Science Arena Publications
Specialty Journal of Electronic and Computer Sciences
ISSN: 2412-7485
Available online at www.sciarena.com
2018, Vol, 4 (3): 12-22

Presenting and Demonstrating New Method for better Resource Allocation in Cloud Computing

Ashkan Samari Masoudi

Master of IT Engineering Islamic Azad University.

Abstract: *Cloud computing is a very promising technology, which helps organizations and companies to reduce their operating costs and increase productivity of them. The unique features of cloud computing have always been coupled with many unknown and new challenges. In accordance to the fact that, the evaluation of practical efficiency of such calculations is not cost-effective, the modeling and evaluation of this type of system is important. In this paper, authors will present new method for better resource allocation in cloud computing network. By using of this method, many servers and gateways are not involved in sending and receiving requests; instead each request will be processed in the nearest server group. As a result, the traffic and inference on the network would be decreased, and simultaneously the scalability of network will increase.*

Keywords: *Cloud Computing, Resource Allocation, Optimization.*

INTRODUCTION

Load balancing is a network method for distributing workloads on a computer or a number of them; network lines, central processing units, disk drives, or other resources to achieve optimal utilization, maximum throughput, minimum response time, and avoid from system overheat. Using multiple components with load balancing, instead of a single component can increase reliability through redundancy (Darbandi, 2017, Vol. 24). The load balancing service is usually provided with proprietary software or hardware, such as the Domain Name Server (DNS). This service is an essential part of the distributed computing system. The load distribution maybe performed on a network server. In fact, CPU transfers information from the heavily loaded servers to the underutilized servers. The load balancing helps, to design powerful and efficient systems that lead to green computations (Pirfalakeh, 2015).

The load balancing answer which problems:

a) Effective distribution of user processes on virtual machines
b) Effective distribution of virtual machines on physical servers

Load balancing in which problems help us?

a) Significant improvement in performance
b) Having a reverse up plan for system faults
c) Maintain system stability
d) Compatibility with future changes
e) Efficient load distribution
f) To be economical (Darbandi, 2017, Vol. 24)

Load balancing in cloud platform:

Load balancing is the process of reallocating the entire load to the separate nodes from the collective system, for efficient resource utilization and improving the response time, and simultaneously eliminating the conditions, in which some nodes are heavily loaded and some others are underutilization. A dynamic load balancing protocol, does not inherently consider the state or behavior of the system, but depends on the current behavior of the system (Pirfalakeh, 2015).

Objectives of load balancing:

a) Significant improvement in performance
b) Keep system's order
c) Increase flexibility so that it matches the changes
d) Build a fault tolerant system by creating backups (Pirfalakeh, 2015). Fig. 1, shows definition of load balancing in one simple 3*3 network, which allocates resources based on requests from network nodes.

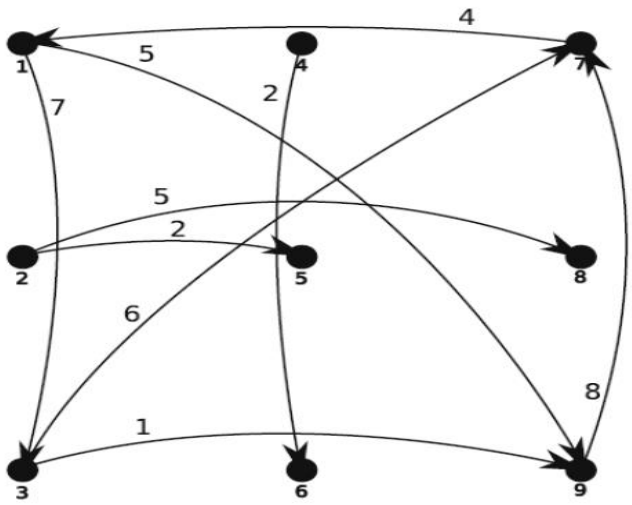

Figure 1: Simple Schematic of Load Balancing in 3*3 Servers Network (Farhadi, 2012).

