Journal of Computer Science

Original Research Paper

Efficient Utilization of Mirroring Servers Using Artificial Neural Networks

[1]Radwan Al-Shalalfa, [2]Hazzaa Alshareef, [1]Azzam Sleit, [1,2]Mousa Al-Akhras and [2]Samah Alhazmi

[1]King Abdullah II School for Information Technology, The University of Jordan, Amman 11942, Jordan
[2]Department of Computer Science, College of Computing and Informatics,
Saudi Electronic University, Riyadh 11673, Saudi Arabia

Article history
Received: 14-08-2021
Revised: 29-12-2021
Accepted: 12-01-2022

Corresponding Author:
Hazzaa Alshareef
Department of Computer Science, College of Computing and Informatics, Saudi Electronic University, Riyadh, Saudi Arabia
Email: h.alshareef@seu.edu.sa

Abstract: Due to the fact that an enormous amount of data is available to any user on the Internet, this leads to increased requests of Internet users to download or process their data among different servers. In fact, the need for increasing the reliability and performance of such servers has become an importance subject to tackle. Replicating servers is a way of reducing the overhead of Internet users' requests as well as increasing the reliability and performance of servers. However, there is still a need to redirect users' requests to a single server from those replications so these servers get unknown by Internet users. This technique is called mirroring servers. In this study, a new model that uses the Artificial Neural Networks (ANN) is proposed to select the appropriate server for any new user's requests. In particular, this method considers the current features of servers with the new user's requests as an input and provides the selected server as an output. The effectiveness of the proposed method is compared with two different techniques: Human's selection after eliminating the required time from a user to make the selection and Round Robin selection. This comparison shows that the proposed method takes the advantages of these two techniques, which are based on the speed of selection for the Round Robin selection method and the selection of the best server according to the mirroring server features that are derived from the manual selection method. The results of this study indicate that the proposed model can improve the use of mirroring servers by 10% better than the Round Robin selection method since in this selection method, most of servers are idle in more than 25% of the time and do not have any more requests to serve.

Keywords: Mirroring Server, Round Robin Selection, Load Balancing, Artificial Neural Networks, Machine Learning

Introduction

Server Mirroring techniques have been used for many years to increase the reliability and the performance of servers that receive heavy requests from a considerable number of users over the Internet (Wang *et al*., 2019; Abdel-Hamid and Gulliver, 2003). The use of the Internet is growing rapidly where this growth raises the need to split the load of users to several servers called mirroring servers. Figure 1 illustrates the concept of mirroring servers.

The mirroring servers' technique performs a load balancing process. This newly introduced technique can solve other related problems other than the load balancing problem. Some of these problems comprise (Cao *et al*., 2005; Liu *et al*., 2004):

- Mirroring servers are used for load balancing by splitting the requested task (s) to different servers to reduce the overhead of these tasks on the network since servers are distributed at different networks
- Reducing the time needed to execute the requested tasks by using the best available resource (server), (c) mirroring servers can provide a fault tolerant system by redirecting jobs to functional and operational server when some servers go down and hence, high data availability is achieved
- In mirroring servers, it is not required from a client to decide when to switch from an overloaded server into a lighter server as the selection process is automated
- Concurrent downloads can be employed to increase the speed of the download process by dividing the

request over a considerable number of servers

Since there is more than one copy (mirror) of the server, there is a need for selecting one server to serve new clients' requests. This selection can naturally affect the overall performance of the mirroring servers. When the selection of the server is not carefully executed, a mirroring server can reach a stage where it cannot serve further requests (i.e., it becomes overloaded) and one of the servers might not operate efficiently. Therefore, historical background about the mirroring servers' behavior is needed, including the need for studying the current servers' status in order to have a correct server selection decision for any new request. In fact, this makes the selection of the server more accurate and provides a longer life time for the overall system (Wang *et al*., 2019; Cao *et al*., 2005).

A recent research (Tavakkol *et al*., 2018) introduces a standard approach "Synchronous Mirroring" to construct highly-accessible and error-tolerant storage systems for big enterprises. This approach ensures strong and secure data cohesion by preserving various proper data replicas and concurrently transmitting each update to all of them. Such powerful coherence supplies error-tolerance guarantees for enterprises systems. To achieve high-performance, the researchers proposed two new methods which enable correct and efficient Synchronous Mirroring over Remote Direct Memory Access (RDMA) (Tavakkol *et al*., 2018).

This study attempts to increase the efficiency and reliability of mirroring server technique and hence, several models and techniques have been proposed to tackle the encountered issue (s) of this technique, which are discussed in the literature section. In this study also, an Artificial Neuron Networks (ANN) system is used to find a relation or an association pattern between the input data and output. The input data represents some of the available server features or the network bandwidth and the output represent the selection of the server that best suited for a given request. Since the ANN system represents one of the Machine Learning (ML) techniques, this in fact makes it one of many available solutions that can be adopted to learn from the available training data. One of the strength points of the ANN system is its ability to find hidden underlying relations between the input data and the desired output as well as memorizing the trained data. The contribution of this study does not solely rely on the ANN system, but also on selecting the features that could affect the selection of the server and such selection can affect the overall system's performance and reliability.

Comparing to other recent studies, the proposed neural networks was more efficient because of the two different techniques applied: (1) Human selection after ignoring the time required by the human to do the selection and (2) round robin selection. Therefore, this comparison exposes that the proposed model takes the advantages of these two techniques which are: (a) The speed of selection for the Round Robin selection method, (b) selecting the best server based on mirroring server features from the manual selection method and (c) overcome their disadvantages which are the selection method from the Round Robin and the need of human intervention for the manual selection. The Proposed model makes the selection decision taking the manual selection as a ground truth. The ANN improves the utilization of the Mirroring Servers by 10% better than the Round Robin selection method, since in Round Robin selection method in more than 25% of the total time most of servers are idle and do not have any more requests to serve.

The rest of this study is organized as follows. In next section, a literature review on the reasons behind introducing the mirroring server technique, including the challenges that might emerge and how researchers have managed to solve similar issues is described in detail. Next, the proposed solution that addresses the selection problem is illustrated and the way of modeling the prospective problem is described. Next, the obtained results of the experimental analysis are highlighted based on the processes of the indicated simulation networks, also the obtained results of the proposed techniques are compared with two different techniques. Finally, the conclusions of this study along with the suggestions pertaining to the future research are drawn.

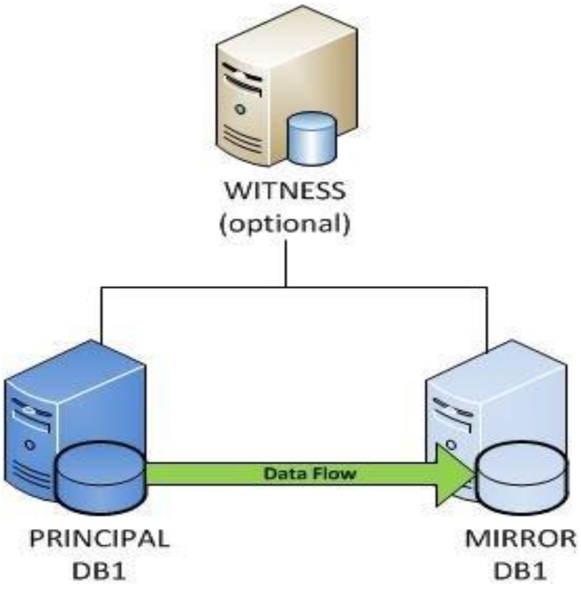

Fig. 1: The concept of mirroring servers

Literature Review

The use of the Internet has rapidly been growing over the past few years, as a result, the network that is accessed from users tend to be heavily concentrated on some major servers, which provide valuable data or processes to most Internet users where the network around these servers is also congested. To distribute the load over multi-servers, the mirror server technique has been widely used. This section describes the reasons behind introducing this technique, the challenges raised after introducing this technique, the performance and reliability of this technique on the client and server, how previous researchers attempted to solve the challenges incurred in the mirroring server techniques, describes some system designs of products of some big companies and finally describes some knowledge about the structure and properties of different Artificial Neural Networks.

