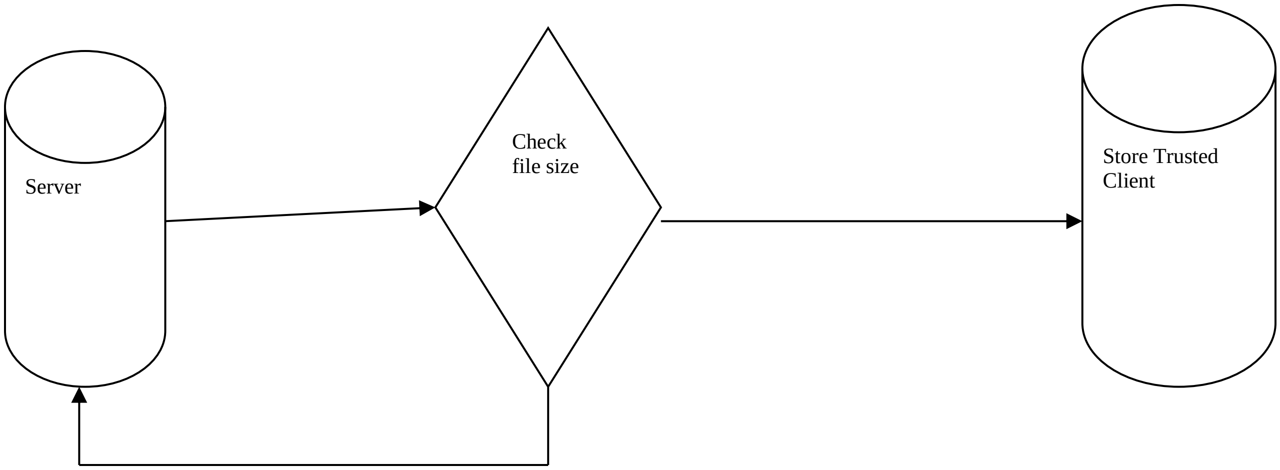

Fig.4 Stores trusted client information.

5. Experimental outcomes:

Fig.5 Getting servers information.

Journal of Informatics Education and Research
ISSN: 1526-4726
https://doi.org/10.52783/jier.v3i2.104
Vol 3 Issue 2 (2023)

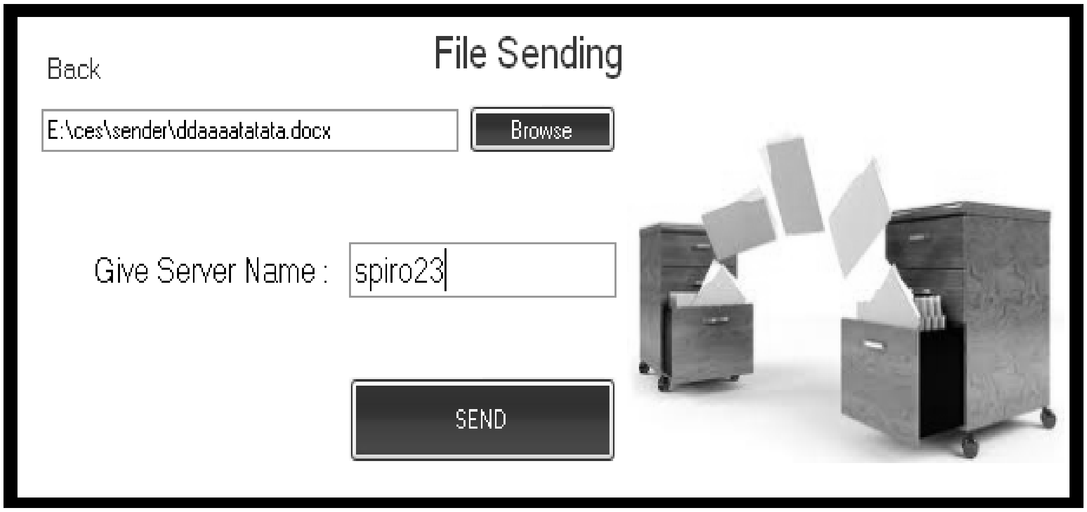

Fig. 6 File sending process.