Challenges that load balancing algorithms faced with them:

a) Emigration of virtual machines: How to distribute data among different machines
b) Develop small datacenters for cloud computing: Creates very useful and small datacenters that help in a geodiversity computing
c) Energy management: Helps in achieving low scalability costs
d) Management of stored data: Distribute data for the optimal use of space for data storage
e) Provision of automated services: Increase elasticity and auto providing of resources (Darbandi, 2017, Vol. 5).

Standards for load balancing in the cloud:

a) Dependent overhead: Determine the amount of involved overhead, while a load distribution algorithm is implemented. This parameter of overhead, due to transfer of tasks, internal processor, and communication of internal processes, are composed. This overhead should be minimized, so that the load balancing methods can work effectively (Darbandi, 2017, Vol. 23).
b) Throughput: To calculate the number of tasks, that their execution has been completed. This parameter is used to improve system performance.
c) Efficiency: This parameter checks the system's effectiveness. Improves reasonable costs such as, reducing response time, so that the delays are acceptable (Darbandi, 2017, Vol. 23).

d) Resource exploitation: Investigate the optimal use of resources, in an efficient load balancing operation, resource exploitation should be optimized.
e) Scalability: It means the ability of an algorithm to perform load balancing in a system with any finite number of nodes. This criterion needs to be improved.
f) Response time: Amount of utilized time to respond to a specific load balancing algorithm, in a system. This parameter should be minimized.
g) Fault tolerance: Is the ability of the algorithm to perform a uniform load balancing with an arbitrary node or broken connection. Load balancing should be a good fault tolerance technique (Pirfalakeh, 2015).
h) Break point: Control of load balancing and data collection from different nodes and systems are designed, to prevent a separate failure point in algorithms. For example, in centralized algorithm, if a controller fails, the whole system will fail. Each load balancing algorithm, must be designed to overcome this challenge (Pirfalakeh, 2015).

This paper is organized as such: The first section is introduction, which is about load balancing, and objectives and problems of such algorithm. The aim of reviewing this technique in the first section, is that, because the novel idea of authors is about better resource allocation, and to some extent relating with load balancing. The second section, will review resource allocation and traffic engineering concepts. The third section, will review about fairness and load balancing, in this section, authors will also review the concept of least-maximum fairness. In the fourth section, authors provide readers with some information about, clustering in cloud computing network, authors believe that by use of clustering of resources in cloud network, and sending requests of users to the nearest cluster of servers, we do not have lots of congestion and interferences of signals in the bandwidth of the network. In the fifth section, you will read about significant parameters in evaluating clusters. The sixth section, will be simulation results and analysis of such results. Finally, the seventh section, will be conclusion and eighth section will be references.

Resource Allocation and Traffic Engineering:

In this paper, authors try to effectively manage and optimized, cloud computing servers by clustering servers, as well as effectively using power control and rate control techniques. In fact, using the above techniques to optimally allocate the resources, is one of the most effective traffic engineering techniques in the network; since resource allocation helps to improve fairness and load balancing (Neamatollahi, Taheri, 2011).
Traffic engineering is a process which, traffic flows in the network can be controlled in such a way that the best use of high performance network resources (appropriate use of resources) is obtained. Proper use of resources means that, the traffic flow is such that, no server is overcrowded, while another server is no longer used or with much lower traffic than its actual capacity.
In fact, traffic engineering is an aspect of network engineering that deals with the evaluation and optimization of the operation of the operational networks. The aspects of traffic engineering optimization, can be achieved through capacity management and traffic management (Neamatollahi, Taheri, 2011).
Routing is actually a solution to an optimization problem. The problem can be defined as: finding the best way to send the origin data toward destination, taking into account a series of limitations, such as network topology and interference. Although optimization goals, vary in different routing algorithms, they all follow the same principal altogether: If the interface node R, is the optimal path P_{XY}, from node X to node Y, the optimal path from node R to node Y, must also be in the same direction. Therefore, all the optimal paths from all resources toward the destination, are the tree, which the root of that, is the destination. Since, there are usually several paths with the same function, from origin to the destination, this is not the only tree. As a result, a routing

protocol is, in fact, finding different trees and using them to form the route of data transfer, from any source to the desired destination (Neamatollahi, Taheri, 2011).
Below are some of these criteria which uses in such network, along with a brief explanation of them:

1) Hop count: The number of required steps to send data between two source and destination nodes.
2) Round-trip-time in each hop: The average round trip time is calculated, by sending probe packets using the Unicast method, between neighboring nodes and calculating the elapsed time until the acknowledgement packet is received.
3) Packet pair delay in each hop: The packet pair delay in each hop, is measured by sending to sets of probe packets in sequences, from one node to the neighbor of that.
4) Expected transmission count: The expected transmission count for a link, is the number of posts needed to successfully send a packet to a link.
5) Expected transmission time: In fact, expected transmission time, the same number of expected transitions, which takes into account, the size of the packets and the quality of links. In a data transmission path, the expected transfer time is equal to the expected total transfer time of all links of the same path (Neamatollahi, Taheri, 2011).

A routing protocol in a new generation of communication networks, can be proactive or reactive. In the proactive routing, the data transmission path between the two nodes occurs before a traffic flow is established between them, while in passive routing, the data transmission path between the two nodes occurs when traffic between the two nodes is established (Neamatollahi, Taheri, 2011).
A routing protocol can be static or dynamic, due to changes in network topology, link quality, traffic load, and etc. In wired networks, static routing is used. In multi-hop wireless networks, such as wireless networking, dynamic routing is used, due to nodes moving, instability of links, topology changes, traffic changes and more. Two of the most well-known dynamic routings, are distance vector routing and link state routing, are actually designed for wired networks. These two protocols, are the basis of a great deal of dynamic routing protocols, for mobile devices and wireless networks. A routing protocol can be operated as centralized, distributed, or a combination of the two (Neamatollahi, Taheri, 2011).

Fairness and Load Balancing:

In cloud network, due to the fact that, there are no residing users (it is possible that, lots of users connect to the network via laptops and mobile phones), and not knowing the exact traffic pattern of each of them, the inherent nature of the traffic requests is unclear, and the precise and predefined traffic pattern cannot be identified. On the other hand, as we mentioned earlier, the basis of work was based on the identification of traffic. In many applications, traffic requests may not be predefined or may be changed over time (Neamatollahi, Taheri, 2011).
As shown in the Fig. 2, the servers which has lots of tasks and operations in its own queue, will not accept any more packets and duties. And also if, any server has lots of duties and processes in its own queue, it will transfer some of the operations and tasks to the servers with lower traffics. For example, as you can see in the Fig. 2, the G Server transfer the G2 and G3 packets to the H server, because these packets has high priority and they should be processed in advance and without any delay. And subsequently the G server accepts H2 and H3 packets from H server.
In the majority of communication and computing networks, network capacity is a limited amount. For this reason, the fair allocation of this capacity to the network nodes, in a way that all nodes in the network can be equally capable of this capacity is a significant issue. Hence, in this section, authors have tried to address the issue of fairness. In this paper, the concept of maximum-minimum fairness is used, which in facts attempts to maximize the minimum bandwidth and resources allocated to network users (Pirfalakeh, 2015)

To improve network performance, another criterion, which is called load balancing on network nodes, should also be improved. This criterion is expressed in terms of the minimum-maximum node load, which minimizes the load on the node that has the maximum load for load balancing on the nodes (Neamatollahi, Taheri, 2011). It should be noted that, in all equations of this section, the forwarded power of all nodes is considered to be constant, and the sent rate of all nodes is considered to be one packet in each time slot (Darbandi, 2017, Vol. 24).