Server Mirroring Approaches

Organizations use a high demanded IT infrastructure to share different important data and such data should be available 24 h a day and 7 days a week for all customers and employees. To archive this, multiple copies of servers have been placed in several locations, mirroring server techniques have been applied in order to manage the process related to these servers. In fact, many researchers have been proposing several methods and techniques for tackling different issues incurred by the server mirroring technique.

Mirror servers have been employed for many years on the Internet as a way to increase the reliability and the performance of processing clients' requests in the presence of frequent access by many clients (Myers *et al*., 1999). While mirroring can provide much higher aggregated throughput to a given data item, individual clients must choose an appropriate mirror server in order to achieve a reasonable performance. Unfortunately, only ad hoc mechanisms for choosing the appropriate mirror server are currently employed (Myers *et al.,* 1999). Therefore, there is a need for an intelligent system that can handle the features related to the current mirroring system to make the best choice of a mirroring server for maintaining the current system performance and reliability with the increase in the number of concurrent users that can be served at the same time.

A major challenge of choosing the most effective server among the available servers is the availability of many features that can be used as criteria for deciding which server to use based on a particular client's request. A number of these features comprises, the available memory for each server, the available bandwidth and the average response time. The effect of these features can be changed during the life time of the mirror servers, that is, many different algorithms and techniques are made available to solve the selection problem.

To improve the performance and reliability of such mirror servers, many researchers studied and proposed different solutions regarding the features of the current mirroring system along with their effects on the total produced performance. Such techniques include Brute Force, Genetic Algorithm (GA) and Round-Robin algorithms in order to select the most effective mirroring server.

Authors in (Cardellini *et al*., 1999) attempted to update a web cluster architecture in which the Domain Name System (DNS) server dispatches users' requests among the servers through the URL-name and along to the IP-address mapping mechanism, which is integrated with a redirection request mechanism based on the HTTP protocol. Additionally, the authors compared the use of different mechanisms and conclude that the combination of centralized and distributed dispatching policies can achieve the highest load balancing (Cardellini *et al*., 1999; Yokota *et al*., 2004). Furthermore, in (Rodriguez *et al*., 2000) the researchers proposed a scheme in which clients access multiple mirror sites in parallel to speedup many documents' downloads while eliminating the problem of the server selection. This is based on the use of a dynamic parallel-access technique that leads to dramatic speedups in downloading multiple documents where the load is shared among servers without the need for a server selection mechanism.

As stated in (Jamin *et al*., 2001), the authored investigated a case of whether the current placement of the mirroring server over the Internet affects the overall system performance or not by using some of the mirror placement and heuristics algorithms. Initially, the authors used an algorithm called the Min k-center algorithm, which finds a set of center nodes (mirror servers) for minimizing the maximum distance between a node and its closest center. Then, they used the Transit Node algorithm to search for the out-degree, which is defined as the number of other nodes that are connected to a single node.

Similarly, the researcher in (Gautam, 2002) attempted to determine the optimal number and locations of proxy servers in a network to minimize the throughput delay and demand constraints by assuming that clients' or users' requests at any location can be frequently be sent along to the nearest server to allow modeling each client–server as an independent queuing network. In order to solve this problem, the authors used a heuristic algorithm called the Decompose Evaluate Join Append Verify and Unplug (DEJAVU) algorithm, which is reported to be superior in using different GAs. Likewise, authors in (Matthur and Mundur, 2003) attempt to avoid the participation of clients in balancing the load across servers. To achieve this, they proposed two new protocols, called the Centralized Control Protocol (CCP) and the Distributed Control Protocol (DCP). In the CCP protocol, servers can periodically send a state information to the central server by indicating their current load. In the DCP protocol, a set of servers form a token passing arrangement for serving each client's request. Similar to the previously indicated researchers. In (McManus, 1999) the study illustrates the use of a heuristic algorithm for selecting the closest available web server from a group of mirrors. This algorithm is based on the Border Gateway Protocol (BGP) that represents multiple path lengths, which can be determined without the introduction of any additional traffic measurement into the network.

Furthermore, in (Chow and Cai, 2003) the authors attempted to enhance the process of downloading a file into the mirror servers by representing two problems that can initially search for the maximum parallel download speed without restricting the number of mirror servers and that can after that, search for the best group of k servers for parallel downloads. They represented the problem as a graph and suggest two ways for solving both problems by applying Brutal Force algorithms from which they obtain the worst-case scenario of O(n) and O(nk) for the first and second problems, respectively, second way the authors used fixed length GA and a variable-length GA (Chow and Cai, 2007). At the same period of the investigational research, the researchers in (Sleit *et al*., 2007) use the GA in order to select a server among the available mirroring servers for distributing a query that can archive load balancing based on two main features in the mirroring server. The two features comprise; the average server processing time and the average reply time. Further, they modeled the problem by a main server called the load balancing server (Sleit *et al*., 2007). The client request is transformed to the load balancing server, which can redirect the request to the appropriate server by using different GAs. This a typical use of GAs in optimization problems like its usage in other studies (Al-Akhras, 2008; AL-Akhras and Saadeh, 2010; AL-Akhras, *et al*., 2011a).

As stated in (Nakaniwa *et al*., 2009), the researchers proposed an optimal mirror allocation model by presenting the topology of a network as an adjacency matrix A, whose elements are weighted by the distance between a pair of nodes. Additionally, they calculated the shortest path matrix Q and formulate the reliability, cost and delay by a matrix Q (Nakaniwa *et al*., 2009). In (Nakamura *et al*., 2009) the authors attempted to study the issues related to mirror servers by proposing two new heuristic algorithms for tackling the encountered issues and for assessing the algorithms when a few real networks are implemented. It was proven from the results that their proposed algorithms can function effectively, which are approximately likewise to the effectiveness of the obtained results in (Nakaniwa *et al*., 2009).

In the same context, the researches in (Maeda and Miwa, 2012) tackled the issues related to mirror servers by defining an issue that is related to a reliable network design based on protecting critical links whose failures can worsen the produced performance. First, they defined such an issue and afterwards, provided an evidence that demonstrates an NP-hard problem. Second, they proved that this problem can be resolved based on a polynomial time by applying a polynomial-time algorithm that can resolve the problem once the number of concurrently failed link is restricted to the other. Additionally, they proved that the solution of the problem is based on a polynomial time whilst a hop count is also restricted to the other (Maeda and Miwa, 2012). A similar study of the NP-hard problem pertaining to mirror servers has later been investigated by authors in (Hillmann *et al*., 2016), where they handled the properties that are related to a content delivery network. In their simulation, many different realistic scenarios have been studied and analyzed and many different performance indicators have been assessed. The obtained results showed an effective enhancement on their investigations and analysis.

In (Irie and Miwa, 2017), the authors investigated the issue of mirror servers by searching for protected links in order to satisfy the conditions, which involve the fragmentation and stretch factors. First, this issue is formulated by them to ensure that it represents an NP-hard problem. Following that, a polynomial-time algorithm for solving such an issue in which the number of the simultaneous link failures is restricted to another one is presented. Furthermore, a polynomial-time approximation algorithm is also applied by the authors based on the approximation ratio, which indicates the number of simultaneous link failures. Moreover, the approximation algorithms are implemented onto the actual networks' topology where the approximation ratio is assessed. Similarly, the authors in (Govindan *et al*., 2018) proposed producing a new mirror server selection policy by restraining the number of retransmissions, TCP connect time, average throughput and Round-Trip Time (RTT). Their produced policy is seamlessly examined through different Android devices. It can be observed from their obtained results that their proposed policy can perform more effectively when the minimized RTT, average throughput, the number of encountered retransmissions and different energy consumptions are considered.