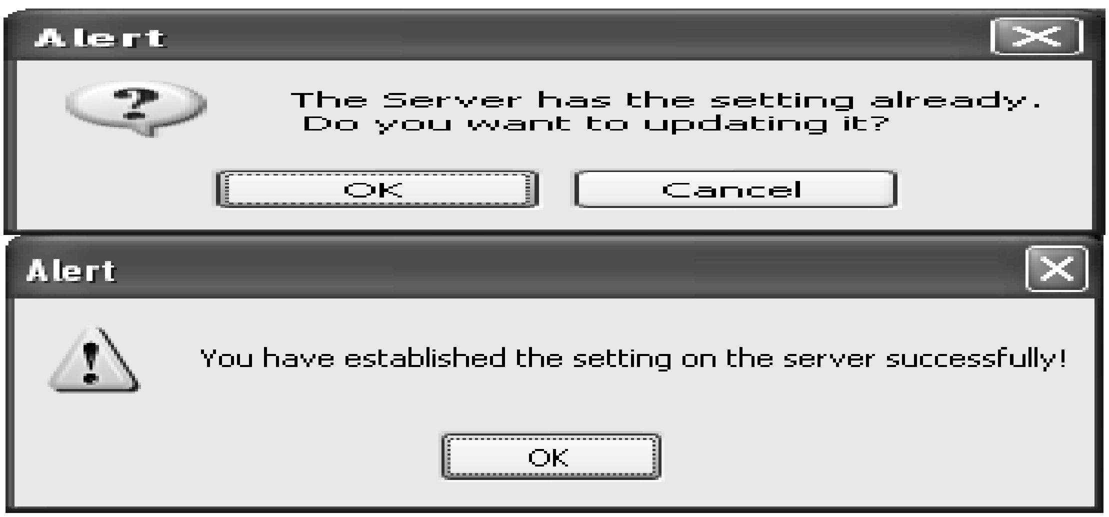

Fig.7 File updated information to the server

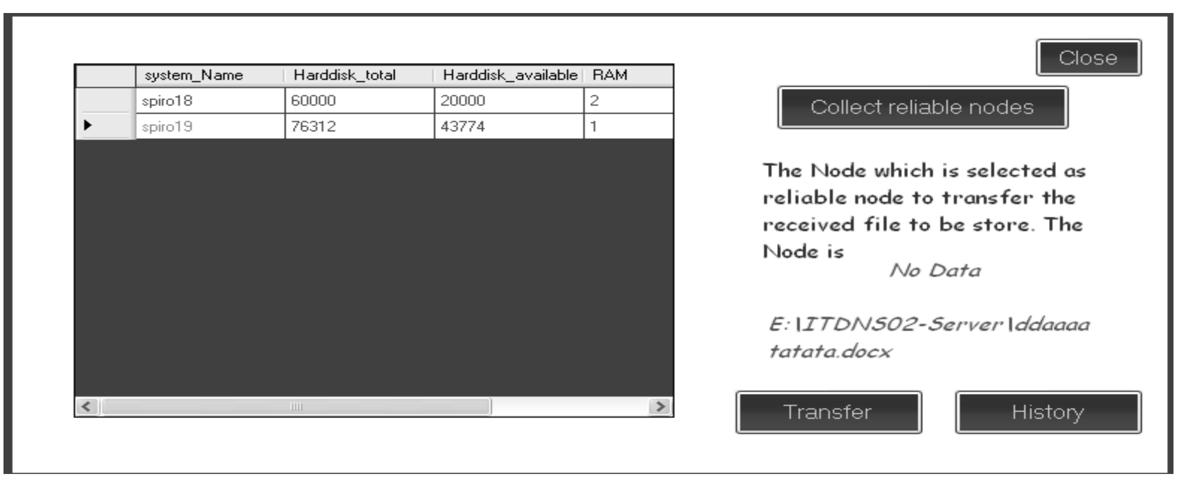

Fig.8 Trusted node information on server side.

ISSN: 1526-4726
https://doi.org/10.52783/jier.v3i2.104
Vol 3 Issue 2 (2023)

6. Conclusion and Future enhancement

The proposed model focused on and achieved system reliability based on system storage concepts. It also ensures any data loss during the transaction; it will automatically identify the client, the type of error, and how to address this error. This system also includes the mechanism for extra storage or an extra node, which will reduce any system crash or error. Due to that, system functions run smoothly without any interference, even if a node failure occurs. Through the system, reliability and system performance improve. In the proposed model, only the system functions (network bandwidth and other network functions) are checked. In the future, the system will be further extended into system hardware such as disc capacity, disc functions, network adapter usage, the link that is used to connect the system to the network, and more hardware functions are considered to improve the system's operations.

7. References

1 Efficient Byzantine-tolerant erasure-coded storage "G. R. Goodson, J. J. Wylie, G. R. Ganger,2004
2 "WEAVER Codes: Highly Fault Tolerant Erasure Codes for Storage Systems" G. R. Goodson, J. J. Wylie, G. R. Ganger,2005.
3 D.Saravanan,Dr. Dennis Joseph," Image data extraction using image similarities", Lecture notes in Electrical engineering, Volume 521, Pages 409-420, ISBN:978-981-13-1905-1,Nov 2018
4 Saravanan, Dr. Vaithyasubramaniyan, Dr. Dennis joseph "Effective Utilization of image information using Data mining technique" , Intelligent systems reference library 172, Volume 172, Pages 207-215, ISBN 978-3-030-32643-2, Nov 2019STAR: An Efficient Coding Scheme for Correcting Triple Storage Node Failures" C. Huang and L. Xu 2005
5 "Failure Trends in a Large Disk Drive Population"E. Pinheiro, W.-D. Weber, and L. A. Barroso. 2007.
6 Vaithyasubramanian, D.Saravanan, C.K. Kirubasankaran, "Communal Fraud Detecting Algorithm for establishing Identity Thefts in online shopping", International journal of e-Collaboration (IJeC), Volume 17, Issue 3, Page 73-84, July 2021
7 D.Saravanan, "Information retrieval using image attribute possessions" Soft computing and signal processing , Advances in Intelligence systems and computing 898, Springer. DOI:10.1007/978-981-13-3393-4_77, Pages 759-767. March 2019
8 "FAB: Building Distributed Enterprise Disk Arrays from Commodity Components"
Y. Saito, S. Frølund, A. Veitch, A. Merchant.2004.
9. D.Saravanan, Y.Immanuel, "Extracting image frames using image property matching techniques", IOP Publishing, Journal of physics: Conference Series Volume 1770(2021)012025, Pages 01-09, March 2021