Definition of maximum-minimum concept:

Maximum-minimum is one of the optimization aspects, in which, trying to maximize the minimum value of the variables under study. Using this type of optimization problem, it is ensured that all the variables examined are of a higher threshold; while optimization aims is to increase that threshold as much as possible (Darbandi, 2017, Vol. 23).

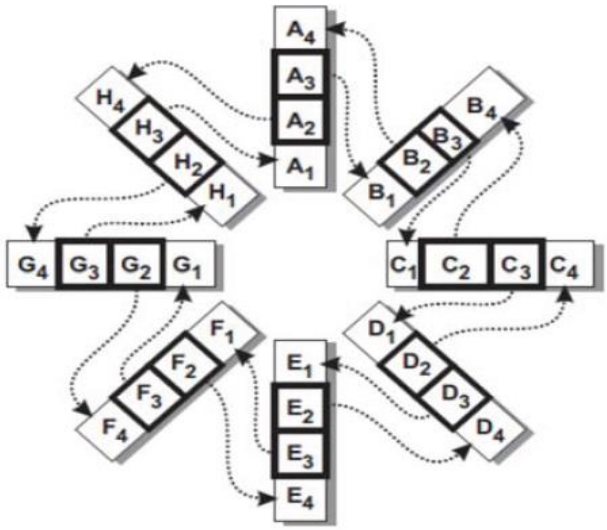

Figure 2: Transferring of duties and processes among servers (Pirfalakeh, 2015).

Definition of minimum-maximum concept:

Minimum-maximum, as the previous case, is one of the most widely used aspects of optimization, which is exactly reverse of maximum minimum problem. In this problem, while trying to minimize the maximum amount of variables which are under consideration, so that the amount of variable does not exceed the threshold level; they try to minimize this threshold level (Farhadi, 2012).

Maximizing capacity with use of fairness:

As we mentioned earlier, in this paper, fairness is defined as maximum-minimum problem. Maximum-minimum fairness, means trying to maximize the minimum capacity, assigned to each node. In fact, in this section, the objective is to maximize the minimum capacity assigned to non-gateway nodes, so that the number of time segments needed to transfer network traffic requests is also reduced (Farhadi, 2012).

Problem formulation can be done in two ways: complete fairness between all non-gateway nodes and relative fairness between all non-gateway nodes.

In complete fairness, the traffic requests of all nodes are considered the same and try to maximize it as much as possible. In another words, by assigning the same capacity to all non-gate nodes, the fairness issue is fully established between all the nodes (Farhadi, 2012).

However, in relative fairness to increase network capacity while we know that all traffic requests are higher that the permissible limit, and this permissible limit is as high as possible, traffic requests of some non-gateway nodes that can be increased, will raised up to the maximum point. In another words, the capacity of some nodes

increases, compared to the other nodes. Among these nodes, we can point to the nodes that are close to gateway node and can move more traffic requests between themselves and the gateway, via single pass transfer. In this case, although the complete fairness which is established between the network nodes is complete fairness is eliminated; but while there is relative fairness between nodes, and at least we are sure that greater than threshold value, the network transmission rate has increased compared to the complete fairness state (Farhadi, 2012).

Formulization of problem:

The following variables and parameters should be considered:

Capacity (i): The amount of capacity between non-gateway node *i* and gateway node.

Min-capacity: Is equal to the least capacity allocated to the network nodes, which is at least one.

Max-min-capacity: Is the variable which is equal to the maximum amount of minimum allocated capacity to the network nodes. This variable is used for normalization (Farhadi, 2012).

Max-active: is equal to the active time slots of one configuration. This variable, is also used for normalization.

Flow (i, j): is the load flow, which is pass through *(i, j)* connection.

B: is the maximum bandwidth of gateway node, or in another word, the maximum amount of capacity that the gateway node can accept. Usually this amount is 300.