The researchers in (Sabareesh *et al*., 2019) proposed a novel Redundant TCP Connector (RTC) method to produce connections that can simultaneously use various existing interfaces of a network so that these interfaces can be dynamically connected to the most effective path of a network whenever required. The reason behind this proposition is to enable a client to know which existing network path could be the most effective when required. The authors in (Sun and Nakhai, 2020) studied a Multi-access Edge Computing (MEC) network that includes multiple users with a single Base Station (BS). On this basis, they improve an Online Mirror-prox Optimization (OMO) algorithm in order to reduce the entire delay encountered in a network for achieving the task completion. The obtained results of their produced simulation show effectiveness of their produced algorithm and an enhancement on the consumption of lower battery of users' devices (Sun and Nakhai, 2020; Sadrhaghighi *et al*., 2021; Raghul *et al*., 2017; Yu *et al*., 2019; Hamdan *et al*., 2021).

It can be inferred from the literature that the researchers have focused on issues related to mirror servers and hence, they have attempted to develop different methods and techniques for tackling such issues based on different disciplines of their research. Results reported by previous researchers showed that the proposed approaches have achieved most effective performances and better than other parametric sides based on their own perspectives, deductions and experimental analysis. In light of these results, it is worth to highlight the investigations of the many features conducted by the current research when tackling the issues involved in mirror servers.

System Design Approaches

In this section several system mirroring designs of some products by some companies are analyzed.

Amazon Web Services (AWS)

Inbound and outbound traffic from the network interfaces associated to the user’s Amazon Elastic Compute Cloud (EC2) instances is copied using Traffic Mirroring. The user can transmit the mirrored traffic to another EC2 instance's network interface or to a Network Load Balancer with a UDP listener. The traffic mirror source and target (monitoring appliance) can both be located in the same Virtual Private Cloud (VPC). They could also be in separate VPCs connected by intra-Region VPC peering or a transit gateway.

Consider the scenario in Fig. 2 below, in which you wish to mirror traffic from two sources (Source A and Source B) to a single traffic mirror destination (Target D). The steps listed below must be followed:

- Identify the traffic mirror sources (A and B)
- Configure the traffic mirror target (D) and traffic mirror filter (A).
- Configure the traffic mirror session for Source A, Filter A and Target D
- Configure the traffic mirror session for Source B, Filter A and Target D

Any traffic that matches the filter rules is wrapped in a VXLAN header when you create the traffic mirror session. It is then delivered to the intended recipient (AWS Documentation, 2021).

Microsoft’s SQL Server Database Mirroring

The operation of a database mirroring session might be synchronous or asynchronous. Transactions commit without waiting for the mirror server to write the log to disk in asynchronous mode, which improves performance. A transaction is committed on both partners in synchronous operation, but at the penalty of higher transaction latency.

There are two operational modes for mirroring. high-safety mode and high-performance mode.

High-safety mode allows for simultaneous operation. When a session is started in high-safety mode, the mirror server synchronizes the mirror database with the principal database as rapidly as possible. A transaction is committed on both partners as soon as the databases are synchronized, at the penalty of increased transaction latency.

High-performance mode executes asynchronously. The mirror server attempts to keep up with the principal server's log records. The mirror database may be a little behind the primary database. The distance between the databases, on the other hand, is usually modest. However, if the principal server is overloaded or the mirror server's system is overburdened, the gap can grow large.

The principal server sends a confirmation to the client as soon as it transmits a log record to the mirror server. It does not wait for the mirror server to acknowledge it. This means that transactions are committed without having to wait for the log to be written to disk by the mirror server. The principal server can run with minimal transaction delay thanks to this asynchronous activity, but there is a danger of data loss. All database mirroring sessions support only one principal server and one mirror server. This setup is depicted in Fig. 3 (a).

A third server instance, known as a witness, is required for high-safety mode with automated failover. The witness, unlike the other two partners, does not work for the database but it enables automatic failover by ensuring that the primary server is operational. If the mirror and the witness remain connected to each other after both have been disconnected from the principal server, then the mirror server commences automatic failover. An illustration with a witness is depicted in Fig. 3 (b). (SQL Server Documentation)

Google Search Appliance (GSA)

GSA mirroring is a feature that allows one search appliance's index to be mirrored to one or more other search appliances. There are active-active and active-passive mirroring configurations:

- To achieve high availability serving, use an active-passive architecture in which all search queries are forwarded to the master search appliance
- To enable high capacity serving, use an active-active setup in which search queries are split across the master and mirror search appliances

Figure 4 shows Google Search Appliance setup with replication.

Artificial Neural Network

Since the invention of digital computers, humans have attempted to create machines, which can directly interact with the real world without any intervention. In this sense, the Artificial Intelligence and Artificial Neural Networks (ANN's) approaches design to simulate the way the humans brain analyses information (Al-Akhras *et al*., 2011b). The ANN system has been developed as a generalization for many different mathematical models related to various biological nervous systems. A first wave of interest in neural networks (also known as connectionist models or parallel distributed processing) has emerged after the introduction of simplified neurons in (McCulloch and Pitts, 1943).

The basic processing elements of neural networks are called artificial neurons, or simply neurons or nodes. In a simplified mathematical model of a neuron, the effects of the synapses are represented by connected weights that model the effect of the associated input signals and the nonlinear characteristic exhibited by neurons, which are, in turn, represented by a transfer function. The neuron impulse is afterwards computed as the weighted sum of input signals, which are transformed by the transferring function. The learning capability of an artificial neuron is achieved by adjusting the weights of the connection in accordance to the chosen learning algorithm.

Some researchers are still investigating the neurophysiology of a human's brain, but much attention has currently been paid to the general properties of a neural computation by using simplified neural models. These properties include trainability, generalization, nonlinearity, robustness, uniformity and parallelism (Tebelskis, 1995). The ANN system includes a number of nodes and layers, which are explained in (Kaastra and Boyd, 1996). Also, there are several architectural types of the ANN system where the majority of these types are being used and may comprise feedforwarded ANN and feedback/recursive ANN. The learning methods of the ANN system can be classified into supervised learning, semi supervised learning, unsupervised learning and reinforcement learning. The Backpropagation algorithm (supervised learning) is considered one of the training methods that employs a gradient descent technique for adapting the neural network weights to minimize the mean squared error difference between the neural network output and the desired output (El Hindi and Al-Akhras, 2011; Heaton, 2011).

In the context of this study, using machine learning approach is considered effective when selecting the most effective server for a provided request to the mirror servers and hence, a new load balancer method is proposed in this study and is discussed in the following section.

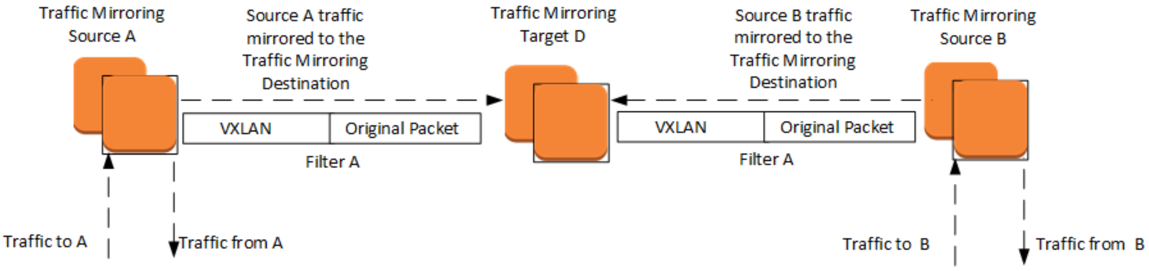

Fig. 2: An Amazon AWS traffic Mirroring Scenario (AWS Documentation, 2021)

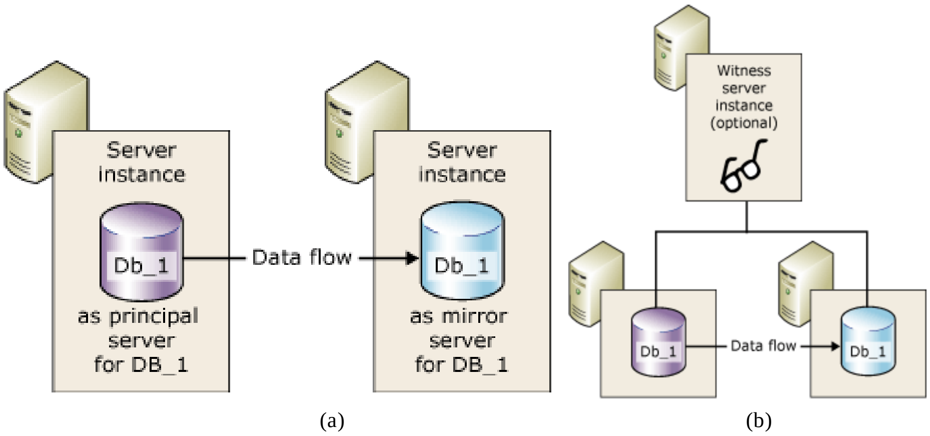

Fig. 3: SQL Server database mirroring (SQL Server Documentation, 2020); (a) Mirroring setup; (b) Witness server with failover setup

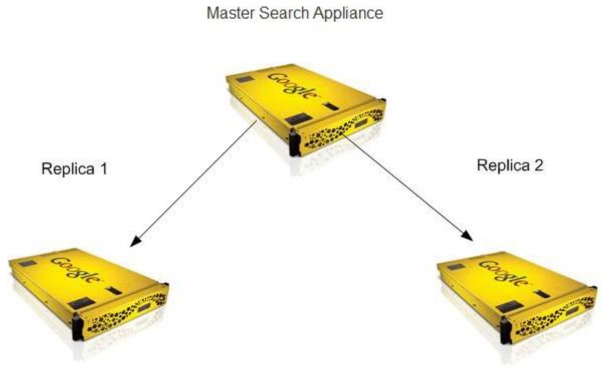

Fig. 4: Google Search Appliance setup with replication (Google Search Appliance Documentation, 2015)

The Proposed Method: The Neural Load Balancer Method

The main goal of this research is to use machine learning approach for selecting the best server for a given request to the mirror servers. Other goals include using existing features for improving the accuracy of the neural network selector by comparing its performance with other selection methods. The results are evaluated by using different performance metrics, such as the system performance, execution time, server delay and memory.

In the proposed system, the Internet servers and clients are simulated as points, which are distributed in two dimensional areas (XY-axis). In these areas, clients and servers are replaced randomly inside the defined area where each client has a direct access to the server. The path from any client to each server has a lot of routers and is viewed by a single line between the client and each server. Any request from the client is passed first through to the main server (decision server), which has a priori knowledge about each server's hardware specification such as the CPU and memory, including the status such as the available memory and CPU along with all assigned requests. This server is responsible for selecting the best server from the available servers. The selector performs the following major steps: (a) Evaluate the new client request, (b) evaluate the current system status, (c) depending on the above two steps, select the best server from the available servers, (d) assign the selected server to the new request.

Training the ANN system starts by generating a training set for this system based on examining different situations of all servers in the mirroring server at different times and by selecting the most appropriate server for each situation. This data is afterwards passed through to a newly created ANN system in order to determine the best values for the hidden parameter and constants. Such data is generated from a selection method called the manual selection method. The back-propagation algorithm is used as a learning algorithm for the proposed neural network where the trained ANN system is used in the main server to select the server for any new situation other than the one that is found in the training set.

Figure 5 depicts the processes that are carried out at the main server. First it receives the request from a client. Received requests are handled on a one-by-one basis in a queuing order. The server examines the current status of all servers of the mirroring environment upon serving the client request e.g., the CPU that is used at each server, the physical memory that is available at each server and the expected time that is required from the server to complete the current running request. These values are afterwards converted into a format that is accepted by the trained ANN system. These values are also passed to the network as an input and output of the trained ANN system, which represents the server that is most appropriate for the input client request. After that, the client request is forwarded to the selected server, which represents the output from the ANN.

The implementation of the proposed solution is divided into two phases: (a) The processes of the main server, the pseudocode of this phase is shown in (Algorithm 1); and (b) the processes of other servers, the pseudocode of the second phase is illustrated in (Algorithm 2).

Algorithm 1: Main server process

```
start simulation
while the simulation is not stopped
begin
    for each request from the client that are not assigned
    begin
        examine all status for each server and the need for the client request
        transfer the collected data to the trained ANN
        examine the output of the ANN then determine the selected server
        pass the client request to the selected server
    end
    increment the simulation time
end
```

Algorithm 2: Servers' processing

```
start simulation
while the simulation not stopped
begin
    for each server in the system
    begin
        determine the number of slots that the server can generate in single round
        serve request assigned to the server as much as the server slots
    end
    increment the simulation time
end
```

The used feed-forward ANN architecture has three layers as follows: (a) The input layer, which accepts the features of the mirroring servers as a number of inputs. This layer contains input neurons that are similar to the number of input features, (b) the output layer, which is responsible for selecting the best server, consists of two neurons, since there are four servers that use the binary presentation for their server's id. Only two neurons are required at this layer that uses the *logsig* transfer function, which produces a value between "0" and "1", (c) the hidden layer between the input and the output layers. A rule of thumb is used to determine the number of hidden neurons as = "(the number of input + the number of output) * (2/3)".

The *trainlm* is used as a training function for the proposed method in this study, since it is the fastest transfer function for back-propagation algorithm and the most recommended for supervised learning. The simulated networks are built to contain a number of servers and broker (main server), which are responsible for delivering the request to the appropriate server that is spread into a predefined area. Furthermore, the number of clients is spread into the same area.

In order to compare these techniques, networks are built to contain a predefined structure of servers and clients, but with different requests. This network is connected with the following main components:

- Servers: Five servers are generated to serve network clients where each of them has the following hardware specifications
    - Memory: Each server has "2–4" Gigabytes (GB) of memory
    - Central Processing Unit (CPU): Each server has "2-5" Gigahertz (GHz) of the CPU
- Clients: 200 clients are generated to request services from the above defined servers.
- Brokers: One of the servers that are defined above is used as a broker

Experimental Analysis

Three different selection techniques have been studied and compared in the simulated networks to have a clear decision about the performance of the proposed technique.

Selection Methods

Round Robin Selection Method: Is an arrangement of choosing all elements in a group equally in some rational order, usually from the top to the bottom of a list and then starting again at the top of the list and so forth (Cao *et al.*, 2005). This technique is used to implement one of the chosen techniques by selecting the first available server (element) then selecting the second (element) till the last server and then starting again from the first available server (element) and so forth. This technique usually does not take any of the server hardware specifications or current status into account in its decision as this raises concerns about the effectiveness of this technique in selecting the best mirroring server.

The reason behind choosing the round robin selection method in this research, because it is still used in many legacy systems due to its simplicity, easy implementation and starvation-free.