As we mentioned earlier, the maximization of minimum capacity of non-gateway nodes and minimization the number of required time slots, should be combined with each other. For this purpose, first of all these two variables should be normalized; it means that we should divide each variable by the maximum amount that it can obtained, to become normalized and after that they linearly combined (Farhadi, 2012).

It should be mentioned that, we used α, ß weights for weighting these two variables. In the sixth section of this paper, you will see that by changing these weights, we obtain interesting results. The relation between α, ß should be as follow (Pirfalakeh, 2015):

$$\alpha + \beta = 1 \quad (1)$$

$$0 \leq \alpha, \beta \leq 1 \quad (2)$$

According to aforementioned definitions, the target function can be defined as follows:

$$maximize: \alpha * \left(\frac{min - capacity}{max - min - capacity}\right) + \beta * \left(\sum h \in sets \frac{active\ (h)}{max - active}\right)^{-1} \quad (3)$$

Based on the following conditions (Farhadi, 2012):

$$\sum_{i \in N} capacity\ (i) \leq B \quad (4)$$

$$\sum_{(i,j) \in L} flow(i,j) - \sum_{(j,i) \in L} flow(j,i) = capacity\ (i);\ \forall i \in N \quad (5)$$

$$\sum_{h \in set\ (i,j)} active\ (h) \geq flow(i,j);\ \forall\ (i,j) \in L \quad (6)$$

$$capacity\ (i) = min - capacity;\ \forall\ i \in N \quad (7)$$

The (3) formula, states that, the capacity of gate node, is limited by amount of *B*. The (5) equation, do routing in the network; the (6) equation, guarantee that the connection is at least active in one-time slot, if one packet

is transferred on one connection. The (7) equation, is guarantee that all nodes has the same capacity of $min - capacity$ (Farhadi, 2012).
This algorithm, has routing and timing of connection.

Relative fairness:

Formulization of problem:
Target function (Darbandi, 2017, Vol. 24):

$$maximize: \alpha * \sum_{i \in N} \frac{capacity\ (i)}{max - capacity} + \beta * (\sum h \in sets \frac{active\ (h)}{max - active})^{-1} \quad (8)$$

Based on these constraints:

$$\sum_{i \in N} capacity\ (i) \leq B \quad (9)$$

$$\sum_{(i,j) \in L} flow(i,j) - \sum_{(j,i) \in L} flow(j,i) = capacity\ (i);\ \forall i \in N \quad (10)$$

$$\sum_{h \in set\ (i,j)} active\ (h) \geq flow(i,j);\ \forall\ (i,j) \in L \quad (11)$$

$$capacity\ (i) \geq min - capacity;\ \forall\ i \in N \quad (12)$$

As the previous case, α and ß are weight of maximizing traffic requests problem and minimizing the needed parts in target function, respectively (each of them is less than one, and the summation of them is one). The (9), (10), and (11) equations, are the same as the (4), (5), and (6) formulas. The (12) formula, let the capacity of network nodes, to increase more than $min - capacity$ (Farhadi, 2012).
It should be mentioned that, in above formulas, $max - capacity$ is the variable, which is equal to the maximum amount of gate node.
Also, it should be noted that, $min - capacity$ is not still a variable, but the amount of that is determined by solving the problem of complete fairness. This algorithm is also, has routing and timing of connection (Farhadi, 2012).

Clustering in Cloud Network:

Up to this point, we review about load balancing and different aspects of this technique in the first section. In the second section, authors explain about resource allocation and traffic engineering, which become an interesting topic in nowadays communication, through new generation of networks. After that, in the third section, authors consider about concept of fairness and maximum-minimum fairness, and also authors formulized such techniques for employing in networks for gaining more power and better utilizing of resources. Now, in the fourth section, authors wants to explain their novelty (Darbandi, 2017, Vol. 5). Authors state that, if cloud providers, clustering their servers and other resources in different geographical locations and respond the demands of users by means of nearest servers, they do not have lots of congestion over their bandwidth, or lots of servers and gateways do not become busy, because of transferring one request to the very far datacenter and after that getting the result from that specific server, and again submit the result of that processing to the user (Neamatollahi, Taheri, 2011). Authors, suggest to the cloud providers to set and establish their datacenters in different geographical locations and by schematics like triangle or star or other schematics and respond the cloud user demands by the means of nearest servers. So that there is no more delay, because of round-trip-time