Manual Selection Method: Is an arrangement of selecting all elements into a group according to some criteria that could change from one time to another depending on a user's interaction. Two different criteria are used to select the best server (element):

- CPU time: Selecting the best server depending on the time that is taken from the server to complete the current assigned request
- CPU time and request time: Selecting the best server depending on the time that is taken from the server to complete the current assigned request and also this new request

Among the possible features, there are the following features; the available memory for each server, available bandwidth and average response time. The effect of these features can be changed during the life time of the mirror servers, that is why many different algorithms and techniques are exist which attempt to solve the selection problem. To improve the performance and reliability of such mirror servers, many researchers have been working to study the current mirroring system features and their effect on the total performance. Such solutions have used many different optimization algorithms such as Brute Force, Genetic Algorithm (GA) and Round-Robin to select the best mirroring server.

Neural Load Balancer Selection Method: The ANN system is trained by depending on the results of the second technique (manual) as a ground truth.

The simulated networks use the three techniques to select the best server among the other servers for the randomly generated requests that are derived from different clients. During the live time of the simulation, these requests are generated based on the following five experiments:

Experiment 1: Is executed within 10 sec from the simulation time and serves 200 randomly generated requests, which have started at different times during the simulation.

Experiment 2: Is executed within 20 sec from the simulation time and serves 500 randomly generated requests, which have started at different times during the simulation.

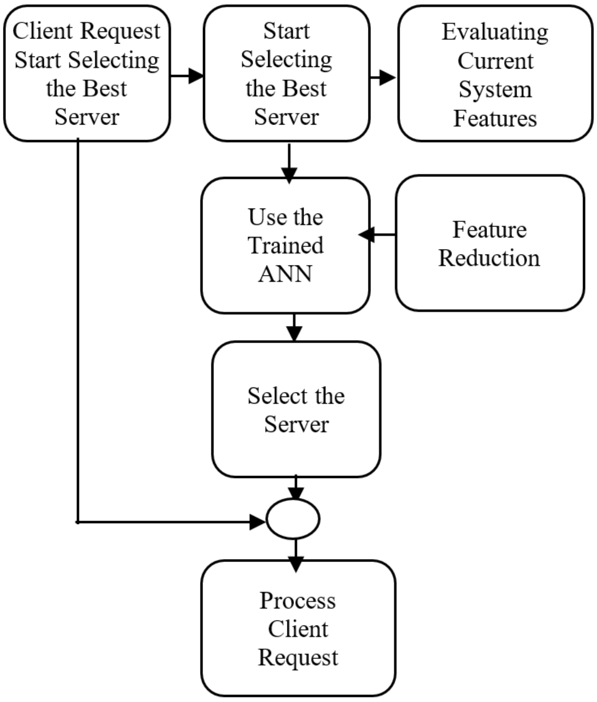
Client Request Start Selecting the Best Server
Start Selecting the Best Server
Evaluating Current System Features
Use the Trained ANN
Feature Reduction
Select the Server
Process Client Request

Fig. 5: The process of the main server

Experiment 3: Is executed within 30 sec from the simulation time and serves 1000 randomly generated requests, which have started at different times during the simulation.

Experiment 4: Is executed within 40 sec from the simulation time and serves 10000 randomly generated requests, which have started at different times during the simulation.

Experiment 5: Is executed within 50 seconds from the simulation time and serves 100000 randomly generated requests, which have started at different times during the simulation.

The first three experiments have been executed in the three selection methods and the remaining experiment has been executed in the Round Robin and Neural Load Balancer selection methods since it takes time by the user in the manual selection to select the best server. All of the above experiments are randomly generated more than once, but the results from executing them possess an approximation of the same result.

Results and Discussion

When running the previously mentioned experiments, three metrics have been taken into consideration in order to compare between the results of each technique. These metrics are:

- Waiting time: This measures the number of slots that have been taken from any assigned request to be served at any server including the idle slots (waiting for another request to complete after it has started its execution). The calculation of these slots starts after the request is being served by a server. This can measure how much the server is overloaded. In an overloaded server, the request spends much of its time in an idle status and is not executed at all. This, in fact, may lead for encountering a delay in the request and overhead at the server's side
- Memory: it measures the amount of available memory at the server when attempting to assign a new request to the server
- Turnaround Time: Measures the amount of time the request is spent at the assigned server after it starts its execution, each server has different CPU power, which means that it can serve different numbers of requests at the same time and this implies that each server can provide different number of slots

According to the definition of each of the used techniques, some behaviors are highlighted as follows:

- The Round Robin selection method selects the servers in the same order, it will most likely split the requests equally between them. If it is assumed that each server will be available at any round, this will cause the same server with a delay since any request at any server will wait for the same number of requests
- The manual selection method selects a server according to the user's criteria, it will reflect these criteria in the requested result. For instance, if the CPU is used as a criterion for selecting the best server, then the requests are split into different groups with different numbers of requests at each group according to each server's power where the most powerful server possesses the largest group. The main drawback of this method is the delay during the selection process since the user is slower than the computer and this causes an overhead and makes the method inapplicable in real scenarios
- The ANN system is used to find an association between the input data and the output, it reflects the behavior of the training data. If the manual data is being used for training the ANN system, then this system will most likely be functional and similar to the manual selection method. This means that ANN method will take the benefits of the manual method and eliminate its main problem, which is the delay caused by the human selection process

Figure (6-11) demonstrate the obtained results, which are derived from the first three experiments since they are executed in all of the three experiments.

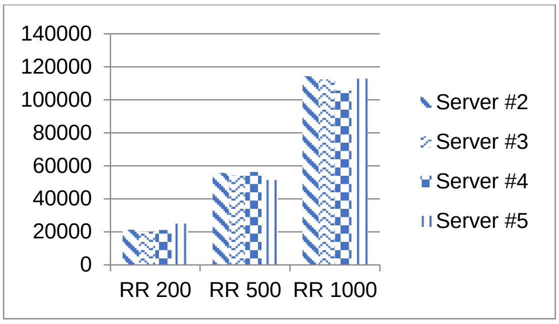

Fig. 6: The waiting time at each server after using the Round Robin selection method

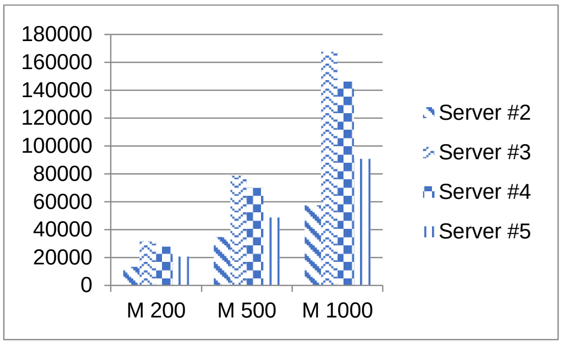

Fig. 7: The waiting time at each server after using the manual selection method

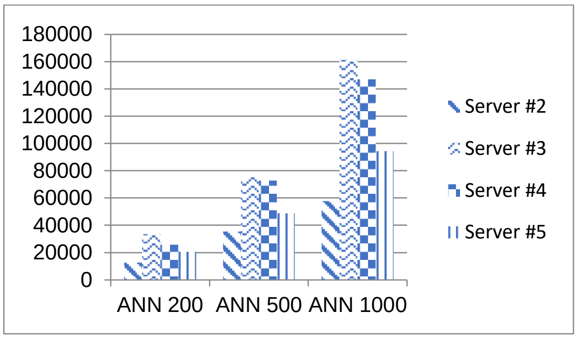

Fig. 8: The waiting time at each server after using the ANN selection method

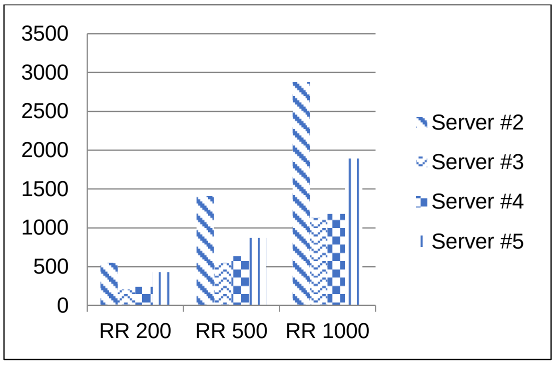

Fig. 9: The turnaround time at each server after using the Round Robin selection method

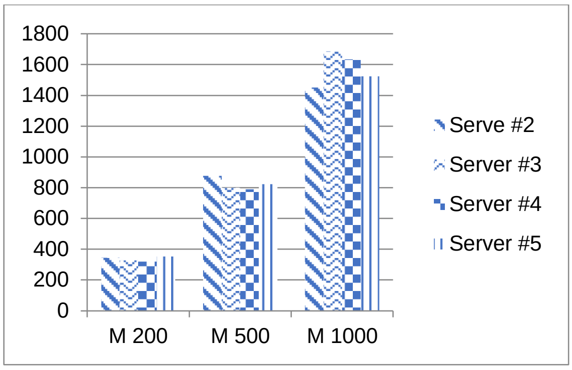

Fig. 10: The turnaround time at each server after using the manual selection method

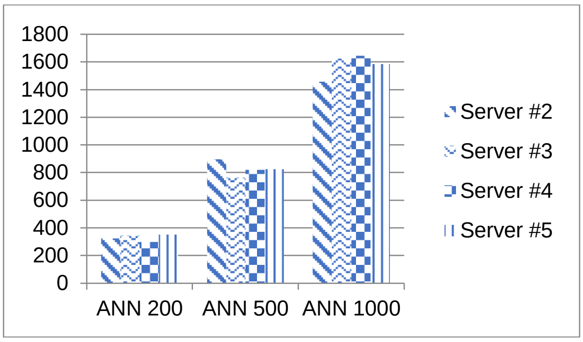

Fig. 11: The turnaround time at each server after using the ANN selection method

Waiting Time

Since many requests are assigned to each server, the average waiting time is calculated for each request. Figure 6 shows the average waiting time at each server in each experiment that is executed at the simulation network according to the Round Robin selection method. From Fig. 6, it can be observed:

- All servers have approximately the same average waiting time, since they have the same number of requests and this means that any request at any server will wait for the same number of slots according to the power of the server. Therefore, requests have the same amount of delay in terms of slots
- The difference between the servers does not reflect the CPU power of the server, since each request needs different numbers of slots, this means that the request with the least needed slots will complete first and will reduce the total number of assigned requests and this makes a difference between the servers in the number of remaining requests in each one
- Since the Round Robin selection method finds the most available server, this implies that the most powerful server will have more requests than the least powerful server, which will also cause difference

between the servers in terms of the waiting time

Figure 7 illustrates the waiting time at each server in each experiment that is executed in the simulation network according to the manual selection method for obtaining the best server. It can be observed from Fig. 7 that each server has a different average waiting time since each server has different numbers of requests, which is caused by a user's selection. This means that any request at each server will wait for different numbers of requests to be served before the cycle comes through to it, which will cause the difference in the waiting time. Most powerful server has the most average waiting time and the least powerful server has the least average waiting time. Figure 8 demonstrates the waiting time at each server in each experiment that is executed in the simulated networks based on the trained ANN selection method as for obtaining the best server.

The following observation about the average waiting time can be obtained from Fig. 8:

- All servers have different numbers of average waiting times, since the ANN network attempts to learn its behavior from the training set that were collected based on the manual selection method. The behavior of this trained ANN approximates the behavior of the manual selection method, which means different numbers of requests are assigned at each server.
- These differences in the average waiting time are not the same as the manual selection method since this data has not been seen by the trained ANN.

Turnaround Time

Since many requests have been assigned to each server, the average turnaround time is calculated for each request for each server at each experiment as illustrated in Fig. 9 which uses the Round Robin selection method as an appropriate selection method for obtaining the best server.

The following points can be observed from Fig. 9 regarding the turnaround time at each server are observed when using the Round Robin selection method:

- All servers have different average turnaround times since they have the same number of requests, which means that when a server is more powerful, it will complete its assigned requests first and the least powerful server will complete its assigned requests last
- Average turnaround time observations do not contradict with the waiting time observations as in each server, different number of slots can be generated at any time depending on the server's CPU power. This, in fact, implies that if a server can generate five slots at a time, another server will accordingly generate 10 slots at the same time. This also implies that the first server needs two times the number of slots as the first one, which is the turnaround time
- The most powerful server has the least turnaround time and the least powerful server has the most turnaround time

Figure 10 shows the turnaround time at each server in each experiment according to the manual selection method. From Fig. 10 the following observations can be made about the turnaround time.

Each server has approximately the same average turnaround time since each server has different numbers of requests caused by the user selection. Consequently, in order to enable a server to complete its request, user's criteria should be considered for selecting the appropriate server, which leads to the fact that the most powerful server to acquire represents the greatest number of requests and the least powerful server to acquire represents the least number of requests. Therefore, servers will approximately complete all of the requests at the same time.

The differences among the turnaround times also reflect the power of the server since each request needs different numbers of slots that will allow the most powerful server to complete its assigned request faster than the least powerful server.

Figure 11 demonstrates the turnaround time at each server in each experiment that is executed in the simulation network using the trained ANN system as a selection method for obtaining the best server. Based on this figure, the following observation can be made:

- All servers have approximately the same average turnaround time since the ANN system attempts to simulate the behavior of the trained data collected using the manual selection method which includes different numbers of requests to be assigned for each server
- Those differences in the average execution time are not the same as the manual selection method since this data has not been seen by the trained ANN system

Memory

Since many requests have been assigned for each server, the average available memory is calculated at each server when assigning the new request. This value is considered as the available memory for the corresponding server at each experiment. Figure 12 depicts the available memory at each server in each experiment that is executed in the simulation network, which uses the Round Robin selection method.

Based on Fig. 12, all servers have different available memory since they possess the same number of requests. After that, the available memory at each server will depend on the total physical memory that is installed at each server. Server 2 has the largest physical memory, while Server 3 has the smallest physical memory.

Figure 13 shows the available memory at each server in each experiment that is executed in the simulation network based on the manual selection method. Based on Fig. 13, the available memory at each server is observed using the manual selection method. All servers have different available memories since they possess the same number of requests. The available memory at each server depends on the total physical memory installed at each server. The manual selection method attempts to minimize the differences among the servers' available memory.

Figure 14 shows the available memory at each server in each experiment that is executed in the simulation network based on the ANN method to select the best mirror server.

From Fig. 14, the available memory at each server is observed when using the manual selection method. All servers possess different available memories. However, it can be seen that the behavior pertaining to this technique is similar to the manual selection method.

A Summary of the Results

Based on the previously indicated experiments, the following comparison results among the three different selection methods are obtained. First, the Round Robin selection method makes the difference among the servers' waiting times the minimum among all compared selection methods. This means that all requests in all servers possess an approximation of the same waiting time, but in the manual and ANN selection methods, the difference between the servers' waiting time is variant, which implies that the waiting time of the requests is different and depends on the server's hardware such as the CPU and memory. These results are illustrated in Table 1. Second, the manual selection method makes the difference among the servers' turnaround time the minimum with the entire selection methods, which means that all servers will complete all assigned requests at the same time as the ANN system. However, the servers in the Round Robin selection method possess a powerful hardware specification, which completes all assigned requests as an initial stage where the least powerful servers complete the assigned requests at the final stage. Table 2 highlights the turnaround time for the entire servers of the third experiment.

In general, the manual selection method and the ANN selection method can increase and decrease the waiting time for all requests in order to minimize the differences among the turnaround times for the entire servers.