of packets in network, also there is no more inference between the packets and signals, when they are submitted to far cloud servers (Darbandi, 2017, Vol. 23). By using of such idea, cloud providers and also users, will consume less energy to transmit one packet through communication media to the far server (Pirfalakeh, 2015). Also, by applying such technique to the network, cloud service providers will have more scalability. When cloud providers set their datacenters all around the world and in different geographical points, in each datacenter, they can set one of the servers as reference server, and this server decide which of the servers in that specific datacenter do the process of one packet or answer the demand of one user. By using of this idea, users will face with more distributed network, and there is no need for any users of the network to have comprehensive information about all aspects of the network (Neamatollahi, Taheri, 2011).
In the next section, authors will review about evaluating criteria's of the clusters.

Clusters Assessing Criteria's:

Suppose there are N servers $\{S_1, S_2, \dots, S_N\}$ in the cloud network, and among them K servers $\{S_1, S_2, \dots, S_K\}$, which $K \ll N$ choose as head-clusters, which respectively arrange $\{C_1, C_2, \dots, C_K\}$ clusters. The number of servers of such clusters are $\{n_1, n_2, \dots, n_K\}$. With knowing such hypothesis, in the following, authors present qualitative evaluation criteria's of clustering in cloud computing network (Neamatollahi, Taheri, 2011).

The centrality of the cluster-head in the cluster:

In order to evaluate this criterion, first of all we should calculate the average distance between each cluster and its members, and then average the results obtained for the whole cluster head. If the distance of each server i to the correspondence head-cluster is shown by L_i ($L_i \geq 0$), (when the server S_i, is head cluster then L_i become zero), then (Neamatollahi, Taheri, 2011):

$$centrality = \frac{\sum_{j=1}^{K}((\sum_{i \in cluster\ j} I_j)/n_j)}{K} \quad (13)$$

The smaller the $centrality$, it means that, generally head-clusters located in the center of cluster, and for the cluster-head that is closer and closer to the center of the cluster (Neamatollahi, Taheri, 2011):

- The connection of other servers to the cluster head requires less power consumption, because the energy used to send messages in these networks is directly related to the square of distance between the sender and the receiver.
- The inference of the signals between the heads of the clusters is reduced. If we assume that the cluster heads are not at the center of the cluster, it is likely that the cluster heads are located in the domain of each other, and the generated interference cause wasting of network resources (Darbandi, 2017, Vol. 5).

Distribution of head clusters throughout the network:

To evaluate this criterion, we must calculate the mean distance between each cluster head and other cluster heads, and then calculate the total average as follows (Neamatollahi, Taheri, 2011):

$$Distribution = \frac{\sum_{j=1}^{K}(\sum_{i=1, i \neq j}^{K} distance\ (S_j, S_i))/(K-1))}{K} \quad (14)$$

The higher the $Distribution$, the clusters is generally distributed more better in the environment, and the better head clusters distributed in the environment (Neamatollahi, Taheri, 2011; Darbandi, 2017, Vol. 24):

- The average distance between each server and its correspondence head cluster is lowered, so the energy consumption of the server for sending messages to the cluster is reduced. This will increase the scalability of the number of servers.

- The loss of network resources, due to interference of head cluster signals is reduced together (Neamatollahi, Taheri, 2011; Darbandi, 2017, Vol. 23).

The difference in the number of servers in the clusters:

To evaluate this criterion, the largest difference in the number of servers within the cluster should be calculated at any time (Neamatollahi, Taheri, 2011):

$$Cluster\ load = \max(n_i - n_j), \forall i,j, 1 \le i,j \le K \quad (15)$$

The lower this value, this means that the servers are more uniform in the clusters, and the difference in the number of servers in the cluster is less. Therefore, a better load distribution has been made among the cluster heads.