Table 1: The waiting time for each server of the third experiment

Server Number	Manual	Round Robin	ANN (the proposed system)
Second	57564	114392	57807
Third	167686	112475	161500
Fourth	146288	105692	147056
Fifth	90775	112757	94325

Table 2: The turnaround time for each server of the third experiment

Server number	Manual	Round robin	ANN (the proposed system)
Second	1451	2879	1458
Third	1686	1132	1623
Fourth	1635	1182	1644
Fifth	1524	1892	1584

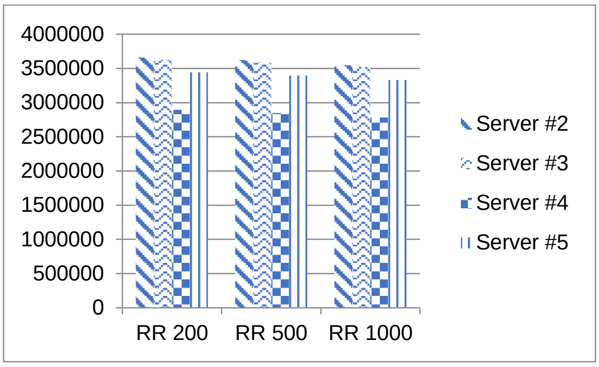

Fig. 12. The available memory at each server after using the Round Robin selection method

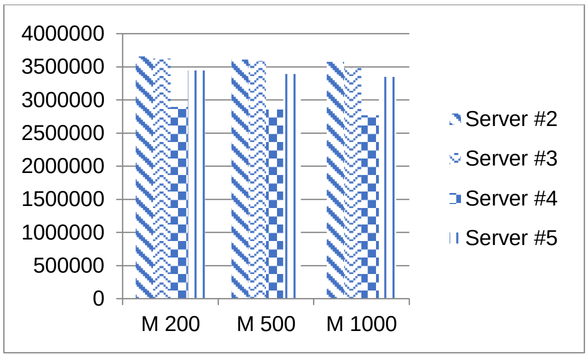

Fig. 13: The available Memory at each server after using the manual method

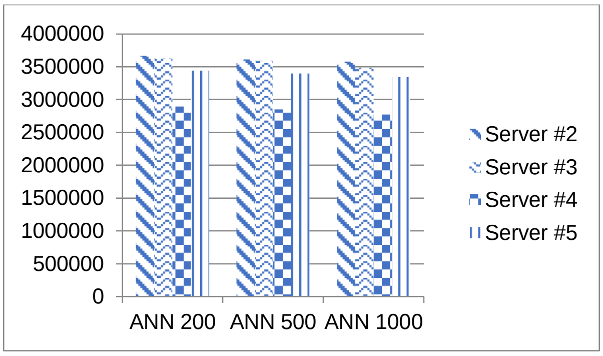

Fig. 14: The available memory at each server after using the ANN selection method

Conclusion

In this study, the features of the servers that affect the decision of selecting the best server in the mirroring server system are investigated. A new ANN system has been produced in this study in order to select the appropriate server for new user's requests. This proposal reflects the need for increasing the reliability and performance of different servers, which are encountered by considerable issues that should be effectively tackled. Additionally, the proposed ANN system is generated from the manual selection method, which is considered one of the best selection methods after eliminating the delay of human selection. The outcomes of this study highlight the potential roles of planning to experiment different ANN topologies and learning functions by comparing them with the currently used topology. Comparisons of the effectiveness of the proposed ANN system are carried out in this study with two different techniques, which are related to the manual human selection after eliminating the required time for making a human's own selection, along with the inclusion of the Round Robin selection method.

It can be inferred from such comparisons that the proposed model takes the advantages of these two techniques, which are based on the speed of selection for the Round Robin selection method and the selection of the best server according to the mirroring server features that are derived from the manual selection method. These can be seen from the obtained results obtained from the ANN system, which improved the use of mirroring servers by (10%) more effectively compared to the Round Robin selection method. On the other hand, the results have also indicated that the Round Robin selection method utilized more than (25%) of the total time in which most of the servers remain idle by not having further requests to serve.

The proposed work in this study can be further extended by including features related to the mirroring servers such as the distance and networking traffic and by studying their effects on the selection of the server by testing the proposed system pertaining to this study on some of the existing network simulators such as the Network Simulator-3 (NS-3) and the Global Mobile Information System Simulator (Glo Mo Sim). Moreover, it is suggested that the local minima problem of the Back Propagation Neural Network (BPNN) training function can be resolved by initializing the weights and by training the feed forward neutral network using optimization techniques such as Genetic Algorithms and Particle Swarm Optimization.

Acknowledgement

The authors of this manuscript would like to express their appreciations and gratitude to their universities for supporting this research.

The authors would like to thank editors for their efforts in handling the manuscript and all reviewers for the constructive comments which improved the original submission.

Author's Contributions

Radwan Al-Shalalfa: Acquisition of data, investigation, software, original draft preparation, approved the version to be submitted and any revised version.

Hazzaa Alshareef: Conceptualization, investigation, methodology, analysis and interpretation of data, review & editing, approved the version to be submitted and any revised version.

Azzam Sleit: Conceptualization, design, investigation, analysis and interpretation of data, review and editing, approved the version to be submitted and any revised version.

Mousa Al-Akhras: Conceptualization, design, investigation, analysis and interpretation of data, review and editing, approved the version to be submitted and any revised version.

Samah Alhazmi: Conceptualization, methodology, design, original draft preparation, approved the version to be submitted and any revised version.

Ethics

This study is original and innovative and contains unpublished material. The corresponding author confirms that all the other authors have read and approved the manuscript and no ethical issues involved or conflicts of interest to release.

References

Abdel-Hamid, Y. S., & Gulliver, T. A. (2003, December). Improved parallel access to multiple Internet mirror servers. In 2003 46th Midwest Symposium on Circuits and Systems (Vol. 1, pp. 446-449). IEEE. doi.org/10.1109/MWSCAS.2003.1562314

Al-Akhras, M. (2008, July). A genetic algorithm approach for voice quality prediction. In 2008 5th International Multi-Conference on Systems, Signals and Devices (pp. 1-6). IEEE. doi.org/10.1109/SSD.2008.4632900

Al-Akhras, M., & Saadeh, M. (2010). Automatic valuation of Jordanian estates using a genetically-optimised artificial neural network approach. WSEAS Transactions on Systems, 9, 905-916.

Al-Akhras, M., Dalhoum, A. L. A., Saadeh, H., & Bader, H. (2011a). A genetical-optimised artificial neural network approach for automatic detection of blood vessels using Gabor filter. The Mediterranean Journal of Computers and Networks, 7(2), 202.

Al-Akhras, M., ALMomani, I., & Sleit, A. (2011b). An improved E-model using artificial neural network VoIP quality predictor. Neural Network World, 21(1), 3. http://www.nnw.cz/doi/2011/NNW.2011.21.001.pdf

AWS Documentation. (2021). How Traffic Mirroring works, https://docs.aws.amazon.com/vpc/latest/mirroring/traffic-mirroring-how-it-works.html

Chow, E., & Cai, Y. (2003). Algorithms for Selecting Multiple Mirror Sites for Parallel Download. In IMSA-2003 conference paper (pp. 400-213).

Cao, J., Yang, L. T., Guo, M., & Lau, F. (2005). Parallel and Distributed Processing and Applications: Second International Symposium, ISPA 2004, Hong Kong, China, December 13-15, 2004, Proceedings (Vol. 3358). Springer.