Join of the servers at the nearest cluster:

If the number of servers that are members of the closest head cluster is m:

$$Proximity = \frac{m}{N-K} \quad (16)$$

The higher this value, the greater of the percentage of head cluster servers are members of closest head cluster. If the server join their closest head cluster, the interface of their signals with the servers of other clusters decreases (Neamatollahi, Taheri, 2011).

Simulation Results:

To do simulations, we have the following assumptions (Farhadi, 2012):

- The number of network servers is ten, with the location of all servers randomly selected with a uniform distribution function.
- The gateway server has a fixed location in all cases.
- The maximum capacity of gateway server, or in another word, the maximum amount of traffic requests that the gateway server can accept (β), is 300, with the assumption that the gateway server interface, to the internet, is Giga Ethernet type, and the distance between both a packet owned to a server is 1.8 seconds, then the maximum number of time slots is 300.
- In order to cite any of the results, ten samples are generated randomly and the average of the numbers obtained from them are listed in the table:

Table 1: the results of the complete fairness algorithm for different α and β (Farhadi, 2012)

α	Minimum server capacity	Total capacity of the network	Number of time slots in single-channel scenario	Number of time slots in multi-channel scenario	Load variance of each server
0	1	9	9.3	2	1.74
0.1	3.46	31.26	33.51	4.13	7.28
0.2	8.9	80.19	80.62	5.52	14.9
0.3	11.51	103.58	110.37	6.41	24.35
0.4	14.98	134.82	140.98	7.78	38.53
0.5	20.92	182.28	193.4	9.21	75.09
0.6	23.02	207.19	210.79	13.4	167.28
0.7	27.84	250.54	252.41	15.13	188.89
0.8	30.39	273.5	281.54	17.82	243.03
0.9	32.37	291.33	299.25	19.45	288.22
1	33	297	683.79	43.79	450.36

As you can see, by increasing the α value, the objective function more focuses on maximizing the minimum servers' capacity than minimizing the number of time slots. Therefore, with increasing α, the minimum server capacity and total network capacity will increase. Clearly, the flow of more traffic requests, will require more time slots (Farhadi, 2012).

The result of the table confirm this claim that, with increase in α, the number of required time slots increases both in single-channel scenario and in multi-channel scenario. Also, with increasing of α, due to more load on the network, the average load balancing on the servers also increases. In $\alpha = 0$ state, the objective function, only tries to minimize the number of time slots, needed to send data. Since the minimum capacity of each node is considered to be 1, the total capacity in this case is always equal to 9 (Farhadi, 2012; Darbandi, 2017. Vol. 24).

In $\alpha = 1$ state, the objective function only tries to maximize the minimum nodes capacity and has no emphasis on reducing the number of time slots.

In the table 2, the results of the load balancing algorithm are modeled on servers with complete fairness for different α values and with γ=β assumption (Farhadi, 2012; Darbandi, 2017, Vol. 23).

Table 2: The result of load balancing algorithm on servers with complete fairness for different α values and with γ=β assumption (Farhadi, 2012)

α	Minimum server capacity	Total capacity of the network	Number of time slots in single-channel scenario	Number of time slots in multi-channel scenario	Load variance of each server
0	1	9	13.45	3.16	0.57
0.1	3.72	33.57	38.88	4.65	1.93
0.2	9.72	87.3	91.92	6.99	4.65
0.3	12.26	110.25	134.36	9.12	6.82
0.4	16.89	152.2	159.87	12.7	11.03
0.5	22.26	199.8	208.6	16.08	38.49
0.6	23.5	212.05	229.34	17.36	49.21
0.7	28.95	261.2	243.02	20.83	67.32
0.8	31.68	287.12	298.15	24.23	89.84
0.9	32.81	295.1	299.93	29.31	93.22
1	33	297	571.34	52.8	292.36

As you can see, with increasing of α; minimum server capacity, total network capacity, the number of time slots required to be sent in both single-channel and multi-channel scenarios and the average load variance on servers, increases (Farhadi, 2012).