Cardellini, V., Colajanni, M., & Yu, P. S. (1999, June). Redirection algorithms for load sharing in distributed Web-server systems. In Proceedings. 19th IEEE International Conference on Distributed Computing Systems (Cat. No. 99CB37003) (pp. 528-535). IEEE. doi.org/1109/ICDCS.1999.776555

el Hindi, K., & Al-Akhras, M. (2011). Smoothing decision boundaries to avoid overfitting in neural network training. Neural Network World, 21(4), 311. http://nnw.cz/doi/2011/NNW.2011.21.019.pdf

Gautam, N. (2002). Performance analysis and optimization of web proxy servers and mirror sites. European Journal of Operational Research, 142(2), 396-418. doi.org/10.1016/S0377-2217 (02)00210-2

Google Search Appliance Documentation. 2015. Configuring GSA Mirroring, https://static.googleusercontent.com/media/www.google.com/en//support/enterprise/static/gsa/docs/admin/current/gsa_doc_set/mirroring/mirroring.pdf

Govindan, K., Arunachalam, K., & Subramaniam, K. (2018, April). Optimal server selection policy for improved network efficiency in smart phones. In 2018 IEEE Wireless Communications and Networking Conference (WCNC) (pp. 1-6). IEEE. doi.org/10.1109/WCNC.2018.8376947

Hamdan, M., Hassan, E., Abdelaziz, A., Elhigazi, A., Mohammed, B., Khan, S., ... & Marsono, M. N. (2020). A comprehensive survey of load balancing techniques in software-defined network. Journal of Network and Computer Applications, 102856. doi.org/10.1016/j.jnca.2020.102856

Heaton, J. (2011). Introduction to the Math of Neural Networks (Beta-1). Heaton Research Inc. http://www.heatonresearch.com., 2011.

Hillmann, P., Uhlig, T., Rodosek, G. D., & Rose, O. (2016, June). Modeling the location selection of mirror servers in content delivery networks. In 2016 IEEE International Congress on Big Data (BigData Congress) (pp. 438-445). IEEE. doi.org/10.1109/BigDataCongress.2016.68

Irie, D., & Miwa, H. (2017, August). Network Design Method by Link Protection to Keep Connectivity and Communication Quality to Servers. In International Conference on Intelligent Networking and Collaborative Systems (pp. 423-433). Springer, Cham. doi.org/10.1007/978-3-319-65636-6_38

Kaastra, I., & Boyd, M. (1996). Designing a neural network for forecasting financial and economic time series. Neurocomputing, 10(3), 215-236. doi.org/10.1016/0925-2312 (95)00039-9

Liu, H., Jia, X., Li, D., & Lee, C. H. (2004). Optimal placement of mirrored web servers in ring networks. IEE Proceedings-Communications, 151(2), 170-178. https://digital-library.theiet.org/content/journals/10.1049/ip-com_20040183

Maeda, N., & Miwa, H. (2012, July). Detecting critical links for keeping shortest distance from clients to servers during failures. In 2012 IEEE/IPSJ 12th International Symposium on Applications and the Internet (pp. 320-325). IEEE. doi.org/10.1109/SAINT.2012.58

Matthur, A., & Mundur, P. (2003, July). Dynamic load balancing across mirrored multimedia servers. In 2003 International Conference on Multimedia and Expo. ICME'03. Proceedings (Cat. No. 03TH8698) (Vol. 2, pp. II-53). IEEE. doi.org/10.1109/ICME.2003.1221551

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5(4), 115-133. doi.org/10.1007/BF02478259

McManus, P. R. (1999). A passive system for server selection within mirrored resource environments using AS path length heuristics. https://www.ducksong.com/patrick/proximate.pdf

Myers, A., Dinda, P., & Zhang, H. (1999, March). Performance characteristics of mirror servers on the internet. In IEEE INFOCOM'99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No. 99CH36320) (Vol. 1, pp. 304-312). IEEE. doi.org/10.1109/INFCOM.1999.749296

Nakamura, R., Hashimoto, A., & Miwa, H. (2009, November). Methods of Locating Mirror Servers with High Connectivity and Small Distances. In 2009 International Conference on Intelligent Networking and Collaborative Systems (pp. 353-356). IEEE. doi.org/10.1109/INCOS.2009.53

Nakaniwa, A., Takahashi, J., Ebara, H., & Okada, H. (2000). Reliability-based optimal allocation of mirror servers for Internet. In Globecom'00-IEEE. Global Telecommunications Conference. Conference Record (Cat. No. 00CH37137) (Vol. 3, pp. 1571-1577). IEEE. doi.org/10.1109/GLOCOM.2000.891903

Raghul, S., Subashri, T., & Vimal, K. R. (2017, March). Literature survey on traffic-based server load balancing using SDN and open flow. In 2017 Fourth International Conference on Signal Processing, Communication and Networking (ICSCN) (pp. 1-6). IEEE. doi.org/ 10.1109/ICSCN.2017.8085416

Rodriguez, P., Kirpal, A., & Biersack, E. W. (2000, March). Parallel-access for mirror sites in the internet. In Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No. 00CH37064) (Vol. 2, pp. 864-873). IEEE. doi.org/10.1109/INFCOM.2000.832261

Sabareesh, D. S., Reddy, G. V. P., Jaiswal, S., Ppallan, J. M., Arunachalam, K., & Wu, Y. (2019, April). Redundant tcp connector (rtc) for improving the performance of mobile devices. In 2019 IEEE Wireless Communications and Networking Conference (WCNC) (pp. 1-7). IEEE. doi.org/10.1109/WCNC.2019.8885889

Sadrhaghighi, S., Dolati, M., Ghaderi, M., & Khonsari, A. (2021, June). SoftTap: A software-defined TAP via switch-based traffic mirroring. In 2021 IEEE 7th International Conference on Network Softwarization (NetSoft) (pp. 303-311). IEEE. doi.org/10.1109/NetSoft51509.2021.9492588

SQL Server Documentation. (2020). Database Mirroring, https://docs.microsoft.com/en-us/sql/database-engine/database-mirroring/database-mirroring-sql-server?view=sql-server-ver15

Sleit, A., Al-Mbaideen, W., Alzabin, N., Dawood, H., & Alqarute, K. (2007). Efficient query processing over mirror servers using genetic algorithms. Neural Network World, 17(4), 311.

Jamin, S., Jin, C., Kurc, A. R., Raz, D., & Shavitt, Y. (2001, April). Constrained mirror placement on the Internet. In Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No. 01CH37213) (Vol. 1, pp. 31-40). IEEE. doi.org/10.1109/INFCOM.2001.916684

Sun, Z., & Nakhai, M. R. (2020, June). An Online Mirror-Prox Optimization Approach to Proactive Resource Allocation in MEC. In ICC 2020-2020 IEEE International Conference on Communications (ICC) (pp. 1-6). IEEE. doi.org/10.1109/ICC40277.2020.9149032, 2020

Tavakkol, A., Kolli, A., Novakovic, S., Razavi, K., Gómez-Luna, J., Hassan, H., ... & Mutlu, O. (2018). Enabling efficient RDMA-based synchronous mirroring of persistent memory transactions. arXiv preprint arXiv:1810.09360. https://arxiv.org/abs/1810.09360

Tebelskis, J. M. (1995). Speech recognition using neural networks. Carnegie Mellon University..

Wang, C., Liu, X., Zhou, X., Zhou, R., Lv, D., Wang, M., & Zhou, Q. (2019, August). FalconEye: A High-Performance Distributed Security Scanning System. In 2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech) (pp. 282-288). IEEE. doi.org/10.1109/DASC/PiCom/CBDCom/CyberSciTech.2019.00059

Yokota, H., Kimura, S., & Ebihara, Y. (2004, March). A proposal of DNS-based adaptive load balancing method for mirror server systems and its implementation. In 18th International Conference on Advanced Information Networking and Applications, 2004. AINA 2004. (Vol. 2, pp. 208-213). IEEE. doi.org/10.1109/AINA.2004.1283788

Yu, R., Kilari, V. T., Xue, G., & Yang, D. (2019, April). Load balancing for interdependent IoT microservices. In IEEE INFOCOM 2019-IEEE Conference on Computer Communications (pp. 298-306). IEEE. doi.org/10.1109/INFOCOM.2019.8737450