In addition, with increasing of α, load balancing on the server worsens; because in the objective function, the weight assigned to the load balancing criterion decreases on the servers.

With the comparison of this table to table 1, the average load balancing for each α, on servers is improved.

Since in this algorithm, the objective function, except for trying to establish fairness and minimizing the number of required time slots, also wants to minimize the load of servers, the weight assigned to the criterion for minimizing the number of time slots decreases, and therefore in comparison with table 1, for all α, increases the number of time slots (Farhadi, 2012).

In $\alpha = 0$ state, each node is assigned a minimum capacity of 1; but since the objective function is not exclusive to the criterion of minimizing the number of time slots, the number of time slots in this case is increased, compared to table 1 (Farhadi, 2012).

In $\alpha = 1$ state, the objective function, only tries to maximize the minimum servers capacity. For this reason, as in previous cases, the algorithm allocates the largest number of possible time slots.

Conclusion:

Cloud computing shows new way of servicing for the users, without any need for them to buy very powerful systems and servers. They can use unlimited speed of processing and unlimited capacity of storing, according to their need. This feature enables users to save lots of money, because they do not need to buy software licenses for their software and applications that they use on their system. Managing of such huge network is of great importance, and if any cloud server, overheats or crashes, it may cause unbelievable loss in number of users and amount of revenue for the cloud provider. As a result, in this paper, authors discuss about such important criteria and they try to purpose new idea for better managing of servers. Authors suggest that, if cloud providers, cluster their servers and arrange and set their servers in architectures or schematics like star and/or triangle and/or other schematics, and send the requests of every user to the nearest cluster, it may prevent the whole network from crashing, because there is no more extra load on the network and lots of resources are not get busy with sending and receiving of requests to very far servers. Also, by assigning each task to the nearest server, users do not face with low speed of servers, because of lots of congestion and inferences of signals on network media's.

References

1. Abedi M.; "involving Kalman filter technique for increasing the reliability and efficiency of cloud computing", International World Competition 2012; Los Vegas, USA.
2. Ahmadi, Nasrin; "Modeling of Reliability in Cloud Computing using Petri nets", MSc. Thesis, University of Science & Culture, 2016.
3. Darbandi, Mehdi; "Kalman Filtering for Estimation and Prediction Servers with Lower Traffic Loads for Transferring High-Level Processes in Cloud Computing"; Published by HCTL International Journal of Technology Innovations and Research, (ISSN: 2321-1814), Vol. 23, Issue 1, pp. 10-20, Feb. 2017, DOI: 10.5281/Zenodo.345288.
4. Darbandi, Mehdi; "Proposing New Intelligence Algorithm for Suggesting Better Services to Cloud Users based on Kalman Filtering"; Published by Journal of Computer Sciences and Applications (ISSN: 2328-7268), Vol. 5, Issue 1, 2017; PP. 11-16; DOI: 10.12691/JCSA-5-1-2; USA.
5. Darbandi, Mehdi; "Proposing New Intelligent System for Suggesting Better Service Providers in Cloud Computing based on Kalman Filtering"; Published by HCTL International Journal of Technology Innovations and Research, (ISSN: 2321-1814), Vol. 24, Issue 1, pp. 1-9, Mar. 2017, DOI: 10.5281/Zenodo.1034475.
6. Neamatollahi P., Taheri H.; "Proposing Criteria's for Qualitative Evaluating of Clustering in WSN", 11th Fuzzy System Conference, Zahedan, Iran, 2011.
7. Shahbazi P., "New Novel idea for Cloud Computing: How can we use Kalman filter in security of Cloud Computing", International IEEE Conf. AICT., Oct. 2012, Georgia, Tbilisi.